New Feature Extraction Methods For Classifying Proteins From Amino Acid Sequences

A dissertation submitted in partial fulfillment
of the requirements of M.Tech (Computer Science)
degree of Indian Statistical Institute, Kolkata
by

Ruma Chakraborty

Under the supervision of

Dr. Sanghamitra Bandyopadhyay
Machine Intelligence Unit

Indian Statistical Institute
203, Barrackpore Trunk Road
Kolkata-700108.

July 7, 2003

Indian Statistical Institute

203, Barrackpore Trunk Road,

Kolkata-700108.

Certificate of Approval

This is to certify that this thesis titled "**New Feature Extraction Methods for Classifying Proteins From Amino Acid Sequences**", submitted by Ruma Chakraborty towards partial fulfillment of requirements for the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata, embodies the work done under my supervision.

Sanghamitra Bandyopadhyay

Dr. Sanghamitra Bandyopadhyay,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata- 700108

Acknowledgement

I am extremely thankful to Professor Sanghamitra Bandyopadhyay for her constant guidance during all stages of my dissertation. She has introduced me to the field of Bioinformatics. I am thankful to her for her constant help, suggestions, criticism, assistance and enthusiasm in this work. She has given me the right direction at every step, which has benefited me immensely in giving this work its present shape.

Ruma Chakraborty

CONTENTS

Chapter 1

Introduction

An interdisciplinary area of Computer Science and Molecular Biology that has developed in the recent years is called Bioinformatics [2]. This was necessitated by the ever-increasing amount of raw data generated and routinely collected by molecular biologists. This is as a result of Human Genome Project and similar efforts, along with dramatic evolution of technology for information storage and retrieval. In response to this problem a number of researchers have developed techniques to interpret the data and discover concepts in the DNA, RNA and protein databases. An important problem in the domain of Bioinformatics is the classification of protein sequences. Proteins are chains of amino acids and form the basic building blocks of a living organism. Classification of a protein sequence allows one to infer the structure and function of proteins. Perhaps, the most important practical application of such knowledge is in drug discovery. A primary challenge in classifying protein sequences lies in the proper extraction of a feature vector. Evidently a good input representation (extraction of feature) is crucial for proper classification of the proteins. In this dissertation we propose new feature extraction methods for classifying proteins from amino acid sequences. This chapter is organized as follows. In Section 1.1 we give the basic concepts of Molecular Biology. This section deals with the basic structure and function of proteins and nucleic acids, the mechanism of molecular genetics and other related terminologies that we come across in the research works related to Bioinformatics. Section 1.2 gives an overview of the existing biological sequence databases. In Section 1.3 we detail out what Bioinformatics is and the role of a computer scientist in the field of molecular biology. Finally Section 1.4 deals with conclusions and the organization of the thesis.

1.1 Basic Concepts of Molecular Biology

All living things are made up of tiny living parts called the *Cells*. Similar cells join to form T*issues*. Similar tissues organize themselves to form *Organs*. Similar organs arrange themselves to form an *Organism*. Thus in the molecular level both complex and simple organisms have a similar chemistry. The main actors in the chemistry of life are molecules called *proteins* and *nucleic acids*. Roughly speaking, proteins are responsible for what a living being is and what it does. The distinguished scientist Russell Doolittle once wrote, "*we are our proteins*". Nucleic acids on the other hand, encode the information necessary to produce proteins and are responsible for passing along this "recipe" to subsequent generations. Molecular biology [1, 23, 24] research is basically devoted to the understanding of structure and function of proteins and nucleic acids. In the following section we provide a preliminary discussion on the structure and function of proteins.

1.1.1 Proteins

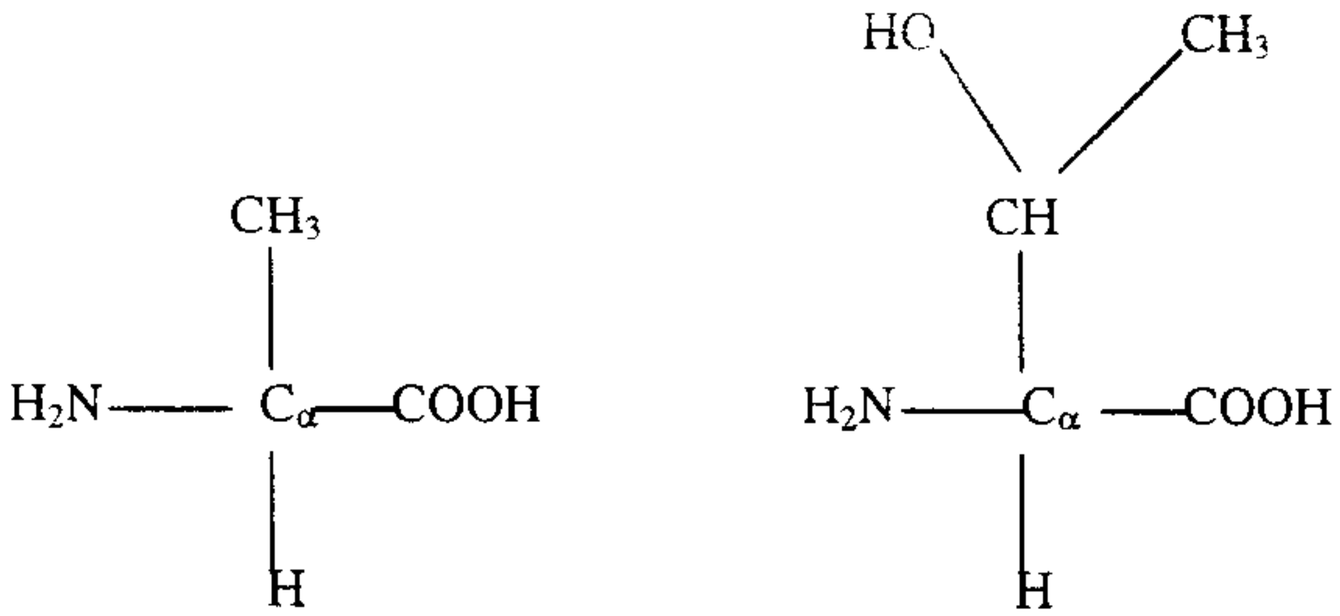


Figure 1.1 Examples of amino acids: alanine (left) and threonine (right).

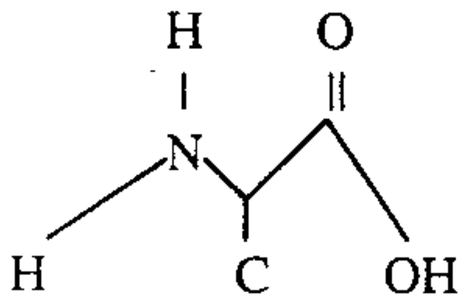
Proteins [25] are basically the tissue building blocks of a living being. Proteins are large organic molecules and are among the most important components in the cells of an organism. They are more diverse in structure and function than any other kind of molecule. *Structural proteins* act as tissue building blocks, whereas there are other proteins known as *enzymes* which act as catalyst of chemical reactions occurring inside any living organism. Other than these, antibodies, hormones, transport molecules, hair, skin, muscle tendons, cartilage, claws, nails, horns, hooves, and feathers are all made of proteins.

A protein is a chain of simpler molecules called *amino acids*. Examplesof amino acids can be seen in Figure1.1. In nature there are 20 different amino acids, which are listed in Table 1.1. Every amino acid has one central carbon atom, which is known as alpha carbon, or C_α. To the C_α are attached a hydrogen atom, an amino group NH_2, a carboxyl group (COOH), and a side chain. It is this side chain that distinguishes one amino acid from another. Side chains can be as simple as one hydrogen atom (the case of amino acid glycine) or as complicated as two carbon rings (the case of tryptophan). The combination of two amino acids is known as a *peptide linkage*. Figure1.2 shows how a sequential condensation produces a chain of bonded amino acids known as polypeptide chain. The linear sequence of amino acids in polypeptide chain is known as the primary structure of a protein. The enormous diversity of proteins is due to the many ways in which amino acids can combine in these chains. A set of n amino acids can form 20^n polypeptides; so 20^{100} combinations are possible for a protein of *100* amino acids. This number is larger than the total number of known atoms in the universe. Typical proteins contain about *300* residues, but there are proteins with as few as *100* or with as many as *5,000* residues.

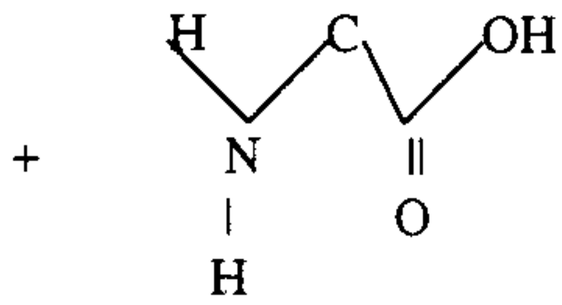
Within long polypeptide chains, certain section twist into coils and fold into sheets. These shapes are known as the *secondary structure* of proteins. Secondary structures are formed by hydrogen bonds between the carboxylic acid group and the amino group of the amino acids, which are not adjacent to the polypeptide chain. If the two amino acids are part of a single chain, a twisted-helix shape is formed. When a protein is built from multiple chains of polypeptides, multiple bonds can form across such chains. This creates pleated sheets. Proteins actually fold in three dimensions, presenting *secondary*, *tertiary*, and *quaternary structures*.

One letter Code	Name of the Amino Acid
A	Alanine
C	Cysteine
D	Aspartic Acid
E	Glutamic Acid
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
K	Lysine
L	Leucine
M	Methionine
N	Asparagine
P	Proline
Q	Glutamine
R	Arginine
S	Serine
T	Threonine
V	Valine
W	Tryptophan
Y	Tyrosine

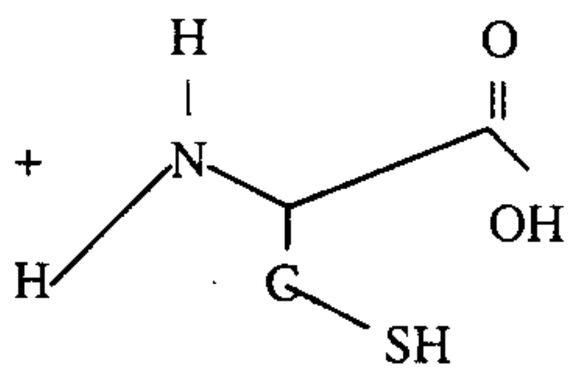
Table 1.2: The twenty amino acids commonly found in proteins



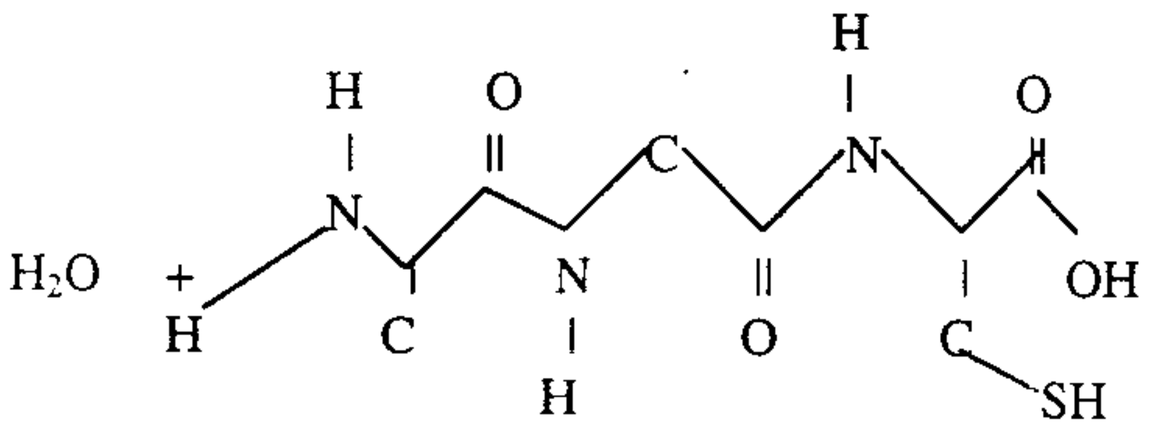
Alanine (A)



Glycine (G)



Cysteine(C)



Polypeptide Chain AGC plus *2* molecules of water

Figure 1.2 Condensation Reaction producing polypeptide chain

But how do we get proteins? Proteins are produced in a cell structure called ribosome. In a ribosome the component amino acids of a protein are assembled one by one. To explain how this happens we need to explain what nucleic acids are.

1.1.2 Nucleic Acids

Nucleic acids [26] are the second most important topic of interest in Molecular Biology. Though nucleic acids too are as important as proteins in molecular biology researches but it is not a concern in the present work. So only the salient points related to nucleic acid structure and functions are discussed here. Nucleic Acids basically encode the necessary information to form proteins and are responsible for passing along this recipe to successive generations. Living organisms contain two kinds of nucleic acids: *ribonucleic acids*, abbreviated RNA, and *deoxyribonucleic acid*, or DNA. The basic unit of both these is a sugar molecule. We describe DNA first. Figure1.3 (a) shows a basic DNA molecule. The bases here are nitrogenous compounds, which distinguishes one molecule from the other. Nitrogenous compounds are of four types:

- Adenine (A),
- Guanine (G),
- Cytosine(C),
- Thymine (T).

Thus a DNA molecule may have millions of nucleotides (sugar + phosphate + base) and can be represented as strings of alphabets A, G,C, T in any combination and of any length. The unit of DNA molecule is bp (base pair) and it represents the number of A, G, C, T pairs present in DNA molecules.

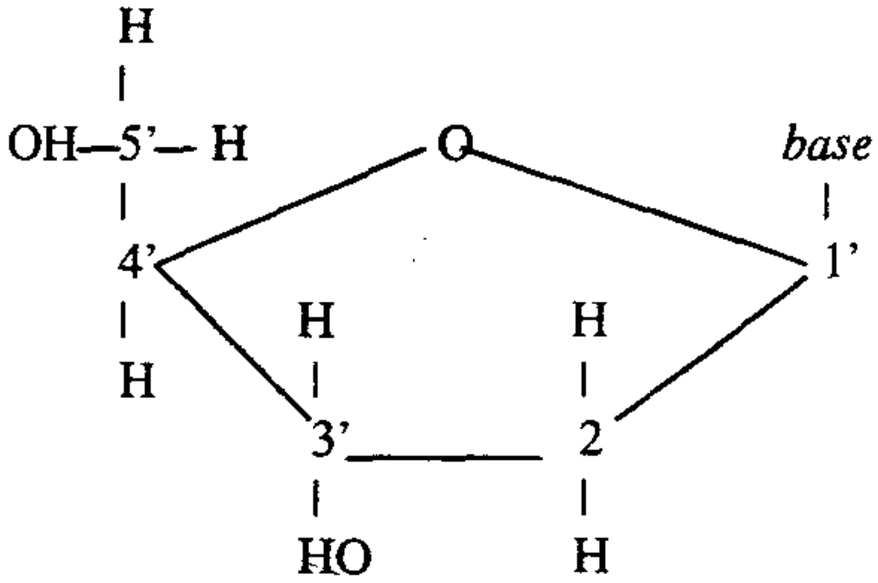
The other forms of nucleic acids are the RNAs. Figure1.3 (b) shows basic unit of RNA. The basic differences between a DNA and RNA are as follows:

- DNA is double strand (chain) of simpler molecules tied in helical structure. RNA is a single strand.
- The base that replaces T of DNA is Uracil (U). The sugar molecule as seen in RNA is Ribose instead of 2'- deoxyRibose (found in DNA). Thus we have RNA as strings of A, G, and C, U in any combination and of any length.

