# PREDICTION OF PROTEIN SUBCELLULAR LOCALIZATION USING GLOBAL PROTEIN SEQUENCE FEATURE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

BURÇİN BOZKURT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

AUGUST 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Volkan Atalay
Supervisor

Examining Committee Members

Prof. Dr. Faruk Polat

Prof. Dr. Kemal Leblebicioğlu

Assoc. Prof. Dr. Volkan Atalay

Asst. Prof. Dr. Rengül Atalay

Dr. Özlen Konu

# ABSTRACT

PREDICTION OF PROTEIN SUBCELLULAR LOCALIZATION USING GLOBAL PROTEIN SEQUENCE FEATURE

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Volkan Atalay

August 2003, 70 pages

The problem of identifying genes in eukaryotic genomic sequences by computational methods has attracted considerable research attention in recent years. Many early approaches to the problem focused on prediction of individual functional elements and compositional properties of coding and non coding deoxyribonucleic acid (DNA) in entire eukaryotic gene structures. More recently, a number of approaches has been developed which integrate multiple types of information including structure, function and genetic properties of proteins. Knowledge of the structure of a protein is essential for describing and understanding its function. In addition, subcellular localization of a protein can be used to provide some amount of characterization of a protein. In this study, a method for the prediction of protein subcellular localization based on primary sequence data is described. Primary sequence data for a protein is based on amino acid sequence. The frequency value for each amino acid is computed in one given position. Assigned frequencies are used in a new encoding scheme that conserves biological information based on *point accepted mutations* ($PAM$) substitution matrix. This method can be used to predict the nuclear, the cytosolic sequences,

the mitochondrial targeting peptides (mTP) and the signal peptides (SP). For clustering purposes, other than well known traditional techniques, "principle component analysis (PCA)" and "self-organizing maps (SOM)" are used. For classification purposes, "support vector machines (SVM)" , a method of statistical learning theory recently introduced to bioinformatics is used. The aim of the combination of $feature\ extraction$, $clustering$ and $classification$ methods is to design an acccurate system that predicts the subcellular localization of proteins presented into the system. Our scheme for combining several methods is *cascading* or *serial combination* according to its architecture. In the cascading architecture, the output of a method serves as the input of the other model used.

Keywords: support vector machines, global protein subcellular localization, clustering, PAM250

# ÖZ

EVRENSEL PROTEİN DİZİ OZELLİĞİNİN KULLANILARAK PROTEİN HÜCRESEL SINIFLANDIRILMASININ TAHMİNİ

Bozkurt, Burçin

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Assoc. Prof. Dr. Volkan Atalay

Ağustos 2003, 70 sayfa

Hesaplanma yöntemleriyle çok hücreli gen dizinleri arasından gen tanıma, son yıllarda popüler bir araştırma konusu olmuştur. Bu konuyla ilgili olarak eski yaklaşımlar, özel fonksiyonel elemanlara, kodlanan ve kodlanamayan DNA'ya odaklanmıştır. Yeni yaklaşımlar birçok tipteki bilginin bütünleşmesini sağlamıştır. Herhangi bir proteinin yapı bilgisi, ilgili proteinin fonksiyonunu tanımlamada esastır. Ek olarak, bir proteinin hücresel bölgelendirilmesi, protein karakteri ile ilgili bilgi edinmede kullanılır. Bu çalışmada, proteinlerin temel dizi bilgisi kullanılarak hücresel bölgelendirilmesini sağlayan bir yöntem sunulmuştur. Bir proteinin temel dizi bilgisi, amino asit dizilimine dayanmaktadır. Bir proteinde bulunan her amino asitin frekansı hesaplanır ve hesaplanan deger amino asitle ilişkilendirilir. Bir protein için hesaplanan her değer, biyolojik bilgiyi koruma amaçlı olarak kullanılan kodlama şablonunda kullanılır. Kodlama şablonu Nokta Kabüllü Mutasyon matrisine dayanır. Bu çalışma, çekirdek, cytosolic, mitokondri ve signal peptid hücresel bölgelerini tanıma amaçlı tasarlanmıştır. Kümeleme yöntemi olarak Temel Bileşenler Analizi ve Özdüzenlemeli Harita kullanılmıştır.

Sınıflandırma yöntemi olarak Destek Vektör Makinaları kullanılmıştır. Özellik ayırma, kümeleme ve sınıflandırma yöntemlerinin birleştirilmesinin amacı protein hücresel sınıflandırılmasını doğru yapan bir tahmin sistemi tasarlamaktır. Üç adet yöntemi birleştiren bu tasarım seri birleşen mimarisine sahiptir. Seri birleşen mimarisinde, bir yöntemin sonucu diğer yönteme girdi olarak kullanılmaktadır.

Anahtar Kelimeler: destek vektör makinaları, global protein hücresel sınıflandırılması, kümeleme, PAM250

To my family

# ACKNOWLEDGMENTS

It is a great pleasure to have the opportunity to express my gratitude to all those who gave me the possibility to complete this thesis.

I would like to give my sincere thankfulness to my thesis supervisors, Assoc. Prof. Dr. Volkan ATALAY and Asst. Prof. Dr Rengul ATALAY, for their support and guidance throughout my study. Without the opportunities that they provided, I would not be able to come to this point. They have been a mentor for not only my research but also my life. I learn the joy of research from them.

I am also thankful to the faculty members and staff at the Dept. of Computer Engineering, METU, for their support especially for using Image Processing and Pattern Recognition Laboratory.

I should not forget all my teachers during my whole education, for exposing me to the beauty of learning.

My endless thanks are to my family, for being a constant source of strength during this endeavor. It is my family who makes my life so wonderful. Not only this thesis but everything in my life was possible because of them. It is not easy to put my thankfulness into words. My lovely mother Meryem, my lovely father Hüseyin, my lovely brother Burak and his lovely family (Emine and Zeynep) and my lovely sister Gülefşan, thanks a lot for your love, encouragement, patience. Thanks a lot for always being with me. Life would have no meaning without you.

My lovely father, it is not possible for you to read these lines. You have already gone. But, I work hard by the strength you gave and you made it

possible to finish my thesis. I miss you a lot ...

I hope my study could be used for cancer cell identification and individual medical treatments for lung cancer could be found in future. Perhaps then children do not lose their fathers due to lung cancer...

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation and Problem Definition

By the exposive growth in molecular biology, the genome sequences of human and other organisms are obtained. With the wealth of sequence-based genomic information, there appear two major paradigms for how biological computation is being leveraged [1, 2]. In the analysis paradigm, data, generally obtained from manual time-consuming experiments, are interpreted and analyzed in order to determine how an organism develops and functions. In the prediction paradigm, data sets with known characteristics are analyzed with the goal of learning models that are predictive of those characteristics, and then these models are applied to new biological data to predict derived properties. Analysis, made by the molecular biologists, has the advantage of often being very precise, but with the drawback of often requiring expensive and time-consuming experiments that occasionally fail to uncover the desired observation. Prediction has the advantage of being purely computational and thus relatively inexpensive, but often performs poorly, when the sought items are well outside the exemplars the computational method was trained on.

Subcellular localization is important for the function of proteins. To cooperate in a function, proteins must be in the proper subcellular localization, such

as a nucleus or a membrane. Thus, to predict the function of a protein, it is helpful to consider the subcellular localization site.

In this study, we have focused on the problem of identifying subcellular localization of protein sequences. The problem definition can be given as follows: given protein sequences compartmentalized at specific regions of the cell (nucleus, membrane, mitochondria, etc.), predict the localization of a protein based on the composition of its corresponding amino acid sequence. Subcellular localization information encoded in the amino acid sequence of a protein is intended to be extracted in the scope of the subcellular localization prediction problem.

This study is based on both analysis and prediction paradigms. In the scope of analysis, examination and interpretation of characteristics of the proteins encoded in human genomic sequences are to be done. In fact, we have not performed any examination because available data sets used in the TargetP [3] system are used in our study. In the scope of further improvements, our proposed system could possess the examination and interpretation of characteristics of the proteins. In the scope of prediction, the study of an semi-automatic and reliable prediction system for protein subcellular localization is performed. Molecular biologists intend to be freed from monotonous screening of large text collections and to be allowed to carry out more advanced analyses of the functionality of genome regions (proteins). There are many biological features that affect the expression of a protein. Subcellular localization of a protein is one of most important biological features. Proteins are located in functionally seperate membrane enclosed compartments in an eukaryotic cell. Each protein is active under its specific cell localization where it exerts its full function. Therefore, determination of subcellular localization of a newly identified protein is invaluable for understanding its function. Furthermore, determining the function has been the origin in the evolution of diseases as well as for the protein design with desired properties and protein engineering by altering an existing protein's function. New curing techniques could be developed in the light of the prediction of the subcellular localization of proteins. Cancer, as an example, could be given

as one of the most deadly kind of disease. Specifying therapies according to tumour types in cancer treatment could also be possible by studying subcellular localization.

## 1.2 Purpose and Improvements Achieved

In this work, we propose an approach to the prediction of protein subcellular localization, motivated by the identification of functions of proteins. This approach formalizes prediction of protein subcellular localization problem as follows: given protein sequences compartmentalized at specific regions of the cell (nuclear, cytosolic, mitochondrial, and signal peptides), classifying of proteins into four regions (classes) using global protein sequence feature.

The aim of this work is to design and develop a system called *prediction of protein subcellular localization* that predicts the subcellular localization of proteins in eukaryotic organisms based on the use of amino acid combination. The amino acid composition in the full or partial sequences can be taken as a global feature and the alignment order of amino acids can be taken as a local feature. In this study, we deal with the prediction of the subcellular localization based on amino acid combination using only global features.

Figure 1.1 presents the overview of the proposed system. The data set is separated into training and test data. First, training data are tokenized in order to transform each protein sequence segment into a column vector of scores calculated from PAM similarity matrix. Column vectors have dimension 400. In order to cope with the problem of excessive dimensionality, PCA approach is used to reduce the dimension from 400 to 20 by combining features. The most effective 20 features are obtained from the data set and used as inputs in the rest of the other methods applied. This reduction provides the extraction of valuable information necessary for the prediction and eliminates the information that does not change the outcome of our system drastically.

After the dimensionality is reduced from 400 to 20, an intermediate step is included in the proposed system. This step requires the natural grouping of

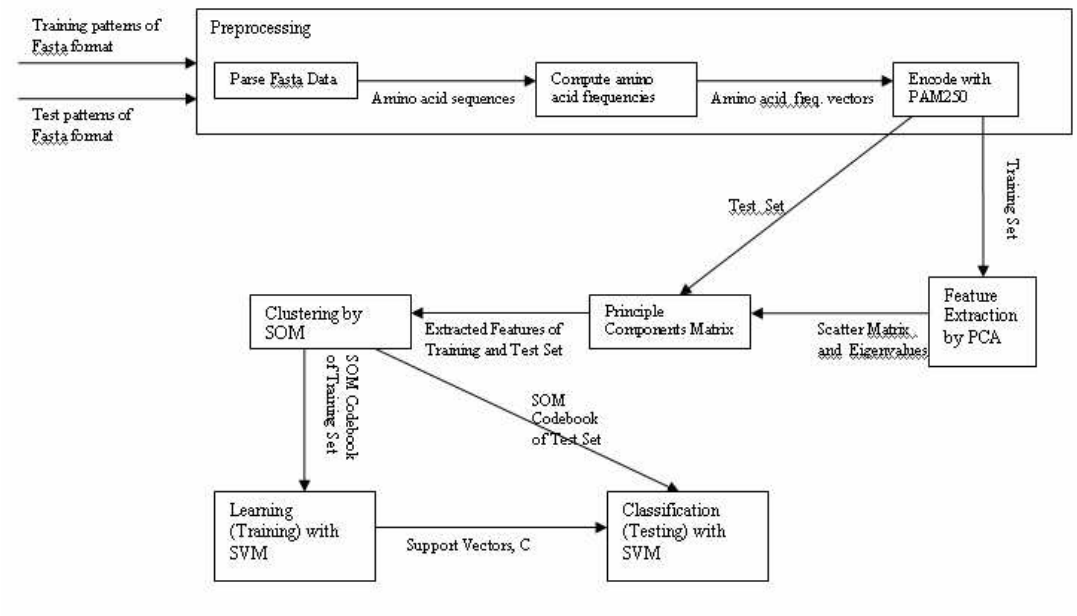


Figure 1.1: Flow diagram of the proposed system.

protein sequences with respect to their subcellular localizations. Natural groups are represented as "clusters" via Self Organizing Map (SOM). SOM is selected as a clustering method. The main reason of the selection is that the number of clusters are not known a priori. Those clusters are organized as a set of nodes in a hypothetical network with a simple neighborhood structure on the nodes. The neighborhood structure is used in the construction of the features on which the classification is applied. Each node is represented with reference vectors adjusted during clustering. For each protein, the reference vectors of the node that have the smallest difference among the vectors of other nodes in the network are taken as a feature for classification. The reference vectors applied to Support Vector Machine (SVM) increase the accuracy of the SVM classification.

After the formation of clusters, the classification method is used in the prediction of subcellular localizations. SVM is applied to determine the label of the proteins with unknown subcellular localization. SVM is chosen due to the fact that SVMs, considered as *supervised* computer learning method, offer many advantages over the *unsupervised* methods. In addition, the results of this study

on the power of SVM for classification and prediction are investigated. Only few parameters are to be chosen and we do not have to choose a complicated architecture as in neural networks.

The aim of the combination of $feature\ extraction$, $clustering$ and $classification$ methods is to design an acccurate system that predicts the subcellular localization of proteins presented into the system. Our scheme for combining several methods is *cascading* or *serial combination* according to its architecture. In the cascading architecture, the output of a method serves as the input of the other model used.

Our approach requires to identify the most significant amino acids for each protein (class) sequence by using Principle Component Anlaysis (PCA). This approach preserves as much of the relevant information for each protein as possible and provides discriminative information for each protein (class) sequence in clustering. This approach uses a clustering method to increase the accuracy of SVM classification. Features for classification are obtained from the reference vectors of the nodes that have the smallest difference among the reference vectors of other nodes in the SOM. The vectors applied to SVM increase the number of support vectors used in the construction. In SVM, outcome is interpreted as support vectors. As the number of support vectors increases, the accuracy of the classification increases. It is observed that the number of support vectors increases with the inclusion of clustering. In addition to these and more importantly, our subtle aim is to preserve biological information on the protein sequences. We describe the use of a new encoding scheme for the amino acids that conserves biological function. Each amino acid in a protein sequence segment is transformed into scores taken from $PAM$ substitution matrix. In the literature, the representation of an amino acid constitutes a conceptual difference from scoring methods like $PAM$, where only one scalar (generally binary) is assigned to one amino acid in one given position. By using $PAM$ substitution matrix in our study, biological and chemical differences of each amino acid in protein sequences are presented. Sequence alignment information is preserved

based on amino acids in each protein sequence.

