



Yarmouk University

Department of Computer Engineering

**New Fast Algorithm to Detect and localize
Zinc Finger Protein Sequence**

M.Sc. Thesis

By

Omar Ali Al-Howari

Supervisor

Dr.Awad Al-Zaben

Co- Supervisor

Dr.Khalid Al-Batayneh

December 2010

New Fast Algorithm to Detect and localize Zinc Finger Protein Sequence

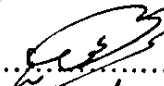
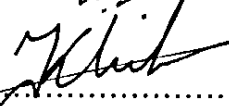
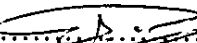


By

Omar Ali Al-Howari

B.Sc. Electrical Engineering, Jordan University of Science and Technology, 1989

**A thesis submitted in partial fulfillment of the requirements for the
degree of Master of Science, in the Department of Computer
Engineering, Yarmouk University, Irbid, Jordan**

Approved By:

<i>Dr. Awad Al-Zaben</i>		<i>Chairman(Supervisor)</i>
<i>Dr. Khaled Al-Batayneh</i>		<i>Member(Co-Advisor)</i>
<i>Prof. Saed AL-Refaae</i>		<i>Member</i>
<i>Dr. Shadi Alboon</i>		<i>Member</i>
<i>Dr. Maryam Nuser</i>		<i>External Examiner</i>

December 2010

DEDICATIONS

I Dedicate this work to who lit the first candle in my life; my father "may Allah bless his soul" and my mother "may Allah prolongate her life"

To my Wife Hanan, my sons Hesham, Rama, Leen, and Karam.

To my Advisors Dr. Awad Al-Zaben and Dr.Khaled Al-Batayneh For their support....

ACKNOWLEDGMENTS

First of all many thanks to Allah for giving me the chance to present this research. I would like to take this opportunity to express my gratitude and sincere appreciation towards Dr. Awad Al-Zaben and Dr.Khaled Al-Batayneh my M.Sc. academic advisors for all their support and their helpful suggestions. Also I would like to thank the thesis discussion committee represented by:

Prof. Saed AL-Refae, Dr. Shadi Alboon and

Dr. Maryam Nuser for their active participation and their constructive comments and suggestions throughout this work.

I also want to send my warm greetings to all my friends at Yarmouk University

Omar A. Alhawari

December 2010

TABLE OF CONTENTS

DEDICATION.....	III
ACKNOELEDGEMENTS.....	IV
TABLE OF CONTENTS.....	V
LIST OF FIGURES.....	X
LIST OF TABLES.....	XVI
ABSTRACT.....	XVII

Chapter One: Introduction

1.1 Background.....	1
1.1.1 The DNA.....	1
1.1.2 Amino Acids.....	2
1.1.3 Proteins.....	3
1.2 Zinc Finger.....	4
1.3 The Problem.....	5
1.3.1 Importance of Zinc Finger Searching.....	5
1.3.2 Objectives	7

Chapter Two: Literature Survey

2.1 Review of Searching Algorithms.....	8
2.2 Algorithms.....	8
2.3 Searching and Matching Algorithms of Protein Sequences	8
2.3.1 BLAST and FASTA Tools.....	9
2.3.2 Protein Sequence Alignment	9
2.4 String Matching	10
2.5 Pattern Recognition Using DSP and Neural Network.....	11

Chapter Three: Methodology

3.1 Algorithm Block Diagram	13
3.2 Read Sequence.....	13
3.3 Digital Mapping for Amino Acids	14
3.3.1 Digital Coding For Amino Acids.....	14
3.3.2 Mapping characterization in Time -Frequency Domain.....	17
3.3.2 Short Time Fourier Transform of Mapped Sequences.....	20
3.4 Candidate Zinc Finger Detection	27
3.4.1 Input Segmentation	27
3.4.2 Segmentation Enhancement	27
3.5 Zinc Finger Determination and Localization.....	32
3.5.1 The Most Popular Fast Algorithms.....	32

Chapter Four: The Classifier

4.1 Neural network Classifier	33
4.1.1 Introduction.....	33
4.1.2 Neurons	34
4.1.3 Multilayer Neural Networks	35
4.1.4 Feed-Forward Neural Networks.....	35
4.1.5 Characteristics of the Neural Network	36
4.2 Mapping and loading the data to the Neural Network	36
4.2.1 Input and Target Feature Vectors and Classes	38
4.3 Method of Creation Features for Feeding Data.	38
4.3.1 Statistical features	38
4.4 Structure of Created Neural Network	39
4.4.1 Training process of the Neural Network	40
4.4.2 Receiver Operating Characteristic (ROC).....	43
4.5 The Output of The Neural Network	44

Chapter Five: Results and Discussion

5.1 Introduction.....	52
5.2 The Output of the Segmentation	52
5.3 The Output of The Neural Network:	52
5.3.1 Detection of Zinc Fingers	
5.3.2 Detection the Percentage of the Four Class Types of the Zinc Fingers.....	57
5.3.3 Detection of Amino Acids.....	59
5.4 Statistical Calculations	61
5.4.1 Sensitivity and Specificity	61
5.4.2 More calculations (α and β)	62
5.4.3 Time of Detection the location and number of Zinc Fingers...	63
5.4.4 Confusion Matrix and Receiver Operating Characteristic (ROC).....	65
5.5 Thesis Outcomes	69

LIST OF FIGURES

<u>Figure</u>	<u>Description</u>	<u>Page</u>
Figure 1.1	The genetic composition of humans	1
Figure 1.2	a: Shape of ZFP. b: ZFP linked with the DNA.....	5
Figure 1.3	The location of zinc finger code named CMFH...GFC in some proteins	6
Figure 3.1	The block diagram of the new fast algorithm.....	13
Figure 3.2	Location of zinc finger in a specific protein signal.....	14
Figure 3.3	The new modified weight for the 20 amino acids.....	17
Figure 3.4	Short protein signal after the mapping zinc fingers.....	18
Figure 3.5	Short protein signal after the differentiator.....	18
Figure 3.6	Long protein signal after the mapping	19
Figure 3.7	Long protein signal after the differentiator	19
Figure 3.8	The contour for short protein sequence that has only two zinc fingers	22
Figure 3.9	The contour for middle protein sequence that has only some of zinc fingers.....	2
Figure 3.10	The contour for middle protein sequence that has many of zinc fingers.....	23
Figure 3.11	The contour for long protein sequence and small f_s	24
Figure 3.12	The contour for the same long protein sequence and large f_s ..	24
Figure 3.13	The contour For a very large protein sequence that has a lot of zinc fingers	25
Figure 3.14	The spectrogram for a middle of protein sequence	25
Figure 3.15	The spectrogram for long protein sequence	26

<u>Figure</u>	<u>Description</u>	<u>Page</u>
Figure 3.16	Image plot for protein sequence data after differentiating shows-in red color-the expected to be zinc ones.	26
Figure 3.17	The Segmentation Enhancement Method	29
Figure 3.18	Segmentation method for one protein. It shows 6 expected zinc fingers with there locations at the bottom of the figure...	30
Figure 3.19	Segmentation method for another protein that shows 8 expected zinc fingers with there locations at the bottom of the figure.....	31
Figure 3.20	Segmentation method for long protein sequence that gives all the candidate zinc fingers. It shows the zinc fingers and the expected ones with there locations at the bottom of the figure.	32
Figure 4.1	General neural network function where the weight is adjusted to make the output matched the target	33
Figure 4.2	The basic neuron model	34
Figure 4.3	Data processing before fed to the input of the neural network	37
Figure 4.4	Block diagram of pattern recognition to classify input signal to N classes.....	38
Figure 4.5	The starting box for the feed-forward neural network. It consists two layers; hidden and output layer. The hidden layer has Size=20.....	40
Figure 4.6	The training box for the feed-forward neural network. It shows all the data we need like the time, performance, training state, confusion matrix etc.....	41
Figure 4.7	Box plot for the performance for the feed-forward neural network shows test and validation errors have almost similar characteristics.	42
Figure 4.8	The validation and the gradient box for the feed-forward neural network. It shows that the training process stopped when the validation error increased six iterations	42
Figure 4.9	All confusion matrices, training, test and validation. They show an average percentage for each test in this sample.....	43
Figure 4.10	The Receiver Operating Characteristic (ROC) curve for 3 classes as an example. The more that these curves concave up, the better is the recognition process	44

<u>Figure</u>	<u>Description</u>	<u>Page</u>
Figure 4.11	The output of the neural network. It shows some of the results that were detected in this algorithm including their class type, location and number of repetition.....	45
Figure 4.12	Test to see the right and wrong detection for 29 zinc finger proteins which shows that they are all correct	46
Figure 4.13	Test to see the right and wrong detection for another zinc finger proteins. It shows that there are two wrong classes	46
Figure 4.14	Percentage of each type of the 4 types of zinc fingers in the testing sets	47
Figure 4.15	Percentage of each type of the 4 types of zinc fingers in the training sets	48
Figure 4.16	Percentage of each type of the 4 types of zinc fingers	48
Figure 4.17	The percentages of each amino acids in the training sets.....	49
Figure 4.18	The regression plot for network P[-1,+1] with T[1,5] values...	50
Figure 4.19	The regression plot for network output with T. It shows that for this sample R=0.99672.	50
Figure 4.20	The regression plot for network output with T. It shows that more times of training process leads to more accurate results with regression value R=0.9993.....	51
Figure 5.1	Segmentation method for one protein with the beginning and lasting for expected zinc shown below of them. It contains 6 possible zinc fingers	52
Figure 5.2	Segmentation method for another protein that contains 6 possible zinc fingers.....	53
Figure 5.3	Segmentation method for protein that gives all the candidate zinc fingers and some of them are not zinc ones.....	54
Figure 5.4	The output of the neural network. It shows the some results that was detected in this algorithm	55
Figure 5.5	Test to see the right and wrong detection for 29 zinc finger proteins which are all correct.....	56

<u>Figure</u>	<u>Description</u>	<u>Page</u>
Figure 5.6	Test to see the right and wrong detection for another zinc finger proteins. It shows two wrong classes.	56
Figure 5.7	Percentage of each type of the 4 types of zinc fingers with respect to others in training sets of protein sequences	57
Figure 5.8	Percentage of each type of the 4 types of zinc fingers with respect to others in testing sets of protein sequences.	58
Figure 5.9	Percentage of each type of the 4 types of zinc fingers with respect to others in validation sets of protein sequences.....	58
Figure 5.10	Percentage of each amino acid with respect to the others in all proteins in the training sets.....	59
Figure 5.11	Percentage of each amino acid with respect to the others in all proteins in the validation sets.	60
Figure 5.12	Percentage of each amino acid with respect to the others in all proteins in the testing sets.....	60
Figure 5.13	The performance of the training of neural network	64
Figure 5.14	The performance of the testing of neural network	65
Figure 5.15	The performance of the validation of neural network	65
Figure 5.16	The regression plot for network output with T. It shows that more times of training process leads to more accurate results with regression value $R=0.9993$	65
Figure 5.17	The confusion matrix for the 5 classes with an overall percentage average for all tests shown in blue color.....	66
Figure 5.18	The ROC curve for a small test sample. It shows low sensitivity.	67
Figure 5.19	Increasing the number of the sample is more perfect classifier. It shows high sensitivity.....	68

LIST OF TABLES

<u>Table</u>	<u>Description</u>	<u>Page</u>
1.1	The universal standard genetic code	2
1.2	Example of DNA sequence and the corresponding amino acid name	2
1.3	The 20 Amino Acids and Their Codes	3
3.1	The 20 Amino acids and their digital coding starts from 1 to 30	15
3.2	The 20 Amino acids and their digital coding starts from 6 to 30	15
3.3	The 20 Amino acids and their digital coding depending on the electron-ion interaction potential (EIIP).....	15
3.4	The new coding for this algorithm.....	16
5.1	The positive and negative test values.....	61
5.2	The calculated results for sensitivity and specificity.	62
5.3	α and β calculations.	63
5.4	The performance of the neural network and how fast and accurate was the detection of location and number of zinc fingers.	63

ABSTRACT

New Fast Algorithm to Detect and localize Zinc Finger Protein Sequence

By: Omar Ali Al-Howari

The problem of determination the number and location of Zinc Finger (ZF) in protein sequences is a very important procedure since this motif protein permits interaction with DNA. Zinc-finger proteins regulate the expression of genes and they have an essential role in forming protein-protein interactions.

Protein sequences are very huge numbers that increased rably every time. The literature represented dealing with these motifs in two methods; the signal processing and the neural networks. First; this method represented the starting way for detecting motif protein sequence, which is reading these protein sequences from certain available sites, then converting them into signals.

The signal processing before the candidate stage has primarily focused on assigning numerical values to the protein sequences in a new coding method, and then used a mapping method in time - frequency domain.

Second, a candidate stage was used to make fast decision whether this sequence is a Zinc finger or not with the helpful of mapping, depending on a segmentation method that gets all the candidate zinc fingers in these sequences. Then feature extraction method was used in order to feed these candidate zinc fingers to a classifier stage.

In the third stage, a classification method was used. This stage was used in order to satisfy the confirmation of this ZF, which means rising the probability of detection ZF to high level using high level precise stage; here it was the neural network classifier which gives high sensitivity and enhance positive predictive value.

The final stage was the determination of the number and location of the required Zinc Fingers through out the specific protein sequences with high precision, hoping that these results will guide to a new researches and theorems.

Keywords: *Zinc Finger protein , Pattern recognition, Pattern sequence, Motif detection, Neural networks, Text searching, String Matching, Machine Learning, Short time fast Fourier transform.*

Introduction

1.1 Background

1.1.1 DNA

It is well known that the molecule of life consists of trillion of cells. Each human cell contains 46 chromosomes found in 23 pairs to build 2 meters of genetic material called DNA as demonstrated in Figure 1.1.

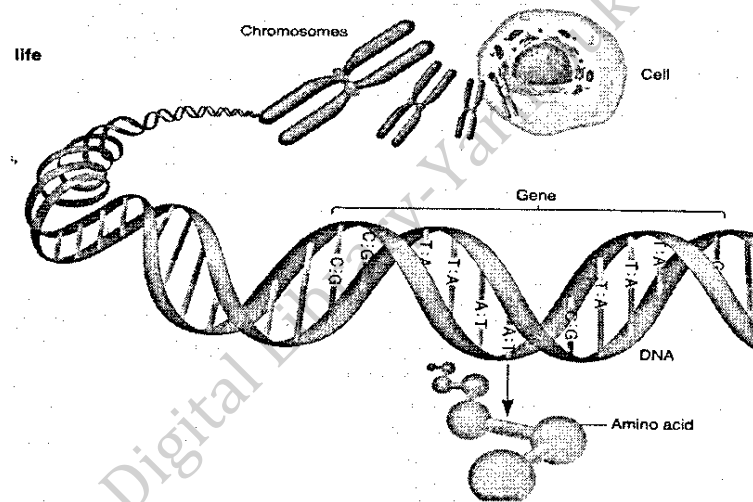


Figure 1.1 The genetic composition of humans [1]

DNA stores the information for protein synthesis of all proteins. That means the main function of the genetic is to code the production of cellular proteins in the correct cell, at the proper time and in suitable amounts. This is an extremely complicated task because living cells make thousands of different proteins. DNA's ability to store information is based on its molecular structure. DNA is composed of a linear sequence of nucleotides contains one nitrogen-containing base, either adenine (A), thymine (T), guanine (G) or cytosine (C). The linear order of these bases along a DNA contains information similar to the way that the groups of alphabetic letters represent words [2].