1.1.3 Promoters, Genes and the Genetic Codes

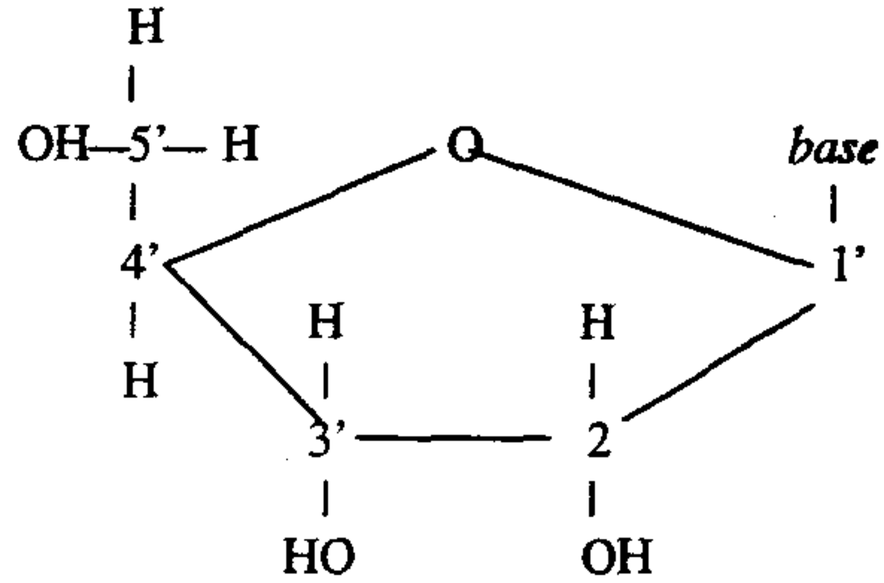
Now we discuss some more important terms in molecular biology: *promoter*, *genes* and the *genetic code*. A gene is a continuous stretch (*~10,1000* bp) of DNA that contains the necessary information to build a protein or RNA molecule. The portion of DNA, which marks the beginning of a gene, is called a promoter. The genetic code on the other hand gives the correspondence between each possible triplet of nucleotides (called CODON) and each amino acid in a tabular form. By this way of representation we can actually have $64(4^3)$ triplets representing *20* amino

acids. Thus some of these do not code for any amino acid. They are the STOP codons. These include UAG, UGA, and UAA. While some other codons may code for the same amino acid (signifying redundancy in the coding). For example, both AAG and AAA code for lysine. Also the codon AUG signifies the start of a gene. In other words, once a promoter site is identified in DNA, the first appearance of AUG in the sequence thereafter indicates the start of a gene.



2'-deoxyribose

Figure 1.3(a)



ribose

Figure 1.3(b)

Sugars present in nucleic acids. Symbols *1'* to *5'* represent carbon atoms.

1.1.4 Protein Synthesis

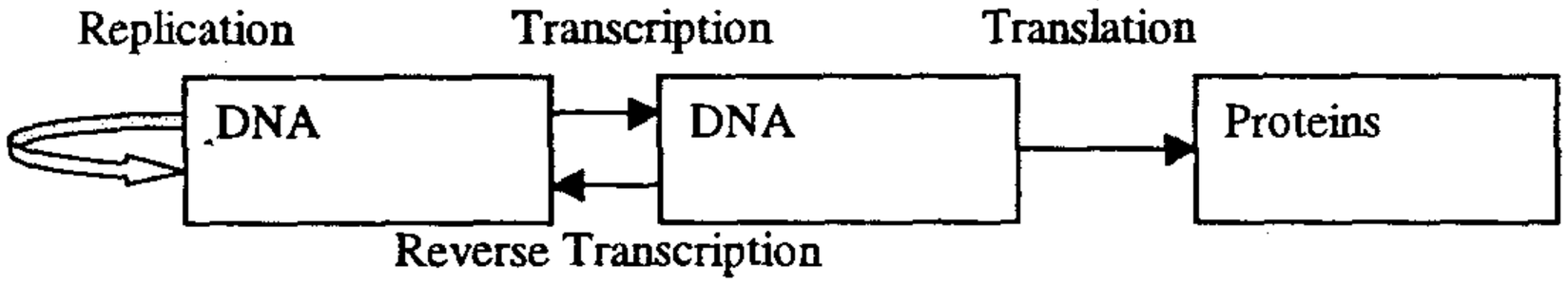


Figure 1.4 Genetic information flow in a cell

As mentioned in the last subsection we can recognize the start of a gene. Having recognized the beginning of gene or gene cluster, a copy of the gene is made on an RNA molecule. This resulting RNA is the messenger RNA, or *m*RNA for short. This process is called *transcription*. The *m*RNA will then be used in cellular structures called ribosome to manufacture a protein. In this process two kinds of RNA molecules play very important roles. Ribosome is made up of proteins and a form of RNA called ribosomal RNA. The ribosome functions like an assembly line in a factory using as "inputs" an *m*RNA molecule and another kind of RNA molecule called transfer RNA or *t*RNA.

Transfer RNAs are the molecules that actually implement the genetic code in a process called *translation*. They make the connection between a codon and the specific amino acid this codon codes for. Each *t*RNA molecule has, on one side, a conformation that has high affinity for

specific codon and, on the other side, a conformation that binds easily to the corresponding amino acid. As the messenger RNA passes through the interior of the ribosome, a *t*RNA matching the current codon- the codon in the *m*RNA currently inside the ribosome - binds to it, bringing along the corresponding amino acid (a generous supply of amino acids is always "floating around" in the cell). The three-dimensional position of all these molecules in this moment is such that, as the *t*RNA binds to its codon, its attached amino acids falls in place just next to the previous amino acid in the protein chain being formed. A suitable enzyme then catalyzes the addition of this current amino acid to the protein chain, releasing it from the *t*RNA. A protein is constructed residue by residue in this fashion. When STOP codon appears, no *t*RNA associates with it, and the synthesis ends. The messenger RNA is released and degraded by cell mechanisms into ribonucleotides, which will be then recycled to make other RNA. The process is explained in Figure1.4.

1.2 Sequence Databases

As already mentioned, proteins and DNAs are represented in form of sequences of alphabets. To work with these or to have basic understanding of the protein and DNA sequences that presently exist, one can easily refer to the large numbers of DNA, RNA, and protein databases that are available. The Internet is a useful tool to access theses databases and to harbor the biological sequences as well as the wealth of associated information. Some of the representative sequence databases of mentionable importance are:

- PIR (http://pir.georgetown.edu/)
- GenBank (http://www.ncbi.nlm.nih.gov/)
- EMBL (http://www.embl-heidelberg.de/)
- PDB (http://www.pdb.bnl.gov)

Throughout this project a large number of protein sequences obtained from the Protein Information Resource (PIR) Database [47] are used. This is a database maintained and distributed by three institutions: the National Biomedical Research Foundation (in the USA), the Martinstried Institute of Protein Sequences (in Europe), and the Japan International Protein Information Database. The data used in the experiments were obtained from the International Protein Sequence Database, release 75, in the Protein Information Resource (PIR) maintained by the National Biomedical Research Foundation (NBREF-PIR) at the Georgetown University Medical Center. The database currently has 172,684 sequences.

Three datasets are considered; they are the *Globin,* the Ras transforming proteins, and the *Trypsin homology* superfamilies. They form the positive datasets. They respectively contain *896*, *530* and *521* sequences in the database at present. We have taken *500* sequences from each of these superfamilies.

1.3 What is Bioinformatics

The recent flood of data from genome sequencing and functional genomic has given rise to new field, Bioinformatics [2], which combines elements of biology and computer science. Ever since the structure of DNA was unraveled in 1953, molecular biology has witnessed tremendous advances. With the increase in our ability to manipulate biomolecular sequences, a huge amount of data has been and is being generated. There is need to process the information that is pouring from laboratories all over the world, so that it can be of use for further scientific advances. This has created entirely new problems that are interdisciplinary in nature. Scientists from biological sciences are the creators and ultimate users of this data. However, due to sheer size and complexity, of these data the help of many other disciplines is required in biological sciences nowadays. These disciplines particularly include mathematics and computer science.

Bioinformatics is conceptualizing biology in terms of macromolecules (in the sense of physical-chemistry) and then applying "informatics" techniques (derived from disciplines such as applied mathematics, computer science, and statistics) to understand and organize the information associated with these molecules, on a large-scale.

Bioinformatics is a practical discipline. It employs a wide range of computational topics including sequence and structural alignment, database design and data mining, macromolecular geometry, phylogenetic tree construction, prediction of protein structure and function, gene finding, and expression data clustering. The emphasis is on approaches that integrate a variety of computational techniques and heterogeneous data sources.

1.3.1 Aims of Bioinformatics

In general, the aims of Bioinformatics are three-fold.

- First, at its simplest, Bioinformatics organizes data in a way that allows researchers to access existing information and to submit new entries as they are produced, e.g., the Protein Data Bank for 3D macromolecular structures. While data-creation is an essential task, the information stored in these databases is essentially useless until analyzed. Thus the purpose of Bioinformatics extends much further.

- The second aim is to develop tools and resources that aid in the analysis of data. For example, having sequenced a particular protein, it is of interest to compare it with previously characterized sequences. Sequence alignment [27] is one of the common approaches for matching or comparing. Typical examples of sequence alignment programs are FASTA [31, 32] and BLAST [19]. Another popular approach in this regard, is to extract appropriately defined features from protein sequences and then using some pattern recognition techniques for classifying them. Some attempts in this regard may be found in [2,5,16,18]. The present work primarily belongs to this category.

- The third aim is to use these tools to analyze the data and interpret the results in a biologically meaningful manner. Traditionally, biological studies examined individual systems in detail, and frequently compared them with a few that are related. In Bioinformatics, we can now conduct global analyses of all the available data with the aim of uncovering common principles that apply across many systems and highlight novel features.

1.3.2 Computer Science and Biology

Biological data are being produced at a phenomenal rate. For example presently, the GenBank repository of nucleic acid sequences contained 11,546,000 entries. On average, these databases are doubling in size every 15 months. In addition, since the publication of the H.influenza genome, complete sequences for nearly 300 organisms have been released, ranging from 450 genes to over 100,000. Add to this, the data from the myriads of related projects that study gene expression, determine the protein structures encoded by the genes, and detail how these products interact with one another. We can imagine the enormous quantity and variety of information that is being produced. As a result of this surge in data, computers have become indispensable to biological researches. Such an approach is ideal because of the ease with which computers can handle large quantities of data and probe the complex dynamics observed in nature.

The distinct subject areas we have mentioned require different types of informatics techniques. Briefly, for data organization, the first biological databases were simple flat files. However with the increasing amount of information, relational database methods with Web-page interfaces have become increasingly popular. In sequence analysis, techniques include string comparison methods such as text search and one-dimensional alignment algorithms. Motif and pattern identification for multiple sequences depend on machine learning, clustering and data-mining techniques. An example of this is the present work itself. *3D* structural analysis techniques include Euclidean geometry calculations combined with basic application of physical chemistry, graphical representations of surfaces and volumes, and structural comparison and *3D* matching methods. For molecular simulations, Newtonian mechanics, quantum mechanics, molecular mechanics and electrostatic calculations are applied. In many of these areas, the computational methods must be combined with good statistical analysis in order to provide an objective measure for the significance of the results.

1.3.3 Practical Applications of Bioinformatics

Here, we describe some of the major uses of Bioinformatics.

- ***Finding Homologues:***

As described earlier, one of the driving forces behind Bioinformatics is the search for similarities between different biomolecules. Apart from enabling systematic organization of data, identification of protein homologies has some direct practical uses. The most obvious is transferring information between related proteins. For example, given a poorly characterized protein, it is possible to search for homologies that are better understood and with caution, apply

some of the knowledge of the latter to the former. Specifically with structural data, theoretical models of proteins are usually based on experimentally solved structures of close homologies. Our work addresses this application.

- ***Rational Drug Design***

One of the earliest medical applications of Bioinformatics has been in aiding rational drug design. Given the nucleotide sequence of a gene as a drug target, the probable amino acid sequence of the encoded protein can be determined using translation software. Sequence search techniques can then be used to find homologies in model organisms, and based on sequence similarity; it is possible to model the structure of the protein on experimentally characterized structures. Finally, algorithms could be used to design molecules that could bind to the model structure, providing a way for biochemical assays to test their biological activity on the actual protein.

1.4 Conclusions and Scope of the Thesis

With the current deluge of data, computational methods have become indispensable to biological investigations. Originally developed for the analysis of biological sequences, Bioinformatics now encompasses a wide range of subject areas including structural biology, genomic and gene expression studies. As a result, Bioinformatics has not only provided greater depth to biological investigations, but added the dimension of breadth as well. In this way, we are able to examine individual systems in detail and also compare them with those that are related in order to uncover common principles that apply across many systems and highlight unusual features that are unique to some.

The present work can be thought of as a step towards the enormous journey in the world of understanding and organizing the information associated with biomolecules. There have been a lot of existing attempts in the same field. An overview of some of the existing works in this field has been provided in the next chapter.

The scope of the thesis is now briefly mentioned. In Chapter 2, we give an overview of the protein sequence classification problem in a biologically meaningful manner. Here we also briefly describe an existing scheme of feature selection as suggested by Wang et al. In Chapter 3 and Chapter 4 we describe the various methods of encoding that have been chosen in the protein classification problem in this work. In these chapters the problem has been approached in two ways. In the Chapter 3, the protein sequences are represented as signals and have been compared in the frequency domain. This way of looking at the sequences as signals has also been adopted in [28,29,50,51]. In the next chapter a new method for feature extraction from protein sequences in a completely different paradigm has been suggested. This method is based on the positional significance of the amino acids in the protein structures of a particular superfamily. It is a simple but effective method for finding out the relevance of all the possible amino acids in a protein sequence. The performances of the methods have been studied on various superfamilies in both these chapters. In Chapter 5 we have extended our classification technique to a multi-class classification problem. Throughout chapters 3, 4 and 5 we have used both *k*NN [33.52] and the MLP [17,30] as the underlying classifiers. Chapter 6 deals with conclusions and scopes for future work in the field of protein sequence classification.

Chapter 2

Protein Classification

It has been already discussed in the previous chapter that there has been a constant need for the development of an algorithm, which can extract useful information from biological databases. In response to this problem, a number of researchers have developed techniques to interpret the data and discover concepts in the DNA, RNA and protein databases. This field of biological datamining is very important in Bioinformatics. Classification of protein sequences into superfamilies [43] is one of the important areas of research in this field. In general, a superfamily is a group of proteins that share similarity in structure and function. This chapter deals with this particular problem in detail. In Section 2.1 a basic introduction to the problem of protein sequence classification is provided. Subsequently in Section 2.2, we present a study of the basic research works that are being undertaken in this area of protein sequence analysis all over the world. Finally in Section 2.3 the details of one of the recent important works of protein sequence classification problem has been provided. All these will provide us with an overview on how far scientific research has already been done in this field. Section 2.4 concludes the chapter.