## 1.3 Organization of the Thesis

The organization of the thesis is as follows. Chapter 2 presents biological and computational background of this study. The algorithms and computational aspects of computational methods used in the prediction of protein subcellular localization are discussed in detail in seperate sections. In Chapter 3, a literature survey of the related studies is given. In Chapter 4, implementation details of protein subcellular localization prediction are stated. Extensive experimental results obtained from proposed system are given and discussions based on this system are presented. The manuscript terminates with Chapter 5 in which the presented study is discussed and possible future directions are stated.

# CHAPTER 2

# BIOLOGICAL AND COMPUTATIONAL BACKGROUND

As a result of the Human Genome Project [4] and related efforts, DNA, gene and protein data accumulate at an accelarating rate. Mining these biological data to extract useful knowledge is essential in genome processing [5]. We present here an understanding of the biological background for the identification of the basic notions of molecular biology. In addition, we present the computational background for the identification of algorithms used in this study.

## 2.1 Biological Background

### 2.1.1 Historical Introduction

Genetics as a set of principles and analytical procedures did not begin until 1866, when Gregor Mendel performed a set of experiments that pointed to the existence of biological elements called *genes* - the basic units responsible for possession and passing on of a single characteristic. Until 1944, it was generally assumed that chromosomal proteins carry genetic information, and that DNA plays a secondary role. This view was shattered by Avery and McCarty who demonstrated that the molecule deoxy-ribonucleic acid (DNA) is the major carrier of genetic material in living organisms, i.e. is responsible for inheritance. In

1953 James Watson and Francis Crick deduced the three dimensional structure of DNA and immediately inferred its method of replication. It is now known that the DNA is the major carrier of genetic material in living organisms. In February 2001, due to a joint venture of the Human Genome Project and a commercial company, the first draft of the human genome was published.

### 2.1.2 DNA

#### 2.1.2.1 Composition

DNA is a macromolecule contained in the nucleus of every organism's cells. The basic elements of DNA had been isolated and determined by partly breaking up purified DNA. These studies demonstrated that DNA is composed of four basic molecules called nucleotides, which are identical except that each contains a different nitrogen base. Each nucleotide contains phosphate, sugar (of the deoxy-ribose type) and one of the four bases: *Adenine*, *Guanine*, *Cytosine*, and *Thymine* (usually denoted A,G,C,T) (See Figure 2.1)

#### 2.1.2.2 Structure

The structure of DNA is described as a *double helix* as shown in Figure 2.1, which looks rather like two interlocked bedsprings. Each helix is a chain of nucleotides held together by phospho-diester bonds. The two helices are held together by hydrogen bonds. Each base pairs consists of one *purine* base (A or G) and one *pyrimidine* base (C or T), paired according the following rule: G = C and A = T (each "-" symbolizes a hydrogen bond). The DNA molecule is directional, due to the asymmetrical structure of the sugars which constitute the skeleton of the molecule. Each sugar is connected to the strand *upstream* (i.e. preceding it in the chain) in its fifth carbon and to the strand *downstream* (i.e. following it in the chain) in its third carbon. Therefore, in biological jargon, the DNA strand goes from 5' ($fiveprime$) to 3' ($threeprime$). The directions of the two complementary DNA strands are reversed to one another.

#### 2.1.2.3 Replication

The double helix could be imagined as a zipper that unzips, starting at one end. We can see that if this zipper analogy is valid, the unwinding of the two strands will expose single bases on each strand. Because the pairing requirements imposed by the DNA structure are strict, each exposed base will pair only with its complementary base. Due to this base complementarity, each of the two single strands will act as a template and will begin to re-form a double helix identical to the one from which it was unzipped. The newly added nucleotides are assumed to come from a pool of free nucleotides that must be present in surrounding micro-environment within the cell. The replication reaction is catalyzed by the enzyme DNA *polymerase*. This enzyme can extend a chain, but cannot start a new one. Therefore, DNA synthesis must first be initiated with a *primer*, an oligonucleotide (a short nucleotide chain). The oligonucleotide generates a segment of duplex DNA that is then turned into a new strand by the replication process (See Figure 2.1).

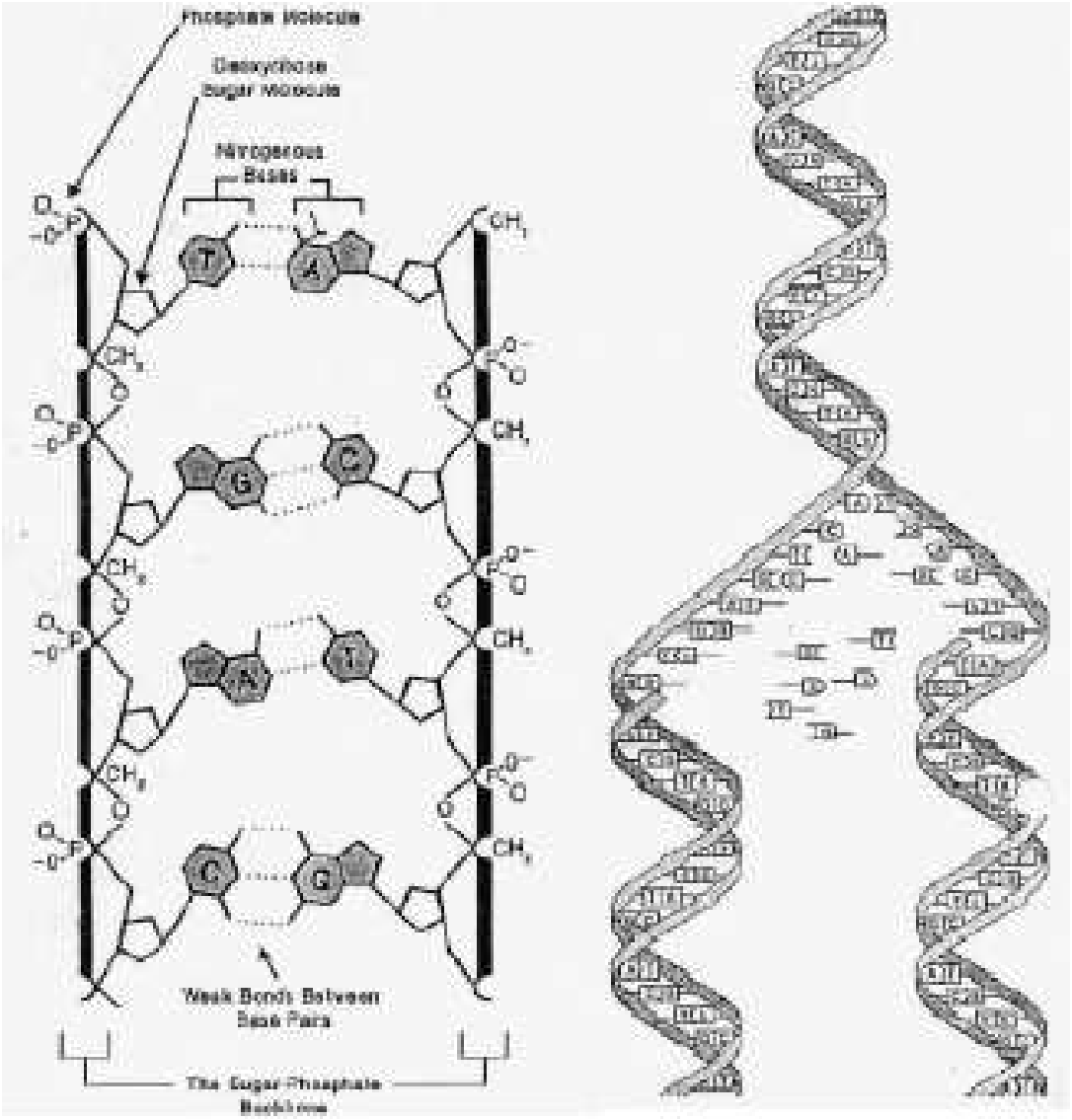


Figure 2.1: On the left, DNA composition. On the right, DNA double helix structure [6].

### 2.1.3 Genes and Chromosomes

The living organisms divide into two major groups: *prokaryotes*, which are single-celled organisms with no cell nucleus, and *eukaryotes*, which are higher level organisms, and their cells have nuclei. Eukariotic cell nuclei contain *chromosomes* - the contiguous structures in which DNA is stored. Each DNA molecule is pack-

aged in a separate chromosome, and the total genetic information stored in the chromosomes of an organism is said to constitute its *genome*. With few exceptions, every cell of a Eukaryotic multi-cellular organism contains a complete set of the genome, while the difference in functionality of cells from different tissues is due the variable expression of the corresponding genes. The human genome contains about $\mathbf{3x10}^9$ base pairs (abbreviated *bp*), organized as 46 chromosomes - 22 different autosomal chromosome pairs, and two sex chromosomes: either XX or XY. The 24 different chromosomes range from $\mathbf{50x10}^6$ to $\mathbf{250x10}^6$ bp. The amount of DNA varies between different organisms. The organism *Amoeba dubia* (a single cell organism), for example, has more than 200 times DNA as human. The living organisms divide into two major groups: *Prokaryotes*, which are single-celled organisms with no cell nucleus, and *Eukaryotes*, which are higher level organisms, and their cells have nuclei. With contemporary knowledge of the biochemical basis of heredity, Mendels abstract concept of a gene can be redefined as a physical entity. A *gene* is a region of DNA that controls a discrete hereditary characteristic, usually corresponding to a single mRNA carrying the information for constructing a protein. In 1977 molecular biologists discovered that most Eukariotic genes have their coding sequences, called *exons*, interrupted by non-coding sequences called *introns*, (See Figure 2.2). In humans genes constitute approximately 2-3% of the DNA, leaving 97-98% of non-genic *junk* DNA. The role of the latter is as yet unknown, however experiments involving removal of these parts proved to be lethal. Several theories have been suggested, such as physically fixing the DNA in its compressed position, preserving old genetic data, etc.

### 2.1.4 RNA

Cells have a second type of nucleic acid - RNA (*Ribonucleic Acid*) which can also carry genetic information. Unlike DNA, which is located primarily in the nucleus, RNA can also be found in the cell's cytoplasm. Like DNA, RNA is also built from purine and pyrimidine nucleotides (*Uracil* taking the place of

Thymine), but forms a single helix (unlike the DNA's double helix).

The messenger RNA ($mRNA$) carries genetic information from the DNA to the *ribosomes* - the intra-cellular constructs where it is translated into a protein. mRNA is synthesized in the nucleus based on a single DNA strand, using the RNA *polymerase* enzyme. mRNA is *transcribed* from a DNA strand only in locations called *open reading frames*. When such transcription occurs, the two DNA strands are split apart, and one of them serves as a mold for the generated mRNA molecule, which is complementary to this strand, and therefore, a replica of the other one. Consecutive triplets of mRNA bases, called *codons*, each determine a certain amino acid as shown in Figure 2.2. In eukaryotes, the mRNA is formed of *coding* and $non - coding$ regions. Coding regions are the regions used to carry real genetic information. Non coding regions do not carry such information.

The coding regions are called *exons*, since they are able to leave the nucleus and reach the Ribosome. The non-coding regions are called *introns* and never leave the nucleus. After being synthesized from a DNA strand, the exons of the RNA molecule are merged together, exiting the nucleus, heading for a ribosome.

The $transfer\ RNA$ ($tRNA$) is a small RNA molecule serving as an adapter between mRNA and amino acids. The molecule is composed of two parts. On one part the tRNA holds an *anticodon*. The anticodon is a sequence of three RNA bases. On the other side, the tRNA holds an amino acid. The many-to-one mapping from anticodons to amino acids defined by tRNA molecules is the *universal genetic code*.

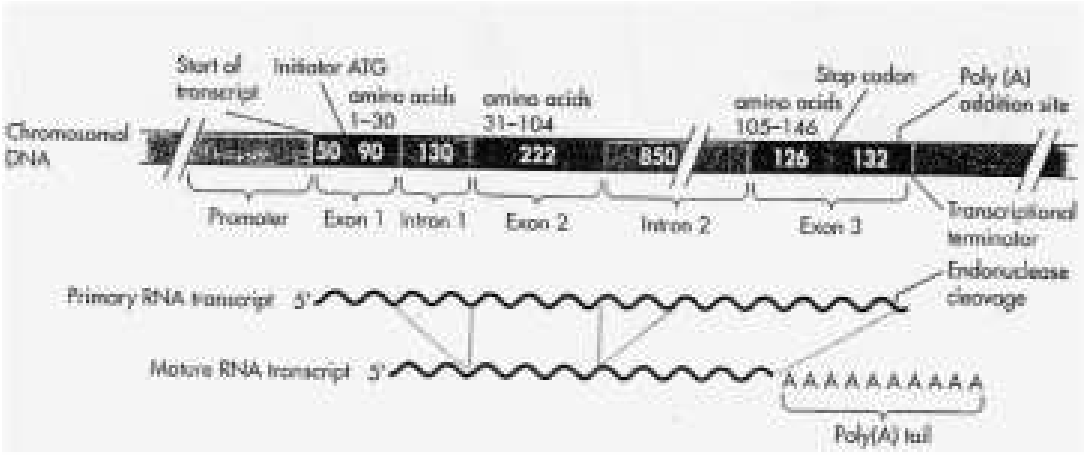


Figure 2.2: Exons and Introns [6].

### 2.1.5 The Central Dogma

The expression of the genetic information stored in DNA involves the translation of a linear sequence of nucleotides into a co-linear sequence of amino acids in proteins. The flow is: DNA → mRNA → Protein. It is shown in Figure 2.3.