DNA sequences within most genes contain the information with respect to the order of amino acids. According to the genetic code; the code of a three-base sequence specifies the particular amino acid. In this way, the DNA can store the information to specify the proteins that made by an organism.

The 20 symbols amino acids composed of 64 different codons with respect to the following equation:

$$\text{No. of - possible - codons} = \text{size - of - DNA - alphabet}^{\text{NO. of - bases - in - codons}} = 4^3 = 64 \dots \dots \dots 1.1$$

Tables 1.1 and 1.2 show these codes and an example of a DNA sequence.

	T	C	A	G
T	TTT Phe (F)	TCT Ser (S)	TAT Tyr (Y)	TGT Cys (C)
	TTC Phe (F)	TCC Ser (S)	TAC Tyr (Y)	TGC Cys (C)
	TTA Leu (L)	TCA Ser (S)	TAA Stop	TGA Stop
	TTG Leu (L)	TCG Ser (S)	TAG Stop	TGG Trp (W)
C	CTT Leu (L)	CCT Pro (P)	CAT His (H)	CGT Arg (R)
	CTC Leu (L)	CCC Pro (P)	CAC His (H)	CGC Arg (R)
	CTA Leu (L)	CCA Pro (P)	CAA Gln (Q)	CGA Arg (R)
	CTG Leu (L)	CCG Pro (P)	CAG Gln (Q)	CGG Arg (R)
A	ATT Ile (I)	ACT Thr (T)	AAT Asn (N)	AGT Ser (S)
	ATC Ile (I)	ACC Thr (T)	AAC Asn (N)	AGC Ser (S)
	ATA Ile (I)	ACA Thr (T)	AAA Lys (K)	AGA Arg (R)
	ATG Met (M)	ACG Thr (T)	AAG Lys (K)	AGG Arg (R)
G	GTT Val (V)	GCT Ala (A)	GAT Asp (D)	GGT Gly (G)
	GTC Val (V)	GCC Ala (A)	GAC Asp (D)	GGC Gly (G)
	GTA Val (V)	GCA Ala (A)	GAA Glu (E)	GGA Gly (G)
	GTG Val (V)	GCG Ala (A)	GAG Glu (E)	GGG Gly (G)

Table 1.1. The universal standard genetic code [3].

DNA sequence	Amino acid Sequence
ATG GGC CTT AGC	METHIONINE GLYCINE LEUCINE SERINE
TTT AAG CTT GCC	PHENYLALANINE LYSINE LEUCINE ALANINE

Table 1.2. Example of DNA sequence and the corresponding amino acid name [3].

1.1.2 Amino Acids

Amino acids are the basic building of the proteins which are essential parts of organisms and participate in every process in cells, since proteins are huge molecules made of large numbers of these amino acids and arranged in a linear chain. These amino acids are typically from 100 to 600 [3], composed of a selection of 20 characters. Table 1.3 gives list of these 20 symbols with their full names; one-letter and three-letter codes.

NO.	1-Letter Code	3-Letter Code	Name	NO.	1-Letter Code	3-Letter Code	Name
1	A	Ala	alanine	11	L	Leu	Leucine
2	C	Cys	cysteine(basic)	12	K	Lys	Lysine
3	D	Asp	asparatic acid(acidic)	13	M	Met	Methionine
4	E	Glu	glutamic(acidic)	14	F	Phe	Phenylalanine
5	R	Arg	Arginine	15	P	Pro	Proline
6	N	Asn	Asparagine	16	S	Ser	Serine
7	Q	Gln	Glutamine	17	T	Thr	Threonine
8	H	His	Histidine	18	Y	Tyr	Tyrosine
9	I	Ile	Isoleucine	19	V	Val	Valine
10	G	Gly	Glycine	20	W	Trp	Tryptophan

Table 1.3. The 20 Amino acids and their codes

Those amino acids are linked together as a chain, and that the true identity of a protein is derived not only from its composition, but also from the precise order of its constituent amino acids [3]. For example, the actual recipe for human insulin consists from a chain of 110 residues represented by:

Insulin=
MALWMRLLPLLALLALWGPDPAAAFVFNQHLCGSHLVEALYLVCGERGFFYTPKTR
REAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSICSLYQLENYCN
[4].

1.1.3 Proteins

Proteins are the building blocks of cells. Most of the dry mass of a cell is composed of proteins. Proteins are made up by combining these 20 different amino acids to long chains. These amino acids are joined together by bonds. Proteins are made up of single or multiple chains of amino acids. Proteins are essential parts of the living organism and play important roles in every cellular function. Proteins are performing both chemical and mechanical functions in the cell. Many proteins are enzymes and catalyze chemical reactions. Proteins are folded into three-dimensional structures and the functions of the proteins are dependent on their structures. To understand the functions of proteins, it is very essential to understand their structures first [3].

The most popular functions for these proteins are:

- Forming the structural components (e.g., skin)
- Catalyzing chemical reactions (e.g., enzymes)
- Transporting and storing materials (e.g., hemoglobin)
- Regulating cell processes (e.g., hormones)
- Protecting the organism from foreign invasion (e.g., antibodies)

1.2 Zinc Finger

- The zinc finger motif was first described in 1985 in the laboratory of Aaron Klug at the MRC laboratory of Molecular Biology in Cambridge, where it was inferred from an analysis of the amino acid sequence of the transcription factor TFIIIA [5].
- They are often complex sets of regulatory elements control the initiation of transcription of structure genes. Upstream of the RNA polymerase II initiation site are different combinations of specific DNA sequences, each of which is recognized by a corresponding site-specific DNA-binding protein. These protein are called transcription factor [1].
- Zinc Finger is a finger shaped fold protein that permits interaction with DNA. The fold is created by the binding of specific amino acids in the protein to a zinc atom. Zinc-finger proteins regulate the expression of genes activate the transcription process ,bind specific DNA sequences and determine the 3D structure [1]. Figures 1.2-a,b below show the shape of this finger and its link with the DNA [5, 6].

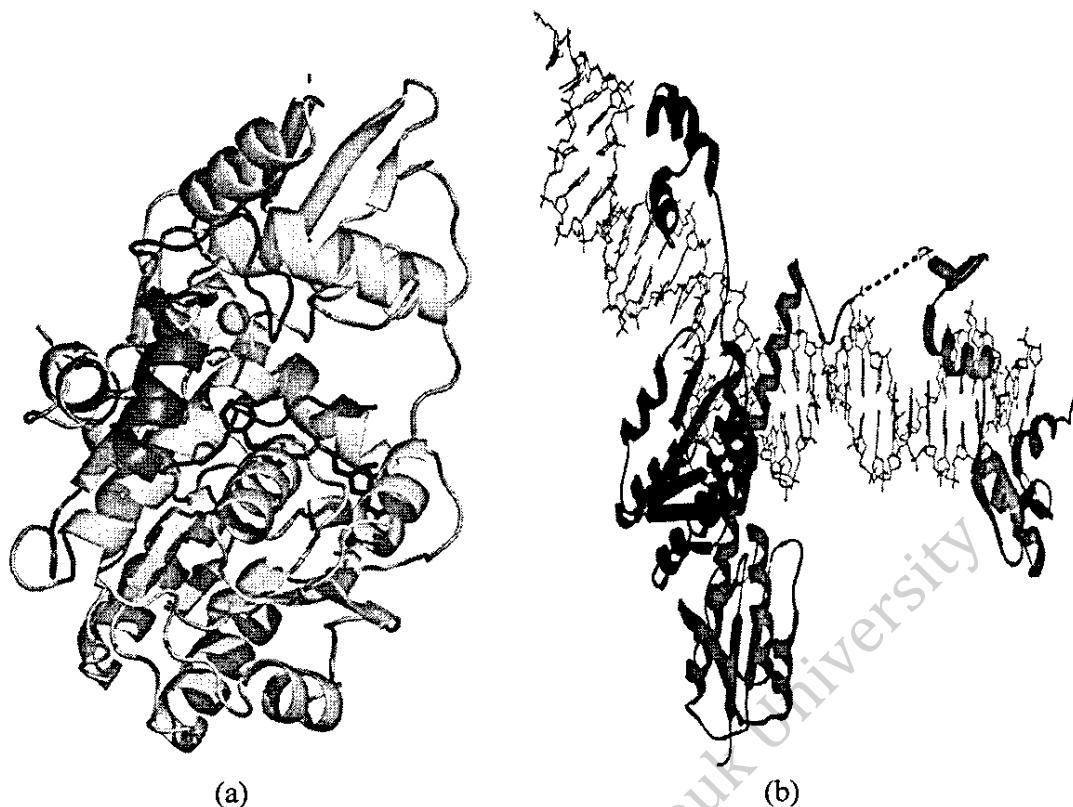


Figure 1.2-a Shape of ZFP [5].

Figure 1.2-b ZFP linked with the DNA [6].

They may be described also as small protein domains that make an interaction between one or more zinc ions to help stabilize their structure. They can be classified into several different structural families and typically functions as interaction modules that bind the DNA proteins or small molecules to describe the structure of the repeated unit of zinc atoms.

The zinc finger proteins have a very important two rules for protein designers. First, they are the most common protein motifs in the DNA binding modules. Second they have an essential role in forming protein-protein interactions.

1.3 The Problem

1.3.1 Important of Zinc Finger Searching

Design of zinc finger proteins (ZFPs) is an important technology for now and future in science of clinical applications such as gene repair and gene regulation. Essentially, the field of zinc finger design is to obtain the full complement of ZFP domains that recognize all DNA triplets with high specificity [7].

Information on the occurrence of zinc finger protein motifs in genomes is important to the developing field of molecular genome engineering. The knowledge of their sequences is important to develop chimerical proteins for genome engineering and to build important sites for genome working.

There is a need to develop a computational resource of zinc finger proteins (ZFP) to identify the binding sites and its location, which reduce the time of task, and overcome the difficulties in selecting the specific type of zinc finger protein and the target site in the DNA sequence. A database is established to maintain the information of the sequence features, including the class, framework, number of fingers, residues, position, recognition site and chemical properties of both natural and engineered zinc finger proteins and dissociation constant of few. ZifBASE can provide more effective way of accessing the zinc finger protein sequences and their target binding sites with the links to their three-dimensional structures. All the data and functions are available at the advanced web-based search interface. [8].

Previously, only the labs with certain technical expertise could take advantage of this powerful ZFP technology. But as ZFPs have taken an important attention recently, they also become interested in this type of technology. Unfortunately, successful design of ZFPs is not always a trivial method. First, the selection of the appropriate target site -depends on knowledge of those domains with high specificity - is not always straightforward. Second, searching in the DNA sequence for a target region with different lengths and composition is difficult and time consuming.

Protein sequence data usually consists of very long strips. Many of these sequences have specific zinc finger proteins that are located in that strip as seen in Figure 1.3.

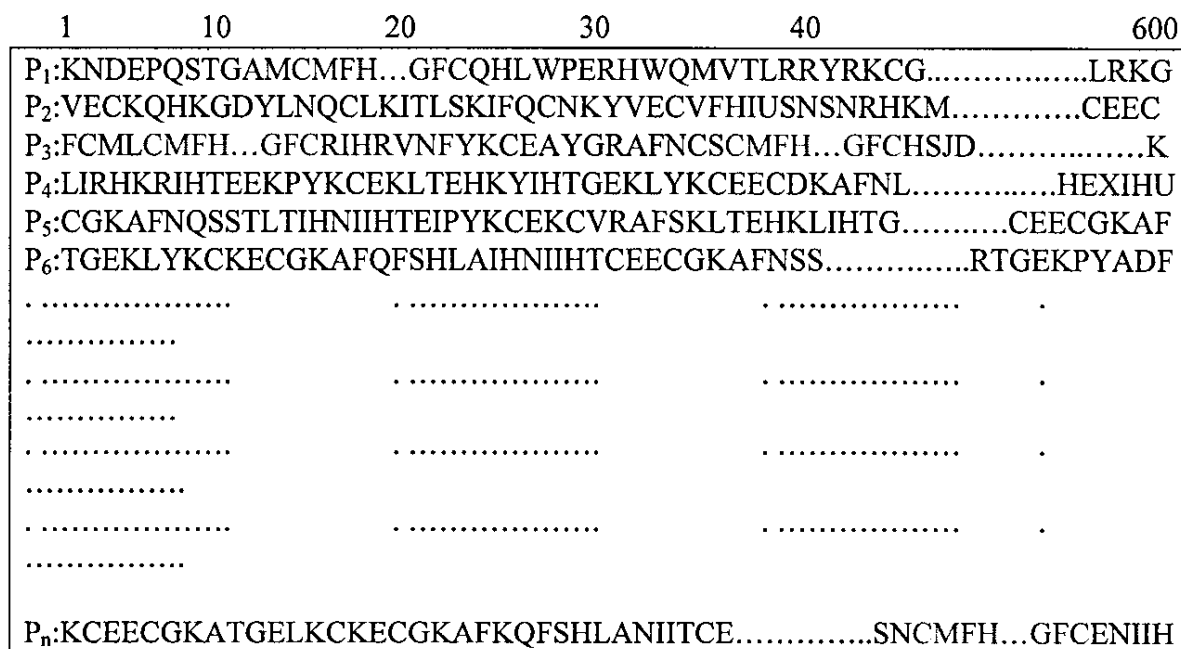


Figure 1.3 The location of zinc finger code named CMFH...GFC in some proteins

1.3-2 Objectives:

Given that protein sequence which consists of very long strips; the objectives of this thesis are to:

- Detect the presence of ZFP in the given sequences.
- Determine the number of ZFP in these sequences.
- Determine the location of each detected ZFP.

Those objectives will be fulfilled by an anticipated steps discussed later in this thesis in details.