2.1 Protein Classification Problem

The problem of protein classification can be formally stated as follows: Given an unlabeled protein sequence S and a known superfamiliy F, we are to determine whether the protein S belongs to the superfamily F or not.

In this section, a brief introduction of the related issues in the problem of protein sequence classification is given. To start with, it can be recalled that the protein signifies a string of alphabets from the set of 20 amino acids. The set (A) of amino acid can be represented by:

A={A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}.

The sequence can be of any length and the amino acids can combine in any order. This gives us an idea of the huge number of possible protein structures. First let us define what a superfamily [43] is. As observed, groups of proteins have similarity in functions and structures and we refer to a group of proteins that share such similarity as a *superfamily*. Thus we have divided all the presently available proteins into a number of superfamilies. An important issue in studying the protein sequences with the aid of several computer science or mathematical techniques in protein sequence classification is how to encode the protein sequences, i.e., how best to represent the protein sequences capable of mathematical manipulation. Suppose we have a protein classifier based on neural network. Note that, this type of classifier has been extensively used in the present dissertation. We know that good input representations make it easier for the network to recognize

the underlying regularities of the proteins. Proteins in form of normal alphabetic sequences are not a good way of input representation for any kind of classification problem. Thus a good input representation is crucial to the success of neural network learning. Similar is the requirement of a good input representation for other classifiers also.

The next question associated with protein sequence classification is how useful it is to biological researches. Protein family classification has several advantages. If we have an unknown protein at hand, the first thing we can do is to classify the protein into a known superfamily. This will help us to make some idea about the structure and function of the newly discovered protein. Perhaps, the most important practical application of such knowledge is in drug discovery. Suppose we have obtained sequence *S* from some disease *D* and by our classification method we infer that *S* belongs to *F*. In order to design a drug for the disease *D* we may try a combination of existing drugs for *F*. Other than its use in drug discovery, classification of proteins provides valuable clues to structure, activity, and metabolic role of the protein in question. Some of the other important uses of protein classification are listed below.

- It improves the identification of proteins that are difficult to characterize based on pair wise alignments;
- It assists database maintenance by promoting family-based propagation of annotation and making annotation errors apparent;
- It provides an effective means to retrieve relevant biological information from vast amounts of data;
- It reflects the underlying gene families, the analysis of which is essential for comparative genomic and phylogenetic; and
- Family and superfamily classification frequently provide identification or probable function assignment for uncharacterized (hypothetical) sequences.

Now we discuss the different classification systems have been developed to organize proteins in recent years. (E.g., in one system the classification into various superfamilies is done based on sequence similarities). Based on the type of classification, different databases have evolved (e.g., PIR Database). Scientists recognize the value of these independent approaches. While each of these databases is useful for a particular need, no classification scheme is by itself adequate for addressing all problems of biological needs.

Among the variety of classification schemes are:

- *Hierarchical families of proteins*, such as the super-families/families in the PIR-PSD [47], and protein groups in ProtoMap [48]
- *Families of protein domains*, such as those in Pfam [42] and ProDom [44].
- *Sequence motifs or conserved regions*, such as in PROSITE and PRINTS [39]
- *Structural classes*, such as in SCOP and CATH; as well as

- *Integrations of various family classifications,* such as iProClass [46] and InterPro [38].

The PIR superfamily/family [43. 47] concept, the original such classification based on sequence similarity, is unique in providing comprehensive and non-overlapping clustering of protein sequences into a hierarchical order to reflect their evolutionary relationships. Proteins are assigned to the same superfamily/ family only if they share end-to-end sequence similarity, including common domain architecture (i.e. the same number, order, and types of domains), and do not differ excessively in overall length (unless they are fragments or result from alternate splicing or initiators). Other major family databases are organized based on similarities of domain or motif regions alone, as in Pfam and PRINTS. There are also databases that consist of mixtures of domain families and families of whole proteins, such as SCOP and TIGRFAMs [45]. However, in all of these, the protein-to-family relationship is not necessarily one-to-one, as in PIR superfamily, but can also be one-to-many. Thus except for PIR superfamily concept classification, the belongingness of protein to a class can be expressed as a fuzzy membership value in that particular class.

Automatic classification of proteins into homogenous superfamilies, by looking at their amino acid sequences has long been a goal for scientists and researchers in the domain of Proteomics. In this aspect one important concept is how to extract high-level features from protein sequences. The best high-level features should be "relevant". By relevant we mean that there should be high mutual information between the features and the output of the classifier, where the mutual information measures the average reduction in uncertainty about the output of the classifier given the values of the features.

Following the completion of the human genome project, which has mapped out the location of every gene in the DNA sequence as well as discovering the function of each gene and how each gene effects our population, the last few years have witnessed consistent improvements in information retrieval, classification and analysis of the proteins and DNA sequences. Most of the advanced research areas of Bioinformatics rely on computational solutions for homologies modeling via sequence or structural similarities. Many statistical, sequence-base approaches have been developed for protein classification and homologies detection. Those methods are very time-consuming and complicated and so far they have only had partial success. They can give a clue to the protein's superfamily but not a definite answer. These include methods based on pair-wise similarity of sequences BLAST. [19], profiles for protein families, consensus patterns using motifs and Hidden Markov Models (HMM)[9,10,11]. Most of these methods attempt to recognize proteins at the super-families level, as they are not yet able to distinguish satisfactory characteristic differences between two sequences within the same super-family.

2.2 Approaches to Protein Classification

Sequence databases are databases comprising one dimensional data structures such as text, digital signals, proteins and DNA. Such objects are often represented as sequences in the databases. For example, a protein is represented as a sequence made from 20 amino acids, each represented as a letter. A digital signal is represented by series of 0s or 1s digits. A DNA is represented as a

sequence of four nucleotides: A, T, C and G as has been explained earlier. With the significant growth of sequence database sizes in recent years, it becomes increasingly important to develop new techniques for data organization and query processing in the sequence databases. Pattern Discovery and Pattern Matching are fundamental operations in the sequence databases. They attempt to discover useful patterns, which can help scientists to find new properties of the databases or predict the function of a new entity and to discover already similar patterns that exist in the database. These often help to classify the newly discovered entity to an appropriate pattern class.

- *Block based approach [18, 34,35]:*

In any basic algorithms for discovering *blocks* or characteristic patterns for a family of proteins the main strategy is as follows.

- The most highly conserved regions of a family of proteins can be represented as "blocks" of locally aligned sequence segments. Each block is considered as a special type of pattern for the protein family.
- If the query sequence belongs to a family with multiple blocks then at least a subset of these blocks should score highly when matching the query with the blocks.

An approach to query processing is a two-phase process:

- Find candidate segments among a small sample *A* of sequences.
- Combine the segments to form candidate patterns and evaluate the activity of the patterns in all sequences of *D* to determine which patterns are solutions of the query.

- *Sequence alignment based approach [19,31,32]:*

```
GA_CGGATTAG
GATCGGAATAG
```

Figure 2.1(a) Sequence alignment of 2 sequences.

```
MQPILLL
MLR_LL_
MK_ILLL
MPPVLIL
```

Figure 2.1(b) Multiple alignment of 4 amino acid sequences

In string matching we are given a sequence X and a string Y and we want to find all occurrences of Y in X. Many techniques have been published in the literature to solve the problem of pattern matching. A commonly used one is based on *multiple sequence alignment*, which is a natural generalization of the two-sequence case. We define an alignment between two sequences as the insertion of spaces in arbitrary locations along the sequences so that they end up with the same size. Having the same size the augmented sequences can be placed one over the other; creating a correspondence between characters or spaces in the first sequence and characters or spaces in the second sequence. If we consider Figure 2.1(a) we cannot help but notice that the two sequences

here actually look very much alike. In multiple sequence alignment let X_1, X_2, ... X_k be the set of sequences over the same set of alphabets. A multiple sequence alignment involving X_1, X_2, ... X_k is obtained by inserting spaces in the sequencesin such a way as to make all of them of the same size. Figure 2.1(b) shows such an alignment being done on four sequences. From the figure it is clear that the scheme gives us some idea about the similarity among these sequences. The basic aim in this is to design an efficient algorithm that takes two sequences and determine the best alignment between them to find out which of the sequences in the database are similar to a query sequence. Dynamic Programming, Tree Alignment and Star Alignment are various ways of creating multiple sequence alignment.

A natural question in all cases of Pattern Matching is whether there are related sequences that share the same pattern. The most widely used tools for sequence similarity search allow matching between arbitrary regions of the query and database sequences. In contrast, many *motif-based search methods* seek database sequences that match a pre-specified pattern. If this pattern is too weak, or not specified with sufficient precision, the number of matches may be very large, most being of no biological relevance. On the other hand, an overly specific pattern may exclude many sequences of interest. Protein families often are characterized by conserved sequence patterns or motifs. A researcher frequently wishes to evaluate the significance of a specific pattern within a protein, or to exploit knowledge of known motifs to aid the recognition of greatly diverged but homologous family members. To assist in these efforts, until recent days, BLAST and FASTA programs have been the major tools to help analyze the protein sequence data and interpret the results in biologically meaningful manner. BLAST returns a list of *high-scoring segment pairs* between the query sequence and the sequences in the databases. This is done after performing an alignment among them.

Faster and more sensitive homology searches based on BLAST [16, 37] have also been devised. One of them is the *pattern-hit initiated BLAST* (PHI-BLAST). In many instances, the program is able to detect statistically significant similarity between homologous proteins that are not recognized using traditional single-pass database search methods.

A method for using a multiple alignment is to identify an average structural "core," a subset of atoms with low structural variation [22]. In can be shown how the means and variances of core-atom positions summarize the commonalities and differences within a family. Thus, because of both the great numbers of structures and of families, it has become desirable (even necessary) to summarize the common features within a family, whilst separating out the variable ones. One of the most basic commonalities shared by each member of a family is a set of atoms, which occupy the same relative positions in space. The focus here is in identifying this set of atoms, and then in characterizing it statistically. How to construct an average core structure for a protein family in such a way that the average is unbiased and the resulting structure has acceptable stereochemistry is an important issue. This core structure can then be used to characterize the structural variability within a family, to define the average relative orientation of domains in multi-domain complexes, and to develop new measures of similarity between members of the same structural family. A core based purely on structural considerations is not the same as one based on sequence considerations, so, clearly, these definitions of "core" do not always coincide. Once a core for a family of structures has been calculated, it is possible to use it to assess the similarity of two structures in the family in a better way.

- *Suffix Tree based approach [20,36]*:

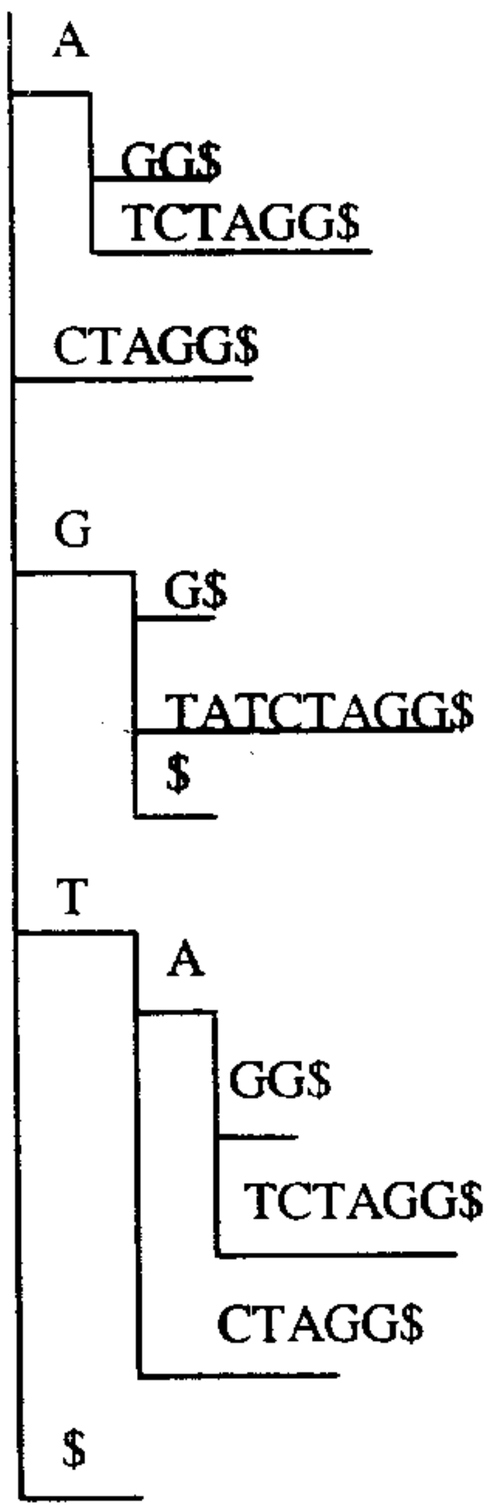


Figure 2.2 Suffix tree for the string GTATCTAGG. A dollar sign marks the end of the string.

This method of finding homologies groups all identical sub strings into a single path of the tree. Formally a suffix tree for string $S = s_1, s_2, \ldots, s_n$ is a rooted tree T with n+1 leaves with the following properties:

- Each edge of T are directed away from the root, and each edge is labeled by a substring from S
- All edges coming out of a given vertex have different labels, and all such labels have different prefixes.
- To each leaf there corresponds a suffix from S, and this suffix is obtained by concatenating all labels on all edges on the path from the root to the leaf.

The disadvantage of using a suffix tree is that it is expensive to store in a straightforward implementation. An example of a suffix tree is shown in Figure 2.2.

In another work the space of all protein sequences have been investigated. The standard measures of similarity (SW, Fasta, Blast), to associate with each sequence an exhaustive list of

neighboring sequences has been combined. These lists induce a (weighted directed) graph [21] whose vertices are the sequences. The weight of an edge connecting two sequences represents their degree of similarity. This graph encodes much of the fundamental properties of the sequence space. We look for clusters of related proteins in the graph. These clusters correspond to strongly connected sets of vertices. Edges between the vertices are weighted with weights that reflect the distance or the dissimilarity between the corresponding sequences, i.e., high similarity translates to a small weight (or distance). To compute the weight of the directed edge from A to B, one compares A against all sequences in the database, and obtains a distribution of scores. The graph is constructed using all currently known measures of similarity between protein sequences.

- *Position Specific matrices based approach*

Also we have Position-specific scoring matrices that have been used extensively to recognize highly conserved protein regions. A position-specific scoring matrix (PSSM) S represents a gapless local alignment of a sequence family. The alignment consists of several contiguous positions, each position represented by a column in the scoring matrix. Position-specific scoring matrices have been used extensively to recognize highly conserved protein regions. Intuitively, a higher segmental score indicates a greater likelihood that the sequence matches the given scoring matrix. The intuition behind scoring matrices is as follows. Amino acids that are abundant in a position in the alignment get high scores, and those that are rare get low scores.

The main novelty of this technique is the method of constructing feature vectors using Hidden Markov Model [8,9,10,11] and the combination of this representation with a classifier capable of learning in very sparse high-dimensional spaces. The system utilizes Support Vector Machines (SVM) classifiers to learn the boundaries between structural protein classes. First the protein sequences of interest are converted to high dimensional feature vectors by extracting them through HMM. Once this transformation has taken place, we then learn SVM discriminators to separate each protein family from the rest. The feature extraction method proposed in this dissertation, has to some extent, been motivated by the HMM based scheme described in [8,9,10].