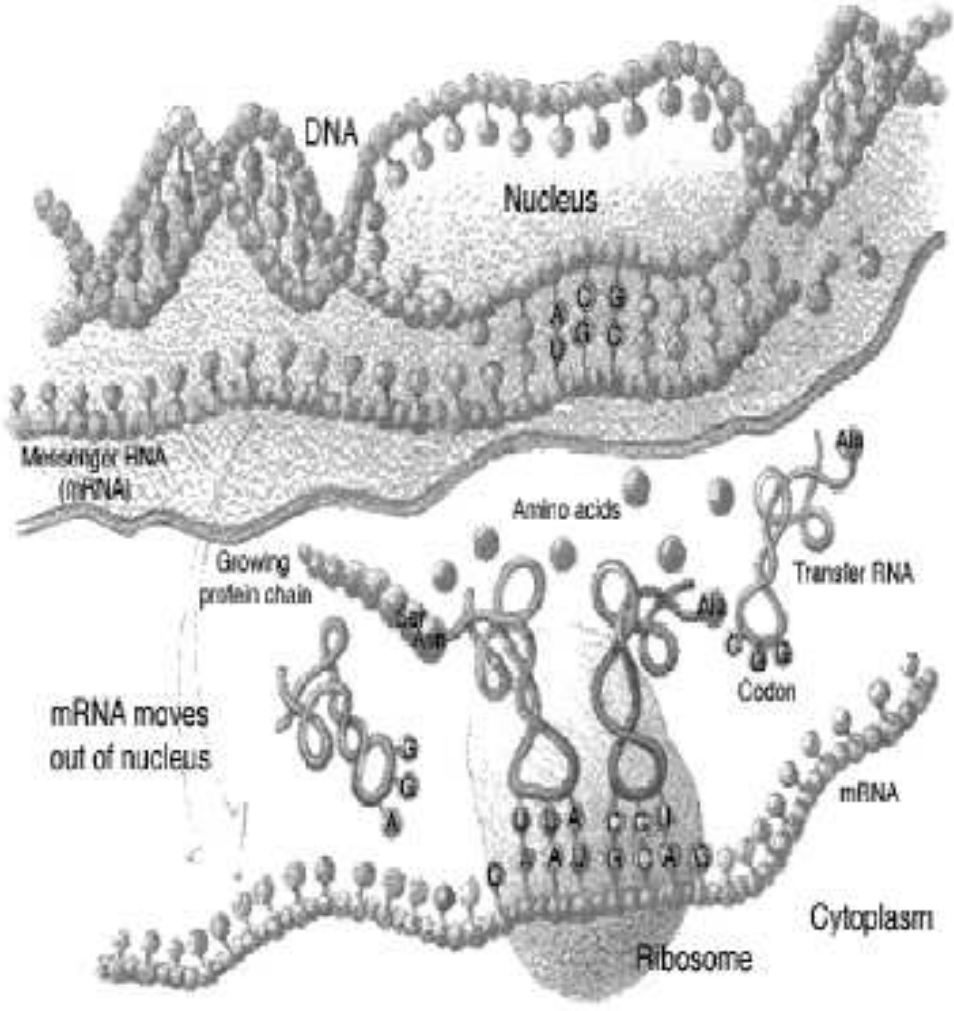


Figure 2.3: From gene to protein [7].

#### 2.1.5.1 Genetic Code

The rules by which the nucleotide sequence of a gene is translated into the amino acid sequence of the corresponding protein, the so called *genetic code*, were deciphered in the early 1960s. The sequence of nucleotides in the mRNA molecule, that acts as an intermediate was found to be read in serial order in groups of three. Each triplet of nucleotides, called a *codon*, specifies one amino acid (the basic unit of a protein, analogous to nucleotides in DNA). Since RNA

is a linear polymer of four different nucleotides, there are $\mathbf{4}^3 = 64$ possible codon triplets. However, only 20 different amino acids are commonly found in proteins, so that most amino acids are specified by several codons. In addition, 3 codons (of the 64) specify the end of translation, and are called *stop codons*. The codon specifying beginning of translation is $AUG$, and is also the codon for the amino acid Methionine. The code has been highly conserved during evolution: with a few minor exceptions, it is the same in organisms as diverse as bacteria, plants, and humans.

#### 2.1.5.2 Transcription

A segment of DNA is first copied into a complementary strand of RNA. This process called *transcription* is catalyzed by the enzyme RNA *polymerase*. Near most of the genes there is a special pattern in the DNA called *promotor*, located upstream of the transcription start site, which informs the RNA polymerase where to begin the transcription. This is achieved with the assistance of transcriptional factors that recognize the promotor sequence and bind to it. Although *ribonucleic acid* (RNA) is a long chain of nucleic acids (as is DNA), it has very different properties. First, RNA is usually single stranded (denoted ssRNA). Second, RNA has a ribose sugar, rather than deoxy-ribose. Third, RNA has the pyrimidine based $Uracil$ (abbreviated U) instead of Thymine. Fourth, unlike DNA, which is located primarily in the nucleus, RNA can also be found in the cellular liquid outside the nucleus, which is called the *cytoplasm*. In Eukariotic organisms, to produce a protein the entire length of the gene, including both its introns and its exons, is first *transcribed* into a very large RNA molecule - the *primary transcript*. At the end of the gene the transcription stops, and a few dozens of Adenine (A) nucleotides are added to the end of the RNA molecule for protection ($poly - Atail$ ). 5 CAP lays an important part in the initializing of protein synthesis by the protecting the growing RNA transcript from degradation. Before this RNA molecule leaves the nucleus, a complex of RNA processing enzymes removes all the intron sequence, in a pro-

cess called splicing, thereby producing a much shorter RNA molecule as shown in Figure 2.4. Typical eukaryotic exons are of average length of 200bp, while the average length of introns is around 10000bp (these lengths can vary greatly between different introns and exons). In many cases, the pattern of the *splicing* can vary depending on the tissue in which the transcription occurs. For example, an intron that is cut from mRNAs of a certain gene transcribed in the liver, may not be cut from the same mRNA when transcribed in the brain. This variation is called alternative splicing, and it contributes to the overall protein diversity in the organism. After this RNA processing step has been completed, the RNA molecule moves to the cytoplasm as a *messanger* RNA molecule (mRNA), in order to undergo *translation*.

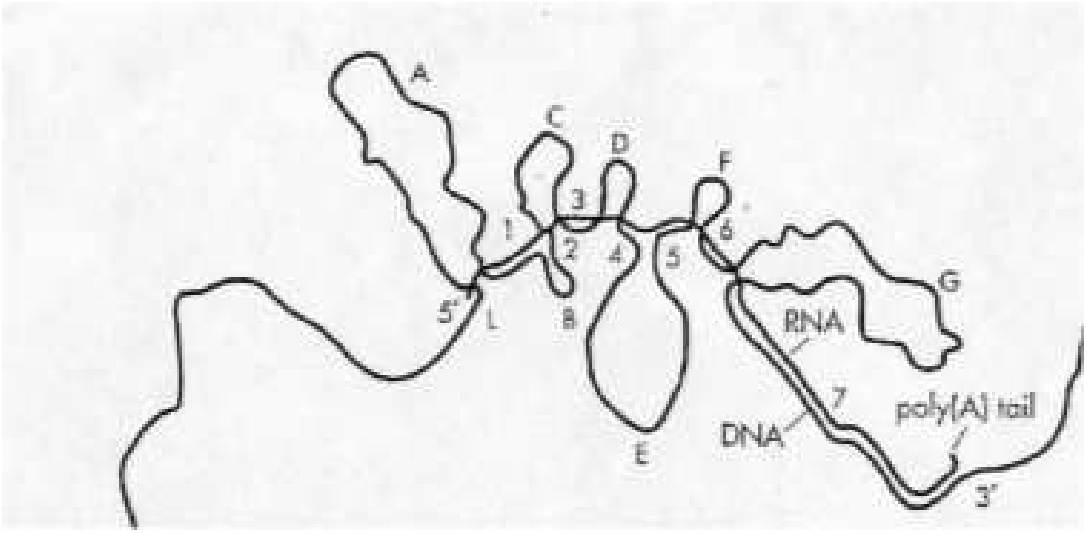


Figure 2.4: An m-RNA is attached to a single-strand DNA from which it is transcribed [6].

#### 2.1.5.3 Translation

In principle, each RNA sequence can be translated in any one of three reading frames in each direction, making a total of 6 possible open reading frames - ORFs, depending on where the process begins. In almost every case, only one of these reading frames will produce a functional protein. However, there are rare cases, especially in viruses, where genes are transcribed from overlapping complementary regions of the DNA.

The *translation* of mRNA into protein depends on adaptor molecules that recognize both an amino acid and a triplet of nucleotides. These adaptors consist of a set of small RNA molecules known as transfer RNA - tRNA, each about 80 nucleotides in length. The tRNA molecule enforces the universal genetic code logic in the following fashion: On one part the tRNA holds an anticodon, a sequence of three RNA bases; on the other side, the tRNA holds the appropriate amino acid. In eukaryotes, the mRNA is formed of coding regions flanked by non-coding regions. Coding regions (exons or parts of exons) used for the protein creation, while the non coding regions - 3 untranslated region and 5 UTR - are

mostly regulatory and are not translated. Note, that along the DNA, the coding region may not be contiguous, as it might span several exons. In Prokaryotes, a gene has only one coding region, flanked by the 3 UTR and the 5 UTR.

Due to the mechanic complexity of ordering the tRNA molecules on the mRNA, a mediator is required. The *ribosome* is a complex of more than 50 different proteins associated with several structural rRNA molecules. Each ribosome is a large protein synthesizing machine, on which tRNA molecules position themselves for reading the genetic message encoded in an mRNA molecule. Ribosomes operate with remarkable efficiency: in one second a single bacterial ribosome adds about 20 amino acids to a growing poly-peptide chain. Many ribosomes can simultaneously translate a single mRNA molecule.

### 2.1.6 Proteins

A protein is linear polymer of *amino acids* linked together by peptide bonds. The average protein size is around 200 amino acids long, while large proteins can reach over a thousand amino acids. To a large extent, cells are made of proteins, which constitute more than half of their dry weight. Proteins determine the shape and structure of the cell, and also serve as the main instruments of molecular recognition and catalysis. Proteins have a complex structure, which can be thought of as having four hierarchial structural levels.

Proteins are not sufficiently described by just stating their amino acid sequence. An important feature of protein function is the folding structure of the molecule. This is the reverse formulation of the central dogma: The function of a protein depends on its structure which is determined by the amino acid sequence which is a translation product of the mRNA copied from the DNA.

Proteins carry out almost all of the functions of the living cell. Neither growth nor development would be possible without proteins. Each of the huge number of protein functions demands its own protein structure, because interaction with other molecules depends on a specific three-dimensional build.

Four different levels of protein structure can be distinguished:

- Primary Structure: refers to the "linear" sequence of amino acids. Proteins are large polypeptides of defined amino acid sequence. The sequence of amino acids in each protein is determined by the gene that encodes it. The gene is transcribed into a messenger RNA (mRNA) and the mRNA is translated into a protein by the ribosome. Primary structure is sometimes called the "covalent structure" of proteins because, with the exception of disulfide bonds, all of the covalent bonding within proteins defines the primary structure. In contrast, the higher orders of proteins structure (i.e. secondary, tertiary and quartenary) involve mainly noncovalent interactions.

- Secondary Structure: is "local" ordered structure brought about via hydrogen bonding mainly within the peptide backbone. The most common secondary structure elements in proteins are the alpha-helix and the beta-sheet (sometime called b pleated sheet).

- Tertiary Structure: is the "global" folding of a single polypeptide chain. A major driving force in dertemining the tertiary structure of globular proteins is the hydrophobic effect. The polypeptide chain folds such that the side chains of the nonpolar amino acids are "hidden" within the structure and the side chains of the polar residues are exposed on the outer surface.

- Quartery Structure: involves the association of two or more polypeptide chains into a multi-subunit structure. Quartenary structure is the stable association of multiple polypeptide chains resulting in an active unit. Not all proteins exhibit quartenary structure. Usually, each polypeptide within a multisubunit protein folds more-or-less independently into a stable tertiary structure and the folded subunits then associate with each other to form the final structure.

## 2.2 Computational Background

Developing computational methods for identifying genes in genomic sequences has been an active field of research. Computational biology research is focused on the analysis and interpretation of data and on the development of new algorithms and statistics. The results of the research has been basics for the computational methods to identify genome sequences. [8].

The methods mainly consist of scanning the DNA sequence, obtaining some statistics and use these statistics for further interpretation and identification of functional parts of the genomic sequence. Common obstacles these methods encounter are: as more genomic data become available, reliable discrimination of functional parts require much more sophisticated analysis/interpretation and wide variety in size and complexity [8].

As mentioned before, our proposed system focuses on the prediction of the subcellular localization of proteins that has been an important problem in bioinformatics. Localization prediction is particularly useful in analyzing and annotating sequences of known gene products. Because many cellular functions are compartmentalized at specific regions of the cell, prediction of the localization of a protein can provide valuable information concerning its possible functions.

Since subcellular localization is in most cases encoded in the amino acid sequence of a protein, there are many sequence analysis methods for predicting the subcellular localization of proteins. This section discusses several methods with reference to their most commonly used implementations and gives details of the computational aspects of the methods used in our study.

Some of the most popular tools for protein localization prediction use neural networks. A neural network is composed of computational units. Neural networks have the advantage that a strict input model is not required. This can be helpful when a model for homology is unknown. However, the internal operations of a neural network are not easily observed; the basis for results is therefore not readily analyzable.

Some of the other tools for protein localization prediction use discriminant

function analysis. A discriminant function is a function of a set of variables (measurements) that is evaluated over various samples with the purpose of classifying them. In the case of protein localization prediction, the variables are used to represent properties of the amino acids in the sequences.

Hidden Markov Models (HMMs) are often used to predict the amino acid sequences of proteins based on probabilities of finding specific amino acids at each position. Biological sequences can be modelled as the output of a stochastic process in which the probability for a given nucleotide to occur at position p depends on the nucleotide occupying the k previous positions. In HMM, system is represented with a set of discrete states and transitions between those states. Each transition has an associated probability. States are hidden meaning that the states can not be directly observed.

Finally, some of the most robust methods for predicting protein subcellular localization take an integrative approach by combining several of the above mentioned methods. Our proposed system can be represented as an integrative approach using PCA, SOM and SVMs.

PCA is used to reduce the dimension of input data. The most effective features are obtained from the data set and used as inputs in the rest of the other methods applied. This reduction provides the extraction of valuable information necessary for the prediction and eliminates the information that does not change the outcome of our system drastically.

After the dimensionality is reduced, an intermediate step is included in the proposed system. This step includes the natural grouping of protein sequences with respect to their subcellular localizations. Natural groups are represented as "clusters" via SOM method. SOM is selected as a clustering method. The main reason of the selection is that the number of clusters are not known. Those clusters are organized as a set of nodes in a hypothetical network with a simple neighborhood structure on the nodes. The neighborhood structure is used in the construction of the features on which the classification is applied. Each node is represented with prototype (reference) vectors adjusted during training. For

each protein, the prototype vectors of the node that has the smallest difference among the vectors of other nodes in the network are taken as a feature for classification. The prototype vectors applied to SVM increase the accuracy of SVM classification.

After the formation of clusters, the classification method is used in the prediction of subcellular localizations. SVM is applied to determine the label of the proteins with unknown subcellular localization. SVM is chosen due to the fact that SVMs, considered as *supervised* computer learning method, offer many advantages over the *unsupervised* methods. In addition, only few parameters are to be chosen and SVM does not have a complicated architecture as in neural networks.