© Arabic Digital Library - Yarmouk University

Literature Survey

2.1 Review of Searching Algorithms

There are some algorithms that describe the extraction method of a specific protein from along chain sequence of amino acids. Most of these algorithms share the following drawbacks:

- 1) Techniques that are based on software programming usually give the output as all proteins that contains Zinc Code. However, to determine the location of this code; manual searching has to be done.
- 2) There are old slow searching methods of letter by letter detection. Nowadays they use new methods but not applied for ZFP detection.
- 3) Some of these methods are used a representation of DNA nucleotides letters either by 2-D graphical or a 3-D spacing plot, which is also complex and slow.
Many of these methods work at a DNA sequence of the 4 nucleotides (A, T, C, and G) not the 20 amino acids.

Some algorithms represent the multiple sequence alignments methods, but these ones require hardworking and long time. It is also computationally intensive, often needs manual adjustment, and can be particularly difficult for a set of deviating sequences. The challenging task is therefore to discover patterns directly from unaligned protein sequences. Sequential pattern allows retrieval of frequently occurring events or subsequences as patterns detection.

2.2 Algorithms

As the protein sequence consists of large number of letters like a text shape or a pattern sequence, we'll review three types of algorithms; protein sequences, string matching and pattern recognition using DSP and neural net works.

2.3 Searching and Matching Algorithms of Protein Sequences:

There are several methods that represent searching and matching of Protein sequences. These methods are popular and important but have some drawbacks.

2.3.1 FASTA and BLAST Tools:

FASTA is the name of a popular sequence alignment and database scanning program created in 1988. Because FASTA is easy to parse, this format has become hugely popular and is now the default input format for much sequence analysis software.[3].

BLAST (Basic Local Alignment Search Tool) is a great sequence comparison tool that quickly tells you which of the other known proteins out there has a sequence similar to the required one. You can then use this information for a variety of purposes; including the prediction of protein function, 3-D structure and domain organization, or the identification of homologues (similar proteins) in other organisms.[3]

However, the popular BLAST tool represents the simplest nearest neighbor approach and exploits local alignments to measure sequence similarity. The BLAST technique compares the wanted protein with a protein database of labeled sequences and produces normalized alignment scores for each comparison by calculating the corresponding expectation values (E-values). The classification procedure is based on the selection of the label of the sequence that produces the best pair wise alignment score (i.e. minimum E-value) [16].

2.3.2 Protein Sequence Alignment

Talking about BLAST and FASTA, leads us to talk about the protein sequence alignment. Each protein is characterized by its sequence. To predict the biological functions of one protein and the roles of its residues, they usually compare the sequence of this protein with similar protein sequences whose functions have been examined experimentally. In bioinformatics, a protein sequence alignment is the procedure of comparing two (pair-wise) or more (multiple sequence alignment) sequences by searching for a series of individual residues or residue combinations that are in the same order in the sequences [3]. Each sequence is presented as a row in a page. The aligning process is a way of arranging these sequences such that similar or identical residues are placed in the same column. This method is depending on a visual method that has limited applications.

In 2001 Bill C.H. Chang, et al, proposed a pattern matching algorithm of fuzzy sequence pattern matching - for sequence data [10]. The proposed algorithm obtains a “similar” match by the use of fuzzy membership function as a case of an approximate matching. The sequential dataset, “Zinc Finger domain proteins”, can be used for simulation but the result shows that the proposed algorithm is an approximate pattern matching.

In 2002 Wentian Li Pedro, et al, presented a recursive segmentation procedure that partitions a DNA sequence into domains with a homogeneous composition of the four nucleotides A, C, G and T [11]. This procedure can also be applied to any sequence converted from a DNA sequence, such as a binary strong (G+C)/weak (A+T) sequence, to a binary sequence indicating the presence or absence of the nucleotide.

In 2005, a paper deals with the programming method of local search method for tandem duplication trees was presented [12]. They used these restricted rearrangements in a local search method which improved an initial tree via successive rearrangements. This method was applied to the optimization and minimization evolution criteria.

In 2007, Gerard Rambally represented a method for visualization approach to motif discovery in DNA sequences [13]. Each nucleotide base (A, T, C, G) in a DNA sequence is

assigned a unique integer as a function of its immediate subsequent base, allowing the DNA sequence to be mapped to a corresponding numeric sequence. This numeric sequence is then plotted in 3-D space. After plotting multiple DNA sequences in the same 3-D space, approximately identical regions of the plots are aligned by translation and rotational transformations.

In 2008 Many papers described the representation of DNA. Francis Chin and Henry C.M. made a motif representation with nucleotide dependency [14]. In this paper, new representation called scored position specific pattern (SPSP), which is a generalization of the matrix and string representations, is introduced, which takes into consideration the dependent occurrences of neighboring nucleotides.

Also in 2008 a paper for Liu Xikui, Li Yan of a 2-D graphical representation of DNA sequence is presented [15]. Each letter represented in a graph called Nandy axis system, so that a DNA sequence is denoted on a plane. In this paper, he presented a novel graphical representation of DNA sequences by taking only four special vectors in 2-D (X & Y) Cartesian coordinate system to represent the four nucleic acid bases in DNA sequences(Not a ZFP).

In 2009 there was a programming method about dynamic programming approach for searching a set of similar DNA sequences [9]. This is a programming approach in optimal estimation of similarity distance between DNA sequences which is performed through alignment process. This is an optimal alignment process that was done by using dynamic programming method running in time complexity, then filtering common process technique introduced to improve this optimal alignment process.

There are also many programming searching sites, one named ZifBASE: a database of zinc finger proteins and associated resources [8]. This presented a programming search technique for ZFP site in a DNA sequence used to get some database of zinc finger proteins.

Also the most famous one is "<http://blast.ncbi.nlm.nih.gov/Blast.cgi>"[16]. A data base site called National Center for Biotechnology Information, Basic Local Alignment Search Tool (BLAST). This site doesn't give the location and repetition straight forward that you must enter to the specific protein to see the location of this zinc finger.

2.4 String Matching:

Alfred V. Aho and Margaret J. presented a simple algorithm to locate all occurrences of any of a finite number of keywords in a string of text by string matching [17]. This algorithm consists of a finite state pattern matching machine from the keywords and then using this pattern matching machine to process the text string in a single pass. This construction method of the pattern matching machine takes long time proportional to the sum of the lengths of the keywords and there is what is called a failure transition.

In 1993 Auda and Hazem Raafat described a method that used Neural Networks as an application of Automatic Text Reader [18]. The idea was based on the way by which humans read. The system's input is real news paper texts. The system predicts the size of the font,

and uses it in separating lines, words and sub-words. Then, it scans the text to recognize its individual characters using a set of nine Neural Networks according to a certain procedure. The whole text is then rebuilt and stored to be used by any application.

In 1995 Pericles A. Mitkas produced a method of prototype of a text search processor that used a programmable analog signal processing [19]. This is a text search based on acoustic charge transport technology. Several text search operations can be performed, including operations with "don't care" terms.

In 1997 H. Leet and F. Ercalt presented complex matching algorithm for parallel string matching [20]. The principle of this method is that if you have given a text T of length n and a pattern P of length m, first it finds the exact matching between T and P in a 2-dimensional of size $(n - m + 1) \times m$. Second finds the approximate matching between T and P in $O(k)$ time on a 2D, where k is the maximum edit distance between T and P. Third allows only the replacement operation in the calculation of the edit distance and finds an approximate matching between T and P in constant-time on a 3D.

Also Hiromichi Fujisawa wrote his experiences as Forty years of research in character and document recognition---an industrial perspective [21]. This paper described major technical achievements in the area of character classifiers, character segmentation algorithms, and linguistic processing.

2.5 Pattern Recognition Using DSP and Neural Networks.

In 2009 Juan V. Ginori , et al, present the use of digital Signal Processing (DSP) applications in Bioinformatics where new effective methods for genomic sequence analysis, such as the detection of coding regions was used. Based on representation of the nucleotide by numerical values, converting the nucleotide sequences into time series. then applying mathematical tools to the identification of protein coding DNA regions, identification of reading frames, and others[22].

In 1989 a method is presented by L.H Holley and M. Karplus for protein secondary structure prediction based on a neural network. A training phase was used to teach the network to recognize the relation between secondary structure and amino acid sequences on a sample set of 48 proteins of known structure. On a separate test set of 14 proteins of known structure, the method achieved a maximum overall predictive accuracy of 63% for three states: helix, sheet, and coil. A numerical measure of helix and sheet tendency for each residue was obtained from the calculations. [23].

Also, in 1995 Steve Fairchild et al, presented an algorithm for a three-dimensional structure of a protein from its amino acid sequence They used an integrated approach with an artificial neural network to predict the spatial proximity of the amino acids in the sequence. Mathematical routines are developed to view the protein structure and to visualize and evaluate the results of the neural network. [24].

In 1998 Gilbert White and William Seffens trained a neural network (NN) on amino and nucleic acid sequences to test the NN's ability to predict a nucleic acid sequence given only

an amino acid sequence. They used a multi-layer back propagation network of one hidden layer with 5 to 9 neurons with different network configurations. [25].

In 2008 Sitanshu Sahu and Ganapati Panda presented new algorithm for prediction of protein function from its sequence by the identification of hot spots in proteins using an efficient time-frequency filtering approach known as the S-Transform filtering. The S-Transform is a linear time-frequency representation and is especially useful for the filtering in the time frequency domain. [26].

Dariusz Plewczynski, et al, trained a neural network on the detection of signal peptides in proteins. The neural network is trained on sequences of known signal peptides extracted from the protein database and is able to work separately on some types of proteins. A query protein is dissected into overlapping short sequence fragments and then each fragment is analyzed with respect to the probability of it being a signal peptide. [27].

Methodology

3.1 Developed Algorithm Block Diagram:

In the following implementation, a new approach will be presented, which consists of a number of stages as shown in Figure 3.1.

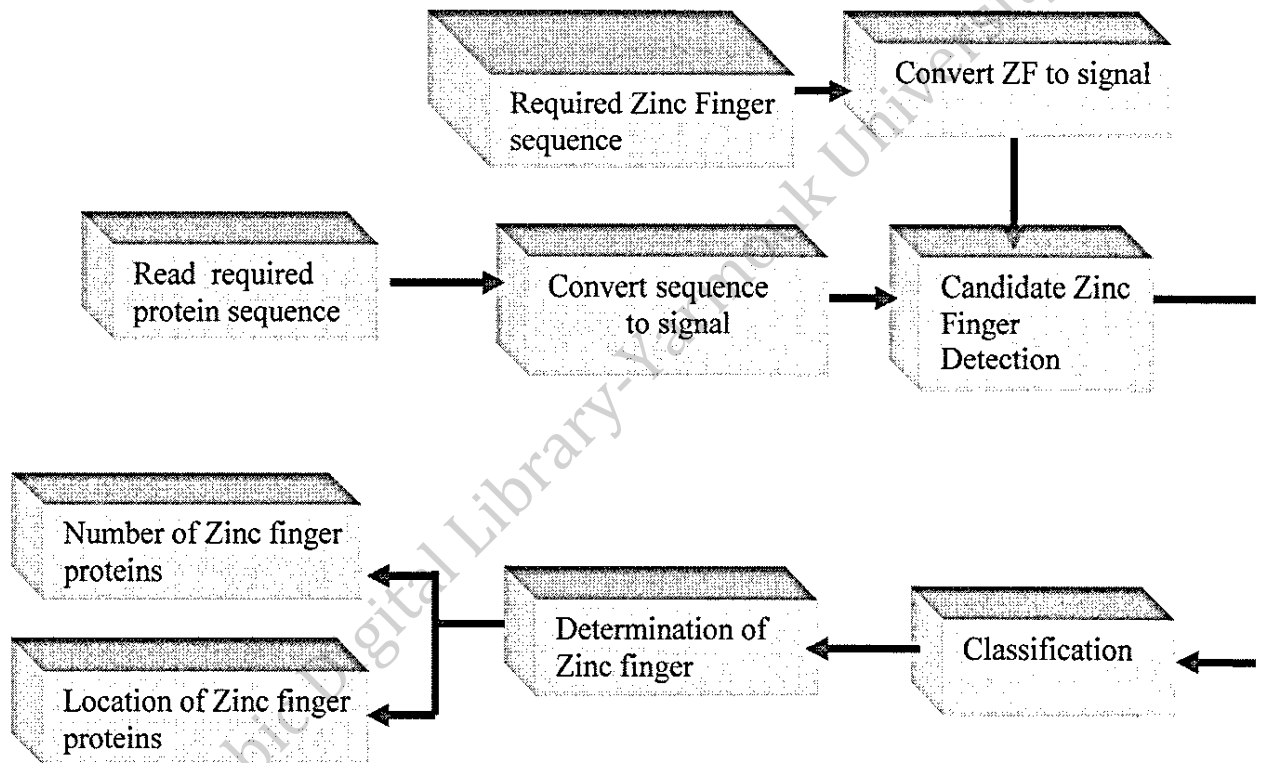


Figure 3.1 The block diagram of the new fast algorithm.

The following is a brief description of each stage of the proposed algorithm.

3.2 Read Sequence:

In this stage data can be read from a specific location that contains the complete protein sequences. Data representation with respect to the above described data acquisition, is commonly converted to digitized representation of amino acid letters. It generally contains sequential information which has the position of each zinc finger inside it that will be fed as an input function in this methodology.

3.3 Digital Mapping for Amino Acids :

Data mapping is a very useful step often prior to designing a classifier, especially when the values of the signal vary in different dynamic ranges. In the absence of mapping, features -that will be discussed later- with large values have a stronger influence on the function in designing the classifier. Data mapping restricts the values of all features within predetermined ranges. Digital mapping is applied in order to reduce the amount of variability in the protein sequence prior to classification, and thus to simplify the further recognition process.

3.3.1 Digital Coding for Amino Acids

A protein sequence is composed of a series of amino acids represented by characters. There are many papers describe the sequence conversion into signals [29]. Some of them gave a digitization code which used the amino acid index and information theory to set up a model of digital coding for amino acids. This digital coding reflects better the amino acid with its chemical properties, physical properties, the electron-ion interaction potential (EIIP), structure and existence. Through the above encoding procedures, a protein sequence is transformed into a serial of digital signals that reflects better the amino acids. For example, Figure 3.2 shows protein signals and the location of a certain zinc finger protein within these signals. Tables 3.1 to 3.3 show a digital coding for amino acids as mentioned in the previous studies [28, 29, 30].