Even with the best alignment of two sequences in hand the basic question remains. Do they share the same biological function or not. It is in general claimed that two sequences with over 30 percent identity are very likely to have the same fold by Sander Sneider 1991. Proteins of the same fold usually have similar biological functions. Nevertheless, one encounters many cases of high similarity in fold, despite a low sequence similarity. Such instances are unfortunately, missed by simple search over the database.

Other than the methods we have talked of briefly in this section, other methods based on HMM, neural network, multiple sequence alignment techniques are also in vogue. In one of the most recent works of flavor similar to what we work on, Wang et al [3,4] have tried to capture the global and local similarities of protein sequences as inputs to a Bayesian Neural Network (BNN) classifier. 2-gram encoding scheme, which extracts and counts the occurrences of two consecutive amino acids in a protein sequence, is used. They have also compared the technique with BLAST, SAM and other iterative techniques to prove the superiority of the proposed method. In this report we have compared the performance of our feature extraction techniques to the 2-gram scheme described in [3,4]. Therefore we described the method of Wang et al [3,4] in detail below.

2.3 Method of Wang et al

In this section the feature extraction technique as given by Wang et al [3,4] is described. The features extracted by the method proposed by them are given as inputs to a Bayesian Neural Network, which serves as the protein sequence classifier in [3,4].

They capture both the *global similarity* and the *local similarity* of protein sequences. The global similarity refers to the overall similarity among multiple sequences whereas the local similarity refers to the *motifs* (or frequently occurring sub strings) in the sequences.

2.3.1 Global Similarity of Protein Sequences

To calculate the global similarity of proteins sequences, they adopted the *2*-gram, also known as the *2*-tuple method. The *2*-gram encoding method extracts various patterns of two consecutive amino acid residues in a protein sequence and counts the number of occurrences of the extracted residue pairs. For instance, suppose we have a protein sequence *S* as PVKTNVK, the *2*-gram amino acid encoding method gives the following result: *1* for PV (indicating PV occurs once), *2* for VK (indicating VK occurs twice), *1* for KT, *1* for TN and *1* for NV.

They have also adopted *6*-letter exchange groups *{e1, e2, e3, e4, e5, e6}*, where e1∈ {H, R, K}, e2∈ *{D, E, N, Q}*, e3 ∈*{C}*, e4 ∈ *{S, T, P, A, G}*, e5 ∈*{M, I, L, V}*, e6∈*{F, Y, V}*. These exchange groups are effectively equivalence classes of the amino acids. For example, the above protein sequence PVKTNVK can be represented as *e4e5e1e4e2e5e1*. The 2-gram exchange group encoding of this sequence is 1 for *e4e5*, 2 for *e5e1*, 1 for *e1e4*, 1 for *e4e2* and 1 for *e2e5*.

For each protein sequence, Wang et al [3,4] have applied both the *2*-gram amino acid encoding and the *2*-gram exchange group encoding of the sequence. Thus there are $20 \times 20 + 6 \times 6 = 436$ possible *2*-gram patterns in total. If all the *436 2*-grams are chosen as the neural network input features, it would require many weight parameters and training data. This makes it difficult to train the neural network- a phenomenon called *"curse of dimensionality"*. Different methods have been proposed to solve the problem by careful feature selection and by scaling of the input dimensionality. They propose here to select relevant features (i.e., *2*-grams) by employing a distance measure to calculate the relevance of each feature.

Let X be a feature and let x be its value. Let $P(x|\ Class = 1)$ and $P(x|\ Class = 0)$ denote the class conditional density functions for the feature X, where *Class_1* represents the target class and *Class_0* is the non-target class. Let $D(X)$ denote the distance function between $P(x|\ Class = 1)$ and $P(x|\ Class = 0)$, defined as:

$$D(X) = \int |P(x|\ Class = 1) - P(x|\ Class = 0)|\ dx \qquad (2.1)$$

The distance measure prefers feature X to feature Y if $D(X) > D(Y)$. Intuitively, this means it is easier to distinguish more between *Class_1* and *Class_0* by observing feature X than feature Y. That is, X appears often in *Class_1* and seldom in *Class_0* or vice versa. In their work, each feature X is a 2-gram. Let c denote the occurrence number of the feature X in a sequence S. Let l denote the total number of 2-grams in S and let *len(S)* represent the length of S. Define the feature value x for the 2-gram X with respect to the sequence S as:

$$x = \textit{the occurrence number of feature X in sequence S} / (\textit{len(S)} - 1)$$

For example, suppose the feature is VK. The feature value of feature VK with respect to S would be $2/(7-1) = 0.33$.

Because a protein sequence may be short, random pairings can have a large effect on the result. $D(X)$ in equation (2.1) is approximated by

$$D(X) = (m_1 - m_0)^2 / (d_1^2 + d_0^2) \qquad (2.2)$$

where, m_1 and d_1 (m_0 and d_0 respectively) are the mean value and the standard deviation of the feature X in the positive (negative, respectively) training dataset. Intuitively, in equation 2.2 the larger the numerator is (or the smaller the denominator is), the larger is the interclass distance, and therefore the easier it is to separate *Class_1* from *Class_0* (and vice versa).

Let $X_1, X_2, \ldots, X_{Ng}$, $Ng << 436$, be the top Ng features with the largest $D(X)$ values. Note that these Ng features occur more frequently in the positive training dataset and less frequently in the negative training dataset. For each protein sequence S (whether it is unlabelled test sequence), they examine Ng feature values for the sequence S. These are input feature values to the Bayesian neural network classifier they have considered.

To compensate for the loss of information due to ignoring the other 2-gram patterns, a linear correlation coefficient *(LCC)* between the values of the *436* 2-gram patterns with respect to the protein sequence S and the mean value of the *436* 2-gram patterns in the positive training dataset is calculated and used as another input feature value for S. A last input is taken based on the local similarity of protein sequences, which refers to frequently occurring motifs in the target protein sequences.

2.3.2 Classification Methodologies

The set of features extracted using the above method basically represent a particular sequence. The next step in this process is to design a classifier that can appropriately classify the protein sequence into a particular superfamily using the extracted feature. In the present work two classifiers have been considered viz., the k nearest neighbors (kNN) classifier and the Multilayer Perceptron (MLP). They are now described in brief.

- *The kNN Classifier [33. 52]:*

Nearest neighbor classification rule can be defined as a rule, which assigns a pattern X of unknown class to the class of its nearest neighbor. Let us consider a set of sample patterns $\{P_1, P_2, \ldots P_t\}$ where it is assumed that each pattern belongs to one of the m existing classes. It is said that $P_i \in \{P_1, P_2, \ldots P_t\}$ is a nearest neighbor to X if

$$D(P_i, X) = min\{D(P_l, X)\}, \qquad l=1, 2, \ldots, t$$

where, D is any distance measure defined over the pattern space. We may call this scheme the one nearest neighbor (1-NN) rule. Therefore it employs only the classification of the nearest neighbor to X. The k nearest neighbor (kNN) rule consists of determining the k nearest neighbors to X and using the majority of equal class in this group as the class of X. In general, the choice of k is an important consideration in kNN classifier.

- *The MLP Classifier [30]:*

The multilayer perceptron consists of multiple layers of simple, two state, sigmoid processing elements (nodes) or neurons that interact using weighted connections. After a lowermost input layer there are usually any number of intermediate, or hidden, layers followed by an output layer at the top. There exist no interconnections within a layer while all neurons in a layer are fully connected to neurons in the adjacent layers. Weights measure the degree of correlation between the activity levels of the neurons that they connect. The total input, x_j^{h+1}, received by the neuron j in layer $h+1$ is defined as:

$$x_j^{h+1} = \Sigma y^h_i w^h_{ji}$$

where, y^h is the state of the i^{th} neuron in the preceding h^{th} layer, w^h_{ji} is the weight of the connection from the i^{th} neuron in the layer h to the j^{th} neuron in layer $h+1$. The output of a neuron in any layer other than the input layer $(h > 0)$ is a monotonic nonlinear function of its total input and is given as:

$$y^h = 1/(1 + exp(-x^h_j)).$$

For nodes in the input layer,

$$y_j^0 = x_j^0$$

An external input vector is supplied to the network by clamping it at the nodes of the input layer. For conventional classification problems, during training, the appropriate output node is clamped to state 1 while the others are clamped to state 0. This is the desired output supplied by the classifier. Here the learning procedure has to determine the internal parameters of the hidden units based on its knowledge of the inputs and the desired outputs.

In the present work the number of hidden nodes and the architecture of the neural net as well as the value of k in kNN were varied in the experimental phase. In using the MLP as the underlying classifier only a single hidden layer is sufficient for the purpose of protein sequence classification. This is because in this case the number of output nodes is always *2* indicating that only a single decision boundary is required to distinguish the target and the nontarget class. The

internal parameters of the MLP that include the learning rate and momentum factor were self-adjusted by the network for the best efficiency. The maximum permissible error was 0.01.

2.3.3 Experimental Results

A series of experiments to evaluate the performance of the proposed MLP and *k*NN classifiers were carried out on a Pentium III PC running the Linux operating system. Table 2.1 and 2.2 show the percentage classification obtained using MLP and *k*NN classifiers respectively. The number of sequences used for training and testing are specified in the tables. As in [3], the training as well as the test set contains 50% positive and 50% negative instances (where the negative instances consist of sequences from various other superfamilies). The MLP has 62 and 2 nodes in the input and output layers respectively, while the number of nodes in the hidden layer is varied. The number of iterations required for classification with MLP was 200 for all the superfamilies. For the *k*NN classifier, the value of k is taken to be *1*, *3* and $\sqrt{n}$, where n is the size of the training data set. It is known that as number of training patterns n goes to infinity, if the values of k and k/n can be made to approach infinity and *0* respectively, then the *k*NN classifier approaches the optimal Bayes classifier [52]. One such value of k for which the limiting conditions are satisfied is $\sqrt{n}$.

Table 2.1: The results by *2*-gram encoding (using MLP).
[Here the MLP has *3* layers with *62*, *30* and *2* nodes in the three layers respectively.]

Superfamily	# of patterns in training	# of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin	500	500	98.6	79.0
	250	250	98.0	71.0
Ras	500	500	99.8	81.0
	250	250	97.7	72.2
Trypsin	500	500	97.2	79.6
	250	250	98.0	69.4

Table 2.2: The results by *2*-gram encoding (using *k*NN classifier).

Superfamily	#of patterns in training (n)	#of patterns in testing	% Accuracy in testing		
			k=1	k=3	k=√n
Globin	500	500	86.4	80.6	76.4
	250	250	85.2	67.6	65.6
Ras	500	500	83.4	78.4	71.8
	250	250	73.2	67.2	66.4
Trypsin	500	500	88.4	80.8	76.4
	250	250	86.2	67.2	65.6

Table 2.3: **The results by *2*-gram encoding (using MLP). [Here the MLP has *3* layers with *62* and *2* nodes in the first and the last layers respectively. The number of nodes in the middle layer is varied]**

Superfamily	# of patterns used in training and testing	# of nodes in the hidden layer	% Accuracy in training	% Accuracy in testing
Globin	500	35	98.6	79.0
		15	98.4	78.6
Ras	500	35	98.4	66.0
		15	99.5	65.4
Trypsin	500	35	99.0	77.2
		15	98.2	77.0

In all above cases a total of *62* features are taken as inputs to the classifier, of which the *2*-gram patterns constitute *60* features. The other *2* inputs to the neural network or the *k*NN classifier include the LCC factor and the one based on local similarities. In Table 2.1 and 2.2 it is noticed that efficiency of this method tends to reduce significantly if the number of patterns used in training set is reduced. This is specifically seen in case of classification by the multilayer perceptron as well as for larger values of k in case of the *k*NN classifier. The *k*NN is seen to perform better than the MLP especially when $k = 1$. However as the value of k increases the performance of the *k*NN classifier is seen to degrade. This indicates that a proper choice of k is required for the classification with the *k*NN classifier. If the members of the superfamily are not very similar (as in the case of Ras) the classification is comparatively poorer. On comparing Table 2.1 and 2.3 it can be clearly seen that when the MLP classifier is used with *30* nodes the performance is better. From Tables 2.1 and 2.3, it is also evident that although the training scores are remarkably high, this is not the case during testing. This indicates a situation of *overfitting* of the data, a serious problem in MLP. This is the reason why the choice of a proper MLP architecture is of crucial importance.

After studying the method in [3,4], a number of reasons for lower efficiency in classification that may arise in some cases have been noted. They are listed as follows:

- This scheme ignores the spatial information of the occurrence of amino acid residues.
- Also the number of possible *2*-gram patterns is *436*, which is a large number. Out of a total of *436* we consider only *60* as inputs to the neural network or the *k*NN classifier. Going by the high reduction in dimensionality of the feature space it is not difficult to notice that there is obviously some amount of loss in information contained in the sequences.
- Training a classifier with so many inputs requires a large number of training samples, which may not be available for any superfamily that may be considered at random. It can

be checked out that in protein databases there are superfamilies, which consist of very few sequences only. In such cases this method might fail.

- It can also be imagined that it is a very time consuming method as the number of inputs included are many, the process of feature extraction is cumbersome and the number of sequences in the training set is also large. All these counts for the excessive time required for this method to work properly.

- Also it can be seen in Table 2.3 that MLP based classification using this scheme often gives very poor results. In this regard the superfamily Ras is particularly noticeable. One reason may be the crucial adjustment in weights and architecture required for the complex classification considered. Manual optimization of architecture is very difficult here. Thus one should go for optimization techniques like Genetic Algorithm or Simulated Annealing etc. It can also be noticed that overfitting is a serious drawback of this method when MLP is used as the underlying classifier. In Table 2.1 and 2.3 it can be seen than the training efficiency is very high as compared to the testing efficiency.

- In Table 2.3 the effect of changing the architecture of the network in the protein classification problem considered is shown. Comparing the results with those obtained in Table 2.1 it is evident that for better classification the number of nodes in the hidden layer should be very close to 30.

2.4 Conclusions

In this chapter the problem of protein sequence classification has been defined. Then an overview of some of the existing methods in this field is given. Lastly, the solution proposed by Wang et al [3,4] to the problem of protein sequence classification has been described in detail. Experimental results to study the method proposed by them have been provided. In the chapters to follow methods, which would overcome the problems that are mentioned in the previous section, have been proposed. The solution to this problem has been tried out in two different paradigms. Firstly in Chapter 3 the analysis of the sequences in the frequency domain has been considered. Secondly in Chapter 4, a scheme is proposed, which take into consideration the probabilities of occurrences of the amino acids residues in the sequences of a given superfamily in different locations; thereby incorporating some position specific information.