In the following subsections, computational aspects of methods that are used in the integrative approach are presented. In Section 2.2.1, a *combination scheme* for the feature extraction ( i.e determination of an appropriate subspace of dimensionality $m$ in the original feature space of dimensionality d ($d \leq m$) ) is presented. Once features are extracted from the given data set and a proper representation is designed, a *classification scheme* (*support vector*) is described in Section 2.2.2. Background information on the support vector analysis is given and the effect of the classifier to the prediction of protein subcellular localization is explained.

### 2.2.1 Feature Extraction

In this study, a *combination scheme* for the feature extraction process is presented. There are several reasons for this combination.

1. To create new features based on transformations or combinations of the original feature set. New features may provide a better discriminative ability than the best subset of given features, despite of the fact that these new features may not have a clear physical meaning.

2. To keep the dimensionality of the pattern representation as small as possi-

ble and minimize measurement cost and maximize classification accuracy.

3. To form clusters or "natural groupings" of the input patterns. Proteins that have similar subcellular localization are grouped together so that classification results are more accurate.

#### 2.2.1.1 PCA

In this study, PCA is used to tackle the first and second items stated above. PCA is known as an unsupervised linear feature extractor that determines an appropriate subspace of dimensionality $m$ in the original feature space of dimensionality $d$ ($d \leq m$). It computes the $m$ largest eigenvectors of the $d \times d$ covariance matrix of the $n$ $d$-dimensional patterns. PCA is the oldest and best known projection technique in multivariate analysis. Pearson [9] first introduced PCA, which had been used in a biological context to recast linear regression analysis into a new form. Then Hotelling [10] developed the technique in a work done on psychometry. Today, PCA is used effectively in data compression, image analysis, visualisation, pattern recognition, regression and time series prediction.

Hotelling defines PCA as:

> For a set of observed $d$-dimensional data vectors $\mathbf{t}_n$, $n = 1, \ldots, N$, the $q$ *principal axes* $\mathbf{w}_j$, $j = 1, \ldots, q$, are those orthonormal axes onto which the retained *variance* under projection is maximal.

The traditional way of computing the principal components is to compute the sample covariance matrix (i.e scatter matrix) of the dataset with sample mean,

$$\mathbf{S} = \frac{\mathbf{1}}{\mathbf{N-1}} \sum_{\mathbf{n}}^{\mathbf{N}} (\mathbf{t_n} - \hat{\mu})(\mathbf{t_n} - \hat{\mu})^{\mathbf{T}}, \qquad \hat{\mu} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{n}}^{\mathbf{N}} \mathbf{t_n}, \tag{2.1}$$

and then to find its eigen-structure

$$\mathbf{SU} = \mathbf{U\Lambda}. \tag{2.2}$$

$\mathbf{U}$ is a $D \times D$ matrix which has the unit length eigenvectors, $\mathbf{u_1}, ..., \mathbf{u_D}$, as its columns and $\Lambda$ is a diagonal matrix with the corresponding eigenvalues,

$\lambda_1, ..., \lambda_D$, along the diagonal. To find the best one-dimensional projection of the data (best in the least-sum-of-squared-error sense), the data is projected onto a line through the sample mean in the direction of the eigenvector of the scatter matrix having the largest eigenvalue. The eigenvectors are the principal components while the eigenvalues are the corresponding variances.

An important property of the principal components is that they constitute the unique set of vectors, up to scaling, that minimise the *reconstruction error*, which is the sum of the squared distances between the data points and their projections on the $L$ principal components, summed over the dataset. The function for the reconstruction error is given as,

$$E_L = \sum_{n}^{N} \parallel \mathbf{t_n^N} - (\mathbf{t_n^N}\hat{\mathbf{U}}_\mathbf{L})\hat{\mathbf{U}}_\mathbf{L}^\mathbf{T} \parallel^\mathbf{2}, \tag{2.3}$$

where $\hat{\mathbf{U}}$ is $D \times L$ matrix with the scaled principal components $\mathbf{u_1}, ..., \mathbf{u_L}$ ($\mathbf{1} \leq \mathbf{L} \leq \mathbf{D}$) as their columns. Under this formulation, which attempts to find the principal components by minimising Equation 2.3, PCA is know as the Karhunen-Loéve transform.

When PCA is examined in multidimensional scaling perspective, it can be shown that multidimensional scaling corresponds to PCA when the Euclidean distance is the dissimilarity measure [11].

PCA fails to separate the clusters due to the nature of the dataset. It is reported in [12] that the data has been generated such that the principal axes of variance are orthogonal to the line separating the two adjacent clusters.

#### 2.2.1.2 Self Organizing Map

In this study, SOM is used to tackle the third item stated in Section 2.2.1. Clustering of proteins that have similar subcellular localizations is maintained by $Kohonen\ self-organizing\ feature\ maps$. SOM is an unsupervised artificial neural network model that relates similar input vectors to the same region of a map of nodes. Since SOM does a topological ordering of the input, topological organization of neurons is an essential feature. It is often used to categorize and

interpret data, by mapping a high-dimensional input data to a lower dimensional space which is usually 2. The self-organizing map algorithm is described as follows:

1. The set of input samples is described by a real vector $x(t) \in \Re^n$ where $t$ is the index of the sample, or the discrete-time coordinate. Each node $i$ in the map contains a model vector $m_i(t) \in \Re^n$, which has the same number of elements as the input vector $x(t)$.

2. The initial values of the components of the model vector, $m_i(t)$, are randomly initialized.

3. Any input item is thought to be mapped into the location, the $m_i(t)$ of which matches best with $x(t)$ in some metric. The self-organizing algorithm creates the ordered mapping as a repetition of the following basic tasks:

   (a) Compare an input vector $x(t)$ with all the model vectors $m_i(t)$

   (b) Identify the *winner node* (best-matching unit) on the map. Winner node is defined by the smallest *Euclidean* distance $\|x - m_i\|$, signified by the subscript $c$:

   $$\|x - m_i\| = min_i\{\|x - m_i\|\}, or$$

   $$c = argmin_i\{\|x - min_i\|\}.$$

   x is mapped onto the node c relative to the parameter values $m_i$

4. The model vectors of the winner and a number of its neighboring nodes in the array are modified towards the input vector according to the learning principle:

   (a) For each sample input vector $x(t)$, the winner and the nodes in its neighborhood are changed closer to $x(t)$ in the input data space.

(b) The net outcome in the principle t is that ordered values for the $m_i(t)$ emerge over the array. If input samples are restricted, the samples must be repeatedly presented.

(c) Adaptation of model vectors in the learning process may take place according to the following equations:

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \;\; for\ each\ i \in N_c(t)$$

$$m_i(t+1) = m_i(t) \;\; otherwise,$$

where t is an integer, $\alpha(t) \in [0, 1]$ is a scalar of learning step, $N_c(t)$ is neighborhood kernel.

### 2.2.2 SVM Classifier

Once algorithms described in the previous sections are applied, a *classification* algorithm, SVM, is applied to determine the label of the proteins with unknown subcellular localization.

We have choosen SVM due to the fact that SVMs have many mathematical features that make them attractive for gene expression analysis, including

- their flexibility in choosing a similarity function,
- sparseness of solution when dealing with large data sets,
- the ability to handle large feature spaces,
- the ability to identify outliers,
- the ability to employ distance functions that operate in extremely high-dimensional feature spaces.

SVMs, considered as *supervised* computer learning method, offer two major primary advantages over the *unsupervised* methods. The advantages are described as follows:

- SVMs can employ distance functions that operate in extremely highdimensional feature spaces.

- Supervised methods like SVMs take advantage of prior knowledge (in the form of training data labels) in making distinctions between one type of protein and another.

SVM implements the following idea: it maps the input vectors $y_i \in \Re^n$ into a high dimensional feature space $\phi(y_i) \in H$ and constructs an Optimal Separating Hyperplane (OSH), which maximizes the margin, the distance between the hyperplane and the nearest data points of each class in the space $H$. Different mappings construct different SVMs. The mapping $\phi$ is performed by a kernel function $K(y_i, y_j)$ which defines an inner product in the space H.

Given a training set of instance-label pairs $(y_i, z_i)$, $i = 1, \ldots, l$ where $y_i \in \Re^n$ and $z \in \{0, 1, 2, 3\}^l$, $l = 800$, the support vector machines require the solution of the following optimization problem:

$$\begin{aligned} &min_{w,b,\xi}\ \tfrac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ &subject\ to\ z_i(w^t\phi(y_i) + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0. \end{aligned} \tag{2.4}$$

Here training vectors $y_i$ are mapped into a higher dimensional space by the function $\phi$. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(y_i, y_j) \equiv \phi(y_i)^T\phi(y_j)$ is called the **kernel function**.

SVMs consist in the combination of two key concepts: Maximal Margin Hyperplanes and Kernel Mappings. Both are explained in Appendix A.

# CHAPTER 3

# LITERATURE SURVEY

## 3.1 Applications on prediction of genomic sequences

Many researchers introduce systems for interpreting long anonymous genome sequence (see [13, 14] for existing methods for the interpretation of genomic sequence). Interpretation includes partitioning into genes, extracting biological features such as coding and non coding regions. One of the first examples of biological feature extraction systems is GRAIL I [15]. It uses neural network to combine multiple coding regions and is accessible through an e-mail server. It marked the entry into the modern era for gene identification software and biologists became aware of computational prediction methods and began to trust them. Following GRAIL I, several prediction systems have been developed. As an example, GeneID [16] starts by identifying first, internal and last coding regions on the basis of coding measures and uses a heuristic, rule-based system to assemble these into models of one likely gene in each sequence. GenLang [17] does not use combinatorial approach but interprets the usual coding measures in a linguistic context as "leaf rules" associated to a cost. A formal grammar model, optimized on a training set, is then used to generate a gene model as the parse that minimizes the total cost. Slonim described GENECLUSTER [18] a visualization and clustering package for revealing informative patterns from

an expression data set. GENECLUSTER uses technique of self-organizing maps (SOM). GENECLUSTER is used to organize genes into different biologically relevant clusters that provide novel insights into the understanding and treatment of leukemia. MORGAN [19], an integrated system for finding genes, uses a variety of techniques, the most distinctive of which is a "decision tree" algorithm. Well established machine learning techniques, decision tree classifiers have been introduced by Salzberg [20] for solving the simpler problem of discriminating coding and non-coding DNA. Once classified, the various components are assembled into an optimal gene model using a dynamic programming approach. GeneParser [21, 22] can be given as an example. It uses dynamic programming approach to find the optimal combination of coding and non-coding DNA. Also, there exists systems that model biological sequences as the output of a stochastic process. GeneMark [23] and GENSCAN [24] introduce a general probabilistic model of gene structure based on hidden Markov Models. GENSCAN explores possible gene models on both DNA strands simultaneously and is capable of parsing sequences containing multiple genes.

## 3.2 Applications on prediction of protein subcellular localization

Most of the systems presented above have some limitations. They only detect protein coding regions and concentrate on one typical gene. Therefore, recent studies are based on prediction of the subcellular localization that is an essential characteristic for the identification and characterization of protein function. Prediction of subcellular localization methods can be classified into two categories: one is based on the amino acid composition and the other is based on the amino acid order. The amino acid composition represents the global features, e.g., the amino acid composition in the full or partial sequences, while the order represents the local features, e.g., the amino acid sequence order, protein N-terminal sorting signals.

A fully automatic and reliable prediction system Subloc [25, 26] for protein

subcellular localization is presented by Hua and Sun. Subloc predicts the subcellular localization of proteins from their amino acid composition based on support vector machines. The dataset included 2427 eukaryotic sequences belonging to four location categories (nuclear, cytoplasmic, mitochondrial and extracellular). The total prediction accuracy reaches to 79.4% for four locations in eukaryotic organisms. It is stated that this approach provides superior prediction performance compared to existing algorithms based on amino acid composition and can be a complementary method to other existing methods based on sorting signals. When we compare this system with our proposed system, it is observed that classification methods used are similar. In addition, both system predicts the localization of proteins from their global sequence features that is amino acid composition. The prediction accuracy of our system reaches to 81.25% while Hua's and Sun's prediction system reaches to 79.4%. The better performance is achieved by the inclusion of encoding scheme, feature extraction and using clustered features for classification.

The most widely used neural network based method for detection of signal peptide sequences is SignalP [27]. SignalP is one of the one-category predictors that predicts eukaryotes signal peptide sequence. Prediction accuracy reaches to 72.4% [28]. SignalP predicts subcellular localization sites using the amino acid order. TargetP [3] use neural network similar to SignalP, predicting the presence or absence of a signal sequence. TargetP integrates several neural network programs to distinguish between secretory, mitochondrial, nuclear and cytosolic signals. It predicts subcellular localization of proteins based on their N-terminal amino acid sequence. The local features of proteins are used in the prediction. In our study, the same data set and classes that is used in TargetP are used. Although the data sets are similar, there are major differences in the implementation, usage of data and accuracy. TargetP integrates several neural network programs while our system uses SVM in classification. TargetP predicts the localization of proteins from their amino acid sequence order. On the other hand, our system predicts the localization of proteins from their global sequence

features that is amino acid composition. The prediction accuracy of our system reaches to 81.25% where TargetP reaches to 85%. The better performance of TargetP is achieved by the usage of the local features of proteins that is amino acid sequence order.

PRED-TMR is a method that predicts transmembrane domains in proteins using solely information contained in the sequence itself [29]. PRED-TMR uses a hierarchical artificial neural network system for the classification of transmembrane proteins. It classifies proteins into two classes: transmembrane protein class and the non-membrane protein class. It is a method that predicts the classsification of proteins from their global sequence features that is amino acid composition. Applied on several test sets of transmembrane proteins, the system gives a perfect prediction rating of 100% by classifying all the sequences in the transmembrane class. When compared with our study, PRED-TMR reaches to 97.7% accuracy. Although the classification features are of different types, PRED-TMR could be adapted to our problem domain and evalutions could be performed by applying PRED-TMR on our data set. The prediction accuracy could decrease with the increase in the number of classes.

MitoProt was developed to predict localization at mitochondria [30]. In a work by Chou and Elrod [31], a more general predictive method, employing discriminant function analysis, is proposed. Their method can predict protein localization in a dozen subcellular regions of eukaryotes based on sequence-derived features: chloroplast, mitochondria, nucleus, cytoplasm, plasma membrane, etc. Their reported overall accuracy is 83.1%.

The most commonly used hidden markov model (HMM) method for signal sequence prediction is SignalP-HMM [32]. Like the original SignalP, SignalP-HMM predicts the absence and presence of a signal sequence.

Some of the robust methods take an integrative approach by combining several of the above mentioned methods. SortPed [33] is developed for using both neural networks and hidden Markov Models for four class problem of subcellular localization. PSORT [34] is one of the most robust tools for predicting subcel-

lular localization. PSORT [35] is an expert system that can distinguish between 17 different subcellular localizations in eukaryotes. The program integrates numerous prediction programs and statistical methods into a "knowledge base" which is used by an "interface engine" to deduce a solution.