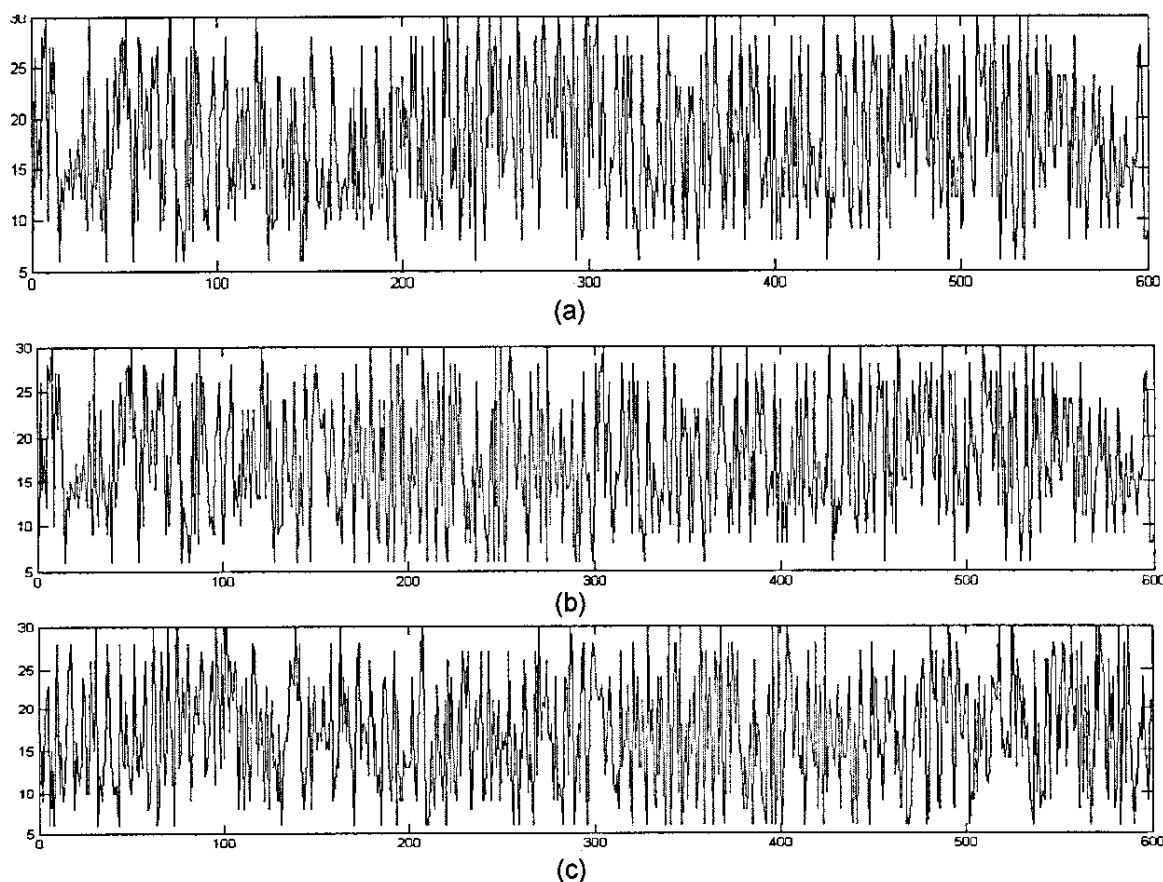


Figure 3.2. a: Shows no location of zinc finger in a specific protein signal. b: Shows the location of zinc finger between 140-300. c: Shows the zinc finger in another location between 290-450, using encoding described in [29].

Amino acid	P	L	Q	H	R	C	S	F	Y	W
Decimal number	1	3	4	5	6	9	11	12	14	15
Amino acid	T	I	M	K	N	A	V	D	E	G
Decimal number	16	18	19	20	21	25	26	28	29	30

Tables 3.1 The 20 Amino acids and their digital coding starts from 1 to 30 [28].

Amino acid	K	N	D	E	P	Q	R	S	T	G
Decimal number	6	8	9	10	11	12	13	14	15	16
Binary number	00110	01000	01001	01010	01011	01100	01101	01110	01111	10000
Amino acid	A	H	W	Y	F	L	M	I	V	C
Decimal number	17	18	20	21	23	24	26	27	28	30
Divided by 60	10001	10010	10100	10101	10111	11000	11010	11011	11100	11110

Tables 3.2 The 20 Amino acids and their digital coding starts from 6 to 30 [29].

Amino Acid Three Letter Symbol	Single Letter Symbol	EIIP(Ry)
Ala	A	0.0373
Arg	R	0.0959
Asn	N	0.0036
Asp	D	0.1263
Cys	C	0.0829
Gln	Q	0.0761
Glu	E	0.0058
Gly	G	0.0050
His	H	0.0242
Ile	I	0
Leu	L	0
Lys	K	0.0371
Met	M	0.0823
Phe	F	0.0946
Pro	P	0.0198
Ser	S	0.0829
Thr	T	0.0941
Trp	W	0.0548
Tyr	Y	0.0516
Val	V	0.0057

Tables 3.3 The 20 Amino acids and their digital coding depending on the electron-ion interaction potential (EIIP) [30].

The Zinc Finger Proteins that are considered can be divided into three major types [31]:

- C2H2 zinc finger; Which is defined by the sequence CX₂-4C...HX₂-4H, where C = cysteine, H = histidine, X = any amino acid. The building structure contains two cysteines and two histidines that interact with a zinc ion.
- C4 zinc finger; Which is defined by the sequence C.X₂-4CX₁₃CX₁₄-15CX.....CX₂C. The first four cysteines interact with a zinc ion and the last four cysteine interact with another zinc ion .
- C6 zinc finger; Which is defined by the sequence CX₂CX₆CX₅-6CX₂CX₆C. In this type six cysteines interact with two zinc ions

The previous types of coding were studied in order to choose the most suitable one to be used in our algorithm. The second type of coding presented in table 3.2 was chosen to be applied in this algorithm with the following modification:

- Many types of the zinc fingers (C2H2, C4, C2HC, etc) have a common factor of certain beginning and ending which has these specific characters C and H.
- The first and last six letters in the zinc finger are implementing in our mapping scheme to make it more easier to detect those types of zinc fingers.

For these reasons; the characters C and H were given the largest weight value of (60) and (-60) respectively, then divide all these values by 60. Table 3.4 shows the new coding scheme to be use in the later stages of the developed algorithm.

Amino acid	K	N	D	E	P	Q	R	S	T	G
Decimal number	6	8	9	10	11	12	13	14	15	16
Divided by60	0.1	0.13333	0.15	0.16666	0.18333	0.2	0.21666	0.23333	0.25	0.26666
Amino acid	A	H	W	Y	F	L	M	I	V	C
Decimal number	17	-60	20	21	23	24	26	27	28	60
Divided by60	0.28333	-1.00	0.33333	0.35	0.38333	0.4	0.4333	0.45	0.46666	1.00

Table 3.4 The new coding for the this algorithm

As shown in Figure 3.3 below the new weight for the 20 amino acids indicates that C has the maximum weight and H has the minimum weight comparing to the others.

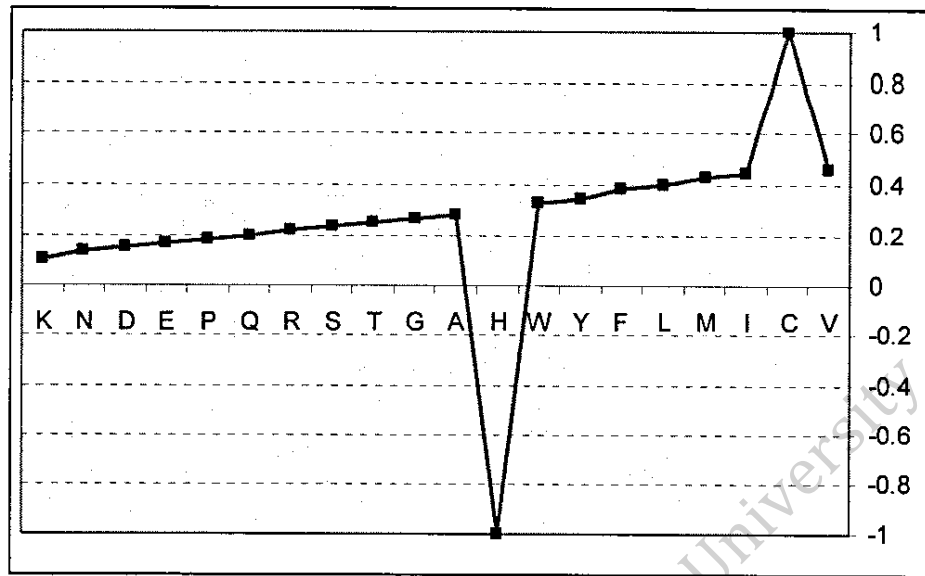


Figure 3.3 The new modified digital coding for the 20 amino acids.

3.3.2 Mapping Characterization in Time Frequency Domain

Most of the signals have a popular way for its first stage in the signal processing including segmentation and classification. There are many different algorithms that represented signal processing especially ECG signals [32]. The most common method is Pan-Tompkins which is used to detect the QRS in the ECG signal [33,34]. This algorithm represented a number of stages to detect a signal by applying a band pass filter followed by differentiator, non linear equation like squaring then window integration and a threshold stage .In this thesis, the protein sequences -after converted to signals - have the following characteristics:

- 1) They are different in lengths, and also have different numbers and lengths of zinc fingers.
- 2) They have many similarities inside them which make it more difficult to detect.
- 3) These signals are more clear than the ECG signal with no noise.

For these reasons, the advantage of the Pan-Tompkins method were chosen, with no need for a band pass filter; but only applying the differentiator stage directly after the coding stage.

Figures 3.4 to 3.7 below show two types of proteins for different lengths after the mapping and the differentiating of the original signals.

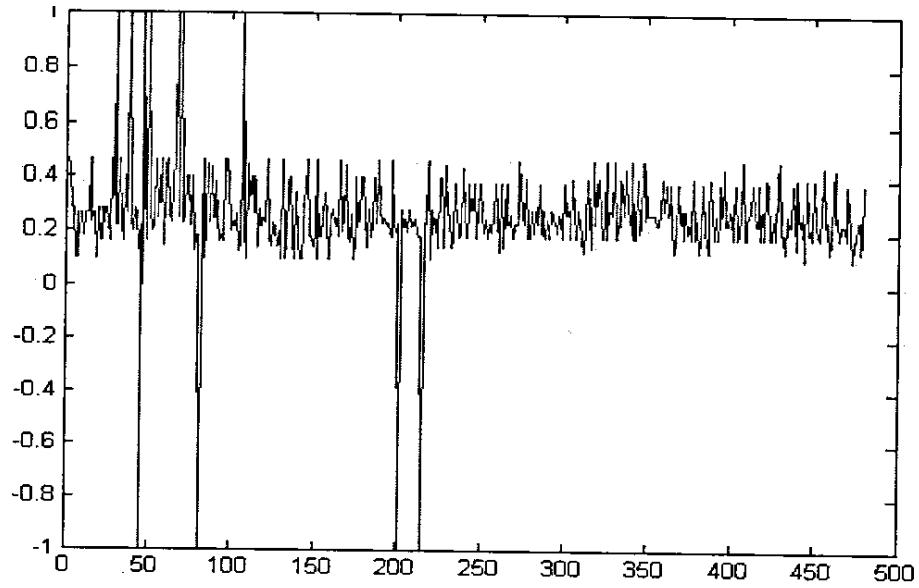


Figure 3.4 Short protein signal after the mapping, where the ZFP is marked by the red region.

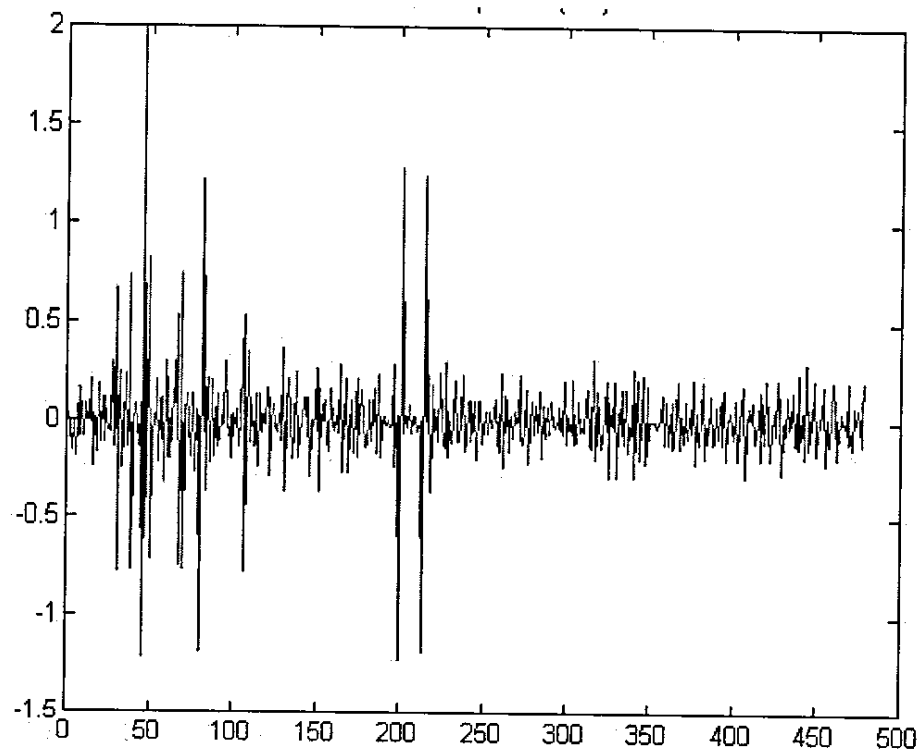


Figure 3.5 Short protein signal after the differentiator, where the ZFP is marked by the red region.

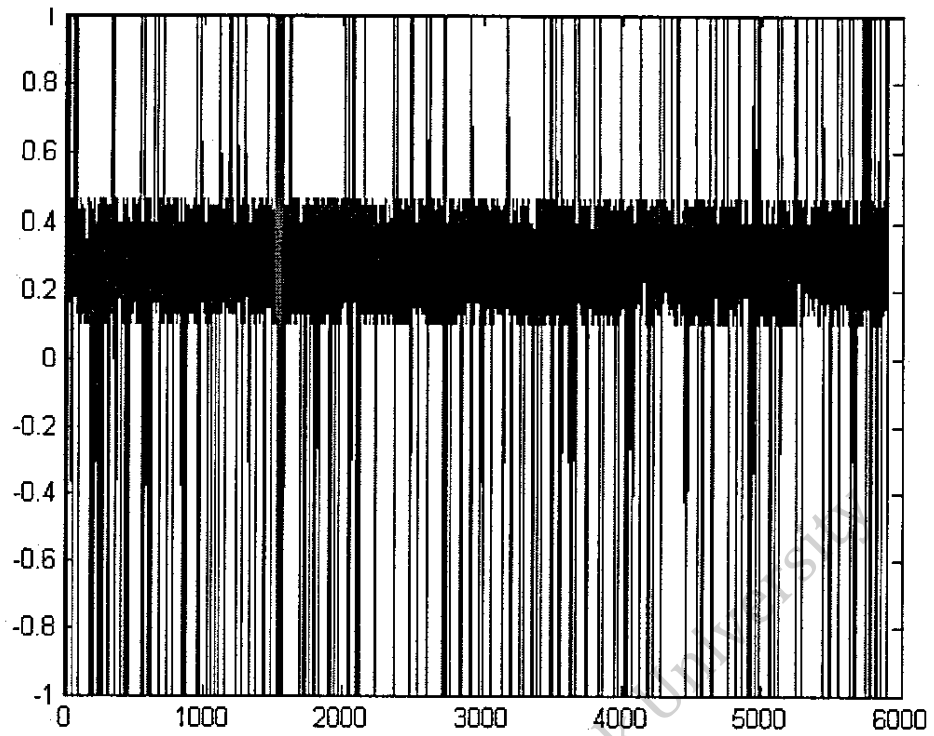


Figure 3.6 Long protein signal after the mapping, where the ZFP is marked by the red region.