Chapter 3

Protein Classification in Frequency Domain

In this chapter concepts from signal analysis techniques [6] in transformed domain have been used to analyze the protein sequences. Examples of signals we encounter frequently are speech, music, picture and video signals. A signal is a function of independent variables such as time, distance, position, temperature and pressure. For example, speech and music signals represent pressure as a function of time at a point in space. In many cases these signals of interest are naturally discrete functions of independent variables. Often such signals are of finite durations. This type of finite extent signals, usually called time series, occurs in business, economics, social sciences, engineering and medicine. Generally biological sequences and protein sequences in particular maintain a periodicity in their structure [40, 41]. So the protein sequences can be represented as signals and studied with the help of existing methods of signal analysis in a transformed domain. In drawing an analogy of protein sequences with such existing concepts of time series signals the first and the foremost thing that comes to our mind is how to represent protein sequences as signals. This has been dealt with in the first section of this chapter. The next two sections deal with two transformations on such protein signals. They are the *Discrete Fourier Transformation (DFT)* and the *Discrete Wavelet Transformation (DWT).* These transformations convert the protein signals from original position domain to frequency domain and aids in better manipulation and analysis of protein sequences [49].

3.1 Signal Representation of Proteins

The first step that is required to study any sequence in the frequency domain is the proper representation of the sequence in form of a signal. The method that has been followed here for representing a protein sequence has been explained below.

- A 2- dimensional coordinate system is taken. The horizontal axis represents the positions of the amino acids in the protein sequence. The *0* of the horizontal axis represents the first position of the amino acid sequence and so on. The vertical axis is calibrated from *1* to *20* units. These represent *20* possible amino acids. Thus, *1* is allocated for A, *2* for C and so on, the numbers increasing in an alphabetic order. Any other way of ordering will also be a valid representation. Note that the amplitude of this signal varies from *1* to *20.*

- By this method if AACT is a protein sequence, it can be represented by points *(1,1), (2,1), (3,2)* and *(4,17)* in the 2-dimensional coordinate system.

- Now the points that are plotted in the above step are joined by means of a curve to get the original protein sequence in the form of a signal. The duration of the sequence is obviously the length of the sequence. Figure 3.1 gives the signal representation of a protein sequence following the method just described.

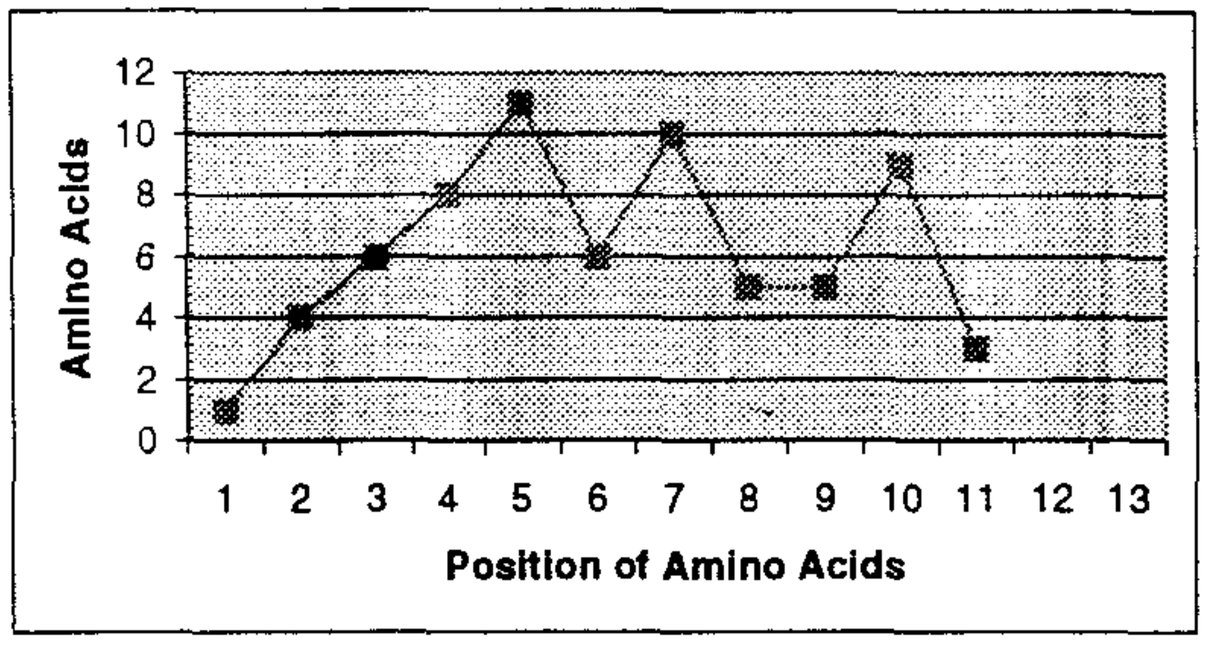


Figure 3.1 Signal representation of the protein sequence AEGIMGLFFKD by the pts *[(1,1), (2,4), (3,6), (4,8), (5,11), (6,6), (7,10), (8,5), (9,5), (10,9), (11,3).]*

After the signal representation of the protein sequence is complete analysis of the protein signal in the transformed domain [5,6] can be attempted. One such transformation is the *Orthonormal Transformation.* Orthonormal transforms form two classes:

(1) The data dependent ones like Karhunen-Loeve (K-L) transform, which need all the data signals to determine the transformation matrix and
(2) The data-independent ones like the DFT, Discrete Cosine Transform (DCT), Haar, or wavelet transform, where the transformation matrix is determined apriori.

The data dependent transforms can be fine-tuned to the specific data set, and therefore they can achieve better performance, concentrating the energy into fewer features in the feature vector. Their drawback is that, if the data set changes, a recomputation of the transformation matrix may be required to avoid performance degradation, requiring expensive data reorganization. Thus in the case of protein sequence classification, for different superfamilies different orthonormal transformations would be needed. Therefore, data-independent transforms are favored. For this two types of methods are used. In Section 3.2 DFT [5,28,29] has been used. To improve on this in the next section Wavelet transformation [7,50,51] is done. As a starting point the Haar Wavelets has been used in this thesis. The choice of other wavelets with improved transformation may be investigated in future.

3.2 Use of DFT for Feature Extraction

An amino acid series (or protein sequence) is a sequence of real numbers, each number representing a value at a position point. The method that has been adopted to get such an amino acid series is described in Section 3.1. The representation is similar to time series representation where the horizontal axis represents the time domain and the vertical axis maps a sequence of real numbers. Given two such time series $x = \{x_1, .. x_n\}$ and $y = \{y_1 ... y_n\}$, a standard approach to compare the sequences is to compute the Euclidean distance $D(x,y)$ between the sequences, where

$$D(x,y) = (\sum | x_i - y_i |^2)^{1/2}$$

If we say that the Euclidian distance is the deciding criteria of the dissimilarity between sequences of the target and the nontarget class, we would like a transform that preserves this distance, is easy to compute and concentrate the energy of the signal in few coefficients.

The distance preservation requirement is met by any *orthonormal transform*, DFT [28,29] being one of them. Among the existing orthonormal transformations, the DFT is chosen because it is the most well known, its code is readily available and it does a good job of concentrating the energy in the first few coefficients, in addition. The DFT has attractive properties among which the crucial ones can be listed as follows:

- For most sequences of our practical interest only the first few frequencies are strong in the frequency domain.
- According to the Parseval's theorem [5, 6], it is specified that the Fourier transform preserves the Euclidian distance in the time or frequency domain.
- The amplitude of the Fourier coefficients is invariable under shifts.

However, performance depends on a large number of false hits. By DFT sequences can be mapped to a lower dimensionality space. The most important thing to keep in mind here is that, the performance gain of this method in any time series data over other methods increases with increase in the number of sequences and the length of sequences.

From previous chapter it can be recalled that similarity queries related to protein sequence classification can be classified into two categories:

- *Whole matching*: All the sequences that are to be compared have the same length.
- *Subsequence Matching:* The query sequence is smaller; we look for a subsequence in the large sequence that best matches the query sequence.

In using the DFT, in the problem of protein sequence classification, we concentrate on whole matching of a sequence with the existing members of a class to classify an unknown protein. A given protein sequence is firstly translated to the frequency domain. Then only the first few frequencies are taken, dropping all other frequencies. This approach addresses two problems of feature extraction as follows:

- *Completeness of feature extraction*: Parseval's theorem, guarantees that the distance between two sequences in the frequency domain is the same as the distance between them in the time domain.
- *Dimensionality Curse*: A large family of interesting sequences exhibits strong amplitudes for the first few frequencies. Using the first few frequencies then avoids the dimensionality problem, while still introducing few false hits.

3.2.1 Discrete Fourier Transform

Here a brief overview of Discrete Fourier Transform (DFT) [5,6] is given. Note that the original signal in taken in the time domain for the sake of understanding the basic concepts of DFT. In the case of a protein signal (sequence) the time is analogous to positions of the amino acids of a protein. The importance of the DFT is the existence of a fast algorithm, *the Fast Fourier Transform (FFT)*, which can calculate the DFT coefficients in $O(n \log n)$ time.

The n point Discrete Fourier transform of a signal $x = [x_t]$, $t=0,.....,n-1$ is defined to be a sequence X of n complex $X_f s$, $f= 0, 1,.....n$, given by

$$X_f = 1/\sqrt{n} \sum x_t \exp(-j2\Pi ft/n) \qquad f=0,1,...n-1.$$

where, j is the imaginary unit $j=\sqrt{-1}$.

The signal x can be recovered by the inverse transform:

$$x_t = 1/\sqrt{n} \sum X_f \exp(j2\Pi ft/n) \qquad t=0,1,...n-1$$ or the various points of time.

X_f is a complex number (with the exception of X_0, which is real, if the signal x is real).

A fundamental observation for this way of representing signals is Parseval's theorem [5,6], which can be stated as follows:

Parseval's Theorem: Let X be the Discrete Fourier Transform of the sequence x. Then we have

$$\sum |x_t|^2 = \sum |X_f|^2$$

That is the energy in the time domain is the same as the energy in the frequency domain.

The Discrete Fourier Transform inherits the following properties from the continuous Fourier transform. Let '$\Leftrightarrow$' indicate Fourier pairs, i.e., $[x_t] \Leftrightarrow [X_f]$ means that $[X_f]$ is the Discrete Fourier Transform of $[x_t]$. The Discrete Fourier Transform is a linear transformation. If

$$[x_t] \Leftrightarrow [X_f] ; [y_t] \Leftrightarrow [Y_f]$$

then, $[x_t + y_t] \Leftrightarrow [X_f + Y_f]$ and $[ax_t] \Leftrightarrow [aX_f]$

Also, the shift in the time domain changes only the phase of the Fourier coefficients, but not the amplitude.

Given the above, Parseval's theorem gives

$$\| x\text{-}y \|^2 \equiv \| X\text{-}Y \|^2$$

This implies that Euclidean distance between the two signals x and y in the time domain is the same as their Euclidian distance in the frequency domain. It is believed that for a large number of time sequences of practical interest, there will be few frequencies with high amplitudes. Thus, if a sequence is characterized on the first few frequencies, there will be a few false hits.

The importance of Parseval's theorem is that it allows translating a query on a protein sequence from spatial domain to the frequency domain. Coupled with the conjecture that a few Fourier coefficients are enough, it allows us to build an effective feature space with low dimensionality.

When applying DFT to protein sequence classification the following resume of the procedure can be suggested:

- Obtain the coefficients of the Discrete Fourier Transform of every sequence in the target and the nontarget class that are considered in a protein sequence classification problem. After doing this some homogeneity in the coefficient values that characterize the sequences of the target class can be noticed.

- Represent each sequence as a point in the $2f_c$-dimensional space (recall that Fourier coefficients are complex numbers). Here fc can be < 5 to characterize the sequence fully. This is a well-established theory of DFT [5].

- These points are used as the features that are made inputs to the neural network or the *k*NN classifier for training.

- For a range of query, obtain the first *fc* Fourier coefficients of the query sequence. Use them to retrieve the matching sequences or the superfamily members that are at minimum distance away from the query sequence.

3.2.2 Experimental Results

In this section, MLP and *k*NN classifiers have been used to analyze the DFT based feature extraction strategy that has been discussed. The implementation parameters and the hardware environment is exactly alike to that used in Section 2.3.3 with a small difference in the MLP architecture only. Here *6* nodes are taken in the first layer corresponding to the first *3* complex conjugate Fourier coefficients (i.e., $fc = 3$). The middle layer consists of *3* nodes. The number of

iterations required for the classification of Globin and Ras superfamilies are *2000*, and *1600* respectively. The classification results obtained by MLP and kNN classifiers using the DFT based features are shown in Table 3.1 and Table 3.2 respectively. Note that the results corresponding to Trypsin are not included since no meaningful result was obtained in this case. Even for Globin and Ras, the results, as evident from Tables 2.1 and 2.2 and Table 3.1 and 3.2, are significantly poorer. This indicates that limiting attention to only the frequency component, while totally ignoring the positional information is insufficient to properly model the protein sequences. In order to overcome the limitations we next investigate the use of wavelets, which takesinto consideration the positional significance of the signal also in the frequency domain.

Table 3.1: The results by DFT encoding (using MLP).
[Here MLP has *3* layers with *6*, *3* and *2* nodes in the three layers respectively.]

Superfamily	#of patterns in training	#of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin	500	500	88.57	79.21
Ras	500	500	60.65	60.25

Table 3.2: The results by DFT encoding (using *k*NN classifier).

Superfamily	#of patterns in training(n)	#of patterns in testing	% Accuracy in testing		
			k=1	k=3	k=$\sqrt{n}$
Globin	500	500	73.5	73.2	67.7
Ras	500	500	75.25	66.3	62.2

3.3 Use of Wavelets for Feature Extraction

Though Discrete Fourier Transform (DFT) has been one of the most commonly used techniques, one problem with DFT is that it misses the important feature of position localization. *Piecewise Fourier Transform* has been proposed to mitigate this problem, but the size of the *pieces* becomes another critical issue. While large *pieces* reduce the power of multi-resolution, small *pieces* are unable to model low frequencies. In this section the use another type of orthonormal transformation viz., Discrete Wavelet Transformation, DWT [50,51], have been investigated in the problem of protein sequence classification. The advantage of using Discrete Wavelet Transformation (DWT) [7] is *multi-resolution* representation of signals. It has the *time-frequency localization* property. Thus, DWT is able to give localizations in both time and frequency. Therefore, wavelet representations of signals bear more information than that of DFT, in which only frequencies are considered. While DFT extracts the lower harmonics, which represent the general shape of a time sequence, DWT encodes a coarser resolution of the original time sequence with its preceding coefficients.

Wavelets are basis functions used in representing data or other functions. Wavelet algorithms process data at different scales or *resolutions* in contrast with DFT where only frequency components are considered. The origin of wavelets can be traced to the work of Karl Weierstrass in 1873. Following a trend in the disciplines of signal and image processing, in this work the use of the Haar wavelets in the problem of feature extraction for protein sequence classification has been advocated. We know the following:

- Euclidean distance is preserved in Haar transformation domain and no false dismissal will occur.

- Haar transform can outperform DFT. This has been confirmed through experiments.

Next a description of the Haar Wavelets is provided and it has been demonstrated how they can be used for the purpose of protein sequence classification.

3.3.1 Haar Wavelets

It can be mentioned that the Haar wavelet is chosen for the following reasons:

- It allows good approximation with a subset of coefficients

- It can be computed quickly and easily, requiring linear time in the length of the sequence and simple coding, and

- It preserves the Euclidean distance.

All these criteria have been confirmed in studies of times series signal analysis with Haar Wavelet Transformation.