As mentioned above, TargetP uses a hidden Markov Model similar to SignalP. However, TargetP can also be used to distinguish between secretory, mitochondrial and chloroplast signals by analysing the output of SignalP, ChloroP [36] and a mitochondrial predictor with a "decision neural network". This decision neural network compares the outputs of the three programs to make a final decision. Overall accuracy of TargetP is approximately 85%.

We present an integrative approach for the prediction of protein subcellular localization using global protein sequence features. A combination of clustering followed by classification is the most important aspect of this study. Proposed system prediction results are very promising compared to the experimental results indicated in the literature. Our ultimate goal is to design a system that applies several methods to analyse biological properties of protein sequence data and perform predictions based on only global protein sequence features yielding comparable results with TargetP.

Our proposed system use a novel encoding scheme which has not been described in the literature. In addition, our system uses principle component analysis method based on the encoded protein sequence data to extract the most effective features. Our system uses self organizing map method for the clustering of the most effective features similar to the visualization and clustering package GENECLUSTER. Finally, different from studies in the literature, our system uses support vector machines method that classifies the protein sequences that are clustered by SOM.

When compared to the studies in the literature, our proposed system integrates PCA, SOM and SVM methods to deduce the subcellular localizations of proteins. It predicts subcellular localizations using solely information contained in the sequence itself. It classifies proteins into four classes: nuclear, cytosolic,

mitochondrial targeting peptides and signal peptides. By the combination of methods, biological information is analysed and evaluated more accurately than any other studies mentioned in this section. In a work by Hua and Sun [25], by employing only SVMs, prediction of four subcellular locations in eukaryotes is performed. The total prediction accuracies reach 79.4%. Our system reports 81.25% overall accuracy. The reason of the better acccuracy performance of the proposed system is based on the use of combination of feature extraction and clustering methods on the encoded protein data and usage of the extracted and featured encoded data for the SVM classification.

# CHAPTER 4

# METHODS AND EXPERIMENTS

In this chapter, the method applied for the subcellular protein localization is explained and a range of experiments are performed to assess the strengths and weaknesses of the proposed method. The chapter consists of five sections. First, we describe the protein sequence data set used in our experiments. Then, we design a set of experiments and present the results. The performance of the proposed method is assessed against another strategy: prediction of protein subcellular localization using only local features. The effects of certain parameters of the method on the performance is discussed and analyzed. Finally, some ideas for improving the performance is proposed.

## 4.1 Data Set and Input Representation

### 4.1.1 Data Set

In this study, the eukaryotic predictor data sets used in the construction and evaluation of TargetP by Emanuelsson et.al [3]. All data is extracted from SWISS-PROT [37]. Detailed information of SWISS-PROT is given in 4.1.1.1.

The prediction of protein subcellular localization is based on the predicted presence of any of the N-terminal presequences the *nuclear* sequences, the *cytosolic* sequences, the *mitochondrial targeting peptides* (mTP) and the *signal*

*peptides* (SP). Sequences were extracted by requiring the keyword EUKARYOTA in the OC (Organism Classification) field. The very few sequences containing B, Z, or X are excluded in order to avoid possible noise from the ambigous positions in the training and testing.

The full data set contains 2738 proteins given in Table 4.1 that consists of the mTP, SP, nuclear, and cytosolic sets. Four classes are presented as amino acid N-terminal sequences. These are in so called 'Fasta' format. Detailed information of Fasta is given in 4.1.1.2.

Table 4.1: Types of protein subcellular locations and the number of each protein.

| Localization Type | Number of protein |
|---|---|
| Mitochondrial (mTP) sequences | 371 |
| Signal Peptides (SP) sequences | 715 |
| Nuclear sequences | 1214 |
| Cytosolic sequences | 438 |

Each experimental data set given in Table 4.1 is further split up into training and standard test sets, containing the same number of proteins from each localization type for training and test sets respectively. No sequence is tested on a network assembly whose training it had participated in.

#### 4.1.1.1 SWISS-PROT

Swiss-Prot is an annotated protein sequence database. It was established in 1986 and maintained collaboratively, since 1987, by the group of Amos Bairoch first at the Department of Medical Biochemistry of the University of Geneva and now at the Swiss Institute of Bioinformatics (SIB) and the EMBL Data Library (now the EMBL Outstation - The European Bioinformatics Institute (EBI)). The Swiss-Prot protein knowledgebase consists of sequence entries. Sequence entries are composed of different line types, each with their own format. For standardization purposes the format of Swiss-Prot follows as closely as possible that of the EMBL Nucleotide Sequence Database. In Swiss-Prot, as in many

sequence databases, two classes of data can be distinguished: the core data and the annotation. For each sequence entry the core data consists of:

- The sequence data
- The citation information (bibliographical references);
- The taxonomic data (description of the biological source of the protein).

The annotation consists of the description of the following items:

- Function(s) of the protein
- Posttranslational modification(s). For example carbohydrates, phosphorylation, acetylation, GPI-anchor, etc.
- Domains and sites. For example calcium-binding regions, ATP-binding sites, zinc fingers, homeoboxes, SH2 and SH3 domains, kringle, etc.
- Secondary structure. For example alpha helix, beta sheet, etc.
- Quaternary structure. For example homodimer, heterotrimer, etc.
- Similarities to other proteins
- Quaternary structure. For example homodimer, heterotrimer, etc.
- Similarities to other proteins
- Disease(s) associated with any number of deficiencies in the protein
- Sequence conflicts, variants, etc

The Swiss-Prot protein sequence database is composed of sequence entries. Each entry corresponds to a single contiguous sequence as contributed to the bank or reported in the literature. In some cases, entries have been assembled from several papers that report overlapping sequence regions. Conversely, a single paper can provide data for several entries, e.g. when related sequences from different organisms are reported.

References to positions within a sequence are made using sequential numbering, beginning with 1 at the N-terminal end of the sequence.

The entries in the Swiss-Prot database are structured so as to be usable by human readers as well as by computer programs. The explanations, descriptions, classifications and other comments are in ordinary English. Wherever possible, symbols familiar to biochemists, protein chemists and molecular biologists are used.

Each sequence entry is composed of lines. Different types of lines, each with their own format, are used to record the various data that make up the entry. Sample sequences are available at [38].

Each line begins with a two-character line code, which indicates the type of data contained in the line. The current line types and line codes and the order in which they appear in an entry, are shown in Table 4.2.

Table 4.2: Current line types and line codes and the order in which they appear in an entry

| Line Code | Content | Occcurence in an entry |
|---|---|---|
| ID | Identification | Once; starts the entry |
| AC | Accession number(s) | Once or more |
| DT | Date | Three times |
| DE | Description | Once or more |
| GN | Gene name(s) | Optional |
| OS | Organism species | Once or more |
| OG | Organelle | Optional |
| OC | Organism classification | Once or more |
| OX | Taxonomy cross-reference(s) | Once or more |
| RN | Reference number | Once or more |
| RP | Reference position | Once or more |
| RC | Reference comment(s) | Optional |
| RX | Reference cross-reference(s) | Optional |
| RA | Reference authors | Once or more |
| RT | Reference title | Optional |
| RL | Reference location | Once or more |
| CC | Comments or notes | Optional |
| DR | Database cross-references | Optional |
| KW | Keywords | Optional |
| FT | Feature table data | Optional |
| SQ | Sequence header | Once |
| | (blanks) sequence data | Once or more |
| // | Termination line | Once; ends the entry |

As shown in Table 4.2, some line types are found in all entries, others are optional. Some line types occur many times in a single entry. Each entry must begin with an identification line (ID) and end with a terminator line (//). A detailed description of each line type is available [38].

#### 4.1.1.2 FASTA format description

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data.

- The description line starts with a greater than symbol (">").

- The word following the greater than symbol (">") immediately is the "ID" (name) of the sequence, the rest of the line is the description.
- The "ID" and the description are optional.
- All lines of text should be shorter than 80 characters.
- The sequence ends if there is another greater than symbol (">") symbol at the beginning of a line and another sequence begins.

The following is an example from cytosolic sequence that is Fasta format:

> $P46156; GALLINACIN1 PRECURSOR.$
> MRIVYLLLPFILLLAQGAAGSSQALGRKSDCFRKSGFCA FLKCP-
> SLTLISGKCSRFYLCCKRIWG

Sequences are expected to be represented in the standard IUB/IUPAC amino acid and nucleic acid codes, with these exceptions: lower-case letters are accepted and are mapped into upper-case; a single hyphen or dash can be used to represent a gap of indeterminate length; and in amino acid sequences, U and * are acceptable letters. Before submitting a request, any numerical digits in the query sequence should either be removed or replaced by appropriate letter codes (e.g., N for unknown nucleic acid residue or X for unknown amino acid residue). The amino acid codes are given in Table 4.3

Table 4.3: Amino Acid Codes.

| Amino Acid Code |
|---|
| A |
| C |
| D |
| E |
| F |
| G |
| H |
| I |
| K |
| L |
| M |
| N |
| P |
| Q |
| R |
| S |
| T |
| V |
| W |
| Y |
| B - Excluded |
| X - Excluded |
| Z - Excluded |

### 4.1.2 Input Representation

We transform each protein given in Table 4.1 to a compact representation to obtain the input to our prediction system. The input to the system is amino acid sequences and primary sequences are extracted from this data. For each protein, an alphabet of size 20 is used since there are 20 kinds of amino acids.

Protein sequences are strings of arbitrary size and amino acids correspond to the letters given in Table 4.3. The amino acid composition in the full or partial protein sequence is taken as a *global feature*.

Let $S$ represent a protein sequence whose length is $len(S)$. $S$ can be represented as a vector of fixed size 20 containing the frequencies of amino acids

used in the representation of a protein sequence. A protein can be represented as follows:

$$\begin{aligned} &S \in \Re^{20} \\ &S : [x_1, \ldots, x_d] \quad \textit{where } d \textit{ is } 20. \end{aligned} \tag{4.1}$$

$x_1, \ldots, x_d$ denotes the frequency of amino acid containing in the protein sequence $S$. The representation is as follows:

$$x_i = (\#\{A_i\}) \ / \ len(S) \quad i = 1, \ldots, 20, \tag{4.2}$$

where the $A_i$ are twenty different amino acids given in Table 4.3 and # indicates the cardinality. Each frequency $x_i$ is between $\{0, 1\}$.

Note that in the rest of the thesis, $S$ denotes a vector representing a protein sequence.

In order to perform further computational analysis, the amino acids need to be encoded. Although, the most popular way of encoding reported in the literature is to present each amino acid in binary form, in this study, an *encoding scheme* is applied on sequences $S$.

The encoding scheme make use of substitution matrices. Substitution matrices are collection of biological information to indicate score values for the alignment of one protein residue against another. PAM [39] substitution matrix is a matrix of similarity scores for all possible pairs of residues (protein). It is derived from amino acid replacements occuring in related proteins. A positive score signifies a common replacement whereas a negative score signifies an unlikely replacement.

Each protein sequence $S$ is encoded with PAM250 substitution matrix. PAM 250 matrix is optimised for sequences separated by 250 PAM, i.e. 250 substitutions in 100 amino acids. The matrix is given in Table 4.4.

Table 4.4: PAM250 matrix

| . | A . | R . | N. | D. | C. | Q. | E. | G. | H. | I. | L. | K. | M. | F. | P. | S. | T. | W. | Y. | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 2 | -2 | 0 | 0 | -2 | 0 | 0 | 1 | -1 | -1 | -2 | -1 | -1 | -3 | 1 | 1 | 1 | -6 | -3 | 0 |
| R | -2 | 6 | 0 | -1 | -4 | 1 | -1 | -3 | 2 | -2 | -3 | 3 | 0 | -4 | 0 | 0 | -1 | 2 | -4 | -2 |
| N | 0 | 0 | 2 | 2 | -4 | 1 | 1 | 0 | 2 | -2 | -3 | 1 | -2 | -3 | 0 | 1 | 0 | -4 | -2 | -2 |
| D | 0 | -1 | 2 | 4 | -5 | 2 | 3 | 1 | 1 | -2 | -4 | 0 | -3 | -6 | -1 | 0 | 0 | -7 | -4 | -2 |
| C | -2 | -4 | -4 | -5 | 12 | -5 | -5 | -3 | -3 | -2 | -6 | -5 | -5 | -4 | -3 | 0 | -2 | -8 | 0 | -2 |
| Q | 0 | 1 | 1 | 2 | -5 | 4 | 2 | -1 | 3 | -2 | -2 | 1 | -1 | -5 | 0 | -1 | -1 | -5 | -4 | -2 |
| E | 0 | -1 | 1 | 3 | -5 | 2 | 4 | 0 | 1 | -2 | -3 | 0 | -2 | -5 | -1 | 0 | 0 | -7 | -4 | -2 |
| G | 1 | -3 | 0 | 1 | -3 | -1 | 0 | 5 | -2 | -3 | -4 | -2 | -3 | -5 | 0 | 1 | 0 | -7 | -5 | -1 |
| H | -1 | 2 | 2 | 1 | -3 | 3 | 1 | -2 | 6 | -2 | -2 | 0 | -2 | -2 | 0 | -1 | -1 | -3 | 0 | -2 |
| I | -1 | -2 | -2 | -2 | -2 | -2 | -2 | -3 | -2 | 5 | 2 | -2 | 2 | 1 | -2 | -1 | 0 | -5 | -1 | 4 |
| L | -2 | -3 | -3 | -4 | -6 | -2 | -3 | -4 | -2 | 2 | 6 | -3 | 4 | 2 | -3 | -3 | -2 | -2 | -1 | 2 |
| K | -1 | 3 | 1 | 0 | -5 | 1 | 0 | -2 | 0 | -2 | -3 | 5 | 0 | -5 | -1 | 0 | 0 | -3 | -4 | -2 |
| M | -1 | 0 | -2 | -3 | -5 | -1 | -2 | -3 | -2 | 2 | 4 | 0 | 6 | 0 | -2 | -2 | -1 | -4 | -2 | 2 |
| F | -3 | -4 | -3 | -6 | -4 | -5 | -5 | -5 | -2 | 1 | 2 | -5 | 0 | 9 | -5 | -3 | -3 | 0 | 7 | -1 |
| P | 1 | 0 | 0 | -1 | -3 | 0 | -1 | 0 | 0 | -2 | -3 | -1 | -2 | -5 | 6 | 1 | 0 | -6 | -5 | - 1 |
| S | 1 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | -1 | -1 | -3 | 0 | -2 | -3 | 1 | 2 | 1 | -2 | -3 | -1 |
| T | 1 | -1 | 0 | 0 | -2 | -1 | 0 | 0 | -1 | 0 | -2 | 0 | -1 | -3 | 0 | 1 | 3 | -5 | -3 | 0 |
| W | -6 | 2 | -4 | -7 | -8 | -5 | -7 | -7 | -3 | -5 | -2 | -3 | -4 | 0 | -6 | -2 | -5 | 17 | 0 | -6 |
| Y | -3 | -4 | -2 | -4 | 0 | -4 | -4 | -5 | 0 | -1 | -1 | -4 | -2 | 7 | -5 | -3 | -3 | 0 | 10 | -2 |
| V | 0 | -2 | -2 | -2 | -2 | -2 | -2 | -1 | -2 | 4 | 2 | -2 | 2 | -1 | -1 | -1 | 0 | -6 | -2 | 4 |
| . | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |

Note that the entries B, X and Z in Table 4.4 are excluded since these entries are discarded in the protein sequences.