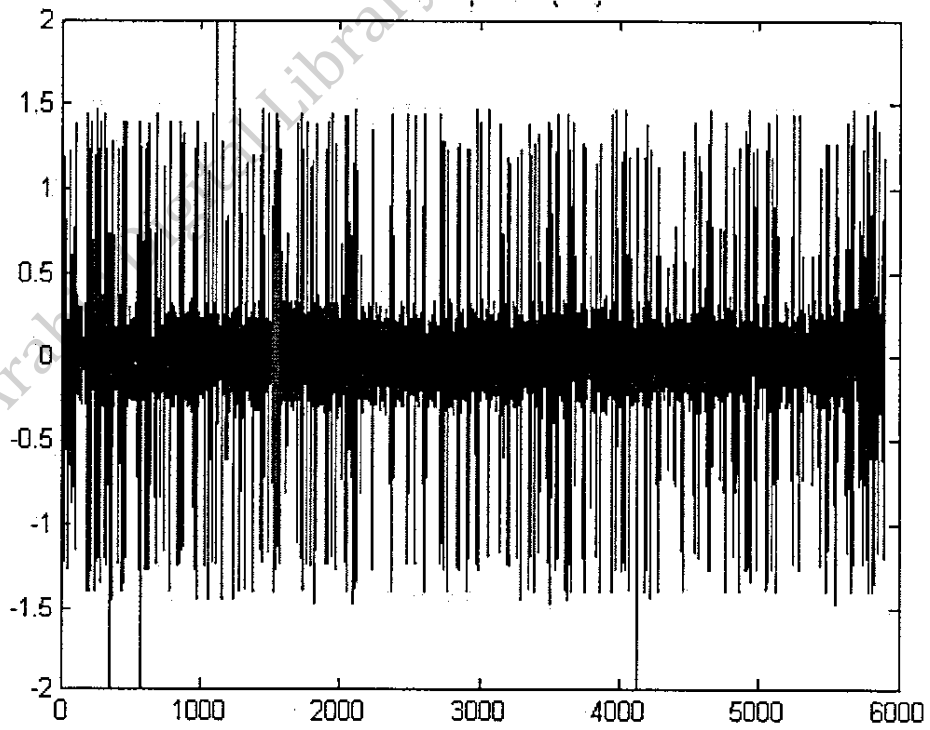


Figure 3.7 Long protein signal after the differentiator, where the ZFP is marked by the red region.

3.3.3 Short Time Fourier Transform of Mapped Sequence

The most interesting work in bioinformatics genomic sequence analysis is the digital signal processing (DSP) applications, such as the method of detecting the coding regions in the DNA sequences. One important rule in the use of DSP principles to analyze genomic sequences is defining a suitable representation of the nucleotide bases by numerical values, then converting the nucleotide sequences into time series. After doing this, most mathematical tools usually employed in DSP are used in solving man problems. First; before choosing a suitable algorithm that will be applied here, we will mention the most popular applications of DSP algorithms which are used in the analysis of genomes [35].

- Discrete Fourier Transform (DFT):
- The Short Time Fourier Transform (STFT):
- Gabor Transform:
- The Discrete Wavelet Transform (DWT):

Mapping in general has great effect in the signal processing. This step is very important especially when dealing with huge data because it makes it easier in the designing of the classifier stage. Mapping guides the data to take a shape which has some features that is simpler to deal with.

Fourier transform approach works best when the signal is composed of a number of discrete frequency components so that time is not a specific issue. The signals that cannot be satisfactorily represented in these ways must be modified using some types of representations like the spectrogram. The Fourier transform is defined as [36]:

$$X(\omega) = F[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \dots\dots\dots 3.2$$

and its inverse is,

$$x(t) = F^{-1}[X(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega \dots\dots\dots 3.3$$

If the window is made short enough to capture rapid changes in the signal, it becomes impossible to get frequency components which are close in frequency during the analysis window duration. On the other hand, if the time window is made long enough to permit good frequency resolution, it is difficult to determine where, in time, the various frequency components act [37].

For clearness, it is impossible to know $x(t)$ for all time, and it is also impossible to know $X(\omega)$ for all frequencies, so it was the need for the spectrogram which is a function that expresses $x(t)$ with $X(\omega)$.

The discrete formula for the short-time Fourier transform (STFT) for a sequence $x(n)$ is [37]:

$$\text{STFT}_x(e^{j\omega}, n) = \sum_{m=-\infty}^{\infty} x(n-m)\omega(m)e^{-j\omega n} \dots\dots\dots 3.4$$

Where $\omega(m)$ is a suitably chosen window sequence.

The disadvantage of the STFT is choosing the time window size. If a specific size for the time window were chosen, then this window must be the same for all frequencies. Taken into consideration that many signals need certain algorithms, as an example the Zinc Finger proteins which haven't the same length for all sequences, so we need to vary the window size to determine more specific determinations in time and frequency.

The spectrogram depends on some parameters that have an affect on the results. While the time lapse between blocks and the frequency discretization do not affect the time or the frequency resolution, but only the pixilation; the block length affects the time and frequency resolution. So frequency resolution becomes better as block length increases, and time resolution goes better as it decreases [37].

The contour function displays a two-dimensional contour plot and fills the areas between contour lines. It is like the intersection of a three dimension surface with a horizontal plane. These lines form loops or terminate at the outer edges of the surface. The following figures show the contour drawing for some protein sequences that have different lengths and different numbers of zinc fingers. Figures 3.8 to 3.16 show the contour and the spectrogram drawing for different sequences.

As you can see, this method is affected by the following factors:

- The length of the protein sequence.
- The length of the zinc finger protein sequence.
- The shape of mapping for this sequence.
- The spectrogram parameters which are: The frequency discretization, sampling frequency, block length, overlap and type of window.

Some of these factors affect the results in a clear way. Increasing or decreasing the sampling frequency has the greatest effect, so it must be chosen in different values for the protein lengths, since neither lengths of proteins or zinc fingers are the same.

Meanwhile the time lapse between blocks affects the pixilation only, the block length affects the time and frequency resolution. As block length increases - Narrow-band spectrogram - the frequency resolution goes better, but if it decreases - Wide-band spectrogram - the time resolution goes better.

In the previous part of this algorithm, and after the mapping, many ideas were jumped to deal with these sequences to reach the required goals. One can ask; what is the fast, simple and more accurate method that can be followed to reach these goals.

From the most fast methods that were mentioned previously like the dynamic programming algorithm, basic local alignment searching tools (BLAST), hidden Markov model, short-time discrete Fourier transform and neural network, the neural network method have been chosen to obtain the results as seen in the next chapter.

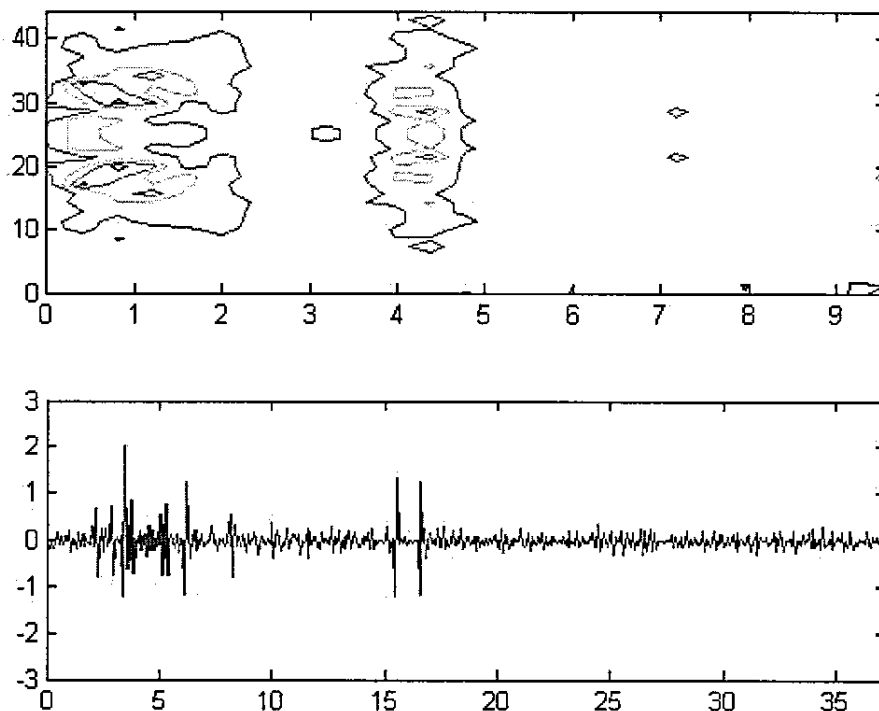


Figure 3.8 The contour for short protein sequence that has only two zinc fingers

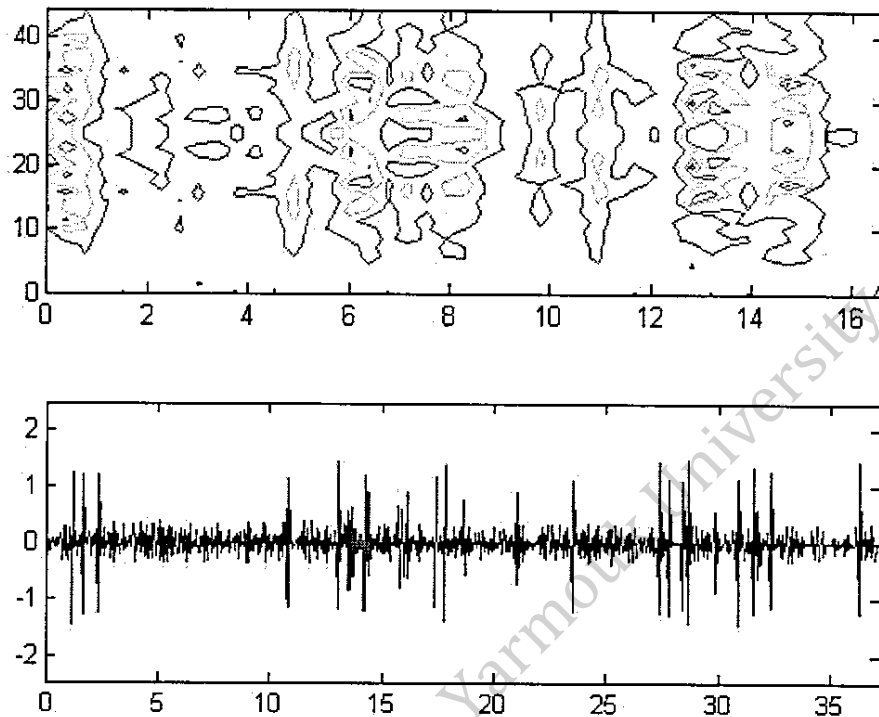


Figure 3.9 The contour for middle protein sequence that has some of zinc fingers.

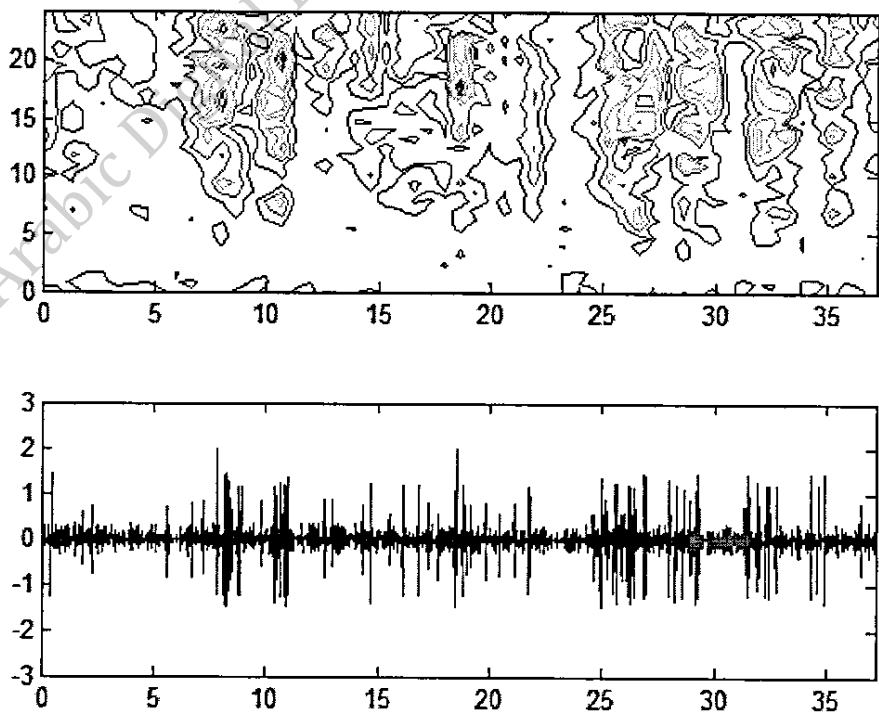


Figure 3.10 The contour for middle protein sequence that has many of zinc fingers.

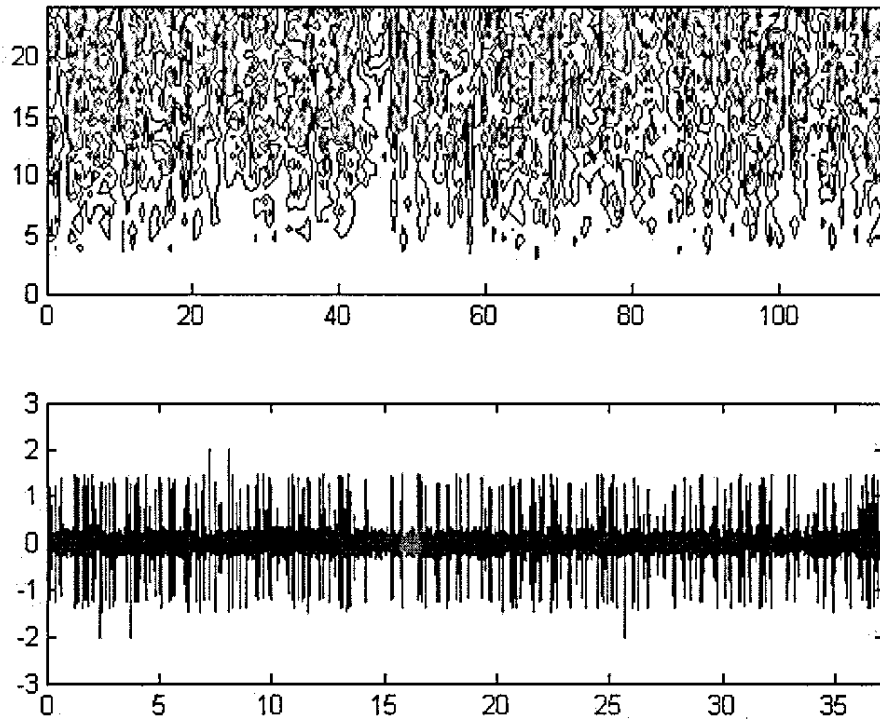


Figure 3.11 The contour for long protein sequence and small f_s .