A formal definition of Haar wavelets is given as:

$$\Psi^{i}_{i}(x) = \Psi(2^{j}x - i) \qquad i = 0, .., 2^{j} - 1$$

where

$$\Psi(t) = \begin{cases} 1 & 0 < t < 0.5 \\ -1 & 0.5 < t < 1 \\ 0 & otherwise \end{cases}$$

together with a scaling function

$$\Psi(t) = \begin{cases} 1 & 0 < t < 1 \\ 0 & otherwise \end{cases}$$

It is necessary to explain how to obtain the Haar transform of time series signal. In defining signals at the beginning of this chapter it has been mentioned that signals are nothing but functions of independent variables such as time. Say such a function is represented as $f(x)$. Haar transform can be seen as a series of averaging and differencing operations on a discrete time function. We compute the average and difference between every two adjacent values of $f(x)$. The procedure to find the Haar transform of a discrete function $f(x) = (9\ 7\ 3\ 5)$ is shown below.

Resolution	**Averages**	**Coefficients**
4	*(9 7 3 5)*	
2	*(8 4)*	*(1 -1)*
1	*(6)*	*(2)*

Resolution 4 is the full resolution of the discrete function $f(x)$. In resolution 2, *(8 4)* are obtained by taking the average of *(9 7)* and *(3 5)* at resolution 4 respectively. *(1 -1)* are the differences of *(9 7)* and *(3 5)* divided by two respectively. This process is continued until a resolution of 1 is reached. The Haar transform $H(f(x)) = (c\ d_0^0\ d_0^1\ d_1^1) = (6\ 2\ 1\ -1)$ is obtained which is composed of the last average value 6 and the coefficients found on the right most column, *2*, *1* and *-1*. It should be pointed out that c is the *overall average value* of the whole time sequence, which is equal to *(9+ 7 + 3+5)/4 =6*. Different resolutions can be obtained by adding difference values back to or subtract differences from averages. For instance, *(8 4) = (6+2 6-2)* where *6* and *2* are the first and second coefficient respectively. This process can be done recursively until the full resolution is reached.

Haar transform can be realized by a series of matrix multiplications as illustrated below:

$$\begin{pmatrix} x_0^1 \\ d_0^1 \\ x_1^1 \\ d_1^1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} X \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Envisioning the example input signal x is a column vector with length $n = 4$ and an intermediate transform vector is another column vector and Haar transform matrix H. The factor *1/2* associated with the Haar transform matrix can be varied according to different *normalization* conditions. After the first multiplication of x and H, half of the Haar transform coefficients can be found which are d_0^1 and d_1^1 in w interleaving with some intermediate coefficients x_0^1 and x_1^1. Actually d_0^1 and d_0^1 are the last two coefficients of Haar transform. x_0^1 and x_1^1 are then extracted from w and put into a new column vector $x^t = [x^1_0\ x^1_1\ 0\ 0\ ]$. x^t is treated as the new input vector for transformation. This process is done recursively until one element is left in x^t. In this particular case, c and d_0^0 can be found in the second iteration. The complexity of Haar transform can be evaluated by considering the number of operations involved in the recursion process.

By transforming the protein sequence into signal the protein sequence can be represented as a discrete function. This has been seen in the Section 3.1. Thus the Haar transformation coefficients of these functions are obtained using the method that has been just described. The following is a resume of the procedure:

- Obtain the coefficients of the Haar Wavelet Transform of every sequence in the target and the nontarget class that are considered in a protein sequence classification problem at the lowest resolution. After doing there should be some homogeneity in the coefficient values that characterize the sequences of the targetclass.

- Use these coefficient values as feature set of the MLP or the *k*NN classifier for training.

- For a range of query, obtain the Haar wavelet coefficients of the query sequence. Use them to retrieve the matching sequences or the superfamily members that are at minimum distance away from the query sequence.

3.3.2 DFT versus Haar Transform

Other than reasons that have been already discussed at the start of this section the motivation of using Haar transform to replace DFT is also based on several evidences and observations in areas of image and signal processing. Some of these are listed below.

- The first reason is on the pruning power. The nature of the Euclidean distance preserved by Haar transform and DFT are different. In DFT, comparison of two time sequences is based on their low frequency components, where most energy is presumed to be concentrated. On the other hand, the comparison of Haar coefficients is matching a gradually refined resolution of the two time sequences. From intuition, Euclidean distance can be highly related to low resolution of signal rather than low frequency components. This property can give rise to more effective pruning, i.e., fewer false alarms will appear.

- Another reason is the complexity consideration. The complexity of Haar transform is *O(n)* whilst *O(nlogn)* computation is required for Fast Fourier Transformation (FFT). Both impose restriction on the length of time sequences, which must be an integral power of 2. Note that, in the present case sequences are padded with *0*s at the end to make their lengths equal to the nearest integral multiple of 2.Although these computations are all involved in pre-processing stage, the complexity of the transformation can be a concern especially when the database is large.

3.3.3 Experimental Results

Similar to Section 2.3.3 the MLP and *k*NN under the same conditions are used as the underlying classification methodologies in classification of protein sequences based on features extracted using the Haar Wavelet Transformation. Table 3.3 and 3.4 shows the results.

The wavelets seem to outperform the DFT coefficients in almost all cases of protein sequence classification. In the case of *k*NN the classification efficiency increases with the increase in the value of *k*. The best classification is obtained by taking $k = \sqrt{500}$ *(≈23)*, where *500* is the number of sequences which has been used as the training data set. In this particular case, the percentage classification is greater than both the DFT as well as the method adopted in [3,4]. In using the MLP the number of nodes in the *3* layers are taken as *2, 2, 2* respectively. The number of iterations required to train the network were *1000, 8500* and *1000* respectively for Globin, Ras and Trypsin. It can be seen from Tables 2.1 and 3.3 that in case of Globin and Trypsin MLP based classification with this method is better than the scheme discussed in [3,4].

Table 3.3: The results by Wavelet encoding (using MLP).
[Here the MLP has *3* layers with *2, 2* and *2* nodes in the three layers respectively.]

Superfamily	#of patterns in training	#of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin	500	500	95.2	82.4
Ras	500	500	91.0	73.0
Trypsin	500	500	96.0	87.8

Table 3.4: The results by Wavelet encoding (using *k*NN classifier).

Superfamily	#of patterns in training(n)	#of patterns in testing	% Accuracy in testing		
			k=1	k=3	$k=\sqrt{n}$
Globin	500	500	73.2	74.4	82.8
Ras	500	500	82.6	85.2	84.4
Trypsin	500	500	73.8	74.0	78.8

3.4 Conclusions

In this chapter DFT has been used as a feature selection process for the protein sequence classification problem. The results obtained by using this transformation, shows limited amount of validity in few cases only. However, generally DFT transformation cannot be thought of as a good measure of feature selection for the case of protein classification [28,29]. The results obtained in many cases are confusing and totally irrelevant.

A better representation of sequences in the frequency domain by Wavelet transformation is thus tried out. As a starting step to feature extraction of protein sequences using wavelets the Haar Wavelet transformation has been used. This is the simplest form of wavelet that exists at present. It can be seen that the results clearly indicate that the feature extraction using wavelet transformation

has good capability in modeling the class characteristics of some superfamilies. This suggests that in future if instead of the simple Haar transform other wavelets transforms like the Daubechies and Coiflet [53] wavelets are used then the classification could be better.

In the chapter to follow a completely different paradigm in the field of feature extraction for the problem of protein sequence classification has been suggested. This chapter would discuss a method that does not involve any transformation to a different domain for the analysis of the protein sequences. Instead feature extraction for this method is done directly from the sequence in its original positional representation.

Chapter 4

An Improved Method of Protein Feature Extraction

In this chapter a method has been proposed, which suggests an improvement over all methods of feature selection that have been discussed till now. The superiority of this algorithm of feature extraction has been demonstrated by various experimental results. This is a hybrid method that combines the statistical as well as machine learning issues to derive the utilities or benefits of both these fields. The proposed technique [54] is explained in detail in the first section of this chapter followed by experimental results and discussions on some salient properties of this technique.

4.1 The Proposed Technique

LSALSDLHAHKLRVDPVN
LLALSDLHHHKLRVIMVN
LSALSALHHAKLRPIMVN
ASALSDAIAHMIRVDMVI

Figure 4.1 Primary structures of four related proteins

In this encoding scheme an attempt to take into account the evolutionary profile [15] information from multiple sequences belonging to a particular superfamily [45] has been made. A hybrid of statistical model and neural nets [17] or other pattern recognition techniques is explored to capture the same. Evolutionary similarity among proteins can be explained as follows. Figure 4.1 shows the primary structure of *4* related proteins. A small piece of each protein is shown. By taking a closer look at the structures, the history of evolution in this protein family can be perceived. Possibly, the ancestor of the *4* proteins in Figure 4.1 looked like the protein in Figure 4.2.

In general, proteins have evolved over time such that although a set of sequences share the same ancestor, structure differences become evident among them because of the evolutionary process. The differences arise due to some biological changes in form of *insertions*, *deletions* and *substitutions*. To understand why, consider what happens to a protein inside a cell when the cell reproduces. Through a process called *mitosis*, the cell makes a copy of it and then splits into two daughter cells. Most of the time a protein of the parent cell is exactly duplicated in the daughter

cell. However, over long periods of time, errors occur in this copying process. When this happens, a protein in the daughter cell becomes slightly different from the parent. It also happens that these proteins suffer similar degradation in their structure and thus a generality of structure is still maintained. As a result of these errors, proteins, which share a common ancestor, are not exactly alike. However, they inherit many similarities in primary structure from their ancestor. This is known as conservation of primary structure in a protein family. These structural similarities make one possible to create a statistical model of a protein family.

LSALSDLHIHKLRVDMVN

Figure 4.2 A possible common ancestor

4.1.1 Statistical Profile

Several statistical profiles already exist for the purpose of protein sequence analysis [9, 10, 11]. In the proposed method, in order to construct a statistical profile of a set of sequences, a *20 X lmax* probability matrix is computed where *lmax=maximum length of a sequence belonging to a particular superfamily.* Here the value at position *(i, j)* indicates the probability of occurrence of the i^{th} amino acid in position *j* of the sequence. To explain this method better, the following example is given.

Sequence*1* L A A T R
Sequence*2* L A H D V
Sequence*3* A L A D R
Sequence*4* L A A T R
Sequence*5* L L A D D

Figure 4.3(a) Five related proteins

Positions	*1*	*2*	*3*	*4*	*5.*
prob (L)	*.8*	*.6*	*.0*	*.0*	*.0*
prob (A)	*.2*	*.4*	*.8*	*.0*	*.0*
prob (H)	*.0*	*.0*	*.2*	*.0*	*.0*
prob (D)	*.0*	*.0*	*.0*	*.6*	*.2*
prob (T)	*.0*	*.0*	*.0*	*.4*	*.0*
prob (R)	*.0*	*.0*	*.0*	*.0*	*.6*
prob (V)	*.0*	*.0*	*.0*	*.0*	*.2*

Figure 4.3(b) A statistical model of the five related proteins that are shown in Figure 4.3(a)

The model shown in Figure 4.3 is a simplified statistical profile, a model that shows the amino acid probability distribution for each position in the family. According to this profile, the probability of L in position *1* is *0.8*; the probability of A in position *2* is *0.4*, and so forth. The probabilities are calculated from the observed frequencies of amino acids in the family of protein

sequences. For example, the value of the *(0,0)* th position of the matrix is *.8* because in first position L occurs in *4* out of a total of *5* sequences.

4.1.2 Network Feature Extraction

Given a profile, the position-specific weight of any amino acid in a given sequence can be obtained by adding the occurrences of the amino acid at a particular place and the respective probability of the occurrence of that amino acid in that place for the entire family.

For example, using this method given a sequence LAADT and the probability matrix that has been developed in Figure 4. 3(b), the weights of the individual amino acids are:

Weight (L) = *1 X 0.8* =*0.8.*
Weight (A) = *1 X 0.4+1 X 0.8* =*1.2*
Weight (D) = *1 X 0.6* = *0.6*

The weights of all other seventeen amino acids are zero. This is because either they appear in irrelevant positions with respect to the already known superfamily members, which form the training data set, or they do not appear at all.

For the classifier (*k*NN or MLP) the weights of all the *20* amino acids are taken as inputs. So for the sequence LAADT the feature vector is *[1.2 0 0.6 0 0 0 0 0 0 0.8 0 0 0 0 0 0 0 0 0 0]* representing the weights of features [A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y].

As is evident, the number of features for any sequence will always be equal to 20. In contrast to the method of Wang et al, which has been described in Chapter 2, where only *60* out of *436* features are selected, here the loss of information as a result of neglecting some possible input features is minimized. Also position specific information is incorporated in the scheme. Moreover the number of features, as compared to the scheme in that method [3,4] is reduced to *20* only.

4.2 Experimental Results

As in Section 2.3.3 we have tested the above-described method by using both MLP and *k*NN as the classifying methodologies. The results obtained are shown in the Tables 4.1 and 4.2. In this classification scheme we have used *3* layers for the MLP with 20, 12 and 2 layer in the 3 layers respectively. The number of iterations required while training the network with Globin, Ras and Trypsin superfamilies were *3000, 1000* and *1000* respectively when the number of members in the training data set was 500 in each.

The results here have shown a constant increase in the accuracy of classification as compared to all other methods that have been discussed till now. It can be concluded that the features extracted by this method are more relevant than any other method as far as protein sequence classification is concerned. This emphasizes the effectiveness of the proposed technique.

It can be noticed that the time required for training by this method is less as compared to the 2-gram encoding scheme. The main observations in general are as follows:

- Firstly, as expected, as the number of sequences in the dataset is increased, both the methods, i.e., the method of Wang et al and the suggested method in this chapter, show improved performance. Comparing Table 2.1 and 2.2 and Tables 4.1 and 4.2 verifies this. However it was observed that if the number of sequences in training phase was less than *200* the method proposed by Wang et al using MLP gives poor performance. This is because the MLP architecture becomes huge due to the large number of input features, involving a lot of adjustments in weights. This is true even for the *k*NN classifier.

- Secondly, it can be seen that if the nontarget class belongs to a particular superfamily (as in Tables 4.4(a) and 4.4(b) using the MLP and *k*NN classifier respectively), then the classification percentage is higher as compared to the case when the non-target class consists of a number of superfamilies (Tables 4.1 and 4.2 using the MLP and *k*NN classifier respectively). This is because in this case the sequences belonging to the non-target class become similar in the training and the testing phases of classification resulting in improved performance. This is noticed more prominently in the results shown in Chapter 5 in case of multi-class classification problem.

- Thirdly, different architectures may be required for classifiers classifying the different superfamilies to give the optimum results. This is because the feature spaces corresponding to the different superfamilies have different complexities. Therefore the difficulty in modeling the target class of the different superfamilies is different, necessitating different architectures. Even the number of iterations required in case of different target classes is different depending upon the complexity of the class.

Table 4.1: The results by the proposed method of encoding (using MLP)
[Here the MLP has *3* layers with *20, 12* and *2* nodes in the three layers respectively.]

Superfamily	# of patterns in training	# of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin	500	500	95.0	83.8
	250	250	96.4	80.0
Ras	500	500	92.8	89.0
	250	250	90.0	76.0
Trypsin	500	500	96.6	93.4
	250	250	96.4	84.4

Table 4.2: The results by the proposed method of encoding (using *k*NN classifier).