Each protein sequence $S$ is encoded with PAM250 substitution matrix and used as the pattern of interest in the rest of the methods applied in the system. This scheme conserves the biological information in the pattern of interest and enables the representation of the chemical differences of each amino acid.

Each amino acid entry $x_i$ given in Equation 4.2 is multiplied by corresponding column $i$ in PAM250 matrix. $\widetilde{S}$ is used as a PAM encoded protein sequence $S$ and $\widetilde{S} \in \Re^{400}$. In the rest of the thesis, $\widetilde{S}$ denotes a PAM encoded protein sequence $S$.

In summary, each protein sequence is presented as a vector of PAM250 encoded amino acid sequence, $\widetilde{S}$, of dimension 400.

## 4.2 Experimental Results

The desribed prediction system is applied to the data set presented in Table 4.1. It is assured that training and testing is done with equal number of protein sequences from each class cytosolic, mTP, SP and nuclear (size-equalized sets). All sequence inclusions andd exclusions are random. From each class, 200 proteins are randomly chosen for training from the data set. Randomly chosen 100

proteins from each class exclusive of those in the training set are used for testing. No sequences are tested on the classification method whose training it has participated in. The training and test input numbers for each class numbered from 0 to 3 are given in Table 4.5.

Table 4.5: Training and test sets for four classes.

| Class | Training Set | Testing Set |
|---|---|---|
| mTP - 0 | 200 | 100 |
| SP - 1 | 200 | 100 |
| Nuclear - 2 | 200 | 100 |
| Cytosolic - 3 | 200 | 100 |

Randomly chosen sets of size 200 from each class cTP, mTP, SP and nuclear are transformed into vectors of size 20 containing frequencies of amino acid compositions, $S$. Then, each vector, represented as $\widetilde{S}$ and having dimension 400, is encoded in PAM250 matrix as stated in Section 4.1.2 and used as input to the proposed system.

### 4.2.1 Application of PCA

As stated in Section 2.2.1.1, PCA is used to extract the most effective features and keep the dimensionality of the pattern representation as small as possible and minimize measurement cost and maximize classification accuracy.

The input data, $\widetilde{S}$, is represented as a vector of size 400. In order to maximize classification acccuracy and use the most effective features, PCA is applied on the vector, $\widetilde{S}$ of size 400. For finding principle components, Mathlab [40] is used as a computational tool. PCA type that uses the *covariance matrix* is applied.

Initially, let $X$ denotes the sample covariance matrix of the data set including all training data with computed sample mean. Then, principle components are extracted using $X$. By the application of PCA, three matrices $PC$, $Variances$ and $Explained$ are obtained. $PC$ is a 400x400 matrix that stores the principle components. $Variances$ is a 400x1 matrix that stores the eigenvalues of the co-

variance matrix of $X$ and $Explained$ is a 400x1 matrix that stores the percentage of the total variances in the observations explained by each eigenvector.

Analysis of $Variances$ and $Explained$ is done to select the most effective principle components from $PC$. The eigenvalues that are greater than a threshold value $10e^{-015}$ are selected for the identification of principle components. The threshold value is selected with respect to the nature of randomly chosen input data set. By the threshold selection, eigenvalues, that have values computationally approximating to 0, are intended to be eliminated. First 20 columns of $PC$ matrix are extracted and used as the 400x20 principle component extraction matrix, $C$ , both for training and test input data.

For each $\widetilde{S}$ in the training data, the $C$ matrix multiplication is done and represented as follows:

$$P = \widetilde{S}C$$

P denotes a 1x20 matrix that is used as an input to our classification algorithm.

### 4.2.2 Application of SOM

After the extraction of most effective features in training data, each $P$ is used in the clustering of proteins that have similar subcellular localizations. Clustering on the extracted features, $P$'s, is applied via a $self-organizing\ feature\ map$.

SOM is an unsupervised artificial neural network that relates similar input vectors to the same region of a map of nodes. Topological organization of neurons in SOM is an essential feature, since it does a topological ordering of the input. SOM method is included in our proposed system. The reason for the inclusion is that natural groupings of the input data for each class increase the acccuracy of classification.

For the self organizing map, SOM-PAK [41] is used as a computational tool. We have tried different map sizes, topologies and neighbourhood functions.

As the first stage, map initialization is done. The map is initialized using random numbers. The lattice type of the map is selected to be *rectangular* and the neighbourhood function type is *Gaussian neighborhood* function. Experi-

ments yield that a SOM map size 20x20 and *Gaussian neighborhood* function give better results than a SOM map size smaller and *Bubble neighbourhood* function. Comparison is done via map visualization provided by SOM-PAK.

As the second stage, the map is trained by the self-organizing map algorithm in SOM-PAK. Training is done in two phases. The first of them is the ordering phase during which the reference vectors of the map units are ordered. During the second phase, the values of the reference vectors are fine-tuned. In the begining the neighborhood function is taken almost equal to the diameter of the map that is 20 and decreases to one during training, while the learning rate decreases to zero. Initial learning rate parameter is taken as 0.05. During the second phase, the reference vectors in each unit converge to their *correct* values. The second phase takes longer than the first one. The learning rate is thereby smaller, that is 0.02. The neighborhood radius is also smaller on the average that is 4.

The first and second stages are used with 10000 and 50000 epochs, respectively. After those two phases of training, the map is ready to be tested and used in classification algorithm. Each input that is to be presented to classification algorithm is presented as follows:

1. All 800 proteins, 200 proteins from each class, are presented to the network at once.

2. Each input vector is compared to weight vectors associated with every node in 20x20 map.

3. The node having the weight vector with the smallest difference to the current input vector becomes the winning node.

4. The weight vector of this winning node is used as an input vector to the SVM classification method.

### 4.2.3 Application of SVM

After clustering, the weight vectors of the winning node for each protein input are presented to the SVM classification algorithm. For the classification, LIBSVM [42] is used as a computational tool.

Our choice of SVM as a classification method is motivated by a growing popularity of support vector machines (SVM) for classification problems. Their practical successes can be attributed to solid theoretical foundations based on VC-theory [43], since SVM generalization performance does not depend on the dimensionality of the input space. However, the quality of SVM models depends on a proper setting of SVM meta-parameters. Therefore, the main issue in this study is how to set these parameter values (to ensure good classification performance) for a given data set.

Given a training set of instance-label pairs $(y_i, z_i)$, $i = 1, \ldots, l$ where $y_i \in \Re^n$ and $z \in \{0, 1, 2, 3\}^l$, $l = 800$, the support vector machines require the solution of the following optimization problem:

$$\begin{gathered} min_{w,b,\xi}\ \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ subject\ to\ z_i(w^t \phi(y_i) + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{gathered} \tag{4.3}$$

Here training vectors $y_i$ are mapped into a higher dimensional space by the function $\phi$. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(y_i, y_j) \equiv \phi(y_i)^T \phi(y_j)$ is called the **kernel function**. The four basic kernels are:

- linear : $K(y_i, y_j) = y_i^T y_j$
- polynomial : $K(y_i, y_j) = (\gamma y_i^T y_j + r)^d,\ \gamma > 0.$
- radial basis function (RBF): $K(y_i, y_j) = exp(-\gamma \|y_i - y_j\|^2)\ , \gamma > 0.$
- sigmoid $K(y_i, y_j) = tanh(\gamma y_i^T y_j + r)$

Here $\gamma$,$r$ and $d$ are kernel parameters.
Initially, the following procedure is used:

- transform data to the format of an SVM software
- randomly try a few kernels and parameters
- test

Test results acccording to randomly choosen kernels and parameters are given in Table 4.6. We could not obtain reasonable accuracy even when test inputs are selected from training data set.

Table 4.6: Initial experiment results.

| #training data | #test data | # features | # classes | Kernel Type | Overall Accuracy |
|---|---|---|---|---|---|
| 400 | 200 | 20 | 0-1 | Linear | 23.5% |
| 400 | 200 | 20 | 2-3 | Linear | 20.4% |
| 400 | 200 | 20 | 0-1 | Polynomial | 45.7% |
| 400 | 200 | 20 | 2-3 | Polynomial | 40% |
| 800 | 400 | 20 | 0-1-2-3 | Linear | 12.4% |
| 800 | 400 | 20 | 0-1-2-3 | Polynomial | 15.2% |

In order to obtain reassonable accuracy in case test inputs are selected from training data set or in case test inputs are excluded from training data, practical recommendations are followed. These are:

- transform data to the format of an SVM software
- conduct simple scaling on the data
- consider each kernel alternative seperately and once the kernel is chosen, adjust two parameters: first, the width $\gamma$ of radial basis function kernels or the degree $d$ polynomial kernels, and second, the error weight C
- use cross-validation to find the best parameter C and $\gamma$
- use the best parameter C and $\gamma$ to train the whole training set

- test

**Scaling.** Scaling weight vectors (nodes corresponding to SOM entries) before applying SVM is very important. The main advantage is to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems. Each attribute of training andd testing data is scaled to the range [-1, +1].

**Kernel selection.** Experiments yield that RBF kernel gives better results than linear and polynomial kernels. The first reason of better performance is that the RBF kernel nonlinearly maps samples into a higher dimensional space, unlike the linear kernel. In our case, class labels and attributes is nonlinear.

The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel.

Finally, the RBF kernel has less numerical difficulties. One key point is $0 < K_{ij} \leq 1$ in contrast to polynomial kernels of which kernel values may go to infinity ($\gamma {y_i}^T y_j + r > 1$) or zero ($\gamma {y_i}^T y_j + r < 1$) while the degree is large.

**Kernel Parameters.** There are two parameters while using RBF kernels: C and $\gamma$. It is not known beforehand which C and $\gamma$ are the best for one problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good $(C, \gamma)$ so that the classifier can accurately predict unknown data (i.e., testing data). Note that it may not be useful to achieve high training accuracy (i.e., classifiers accurately predict training data whose class labels are indeed known). Therefore, a common way is to separate training data to two parts of which one is considered unknown in training the classifier. Then the prediction accuracy on this set can more precisely reflect the performance on classifying unknown data. An improved version of this procedure is $cross - validation$.

In v-fold cross-validation, we first divide the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v - 1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

In our study, a "grid-search" on C and $\gamma$ using cross-validation is applied. 15-fold cross-validation is done on the training data set. 15 subsets of size 100 are obtained and 4 subsets are tested using SVM trained on randomly chosen 15 subsets. "Grid-search" starts from a small C and $\gamma$ and iteratively $C$ and $\gamma$ are increased. Basically pairs of (C, $\gamma$) are tried and the one with the best cross-validation accuracy is picked. We found that trying exponentially growing sequences of C and $\gamma$ is a practical method to identify good parameters (for example, C $=2^{-5}, 2^{-3}, \ldots, 2^{15}$ *and* $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$).

After iterating $C$ and $\gamma$ values, the best $(C, \gamma)$ found is $(2^{3.25}, 2^{-5.25})$ with 77.6% cross-validation accuracy. 77.6% is the average percentage for the iteration of 15 subsets of training set. The total number of iteration for the optimization is 8772. 8772 is the total iteration number for finding the optimised values for $C$ and $\gamma$ values. Finally, the whole training set is trained again to generate the final classifier.

LIBSVM uses $one - against - all$ approach for 4 class prediction in which 4 SVM models are constructed and the $ith$ SVM is trained with all of the examples in the ith class with positive labels, and all other examples with negative labels.

Number of support vectors found for each class are listed in Table 4.7. As number of support vectors increases, the classification accuracy increases. Our goal in the classification is to find the optimal hyperplane for a given training set. There are active and inactive constraints. Active constraints correspond to the points from which optimal hyperplane is constructed. These active constraints are also called support vectors. Classification acccuracy could be evaluated with the analysis of the support vectors. From Table 4.7, *nuclear* is the class that is to be classified with minimum error rate.

Table 4.7: Number of support vectors.

| Class | Support Vectors |
|---|---|
| mTP | 203 |
| SP | 243 |
| nuclear | 278 |
| cytosolic | 225 |

## 4.3 Integrating Methodology

The aim of the combination of $feature\ extraction$, *clustering* and $classification$ methods is to design an accurate system that predicts the subcellular localization of proteins presented into the system. Our scheme for combining several methods is *cascading* or *serial combination* according to its architecture. In the cascading architecture, the output of a method serves as the input of the other model used.

Firstly, protein sequences are represented as sequences, $\widetilde{S}$. Then, PCA method is used to extract the most effective features from sequences, $\widetilde{S}$. The extracted features are the input vectors for the SOM method. SOM is applied to the extracted features and reference vectors for the corresponding sequences are obtained. Finally, the corresponding SOM reference vectors are used in the SVM input representations for each $\widetilde{S}$. This allows SVM to take into account correlations between clustered proteins using SOM and obtain more accurate results. The results are compared with the case of using extracted features directly for classification without clustering the extracted features. It is observed that the number of support vectors for each class given in Table 4.8 is less than the number of support vectors given in Table 4.7. In addition, the number of proteins that are misclassified is greater than that of proposed method.

Table 4.8: Number of support vectors without using SOM.

| Class | Support Vectors |
|---|---|
| mTP | 187 |
| sTP | 200 |
| nuclear | 267 |
| cytosolic | 195 |

### 4.3.1 Cross Validation

Before network training, the data sets are divided into equally sized parts for cross-validation. Each sequence participate either in the training or in the testing of a particular network, not both. For the SVM, the sets are constructed to contain equal numbers of positive and negative training sequences and the negative sequences consisted of approximately equal numbers of all the applicable nonpositive categories. All sequence exclusions are random.

Cross-validation is used to test the classsifier experimentally. Once classsifier is trained using cross-validation, the validation error gives an estimate of the accuracy of the final classifier on the unknown data test set.

In our study, 15-fold cross-validation is applied to SVM classifier. Our data set given in Table 4.1 contains 2318 proteins. From the data set, 26 disjoint sets of equal size 100 is obtained. The number of sets obtained from each class is given in Table 4.9. Randomly choosen 1 set from each class is used for testing and randomly choosen 2 set from each class is used for training. The classifier is trained 15 times, each time with a different set held out as a validation set. Validation set includes two subsets from each class for training and one subset from each class for testing. The estimated performance is the mean of these 15 errors.