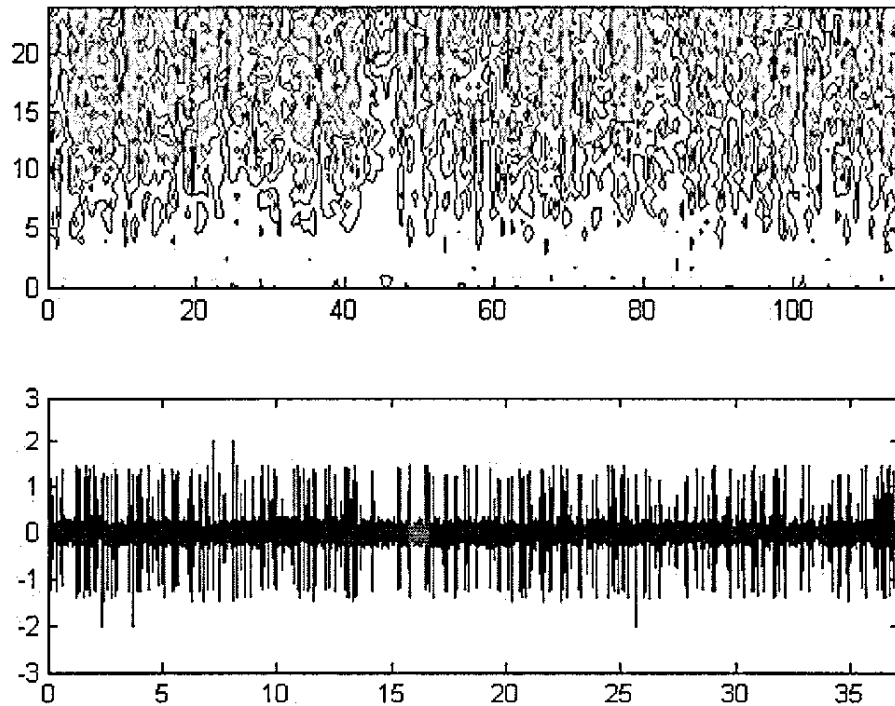


Figure 3.12 The contour for the same long protein sequence and large f_s .

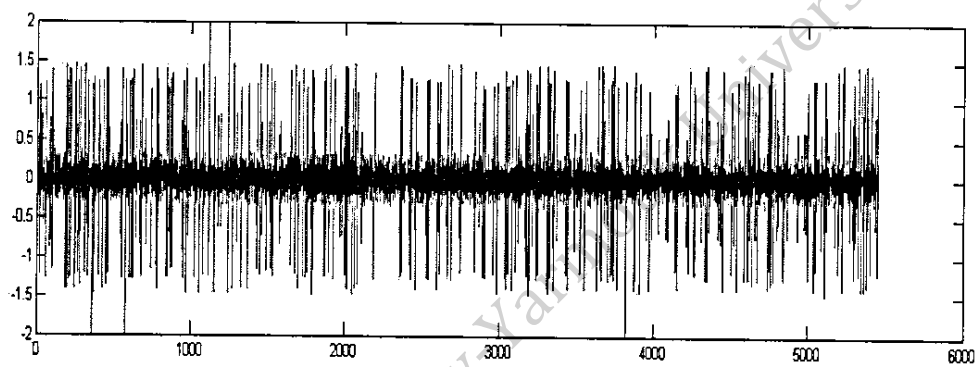
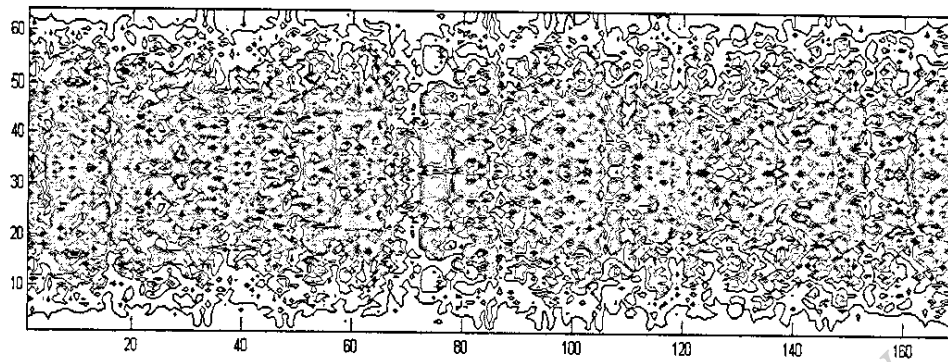


Figure 3.13 The contour for very large protein sequence that has a lot of zinc fingers.

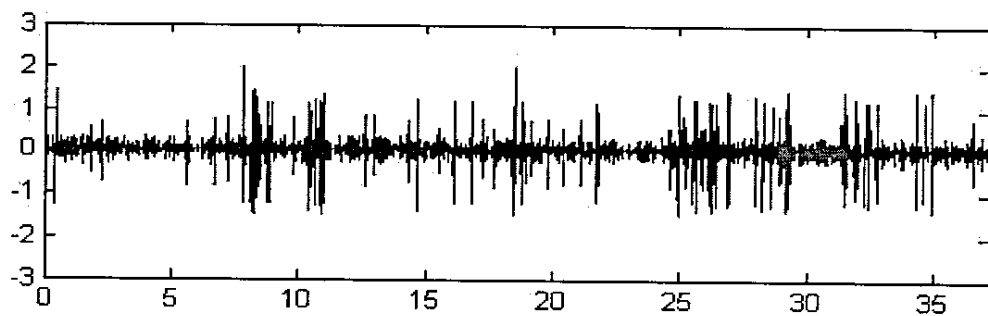
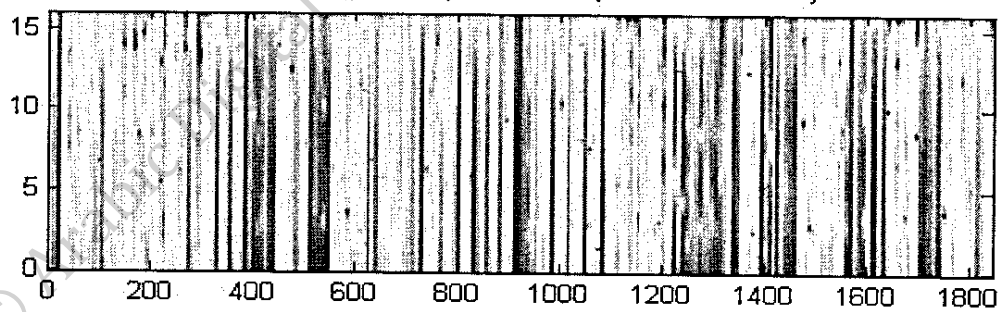


Figure 3.14 The spectrogram for middle protein sequence.

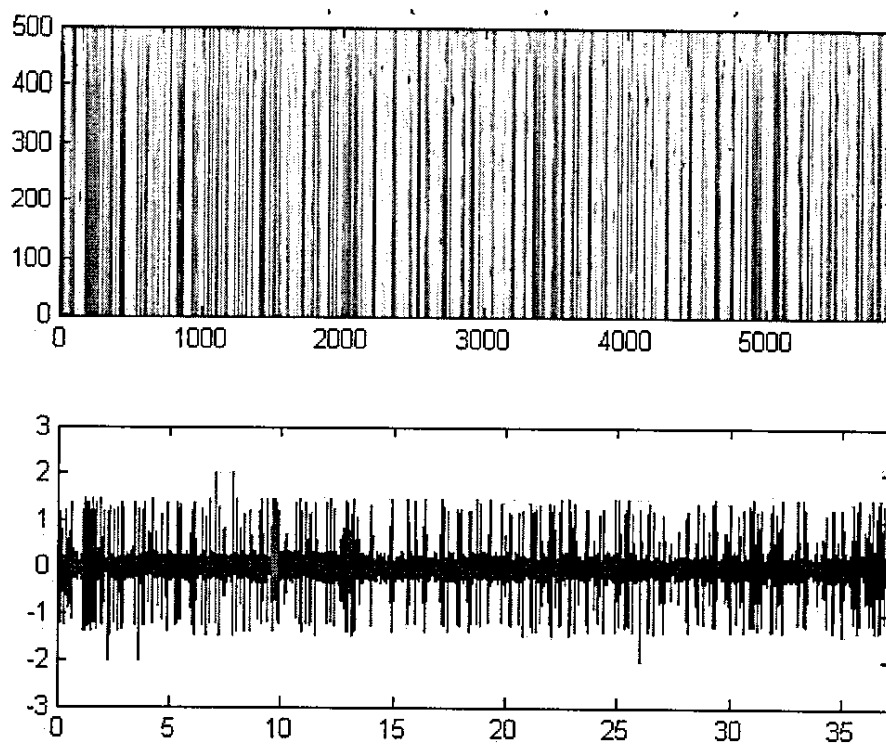


Figure 3.15 The spectrogram for long protein sequence.

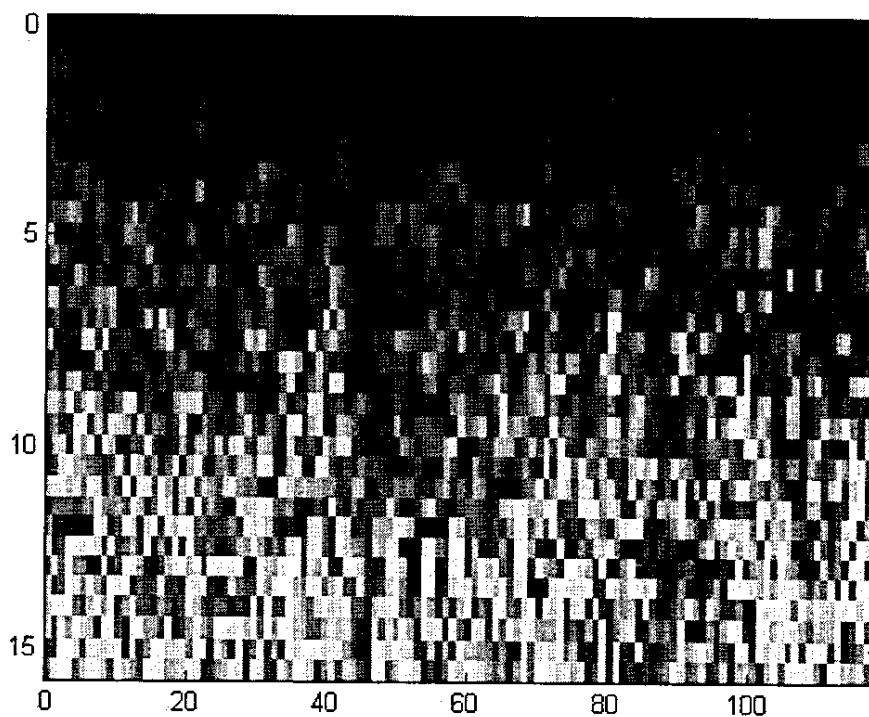


Figure 3.16 Image plot for protein sequence data after differentiating shows-in red color-the expected to be zinc ones.

3.4 Candidate Zinc Finger Detection

Given the signal which represents the protein after mapping, we need to detect the initial location of Zinc Finger. This location is detected with high sensitivity, however with low positive predictive value. This candidate Zinc Finger will be confirmed using the last classification stage. The detection in this stage is based on a previous work [38] and is summarized in the following subsequences.

3.4.1 Input Segmentation

Several methods have been developed for signal segmentation like auto correlation, blind segmentation and others. Some of them use amplitude/threshold separation methods or the most periodic signal peaks.

For a signal that has some certain changes inside it, one can develop a method to separate certain data segments. Such method requires identification of the time instants through this segment in order to separate it. After that, breaking up the data into segments according to these time instants regions is possible.

Auto correlation is one of the simplest examples that use a fixed reference window and a moving test window with the same width. But as you can see, zinc finger signals are time-varying with different lengths, so the auto-correlation method with a fixed reference window generates error and might not be robust because of the losses of many data.

In this thesis, the segmentation depends on the characteristics of the zinc finger that focused on the peaks of the signal and the repetitions of the Cysteine and Histidine amino acids at the starting and ending of the zinc; as mentioned in the previous chapter about the characteristics of the zinc fingers

3.4.2 Segmentation Enhancement

This novel segmentation method is shown in Figure 3.17, which consists of two parts [38]. The first part is an initial segmentation that comes after certain stages which are a differentiator followed by nonlinear squaring stage, and then a threshold decision rule stage is used to identify the event of starting and ending points.

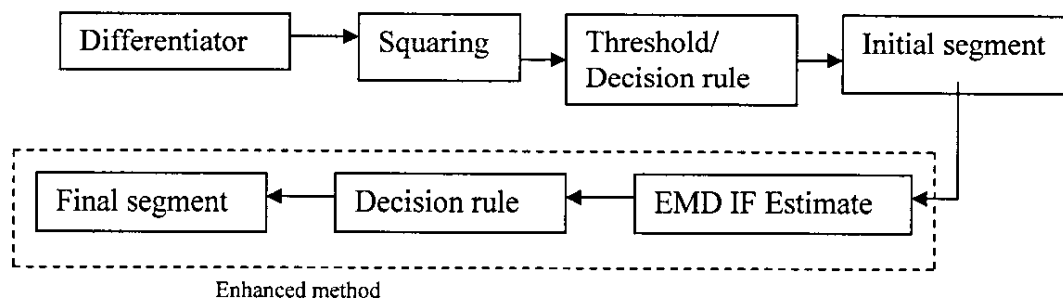


Figure 3.17 The Segmentation Enhancement Method [38].

From this figure 3.18 one can see that this protein contains 6 possible zinc fingers. The location of the first one is from 27 to 68 and the last one is from 758 to 783, while figure 3.19 shows another protein that contains 8 possible zinc fingers with another different locations.