Superfamily	#of patterns in training (n)	#of patterns in testing	% Accuracy in testing		
			k=1	k=3	K≈√n
Globin	500	500	92.0	90.8	90.8
	250	250	85.2	86.0	82.8
Ras	500	500	89.0	90.2	89.8
	250	250	73.2	71.2	58.4
Trypsin	500	500	94.4	93.6	91.2
	250	250	88.4	88.4	75.6

Table 4.3(a): The results by the proposed method of encoding (using MLP).
[Here the MLP has *3* layers with *20*, *12* and *2* nodes in the three layers respectively. The training set consists of mostly similar sequences]

Superfamily	#of patterns in training	#of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin	500	500	93.8	66.6
Ras	500	500	89.4	86.4
Trypsin	500	500	90.4	79.0

Table 4.3(b): The results by the proposed method of encoding (using *k*NN classifier).
[The training set consists of mostly similar sequences]

Superfamily	#of patterns in training(n)	#of patterns in testing	% Accuracy in testing		
			k=1	k=3	k≈√n
Globin	500	500	67.2	68.8	69.4
Ras	500	500	79.8	80.6	83.0
Trypsin	500	500	91.0	90.6	88.4

Table 4.4(a): The results by the proposed method of encoding (using MLP).
[Here the MLP has *3* layers with *20*, *12* and *2* nodes in the three layers respectively. The nontarget class members belong to a single superfamily only].

Superfamilies (Target and Nontarget)	# of patterns in training	# of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin and Ras	500	500	96.8	85.4
Ras and Trypsin	500	500	95.8	83.8

Table 4.4(b): The results by the proposed method of encoding (using *k*NN).
[Here, the nontarget class members belong to a particular superfamily only].

Superfamilies (Target and Nontarget)	# of patterns in training(n)	# of patterns in testing	% Accuracy in testing		
			k=1	k=3	k=$\sqrt{n}$
Globin and Ras	500	500	93.8	94.8	94.8
Ras and Trypsin	500	500	90.2	89.2	85.4

Table 4.5: The results by our method of encoding (using MLP).
[Here the MLP has *3* layers with *20* and *2* nodes in the first and the last layers respectively. The number of nodes in the middle layer is varied]

Superfamily	# of patterns used in training and testing	# of nodes in the hidden layer	% Accuracy in training	% Accuracy in testing
Globin	500	17	96.2	83.0
		5	94.0	73.4
Ras	500	17	95.6	84.2
		5	95.8	84.4
Trypsin	500	17	96.6	89.6
		5	96.4	90.6

When MLP was used as the underlying classifier, one of the goals of this work has been to design networks that avoid over-fitting [13, 14, 15] as far as possible. By avoiding over-fitting, the learning and generalization errors stay almost identical, and therefore training can be continued until it reaches minimum training error. Since the noise in the training and the testing sets are

uncorrelated, the generalization ability on the testing set deteriorates at some point during training. The point at which this generalization ability deteriorates is highly dependent on the initial weights and the dynamics of the learning rule. Hence, it is almost impossible to determine at which point the training should be stopped in order to get an optimal solution. Early stopping is used, where the training is stopped after some fixed number of iterations or we can use a *validation set* to monitor the generalization ability of the network during training. When the performance on the validation set begins to deteriorate the training is stopped. However sacrificing data for the validation set can be crucial for the performance of the proposed model, since the available amount of data is limited in this case. Another method is to choose the network achieving best performance on the test set by always saving that network during training. The best approach of course is to deal with the root of the problem, namely, finding the proper complexity of the network.

- In all the networks that have been used in the proposed scheme less number of adjustable weights than those used by Wang et al's method have been used. Thus over-fitting is avoided to some extent. In Table 2.1 it can be seen that overfitting is a serious problem. In the case of the proposed method whenever it was found that an overfitting actually took place, early stopping was used, which means that training was stopped after the training error reached below some threshold.

- When using neural networks for protein sequence classification the choice of protein database is complicated by potential homology between proteins in the training and testing set. Homologous proteins in the training set can give misleading results since neural networks in some cases can memorize the training set. Tables 4.3(a) and 4.3(b) show such a result where the training set contains similar sequences, indicating poor representation of the data. Here the testing accuracy degrades. The change is more noticeable in cases of certain superfamilies like Globin. Table 4.6(a) and (b) show the confusion matrices that are generated using nonhomologous and homologous members in the training dataset respectively. On comparing these figures it can be clearly seen that when homologous members form a training dataset the number of wrongly classified target class members increases, thus reducing the overall performance or classification accuracy. Also in this case, the number of nontarget class members that are correctly classified increases.

- Furthermore, the size of the training and testing sets can have a considerable influence on the results as can be verified in Table 4.1. However much better classification power of this scheme can be seen as compared to Wang et al when the number of sequences in the training and the test set is only 250.

- In Table 4.5 the effect of changing the architecture of the network in the protein classification problem considered is shown. Comparing these results with those obtained in Table 4.1 it is evident that for better classification the number of nodes in the hidden layer should be very close to 12.

Table 4.6(a) The confusion matrix during testing (using MLP) corresponding to Globin family classification using nonhomologous members in training. Overall accuracy is 83.8%

Actual Class	Predicted Class		Percentage Classification
	Globin	Not Globin	
Globin	246	4	98.4
Not Globin	77	173	69.2

Table 4.6(b) The confusion matrix during testing (using MLP) corresponding to Globin family classification using homologous members in training. Overall accuracy is 66.6%

Actual Class	Predicted Class		Percentage Classification
	Globin	Not Globin	
Globin	91	159	36.4
Not Globin	8	242	96.8

4.3 Conclusions

This chapter exhaustively deals with a new feature extraction method to classify proteins from amino acid sequences in the original positional domain, which is seen to outperform one of the most successful existing schemes of protein sequence classification. This method, which has been suggested in this chapter, shows good classification results in all the cases. Intuitively, the success of this method lies in the way this method finds out the positional significance of the amino acids in protein sequence data. Also in this method there is a minimum loss of information of the entire sequence as compared to the other methods. Dimensionality of the feature space extracted by this method is quite manageable. A lot of variation in the classification of the protein sequences using this method can be considered. In this regard the following ways of classification can be thought of:

- Hybrid of this scheme along with other statistical or mathematical methods can be tried out.
- The incorporation of fuzziness might result in further improvement in classification
- The same method can be used to classify sequences when the sequence may belong to any of the m classes, m being greater than one.

The third view leads us to another aspect of the problem of protein sequence classification. This new arena is the problem of classifying a protein sequence to the proper class when more than one class are present. The multi-class classification problem has been dealt in detail in the next chapter.

Chapter 5

Multiclass Classification of Proteins

Whatever work has been done in the field of protein sequence classification has been mostly restricted to a target and nontarget class classification problem. It is basically a binary classification to decide whether a given sequence belongs to a particular superfamily or not. The same idea can be extended and collaborated with other techniques to tackle multi-class protein sequence classification. This chapter deals with one solution to the problem of multi-class classification problem. In Section 5.1 the problem is first defined and a possible solution to this problem is dealt in detail. Section 5.2 provides some experimental results using the multi-class classification method that has been suggested in the present chapter of the thesis. Section 5.3 concludes the chapter.

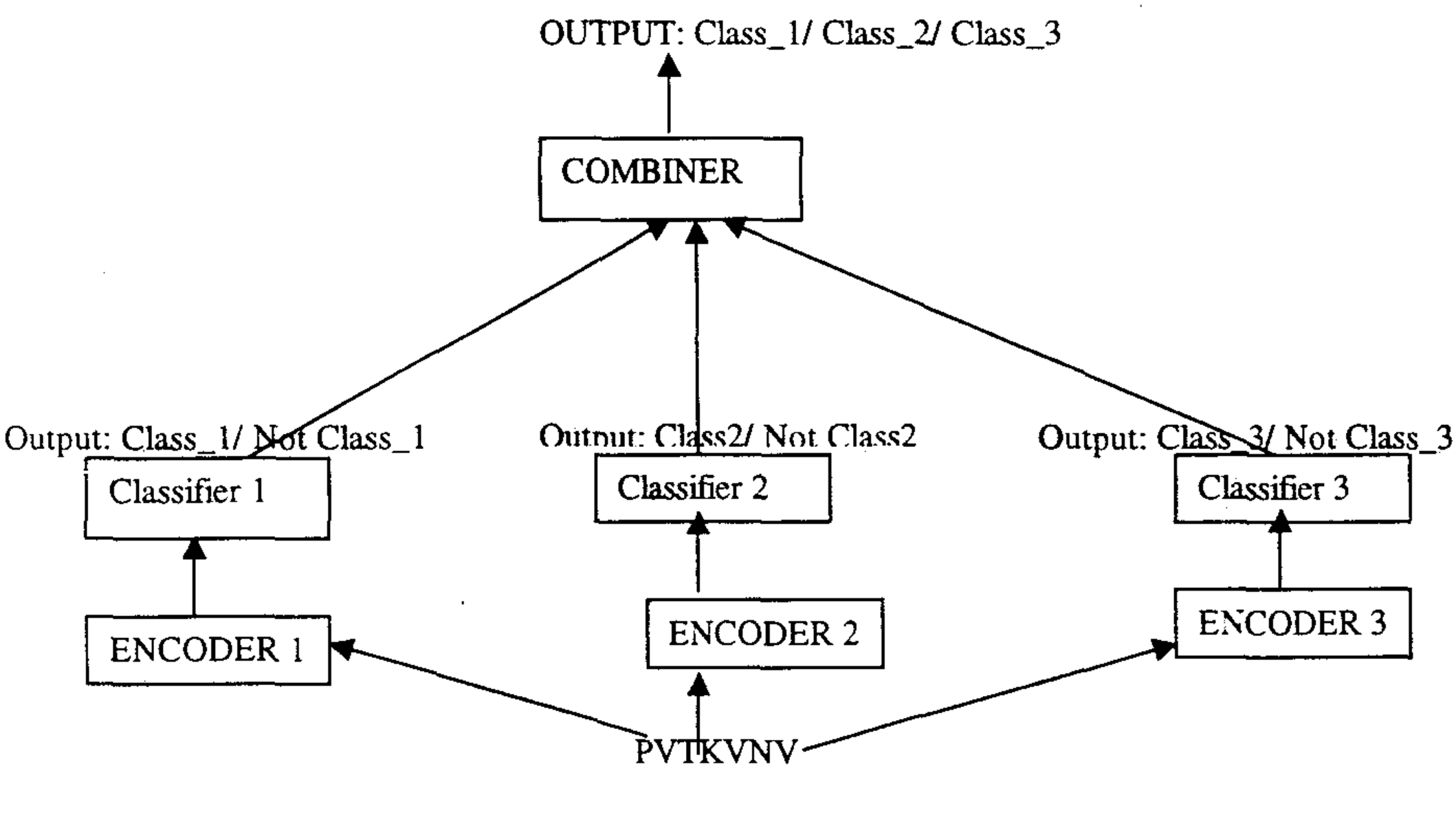


Figure 5.1: Multiclass Classification using a combination of networks assuming only 3 classes.

5.1 The Classification Method

In multi-class classification the problem can be defined as follows. Given an unlabeled protein sequence S and known superfamilies F_1, F_2,...F_m we are to determine to which of these superfamilies the protein S belongs to. Since, in this case, the target class is not fixed we are unable to model the characteristic features of a particular target class as have been done in the classification schemes considered in the preceding chapters. Here the characteristics of the m different target classes have to be modeled. Also, for each unknown sequence S we have to compare the sequence with each of these target class sequences and find out which superfamily sequences S most closely resembles. To do this an ensemble of classifiers have been used. The method that has been adopted is discussed in this section in detail.

In this thesis, a two-layered classifier [12] has been applied to solve this m class classification problem. They are as follows.

- In the first layer, individual classifiers are designed for the prediction of m classes. Note that $classifier_i$ is trained to distinguish between sequences that belong to class F_i and sequences that do not belong to class F_i. The initial goal has been to get as good accuracy of these single class predictions as possible. The method followed for this classification is same as the method of classification that has been described in the last chapter.

- MAXNET [30] has been used in the second layer for enhancement of the initial dominant response of the p^{th} classifier in the previous layer.

. Ensembles of single structure networks are thereby combined to obtain an m class prediction. Figure 5.1 pictorially depicts the entire procedure.

5.1.1 Single Class Classification Networks

Suppose there are m superfamilies: F_1, F_2,.... F_m. Then in the first layer m different classifiers for the m different classes are needed. However, the sequences used to train all the m classifiers are the same. The i^{th} classifier of this layer predicts whether a sequence belongs to the i^{th} class or not. While training the $classifier_i$ all sequences belonging to F_i are labeled as the target class and all sequences not belonging to F_i form the nontarget class. Similar labeling is followed for all the m-1 other classifiers of the first layer.

To note, for each sequence of protein in the dataset m predictions are obtained from the m separate classifiers of the first layer. It is a common assumption that any adaptive method of classification with some built in knowledge about the problem, performs better than the more general classifiers. The same model could have been used for the prediction of all classes. However, since the m superfamilies are different from each other it is possible that the performance can be enhanced if a separate classifier is specifically designed for each of the m superfamilies.

As shown in Figure 5.1 the single structure classifiers have only one output. For the classifier$_i$ if the output is larger than some decision threshold for some sequence *S* it says that the sequence *S* belongs to the i^{th} superfamily: else *S* does not belong to it. For an input/ output interval of *[0:1]* a decision threshold of *0.5* is optimal.

5.1.2 Combining the Single Class Predictions

Usually the classifiers of the first layer output *m* values, one for each of the *m* classes. This type of classification does not necessarily choose one of the *m* structures. For instance. it can (and sometimes does) classify one input pattern as a member of all the *m* classes i.e.. it gives large outputs on all *m* output units in the first layer. In practice, the input is classified as the structure giving the largest output.

We use the same concept to combine the output of the *m* classes. Shown in Figure 5.1 is a combiner. The combiner is nothing but a MAXNET, which takes the output from the *m* classifiers of the first layer for each of the sequences. It ultimately classifies the sequence to belong to that class for which the output of the classifier is the highest.

5.2 Experimental Results

The above-described method has been tested in a three-class (superfamilies) classification problem by using both MLP and *k*NN as the classifying methodologies. Note that it can be easily generalized for classification of protein sequences into more than three classes. The method of feature selection used in this scheme in its first layer is similar to the one described in Chapter 4. The classifying methodology as well as the environment that have bee used is similar to the ones that have been used in Section 4.2. The results are as tabulated in Table 5.1 and 5.2 below.

The first thing that is noticeable by just looking at the results in the above tables is that the multi-class classification scheme is more or less a success in cases where three separate superfamilies are treated as the three target classes. However in the third case we have taken the third target class as a mixture of members belonging to a number of classes other than Globin and Ras. It can be seen that in the third case the accuracy in classification is poorer. The utility of using some other function like (the Sigmoid) instead of taking the simple maximum (as done in MAXNET) in the output layer of the combined classifier for overcoming the limitation may be investigated in future.

Table 5.1: The results of multiclass classification (using MLP in first layer).