Table 4.9: Number of subsets.

| Class | Subset Number |
|---|---|
| mTP | 3 |
| sTP | 7 |
| nuclear | 12 |
| cytosolic | 4 |

## 4.4 Evaluating Results

Performances were in general measured as percentage correctly predicted sequences, and as sensitivity (fraction of positive examples predicted as positives)

$$sens = \frac{tp}{tp + fn}$$

and specificity (fraction of all positive predictions that are true positives):

$$spec = \frac{tp}{tp + fp}$$

where $tp$ = true positives, $fn$ = false negatives (under prediction), and $fp$ = false positives (over prediction) and prediction accuracy (the full group of $tp$ and $tn$ where $tn$ = true negatives)

We performed the 15-fold cross validation method to estimate the prediction accuracy of our method: the data were randomly divided into twenty-six subsets of size 100, and randomly chosen two subsets from each class are used for training while randomly chosen one subset from each class is used for testing. The results for the twenty-two fold cross-validation test are summarized in Table 4.10.

Table 4.10: Prediction Performance.

| Class | Sensitivity (%) | Specificity(%) |
|---|---|---|
| mTP | 76 | 88.3 |
| sTP | 79 | 86.8 |
| nuclear | 90 | 74.3 |
| cytosolic | 80 | 86.9 |

The prediction accuracy for our method is 81.25% with 4.3013 standard deviation of 15-fold cross validation. Since we have used the full data set used

in TargetP [3], our results can be compared with TargetP. The overall prediction acccuracy for TargetP is approximately 85%. The results of these predictions are summarized in Table 4.11.

Table 4.11: Comparison of localization predictor performances.

| | Our Work | | | TargetP | | |
|---|---|---|---|---|---|---|
| | mTP | **sTP** | other | mTP | **sTP** | other |
| Sensitivity(%) | 76 | 79 | 85 | 80 | 96 | 88 |
| Specificity(%) | 88.3 | 86.8 | 80.6 | 67 | 92 | 97 |

### 4.4.1 Prediction of Protein Subcellular Localization using only local features

Our study is compared with the method based on local protein features [44] proposed for the identification of protein subcellular localization using local features of protein sequences. Study based on local protein features gives better results for the same data set used in our study for training and testing.

Our system reports 81.25% overall accuracy while system based on local protein features reports 90% accuracy.

Method based on local protein features gives better results than our prediction system based on global features of protein sequences. The main reason of the difference is that it predicts the subcellular localization of proteins in eukaryotic organisms based on the use of amino acid order. The amino acid order represents the local features such as the sequence order of amino acids that are given by motifs. It is interested in only local features. However, our system is interested in only global feature of proteins.

One of the most important performance measures include the splitting of the available samples to form training and test sets. Utilization of the available samples affects the performance of a classifier. The design of the method based on local features is based on the most frequent motifs for each protein. The most frequent motifs are found and used as features for classification. This can

cause problems when different data set is applied on the method based on local features. Classification algorithm may not be applicable for different kinds of data set. However, our system is designed in a generic way and can give similar accuracy results for different kind of sets.

## 4.5 Parameters effecting the performance

The performance of a prediction system depends on the interrelationship between sample sizes, number of features and complexity of the algorithms used in the prediction. There exist some parameters that affect the accuracy and performance of the preddiction.

One of the parameters that affects the measurement cost and system accuracy is the dimension of training and test data. Data sets in spaces of high dimensions cause computational problems. In our case, dimension of encoded protein sequences is initially 400. By the application of PCA, dimension is reduced to 20. The selection of principle components yields the reduced dimension. Variances approximating to 0 are excluded from principle component matrix. The threshold value 0 can be changed and the number of dimensions should be updated according to the selected principle components.

Another parameter that affects the system accuracy is related with clustering. Actually, a set of parameters are considered in SOM method. The set includes the dimension of the map, the type of the map and neighborhood function. In our study, several dimensions such as $5x5$, $10x10$ and $20x20$ of rectangular map are experimented. A rectangular map with dimension $20x20$ gives better clustering results than a hexagonal map with equal or smaller dimensions.

Finally, the parameters that affect the system accuracy are related with classification. In SVM, classification accuracy depends on the number of support vectors found during training. To increase the number of classified data points that is the number of support vectors, few parameters are to be chosen. Once the kernel type is choosen, we have only to adjust two parameters: the degree of polynomial kernels or the width of radial basis function kernels and the error

weight C. Compared with other kernel types, RBF gives better results. The non-linearity of data is classified via support vectors found in the training. The selection details of the width and error weight is given in Section 4.2.3.

## 4.6 Improving the system

As mentioned before, the data set used for TargetP [3] is used in our system. The occurrence frequencies of the amino acids are obtained from proteins of having four different subcellular localizations. Analysis of data set could be performed to remove inappropriate sequences. Redundant data could be reduced to avoid problems related to redundant data during training and testing.

Recently, SOM entry corresponding to the smallest difference with the training data is taken as a feature of classification. Feature generation algorithm could be changed to increase the performance of the classifier. Extraction of protein based rules using SOM entries could be done and training data applied to the classification algorithm could be generated from the extracted rules and classification performance could be improved.

Further improvement strategies on the selection of SVM parameters, in order to gain better understanding of the relationship between support vectors and the number of training samples, could be presented. As an example, instead of the heuristic "grid-search" method that is used for the selection of best the $C$ and $\gamma$, an advanced method which can save computational cost could be used.

Finally, our own data set could be generated and extracted from SWISS-PROT. The localization types of proteins and the number of proteins from each localization could be specified by our considerations. The nature of data could yield more accurate results. Nowadays, a human protein sequence data preparation is almost finished. After the data set is finalized, it can be applied to our system. Different clustering and classification algorithms could be applied on the new data set.

# CHAPTER 5

# CONCLUSIONS

In this thesis, we propose an approach to the protein subcellular localization problem, motivated by the explosive growth in available protein data. This approach formalizes protein subcellular localization as the identification of the proteins located in a compartment of a cell using global features. Two methods TargetP and prediction system that predicts protein subcellular localization based on local features of protein sequences are compared to discover the effects of the selection of training and testing data.

The proposed method can be summarized as follows. The data set is separated into training and test data. First, training data are tokenized in order to transform each protein sequence segment into a column vector of scores calculated from PAM similarity matrix. Column vectors have dimension 400. In order to cope with the problem of excessive dimensionality, PCA approach is used to reduce the dimension 400 to 20 by combining features. The most effective 20 features are obtained from the data set and used as inputs in the rest of the other methods applied. This reduction provides the extraction of valuable information necessary for the prediction and eliminates the information that does not change the outcome of our system drastically.

After the dimensionality is reduced from 400 to 20, an intermediate step is included in the proposed system. This step includes the natural grouping of

protein sequences with respect to their subcellular localizations. Natural groups are represented as "clusters" via SOM method. SOM is selected as a clustering method. The main reason of the selection is that the number of clusters are not known. Those clusters are organized as a set of nodes in a hypothetical network with a simple neighborhood structure on the nodes. The neighborhood structure is used in the construction of the features on which the classification is applied. Each node is represented with reference vectors adjusted during clustering. For each protein, the reference vectors of the node that have the smallest difference among the vectors of other nodes in the network are taken as a feature for classification. The reference vectors are applied to the SVM classification used in the prediction of subcellular localizations. SVM is applied to determine the label of the proteins with unknown subcellular localization. SVM is chosen due to the fact that SVMs, considered as *supervised* computer learning method, offer many advantages over the *unsupervised* methods. Only few parameters are to be chosen and we do not have to choose a complicated architecture as in neural networks. The usage of reference vectors obtained from SOM clustering increases the number of support vectors found in the classification. As the number of support vectors increases, the accuracy of classification increases. Thus, reference vectors increase the accuracy of classification of protein subcellular localizations. The reason of the better acccuracy performance of the proposed system is based on the use of combination of feature extraction and clustering methods on the encoded protein data and usage of the extracted and featured encoded data for the SVM classification.

The method is applied on the eukaryotic predictor data sets used in the construction and evaluation of TargetP by Emanuelsson et.al [3]. All data is extracted from SWISS-PROT [37]. A series of experiments are carried out to evaluate the performance of the method. Cross validation is done to obtain an average accuracy rate.

Although the system is designed to be supervised, it uses PCA as dimension reduction and an unsupervised method SOM to prepare the most appropriate

data for the SVM classification purposes.

The proposed system is attractive, because it allows us to preserve biological information via an encoding scheme and uses an integrative approach for the prediction. Principle component analysiss decreases the dimensionality so that clustering accuracy and performance increase with decreasing dimension. SOM adjusts each feature of protein sequences extracted via PCA so that features that violate seperability from different classes could be classified more accurately by SVM. In addition, although global features of protein sequences are used, the performance of our system is comparable with TargetP [3] that uses local features of protein sequences. The SVM classification algorithm is efficient in terms of complexity unlike neural networks used in TargetP.

The prediction of protein subcellular localization is one of the central problems of computational biology. Because many cellular functions are compartmentalized at specific regions of the cell, prediction of the localization of a protein can provide valuable information concerning its possible functions.

Because subcellular localization is in most cases encoded in the amino acid sequence of a protein, there are many sequence analysis methods for predicting the subcellular localization of proteins. Some methods are based on using global protein features and some methods are based on using local protein features.

Our proposed system predicts subcellular localization based on global protein features. As a future work, both the local and global protein features can be used in the prediction of subcellular protein prediction. It is known that the successful generalization of a prediction requires samples that contain discriminative properties. The quality of training and test samples can be improved with the inclusion of local and global protein features.

In addition, our own data set can be generated and those data can be applied to our proposed system. A number of methods use subcellular localizations of proteins of eukaryotes. As a future work, *human* proteins can be extracted from SWISS-PROT and prediction of subcellular localization of *human* proteins based on local and global human protein features can be developed. Dealing with

subcellular localizations of *human* proteins can provide valuable information for the identification of functions of *human* proteins.

Finally, it can be possible to include biological knowledge in the proposed system via defining new SVM kernels. SVM kernel functions can be defined with the inclusion of characteristics of protein data and can be used in the classification of subcellular localizations of proteins.

# REFERENCES


[1] Andreas D. Baxevanis and B.F. Francis Quellette. *Bioinformatics*. A John Wiley and Sons, Inc, 1998.

[2] B. Weiss. H.-W. Mewes, H. Seidel. *Bioinformatics and Genome Analysis*. Springer, 2002.

[3] O. Emanuelsson, H. Nielson, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *J. Mol. Biol.*, 300(4):1005–16, 2000.

[4] E.S. Lander, B. Birren L.M. Linton, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, and W. FitzHugh. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[5] J.T.L. Wang, S. Rozen, B.A. Shapiro, D. Shasha, Z. Wang, and M. Yin. New techniques for dna sequence classification. *Journal of Computational Biology*, 6(2):209–218, 1999.

[6] J.D. Watson, M. Gilman, J. Witkowski, and M. Zoller. *Recombinant DNA*. W.H Freeman, New York, 2nd Edition, 1992.

[7] http://www.ornl.gov/hgmis/publicat/tko/index.htm/ as accessed on, August 2003.

[8] Jean-Michel Claverie. Computational methods for the identification of genes in vertebrate genomic sequences. *Human Molecular Genetic*, 6(10):1735–1744, 1997.

[9] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[10] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[11] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[12] M. E. Tipping and C. M. Bishop. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.

[13] R. Staden. The current status and portability of our sequence handling software. *Nucleic Acids Res.*, 14:217–237, 1986.

[14] R. Legouis, J.P. Herdelin, J. Levilliers, J.-M. Claverie, S. Compain, V. Wunderle, P.Millasseau, D. Le Paslier, D. Cohen, and D. Caterina. The candidate gene for the x-linked kallman syndrome encodes a protein related to adhension molecules. *Cell*, 67:423–435, 1991.

[15] E.C. Uberbacher and R.J. Mural. Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach. *Nucleic Acad. Sci.*, 88:11261–11265, 1991.

[16] R. Guigo, S.Knudsen, N. Drake, and T.Smith. Prediction of gene structure. *J. Mol. Biol.*, 226:141–157, 1992.

[17] S. Dong and D.B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.

[18] P.Tamayo. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci.*, 96:2907–2912, 1999.

[19] http://www.cs.jhu.edu/labs/compbio/morgann.html as accessed on, March 2003.

[20] S. Salzberg. Locating protein coding in human dna using a decision tree algorithm. *J. Comput. Biol.*, 2:473–485, 1995.

[21] E.E Synder and G.D. Stormo. Identification of protein coding regions in genomic dna. *Journal of Mol. Biol.*, 248:1–18, 1995.

[22] E.E. Snyder and G.D. Stormo. Identification of coding regions in genomic dna sequences: an application of dynamic programming and neural networks. *Nucleic Acids Res.*, 21:607–613, 1993.

[23] http://genemark.biology.gatech.edu/genemark as accessed on, July 2003.

[24] http://genes.mit.edu/genscan.html as accessed on, June 2003.

[25] S.J Hua and Z.R Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17:721–728, 2001.

[26] http://www.bioinfo.tsinghua.edu.cn/subloc as accessed on, August 2003.

[27] H. Nielson, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and other protein sorting signals. *Protein Eng.*, 10:1–6, 1997.

[28] H. Nielson, S. Brunak, and G. von Heijne. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng.*, 12:3–9, 1999.

[29] C. Pasquier and S.J. Hamodrakas. An hierarchical artificial neural network system for the classification of transmembraane proteins. *Protein Engineering*, 12(8):631–634, 1999.

[30] M.G Claros. Mitoprot: a macintosh application for studying mitochondrial proteins. *Computer Applications in the BioSciences*, 11(4):441–447, 1995.

[31] K.-C. Chou and D. Elrod. Protein subcellular location prediction. *Protein Eng.*, 12:107–118, 1999.

[32] M. Nielsen and A. Krogh. Prediction of signal peptides and signal anchors by a hidden markov model. *Intell. Syst. Mol. Biol.*, 6:122–130, 1998.

[33] Y. Fujiwara and M. Asogawa. Prediction of subcellular localization using amino acid composition and order. *Genome Informatics*, 12:103–112, 2001.

[34] K. Nakai and M. Kanehisa. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics*, 14(4):897–911, 1992.

[35] K. Nakai. Protein sorting signals and prediction of subcellular localization. *Advances in Protein Chemistry*, 54:277–344, 2000.

[36] O. Emanuelsson, Henrik Nielsen, and Gunnar von Heijne. Chlorop, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science*, 8:978–984, 1999.

[37] A. Bairoch and R. Apweiler. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucl. Acids Res.*, 28:45–48, 2000.

[38] http://www.expasy.org/sprot as accessed on, August 2003.