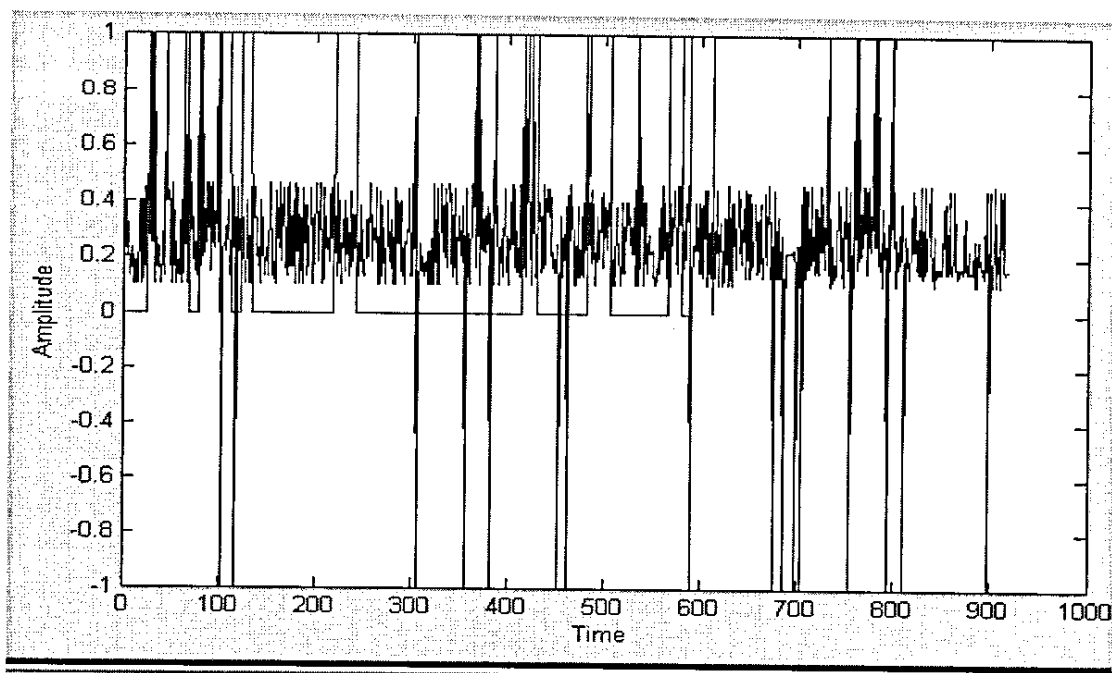
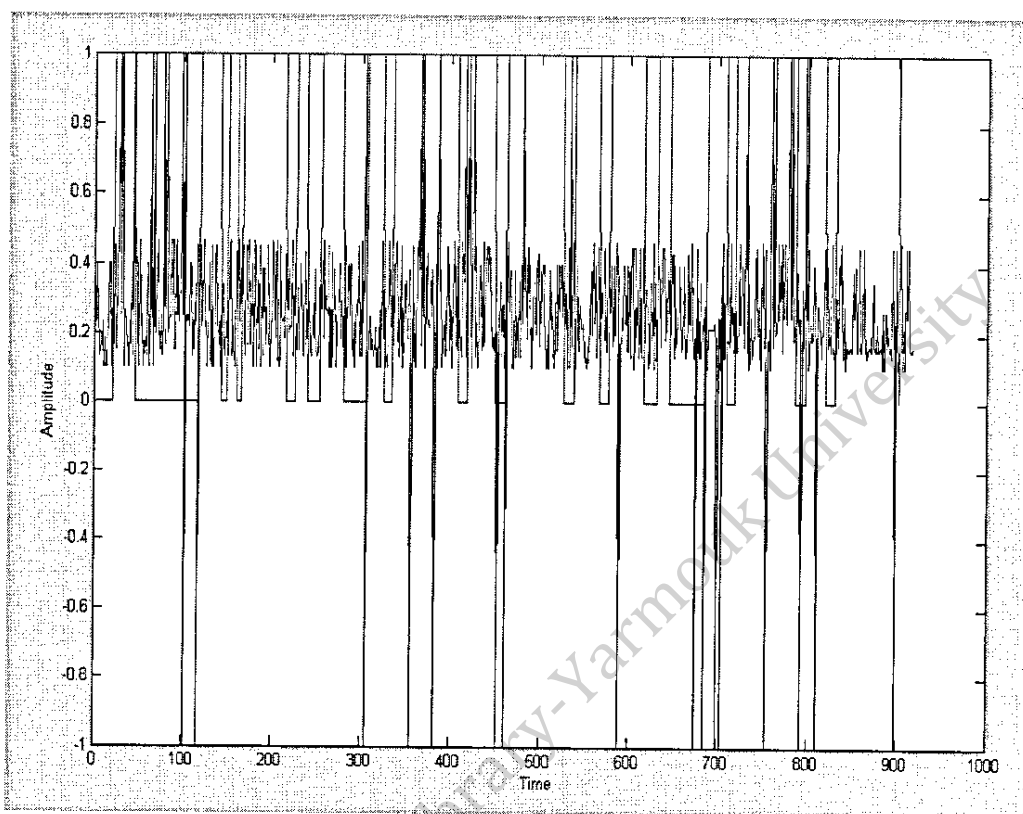


Figure 3.19 Segmentation method for another protein that shows 8 expected zinc fingers with their locations at the bottom of the figure.

While this segmentation has a great effect in this algorithm in speeding up and finding the location of these expected zinc fingers; it cannot determine them with very high accuracy, because it gives all the candidate zinc fingers, so some of them might not be zinc ones as you can see in figures 3.19 and 3.20.

Decreasing the time is a great advantage, and the reason for this is because segmentation method gives the output signal of most of all candidate fingers only and not the signal of all the proteins. Next, it must be followed by a certain stage to confirm these fingers with high sensitivity, so it was the need for the neural network.



ans -

21	46
119	143
150	161
167	199
197	216
227	240
255	280
307	325
336	409
420	434
432	450
463	482
480	505
503	527
539	567
579	617
632	646
687	710
719	743
741	754
752	776
774	787
799	820
832	850
848	872
870	881
879	902

Figure 3.20 Segmentation method for long protein sequence that gives all the candidate zinc fingers. It shows the zinc fingers and the expected ones with there locations at the bottom of the figure.

3.5 Zinc Finger Determination and Localization:

The goals of this stage are to:

- Confirm the detection of ZF.
- Determine the location of ZF.
- Determine the number of sequences of ZF.

3.5.1 The Most Popular Fast Algorithms:

While talking about fast methods, the most popular fast and accurate algorithms can be divided into five categories:

- 1) The dynamic programming algorithm which is the most sensitive among all the based methods [13].
- 2) Sites like basic local alignment searching tools (BLAST) and (www.zincfingertools.org). They are faster and less sensitive than the Smith-Waterman algorithm because of their heuristic features but don't give all of our goals [14,15].
- 3) Profiles algorithms. One typical representative of it is the Hidden Markov Model. It collects statistics from a set of similar sequence and compares the resulting statistics to a single sequence of interest [39].
- 4) Short-time discrete Fourier transform (ST-DFT). It gives the location repeats and identification of the protein coding regions in DNA sequences [28].
- 5) Neural network, Fuzzy, trees, scored position specific, etc. They explicitly model the string and try to use it to judge them through some sequences [7, 9, 11]

The Neural Network is chosen to confirm the initial candidate Zinc Finger and its type; this will give the high sensitivity and enhance the positive predictive value as well be seen later.

The Classifier

4.1 Neural network Classifier

4.1.1 Introduction

Given that an initial candidate Zinc Finger is detected, the goal is to confirm that the detected sequence event is true event. To achieve this goal Neural Network is used.

A neural network is an information processing system which was developed as a generalization of the mathematical model of human learning. It is constructed of neurons (similar to the biological system of the human brain).

Neural networks are formed of simple elements operating in parallel called neurons. The network function is determined largely by the connections between neurons. Any neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between neurons; this trained network is used in various application fields like robotics, speech, securities and telecommunications including pattern recognition, identification and classification methods.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a certain target output. The network is trained, based on a comparison of the output and the target, until the network output matches the target. Many inputs and targets are needed to train a network. Figure 4.1 shows the weights adjustment of the Neural network. [40].

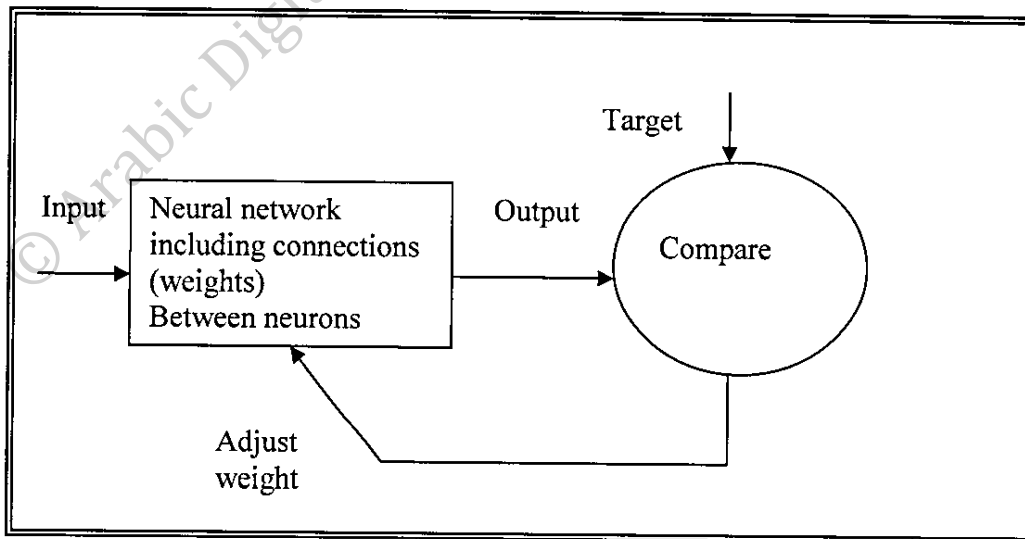


Figure 4.1 General neural network function where the weight is adjusted to make the output matched the target

Neural networks that are used for classification consist of two stages: the encoding stage, where numerical values are computed for each training sequence and the decision stage where the output vectors are used as input vectors to a neural network classifier, and they work for large training sets; since the number of inputs is very large. The three elements that the network consists of can be classified into three parts: The pattern of connection between neurons, the active function applied to the neurons and method of determining weights (learning and training) [41]

4.1.2 Neurons

The main element of the neural network is the neuron. It is a node that carries out information during the processing operations. It consists of a set of inputs x_1 to x_n that is weighted before it reaches the main body of the processing part of the connection weight factor; the amount of information input required to solve the problem is stored in the form of weights. Each signal is multiplied by an associated weight W_0 to W_n , then it is applied to a summing stage as seen in Figure 4.2 [40].

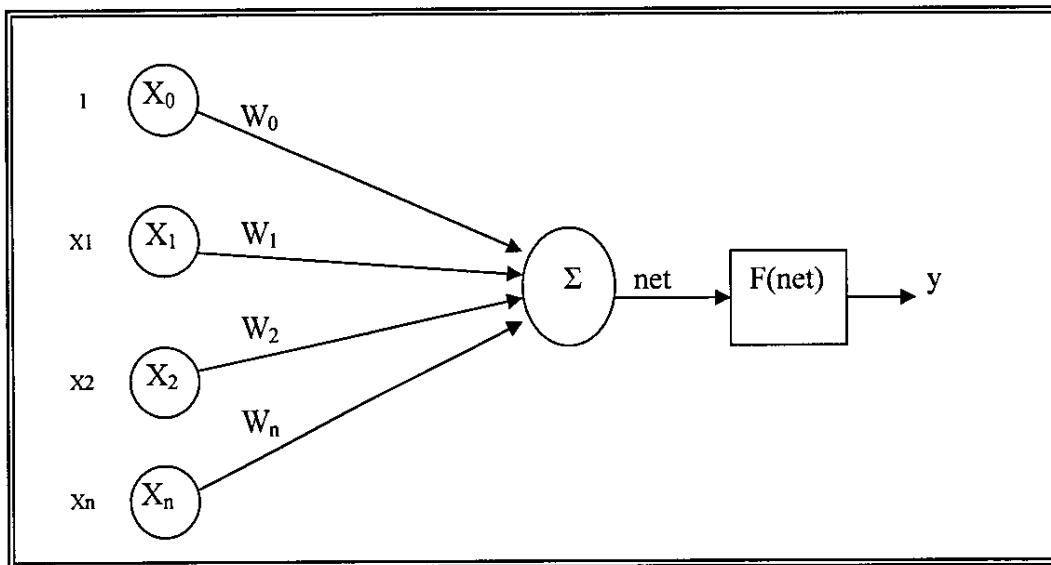


Figure 4.2 The basic neuron model [40].

Where : as $y = F(\text{net})$; the net is:

$$\text{net} = W_0 + x_1 W_1 + x_2 W_2 + \dots + x_n W_n$$

or

$$\text{net} = W_0 + \sum_{i=0}^n x_n W_n \dots \dots \dots 4.1$$

The network performance depends on the number of neurons, in which the training process becomes more efficient, The neural networks consist of three arranged layers: the input, the output and the hidden layers, which may be single or double layers that are all working together with some activation functions as : [40].

The linear function $F(x) = x$ for all x 4.2

The binary step function $F(x) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{if } x < \theta \end{cases}$ where θ is a threshold value.....4.3

The sigmoid or logistic function $F(x) = \frac{1}{1 + e^{-x}}$ 4.4

The bipolar sigmoid function $F(x) = \frac{2}{1 + e^{-x}} - 1$4.5

4.1.3 Multilayer Neural Networks

A network can have several layers. Each layer has an input p , a weight matrix W , a bias vector b and an output vector a . The outputs of each layer are the inputs to the following layer. As an example, layer number two can be analyzed into one layer as a function of layer number one then it can be treated as a single layer network. The following equations represented the relation between these layers.

$$a_1 = f_1(W_{1,1}p + b_1) \quad a_2 = f_2(W_{2,1}a_1 + b_2) \quad \dots \quad a_n = f_n(W_{n,n-1}a_{n-1} + b_n) \quad 4.6$$

So, the output of layer number three is:

$$a_3 = f_3(W_{3,2} a_2 + b_3) = f_3(W_{3,2} f_2(W_{2,1} f_1(W_{1,1}p + b_1) + b_2) + b_3) \dots \quad 4.7$$

4.1.4 Feed-Forward Neural Networks

Feed-forward neural networks are the most popular and widely used networks in many applications. It consists of the input layer which is the input of the network and the hidden layer which consists of any number of neurons, or hidden units placed in parallel. Each neuron performs a weighted summation of the inputs, which then passes a nonlinear activation function (neuron function). The network output is formed by another weighted summation of the output of the neurons in the hidden layer. This summation is called the output layer.