Superfamilies	# of patterns in training	# of patterns in testing	% Accuracy in training	% Accuracy in testing
Globin + Ras + Acetate	750	750	87.467	85.733
Globin + Ras + Trypsin	750	750	88.88	81.33
Globin + Ras + Other Classes	750	750	88.93	70.67

Table 5.2: The results of multiclass classification (using *k*NN classifier with k=1 in first layer).

Superfamilies	#of patterns in training	#of patterns in testing	% Accuracy in testing
Globin + Ras + Acetate	750	750	88.20
Globin + Ras +Trypsin	750	750	85.20
Globin + Ras + Other Classes	750	750	74.40

5.3 Conclusions

The multi-class classification that has been described in this chapter is a new way of looking into the protein sequence classification problem. In this thesis the matter has been dealt in its infancy and thus only the intuitive ways of extending a single class classification problem into a multi-class situation have been tried out. Though this way of approach gives good results there may exist better methods to tackle the same problem. Multi-class classification of proteins may be a promising field of study in the near future. With this hope, we move on to the concluding chapter of this dissertation. The following chapter would serve as an outline of whatever work has been done in the entire dissertation. It deals with a brief discussion on the scopes of future work in the same field as well as a recapitulation of the contributions of this work in the field of biological datamining.

Chapter 6

Conclusions and Scope of Future Work

This chapter deals with conclusions and a review on the further scopes that this work has in the field of Computational Molecular Biology. Section 6.1 deals with the main contributions of this thesis in the field of Bioinformatics. This is followed by a brief discussion on the scopes of future works in the same field.

6.1 Conclusions

The focus of this dissertation is protein sequence classification, which is an important area of research in the field of Biological Datamining. The main contributions of this thesis include the development of a method for extracting the position-specific similarity of sequences that are used as input features of the MLP and *k*NN classifiers. In the experimental studies we have compared the performance of the proposed classification scheme with another similar classifier, which has been described in [3,4]. It has been shown that the proposed way of classifying proteins is simple and at the same time leads to sufficient generality in classification. Thus the aim to classify protein sequences that are similar to but not identical to the patterns in the training set is satisfied. Finally, in the proposed scheme the number of training sequences required for the network to learn is not many. .Thus even if the superfamily based on which the classification is done is not large the classification does not suffer much. This has been a drawback of many other classification methods in the field of protein sequence classification. Further, the biological realities have motivated modern researches into new kinds of classification methods of protein sequences. Hybrids of several statistical models and neural nets are recently being explored. Our present work suggests the beginning of such a method for classification.

A mentionable study of proteins in the transformed domain has also been done in this dissertation. Lastly, we have proposed a solution to the multi-class classification problem of proteins. This is an important problem area in the field of protein sequence classification and can attract a lot of attention of future scientists.

6.2 Scopes of Future Works

This project has explored many exciting ways of feature extraction from protein sequences. In a part of the investigation we have represented and analyzed a protein sequences in the frequency domain. Though the experimental results that were obtained are not completely satisfactory but they open new scopes of phenomenal research areas. Future work could examine the following issues in this regard.

- Examination of other orthonormal transformation in addition to DFT and DWT.
- Betterment in the approach of Wavelet transformation by using more improved and stronger wavelets in the same fashion. For example, Daubechies and Coiflet [53] wavelets can be used instead of Haar.

In Chapter 4 a new method of modeling a superfamily has been suggested which gives better performance than one of the most successful recent methods in the area of protein sequence classification. However there are still lots of ways of improving on the suggested model. Some of them are as follows.

- One of the biggest limitations of such a modeling strategy is that this model is a linear one and is unable to capture higher order correlations among amino acids in a protein molecule. These correlations include hydrogen bonds between non-adjacent amino acids in a polypeptide chain, hydrogen bonds created between amino acids in multiple chains, and disulphide bridges, chemical bonds between C (cysteine) amino acids which are distant from each other within the molecule. In reality amino acids which are far apart in the linear chain may be physically close to each other when a protein folds. Chemical interaction between them cannot be predicted with a linear model. Thus with a proper understanding of the biological implications a betterment of this method is expected.

- Also biologically there are strong dependencies between the probabilities of occurrences of the various amino acids at the various positions. This has not been taken into consideration in the modeling strategies that have been discussed in this thesis. Incorporation of a suitable factor, which would take into consideration the transitional possibility of the amino acids at various positions, is expected to give better results. Thus biological researches can augment research in the field of protein sequence classification by leading to the development of improved statistical models.

- Hybrids of neural nets and various other methods, improvements by dynamic Bayesian nets, several graph matching algorithm in combination with this generalized classifiers can be tried out in future.

- Also improvement of the model with a multiple sequence alignment of the sequences prior to encoding could be a good choice.

- A very likely improvement can be achieved by optimizing the architecture of the network classifier by methods like genetic algorithm or simulated annealing etc.

- The length of the chain is an important point of consideration for the classification problem, which has been considered. Ways to incorporate that in some way or the other must be thought of. It may be noted that the scheme that has been discussed will be applicable if the difference in lengths of different sequences of the same superfamily are in the range of *25* to *50* residues. However, its effectiveness for cases where this difference is greater than that may be limited in view of the fact that positional information loses its significance if the sequences are widely varying in lengths. But

again, sequences belonging to the same superfamily are, in general, of similar characteristics, and hence we feel that this will not pose a serious problem.

Thus, compared with other methods, this system has several advantages including easy implementation, lesser time consumption in training and relative simplicity. The entire process does not require any knowledge about species origins, biological relations between sequences and so on. However non-involvement of biological information in the entire classification scheme is a major drawback of the suggested method

Although crisp (hard) classification has been used in this dissertation incorporation of fuzziness will result in further improvement of classification performance. This is so since the protein sequence in many existing protein databases, as we have discussed in Chapter 2. is likely to have belongingness to several classes with varied degrees of memberships. Thus incorporation of the principles of fuzzy set theory by using neuro-fuzzy classifier is another direction of further research. Last but not the least, use of more improved classification methodology in multi-class classification scheme is indeed a challenging task for future.

Bibliography

[1] J.Setubal and J.Meidanis, *An Introduction to Computational Molecular Biology,* An International Thomson Publication Company, 1997.

[2] N.M.Luscombe and D. Greenbaum, What is bioinformatics? *A proposed definition and overview of the field,* citeseer.nj.nec.com/453368.htm

[3] Jason T.L. Wang, Qi Cheng. Ma, Dennis Shasha and Cathy H.Wu, *Application of Neural Networks to Biological Data Mining: A Case Study in Protein Sequence Classification,* Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD 2000) Boston, Massachussetts, August 2000, pp. 305-309.

[4] Jason T.L. Wang, Qi Cheng. Ma, Dennis Shasha and Cathy H.Wu, *New Techniques for Extracting Features from Protein Sequences,* IBM Systems Journal, Special Issue on Deep Computing for the Life Sciences,Vol-40. no-2, 2001, pp. 426-441.

[5] R.Agrawal, C.Faloutsos and A.Swami. *Efficient Similarity Search In sequence Database.* .Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithm (FODO). Chicago Illionois, 1993, pp. 69-84.

[6] A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing,* Prentice Hall, Englewood Cliffs, N.J, 1975.

[7] K.Chan and A.W.Fu, *Efficient Time Series Matching by Wavelets,* ICDE, 1999, pp. 126-133.

[8] R.Karchin and R.Hughey, *Weighting hidden Markov models for maximum discrimination.* Bioinformatics, 14(9), 1998, pp.772-882.

[9] R.Hughey and A.Krogh. *Hidden Markov Models and sequence analysis: Extension and Analysis of the basic method,* Computer Applications in Biosciences 12(2), 1996, pp.95-107.

[10] David Hausler, Anders Krogh, I.Saira Mian and K. Sjolander, *Protein Modelling using Hidden Markov Models,* Analysis of Globins, Proceedings od the26th Hawaii International Conference on Systems, 1993.

[11] Christian Barrett, Richard Hughey and Kevin Karplus, *Scoring hidden Markov models,* CABIOS, Vol.13 (2), 1997, pp.191-199.

[12] S.K.Riis and Anders Krogh, *Improving Prediction of protein Secondary Structure using Structured Neural Networks and Multiple Sequece Alignments,* Journal of Computational Biology, Vol.3, 1996, pp.163-183.

[13] Rost, B.and Sander, C. *Improving prediction of protein secondary structure by use of sequence profiles and neural networks*. Proceedings of National Academy of Sciences of the United States of America, Vol 90(16), 1993, pp.7558-7562.

[14] Rost, B.and Sander, C. *Combining evolutionary information and neural networks to predict protein seccondary structure,* Proteins, Vol. 9,1994, pp.55-72.

[15] Maclin, R.and Shavlik, C., *Using knowledge-based neural networks to improve algorithms: Refining the Chou-Fasman algorithm for protein folding,* Machine Learning, Vol. 11, pp. 195-215.

[16] Bin Ma, John Tromp and Ming Li., *Pattern Hunter: faster and more sensitive homology search,* Bioinformatics, Vol.18 (3), 2002, pp. 440-445.

[17] P.Baldi and S.Brunak, *Bioinformatics- The machine learning approach,* MIT Press, 1998.

[18] G.Churn, *Pattern discovery in sequence databases*: *Algorithms and Applications to DNA/protein classification,* Department of Computer and Information Sciences, New jersey Institute of Technology, citeseer.nj.nec.com/chirn97pattern.htm, 1996.

[19] S.F. Aaltschul, W.Gish, W.Miller, E.Myers and D.J.Lipman, *Basic Local Alignment Search Tool, Journal of Molecular Biology,* Vol. 215, 1990, pp. 403-410.

[20] Dorohonceanu, B. and Nevill-Manning, *Accelerating Protein Classification using Suffix Trees,* In Proceedings of 8th International Conference on Intelligent Systems in Molecular Biology (ISMB), 2000, pp. 128-133.

[21] Nazaar Zaki Safaai, *Protein Sequence Classification Based on String Weighting Scheme,* citeseer.nj.nec.com/556028.html.

[22] M.Gerstein and R. B.altman, *Average Core Structure and Variability Measures for Protein Families, Application to Immunoglobins.* 1995

[23] J.D. Watson, Hopkins, K.Roberts, Steiz, and Weiner. *Molecular Biology of Gene, volume 1,* Redwooed city, CA: Benjamin/Cummins, 1987.

[24] J.D. Watson, Hopkins, K.Roberts, Steiz, and Weiner. *Molecular Biology of Gene, volume 2,* Redwooed city, CA: Benjamin/Cummins, 1987.

[25] R.F. Doolittle, *Proteins*, Scientific American, Vol. 253(4), 1985, pp. 74-83.

[26] I.Rosenfeld, E.Ziff, and V.van Loon, *DNA for Beginners*, Writers and Readers, 1984.

[27] Chang Zhang and Andrew K.C. Wong, *Towards Efficient Multiple Molecular Sequence Alignment: A System of Genetic Algorithm and Dynamic Programming*, IEEE Transaction on Systems, Man and Cybernatics, Vol.27 (6), 1997.

[28] A.J.Shephard, D.Gorse and J.M. Thornton, *A novel approach to the recognition of protein architecture from sequence using Fourier analysis and neural networks*, Proteins, Vol. 50(2), 2003, pp. 290-302.

[29] A.D.McLachlan , *Multichannel fourier analysis of patterns in protein sequences*, J.Phys.Chem. Vol .97, 1993, pp. 3000.

[30] Jacek M. Zurada, *Introduction to artificial Neural Systems*, West Publishing Company, 1994.

[31] D.J.Lipman and W.R.Pearson, *Rapid and Sensitive protein similarity search*, Science, Vol.227, 1985, pp. 1435-1441.

[32] W.R.Pearson, *Seaching protein sequence libraries: Comparison of sensitivity and selectivity of the Smith-Waterson and FASTA algorithms*, Genomics, Vol. 11, 1991, pp. 635-650.

[33] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*, Addison –Wesley Publishing Company, 1974.

[34] S. Heinkoff and J.G. Heinkoff, *Automated assembly of protein blocks for database searching*, *Nucleic Acids Research*, Vol. 19(23), 1991, pp. 6565-6572.

[35] S. Heinkoff and J.G. Heinkoff, *Protein family classification based on searching a database of blocks*, Genomics, Vol. 19, 1994, pp.97-107.

[36] D.Gusfield,. *Algorithms on Strings, Trees and sequences: Computer Science and Computaional Biology*. Cambridge, UK: Cambridge University Press, 1997.

[37] Altschul, S.F., et al.,. *Gapped BLAST and PSI-BLAST: A new generation of protein database search programs*. Nucleic Acids Research, Vol.. 25, 1997, pp. 3389-3402.

[38] Apweiler, R., et al., *The InterPro database, an integrated documentation resource for protein families, domains and functional sites*, Nucleic Acids Research. Vol. 29, 2001, pp. 37- 40.

[39] Attwood, T.K., et al., *PRINTS and PRINTS-S shed light on protein ancestor*, Nucleic Acids Research, Vol. 30, 2002, pp. 239-241.

[40] V.Ju.Makeev V.G.Tumanyan, *Search of periodicities in primary structure of biopolymers: a general Fourier approach*, Computer Applications in Biosciences, Vol. 12, 1995, pp.49.

[41] J.Heringa and P.Argos, *A method to recognize distant repeats in protein sequences*, Proteins. Vol. 17, 1993, pp.341.

[42] Bateman, A., et al., *The Pfam protein families database,* Nucleic Acids Research, Vol. 30, 2002, pp.276- 280.

[43] Dayhoff, M.O., *The origin and evolution of protein super-families*, Fed. Proceedings, Vol. 35, 1976, pp. 2132- 2138.

[44] Falquet, L., et al., *The PROSITE database*, Nucleic Acids Research, Vol. 30, 2002, pp.235- 238.

[45] Haft, D.H., et al., *TIGRFAMs: a protein family resource for the functional identification of proteins,* Nucleic Acids Research, Vol. 29, 2001, pp. 41- 43.

[46] Huang, H., Barker, W.C., Chen, Y., Wu, C.H., *iProClass: an integrated database of protein family, function, and structure information*, Nucleic Acids Research, Vol. 31, 2003.

[47] Wu, C.H. et al., *The Protein Information Resource,* Nucleic Acids Research, Vol. 31, 2003.

[48] Yona, G., Linial, N., Linial, M., *ProtoMap: automatic classification of protein sequences and hierarchy of protein families,* Nucleic Acids Research, Vol. 28, 2000, pp.49- 55.

[49] M.Akay, *Time Frequency and Wavelets in Biological Signal Processing*, IEEE Press, 1998.

[50] de Trad. CH, Fang Q. and Casic I, *Protein Sequence Comparisons based on Wavelet Transformation Approach*, Protein Engineering, Vol. 15(3), 2002, pp. 193-203.

[51] de Trad. CH, Fang Q. and Casic I, *An Overview of Protein sequence Comparison Using Wavelets*, http://www.eng.monash.edu.an/ieee/ieeebio2001/trad.pdf.

[52] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[53] John J. Benedetto and Michael W.Frazier, *Wavelets- Mathematics and Applications,* CRC, 1994.

[54] R.Chakraborty, S.Bandyopadhyay and U. Maulik, *Extracting Features for Protein Sequence Classification,* An International Conference on Information Technology: Prospects and Challenges (ITPC – 2003), May 23-26, 2003, Kathmandu, Nepal.