[39] W.A. Pearson. Rapid and sensitive sequence comparison with fastp and fasta. *in Methods in Enzymology, ed. R. Doolittle (ISBN 0-12-182084-X, Academic Press, San Diego)*, 183:63–98, 1990.

[40] http://mathlab.com as accessed on, July 2003.

[41] S.F Altschul. Amino acid substitution matrices from an information theoreticc perspective. *J. Mol. Biol.*, 219:555–565, 1991.

[42] C.-C Chang and C.-J. Lin. Libsvm software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.

[43] V.N Vapnik. *Statistical Learning Theory.* New York: John Wiley & Sons, 1998.

[44] M. Ozarar. Prediction of protein subcellular localization based on primary sequence data. Master's thesis, METU, 2003.

# APPENDIX A

# SUPPORT VECTOR MACHINES

## A.1 Large Margin Hyperplanes

The first step in the construction of SVMs is classification by simple linear hyperplanes. Consider the class of hyperplanes $\langle w, x\rangle + b = 0$, $w \in \Re^n$, $b \in \Re$, corresponding to decision functions

$$f(x) = sign(\langle w, x\rangle + b) \tag{A.1}$$

At first, the case of linear separable data is considered. There are many possible ways in which a hyperplane can separate the two classes. Hence, a criterion to choose "the best one", the "optimal" separating hyperplane is required. The idea of learning from examples is to gain an impression of what the elements of a class look like by examining the training points that belong to that class. All new data points are thought to lie somewhere near the known training data. Hence, the hyperplane should be chosen such, that small shifts of the data do not result in changing predictions. If the distance between the separating hyperplane and the training points becomes too small, even test examples very close to the training samples may be classified incorrectly.

A learning algorithm is proposed for separable problems, called the *Generalized Portrait*, that constructs a hyperplane ysielding the maximum margin of separation between the classes:

$$max_{w,b} = min\{\|x - x_i\| : x \in \Re^n, \langle w, x\rangle + b = 0, i = 1, \ldots, l\} \tag{A.2}$$

Next effective way to construct this hyperplane is considered. Beginning with separating hyperplanes constructed on linear separable datasets and then generalizing the results allowing for errors on the training set.

### A.1.1 Hard Margin

**Seperability.** The training set

$$X = \{(x_1, y_1), \ldots, (x_l, y_l) : x_i \in \Re^n, y_i \in \{-1, +1\}\}$$

is called separable by the hyperplane $\langle w, x\rangle + b = 0$, if there exist both a unit vector $w(\|w\| = 1)$ and a constant b such that the inequality holds true:

$$y_i(\langle w, x_i\rangle + b) - 1 \geq 0 \quad i = 1, \ldots l \tag{A.3}$$

The hyperplane $H$ defined by $w$ and $b$ is called a separating hyperplane. The margin $\gamma_i(w, b)$ of a training point $x_i$ is defined as the distance between $H$ and $x_i$ : $\gamma_i(w, b) = y_i(\langle w, x_i\rangle + b)$. The margin $\gamma_s(w, b)$ of a set of vectors $S = \{x_1, \ldots, x_n\}$ is defined as the minimum distance from $H$ to the vectors in $S$:

$$\gamma(w, b) = min_{x_i \in S}\gamma_i(w, b)$$

**Optimal Separating Hyperplane.** Consider the unit vector $w^*$ and the constant $b^*$ which maximize the margin of the training set $\gamma_x(w, b)$ under the condition that inequality A.3 is satisfied. The pair $(w^*, b^*)$ determines the hyperplane that separates the positive from the negative examples and has maximal margin. This hyperplane is called the *Maximal Margin Hyperplane* or the *Optimal Separating Hyperplane*(OSH).

**Construction of OSH.** The Optimal Separating Hyperplane is the solution of the optimization problem

$$\begin{gathered} maximize \quad \gamma_x(w,b) \\ subject\ to\ \gamma(w,b) > 0\ and\ \|w\|^2 = 1 \end{gathered} \tag{A.4}$$

This poses several difficulties. The objective function is neither linear nor quadratic and the constraints are nonlinear. Hence, from an algorithmical point of view this is a difficult optimization problem. But one can find an effective method for constructing the OSH by considering an equivalent statement of the problem:

$$\begin{gathered} minimize\ \tfrac{1}{2}\|w\|^2 \\ subject\ to\ y_i(\langle w, x_i\rangle + b) - 1 \geq 0 \quad i = 1, \ldots, l \end{gathered} \tag{A.5}$$

**Lagrangian.** To solve the optimization problem by the Lagrange method we try to find the saddle point of the Lagrangian

$$L_P(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i[y_i(\langle w, x_i\rangle + b) - 1] \tag{A.6}$$

where $\alpha_i \geq 0$ are the Lagrange multipliers. The Lagrangian $L_P(w,b,\alpha)$ has to be minimized with respect to the primal variables $w$ and $b$ and maximized with respect to the dual variables $\alpha_i$. The dual problem can now be formulated as follows: Find multipliers $\alpha_i$ which solve

$$\begin{gathered} Maximize\ L_D(\alpha) = \textstyle\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle\,, \\ subject\ to\ \alpha_i \geq 0\ \forall i, \ \ and\ \ \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0 \end{gathered} \tag{A.7}$$

To construct the OSH one has to find the coefficients $\alpha_i^*$ that solve the dual problem. As a solution we obtain

$$w^* = \sum_{i=1}^{l} \alpha_i^* y_i x_i \tag{A.8}$$

In the labelling of the Langrangian $P$ stands for "primal" and $D$ for "dual". Notice that $L_P$ and $L_D$ arise from the same objective function but with different constraints. The solution is found by minimizing the primal $L_P$ or by maximizing the dual $L_D$.

The construction of the OSH amounts to maximizing $L_D$ with respect to the $\alpha_i$, subject to A.3 and positivity of the $\alpha_i$. The solution is given by A.8. An

important observation is that the training points enter the algorithm only as entries in an inner product. So the influence of the training set can be brought together in the so called Gram matrix $G = (\langle x_i, x_j \rangle)$. This will play a central role in the next chapter, where we show how to generalize the concept of separating hyperplanes to nonlinear decision surfaces.

**Support Vectors.** Every solution $(w, b)$ of A.5 fullfills the *Kuhn − Tucker complementarity condition*

$$\alpha_i[(\langle w, x_i \rangle + b) - 1] = 0 \; \forall i$$

This condition states, that for a given training point $x_i$ either the corresponding Lagrange multiplier $\alpha_i$ equals zero, or $x_i$ lies on one of the hyperplanes $H_1 = \{x : \langle w, x \rangle + b = +1\}$ or $H_2 = \{x : \langle w, x \rangle + b = -1\}$. These hyperplanes are called the *margin hyperplanes*. They contain the training points with the minimal distance to the OSH, they form the boundary of the margin. The vectors that lie on $H_1$ or $H_2$ with $\alpha_i > 0$ are called *Support Vectors* (SV).

Another application of the Kuhn-Tucker complementarity condition is the computation of the threshold $b$. While $w$ is explicitly determined by the training procedure, the constant $b$ is not.For finding $b$ choose any i for which $\alpha_i \neq 0$; then A.3 becomes an equality to compute $b$. Of course, it is numerically safer to compute $b$ for all $i$ and take the mean value of the individual $b$ values.

**Test Phase.** How to use the separating hyperplane, once we have trained it on the training set? The hyperplane divides the $\Re^n$ into two regions: one where $\langle w^*, x \rangle + b^* > 0$ and one where $\langle w^*, x \rangle + b^* < 0$. To use the maximal margin classifier, we determine on which side the test pattern lies and assign the corresponding class label. Hence, the predicted class of a test point $x$ is the output of

$$\begin{aligned} f(x) &= sign(\langle w^*, x \rangle + b^*) \\ &= sign(\textstyle\sum_{i=1}^{SV} \alpha_i y_i \langle x_i^{SV}, x \rangle + b^*). \end{aligned} \tag{A.9}$$

The optimal separating hyperplane obtained by solving the margin optimization problem is a very simple special case of a SVM, because the computation is

done directly on the input data. Before we generalize the support vector concept to nonlinear classifiers and introduce the *kernel mapping*, we will demonstrate how a separating hyperplane can be adapted to the case of linear non-separable datasets.

### A.1.2 Soft Margin

The algorithm worked out above cannot be used in many real-world problems. In general, noisy data will render linear separation impossible. No feasible solution to the margin maximisation problem can be found, due to the objective function (i.e. the dual Langrangian) growing arbitrarily large. The main drawback of the OSH is, that it allows for no classification errors. Either we get a solution without any training errors or no solution at all. To generalize the concept of the maximal margin hyperplane we relax the separability constraint A.3. Each exceeding of the constraints will be punished by a misclassification penalty (i.e. an increase in the primal objective function).

**Slack Variables.** This punishment can be achieved by introducing positive slack variables $\xi_i (i = 1, \ldots, l)$ in the constraints, which then write:

$$y_i(\langle w, x_i + b) - 1 + \xi_i \geq \quad i = 1, \ldots, l. \tag{A.10}$$

The smallest slack variable for which A.10 is satisfied, i.e

$$\xi_i = max\{0, 1 - y_i(\langle w, x_i \rangle + b)\},$$

measures how much a point fails to have a margin of $1 \mid \|w\|$. The value of $\xi_i$ indicates, where $x_i$ lies compared to the separating hyperplane. If $\xi_i \geq 1$ then $x_i$ is misclassified; if $0 < \xi_i < 1$, then $x_i$ is classified correctly, but lies inside the margin; and finally, if $\xi_i = 0$, then $x_i$ is classified correctly and lies outside the margin or on the margin boundary.

A classification error will be evidenced by the corresponding $\xi_i$ exceeding unity. So $\sum_{i=1}^{l} \xi_i$ is an upper bound on the number of training errors. Simultaneous maximization of the margin and minimization of the number of

misclassifications is achieved by changing the objective function from $\frac{1}{2}\|w\|^2$ to $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i^k$:

$$\begin{gathered} Minimize\ \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^{l}\xi_i^k, \\ subject\ to\ y_i(\langle w, x_i\rangle + b) - 1 + \xi_i \geq 0 \quad i = 1, \ldots, l \\ \xi_i \geq 0 \quad i = 1, \ldots, l \end{gathered} \tag{A.11}$$

To construct the OSH one has to find the coefficients $\alpha_i^*$ that solve the dual problem. As a solution we obtain

$$w^* = \sum_{i=1}^{l} \alpha_i^* y_i x_i \tag{A.12}$$

**Error Weight.** The *error weight* C is a parameter to be chosen by the user, it measures the size of the penalties assigned to errors. In practice, C is varied through a wide range of values and the optimal performance is usually assessed by cross-validation. The optimization problem is convex for any positive integer k, for $k = 2$ and $k = 1$ it is also a quadratic programming problem. This approach is called the *Soft Margin Generalisation* of the OSH, while the original concept with no errors allowed is called *Hard Margin*. The optimization problem to be solved in the Soft Margin case is only a minor variation of A.7. For $k = 1$, the primal Lagrangian for this problem becomes

$$L_P(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(y_i(\langle w, x_i) + b) - 1 + \xi_i) - \sum_{i=1}^{l}\beta_i\xi_i$$

with $\alpha_i$ and $\beta_i \geq 0$. The $\beta_i$ are the Lagrange multipliers introduced to enforce $\xi_i \geq 0$. This results in the following dual formulation:

$$\begin{gathered} Maximize\ L_D(\alpha) = \textstyle\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle, \\ subject\ to\ 0 \leq \alpha_i \leq C\ \forall i,\ \ and\ \textstyle\sum_{i=1}^{l}\alpha_i y_i = 0 \end{gathered} \tag{A.13}$$

The only difference to A.7 is that the langrange multipliers have an upper bound of C. Non-zero slack variables can only occur for $\alpha_i = C$. Points for which $0 < \alpha_i < C$ lie on one of the two margin hyperplanes. The fact that the Lagrange multipliers are upper bounded gives rise to the name *box constraint*

for this formulation, as the vector $\alpha_i$ is constrained to lie inside the box with side length C in the positive orthant. The result of this section is, that the construction of the Optimal Separating Hyperplane can be adapted to the case of non-separable data with only minor changes to the algorithm.

## A.2 Kernel Trick

It has shown how classfiers linear in input space can easily be computed by standard optimization techniques. Although linear classifiers are easy to handle, they pose severe restrictions on the learning task. The target concept may be too complex to be expressed as a linear combination of the given attributes. This problem can be overcome by an approach called *kernel technique* introduced as the method of potential functions. The general idea is to map the input data to a high dimensional space and separate it there by a linear classifier. This will result in a classifier nonlinear in input space.

**Kernel Trick.** Given a mapping $\Phi : L \to H$ from input space $L$ to an (inner product) feature space $H$, the function $k : LxL \to \Re$ is called a kernel function, iff for all $x, z \in L$

$$k(x, z) = \langle \Phi(x), \Phi(z) \rangle_H \tag{A.14}$$

The kernel function behaves like an inner product in $H$, but can be evaluated as a function in $L$. This allows to side-step possible computational problems inherent in evaluating the feature map. Choosing a kernel-function will implicitly define a mapping $\Phi$. Both learning and training step only depend on the value of inner products in feature space. As a consequence, we do not need to know the underlying feature map to be able to solve the learning task in feature space, we only have to choose a kernel function. Most commonly used are

$$polynomal \quad k(x, z) = (\langle x, z \rangle + 1)^P \tag{A.15}$$

$$radial\ basis\ function \quad k(x, z) = exp(-\|x - z\|^2)/2\sigma^2), \tag{A.16}$$

where the parameters p (degree of polynomial) and $\sigma$ (width of rbf) are to be chosen by the user. Complex kernels can be built from simple ones.

**Support Vector Machines.** One merely has to combine the Optimal Separating Hyperplane with the kernel-induced mapping to a high dimensional feature space. Roughly speaking, a Support Vector Machine is a high dimensional Maximal Margin Hyperplane evaluated in the input space. Now, use an inner product in the feature space generated by a kernel function to restate our previous results in the input space. To construct a SVM we have to solve the following optimization problem, where $C = \infty$ leads to a Hard Margin SVM and $C < \infty$ to a Soft Margin SVM.

$$\begin{gathered} Maximize\ L_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j)\,, \\ subject\ to\ 0 \leq \alpha_i \leq C\ \forall i,\ \ and\ \sum_{i=1}^{l} \alpha_i y_i = 0 \end{gathered} \tag{A.17}$$

## A.3 Modification

**Multiple Classes.** There are many ways to apply the above SVM to $n > 2$ classes. Most commonly used is a $one - against - all$ approach, also called $winner - takes - it - all$. Every class is separated by a SVM from the pooled datapoints of all other $n$ hyperplanes and assigns it to the class, where it achieves maximum distance