In training the network, its parameters are adjusted incrementally until the training data satisfy the desired mapping, with maximum number of iterations as much as possible. The nonlinear activation function is set as a default standard function sigmoid, but it can be changed to another function like the step function as an example.

4.1.5 Characteristics of the Neural Network

There are so many different string matching algorithms and their variants. The most popular approach is a simplest string matching algorithm - deals with letter by letter detection – which is easy to understand and implement but it has some problems like slow in application, incorrect results in some times for huge data and the characters in the text may be compared multiple times.

The characteristics of the neural network are:

- Ability of effective training.
- Self-Organization.
- Real Time Operation.
- Good continuous response.

making it very effective in producing a desired output process, and this can make a benefit for searching, matching and recognition process in many applications especially the pattern recognition.

There are widely several applications for the neural network in pattern recognition like:

- Optical Character Recognition (OCR) (Handwritten and printed texts)
- Biometrics (Face recognition ,finger prints and speech recognition).
- Diagnostic systems (Medical and machine diagnostics).
- Military applications (Automated Target Recognition).

4.2 Mapping and loading the data to the Neural Network

The following Figure 4.3 shows the stages for this method. The first step in this method was loading the data, which consists of inputs and targets that - as mentioned previously- has been be taken from data sequences for proteins and genes within specific sites, then using the mapping method that has been discussed previously, which focused on giving the amino acids C an H the maximum absolute values referred to the properties of the zinc fingers.

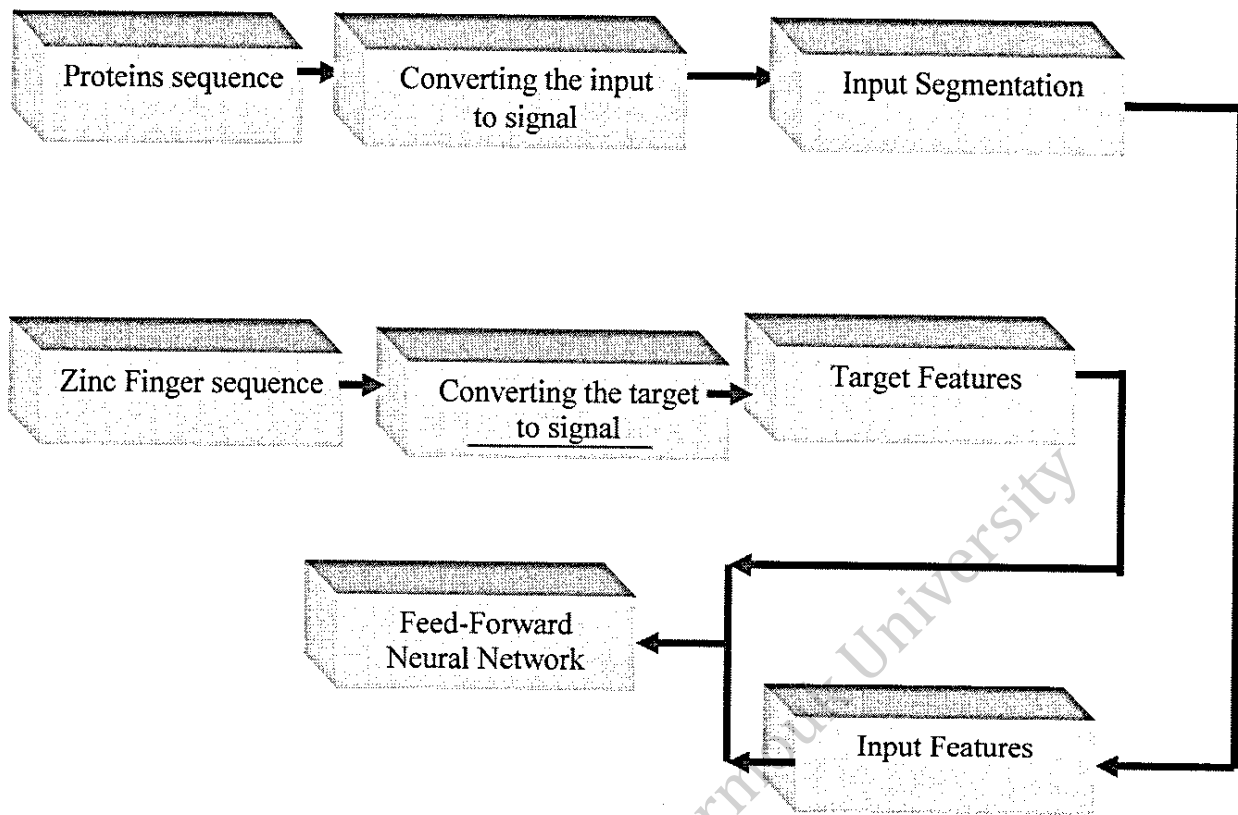


Figure 4.3 Data processing before fed to the input of the neural network

After the second stage, which was the segmentation method that has been discussed previously, the neural network was used as a classifier. It was fed by the feature input and target vectors that have been taken out from the segmentation stage, and will be discussed next.

4.2.1 Input and Target Feature Vectors and Classes

Feature extraction is a special method used in pattern recognition, which reduce the dimensionality of signal or image processing. As can be seen in Figure 4.4, pattern recognition systems have three components: signal segmentation, feature analysis and pattern classification.

Here, feature analysis is achieved in the feature extraction step, in which the zinc vector was transformed into a feature vector. Finally, the signal classification can be done using classifiers such as neural networks which is explained next, where number of classes classified were $N=5$.

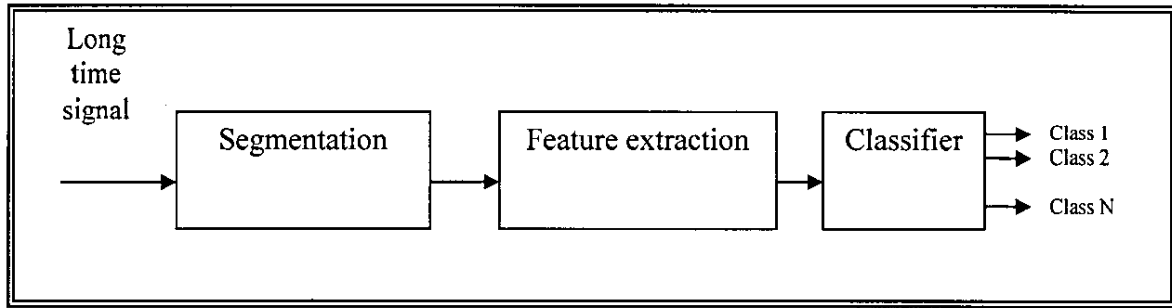


Figure 4.4 Block diagram of pattern recognition to classify input signal to N classes.

4.3 Method of Creating Features for Feeding Data.

There are several methods which are used in feature extraction depending on the type of the signal and the performance of these features that is fed to the classifier. Here this method selected these features depending on statistical values and the characteristics of the zinc fingers.

4.3.1 Statistical features

Feature vectors must be selected to give an acceptable classification output. Every class of these output classes must be separated to make no interference with other to get good determination; since the performance of the neural networks depends on the quality of the training data.

The popular statistical moments of time-series $x[n]$ which were used in this study are:

- Mean = $E[x] = \mu$.
- Variance = $E[(x - \mu)^2]$.
- The first six element of the zinc.
- The last six element of the zinc.

The first and last six elements of the zinc were chosen depending on zinc finger properties types mentioned in [30].

$$F = [\text{First six elements of } (x) \quad \text{Last six elements of } (x) \quad \text{Mean}(x) \quad \text{Variance}(x)] \dots 4.8$$

The protein matrix P_n contains all proteins with different lengths that we want to find the zinc finger through them, from which we get the segmentation and features vectors as seen in equation 4.9

$$P_{\text{proteins_signal}} \Rightarrow \begin{bmatrix} \text{Seg}_1 \\ \text{Seg}_2 \\ \text{Seg}_3 \\ \cdot \\ \cdot \\ \text{Seg}_n \end{bmatrix} \Rightarrow \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \cdot \\ \cdot \\ F_n \end{bmatrix} = \begin{bmatrix} [\text{First6}(x) & \text{Last6}(x) & \text{Mean}(x) & \text{var}(x)] \\ [\text{First6}(x) & \text{Last6}(x) & \text{Mean}(x) & \text{var}(x)] \\ [\text{First6}(x) & \text{Last6}(x) & \text{Mean}(x) & \text{var}(x)] \\ \dots\dots\dots \\ \dots\dots\dots \\ [\text{First6}(x) & \text{Last6}(x) & \text{Mean}(x) & \text{var}(x)] \end{bmatrix} \dots 4.9$$

In the training method of the neural network, the zinc finger types can be divided into 5 different classes. The first 4 classes have the required zinc types and the last class represents other else types.

These all classes are:

- Class 1 which has C4
- Class 2 which has CC-HC
- Class 3 which has CC-CH
- Class 4 which has CH-CC
- Class 5 which has Other else.

The zinc fingers contain all the zinc finger vectors with different lengths that we want to detect. It can yield also to feature vectors with same lengths. For the training of the neural network, first; a sample of 120 vectors was chosen which include the following classes:

- T(1:15) =1 which is C4.
- T(16:30) =2; which is CC-HC.
- T(31:45) =3; which is CC-CH.
- T(46:60) =4; which is CH-CC.
- T(61:120)=5; which is Other else.

4.4 Structure of the Created Neural Network

After this, the neural network was created. In this algorithm, we used a feed-forward network with the default tan-sigmoid transfer function in the hidden layer and linear transfer function in the output layer [42]. Here the information moves only in one direction, forward from the input nodes through the hidden nodes and to the output nodes. This type of neural network have been chosen because it is the simplest type of artificial neural network with no loops in this network and stable behaviors so it has used in solving many real problems like ECG signals. This algorithm was running through number of steps discussed in details.

4.4.1 Training process of the Neural Network

Once the network weights and biases have been initialized, the network is ready for training. The network can be trained with a set of proper inputs and targets. During the training, the weights and biases are self adjusted to minimize the network performance function.

In this step of training the network, the network uses the default algorithm for training. The application randomly divides the input vectors and the target vectors into three sets as follows:

- 60% of the vectors are used for training.
- 20% of the vectors are used to validate that the network is generalizing and to stop training before over fitting. Over fitting generally occurs when a model is very complex and describes an error instead of making a relationship. It depends on some items like the number of parameters, number of data, model structure and the data shape.
- The last 20% are used as test of network generalization which are completely independent on other sets. These are not standard percentages, but they can be modified as needed [42].

Figures 4.5 and 4.6 below show the starting and the training boxes of the neural network. It can be seen that during training progress it allows us to maintain the training process so one can see the states of the performance, training etc. This neural network consists of one output layer and one hidden layer with Size=20.

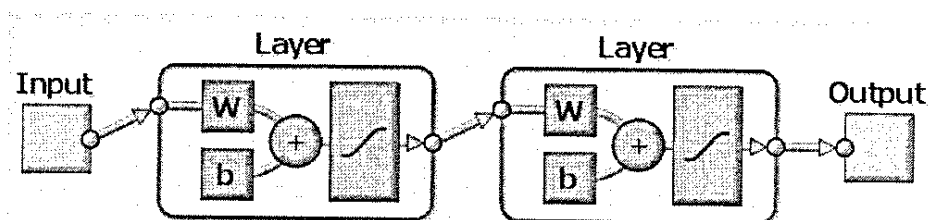


Figure 4.5 The starting box for the feed-forward neural network. It consists of two layers; hidden and output layer. The hidden layer has Size=20.

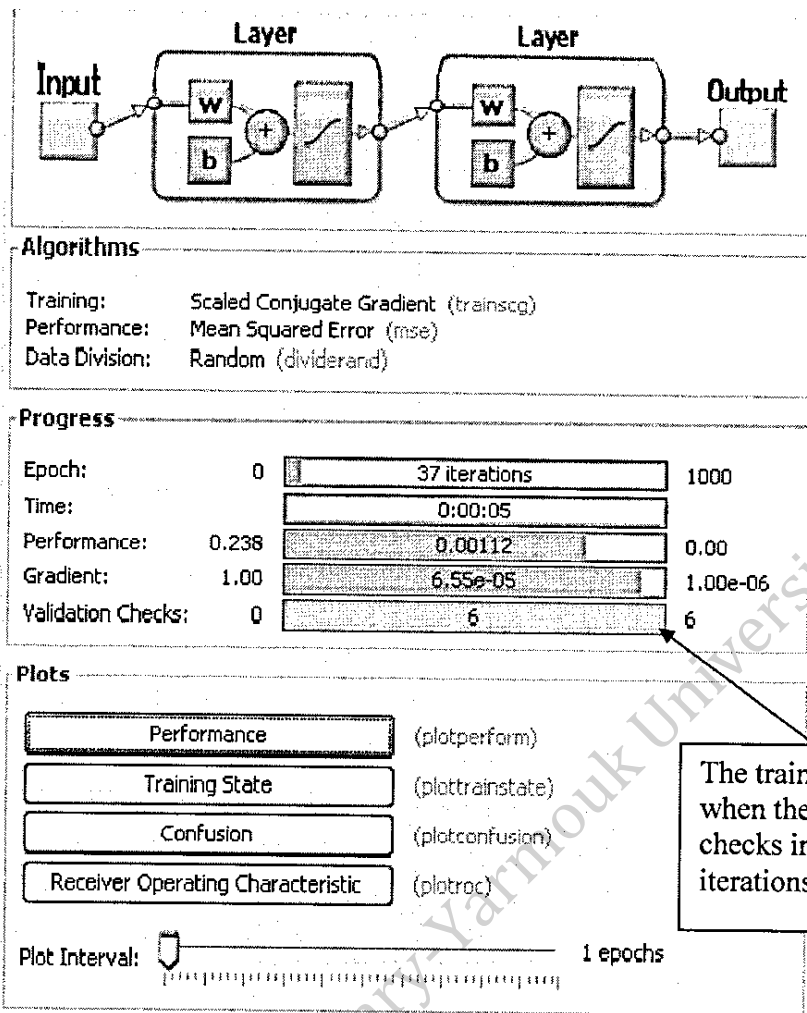


Figure 4.6 The training box for the feed-forward neural network. It shows all the data we need like the time, performance, training state, confusion matrix etc.

Figure 4.7 shows a plot for the performance in the training window. It shows plots of the training, validation, and test errors. In this stage the result can be considered if the following cases occur: The result during this stage will be acceptable for small final mean-square error or the test error and the validations error have similar characteristics. The training stopped when the validation error increased six iterations as shown in Figure 4.8.

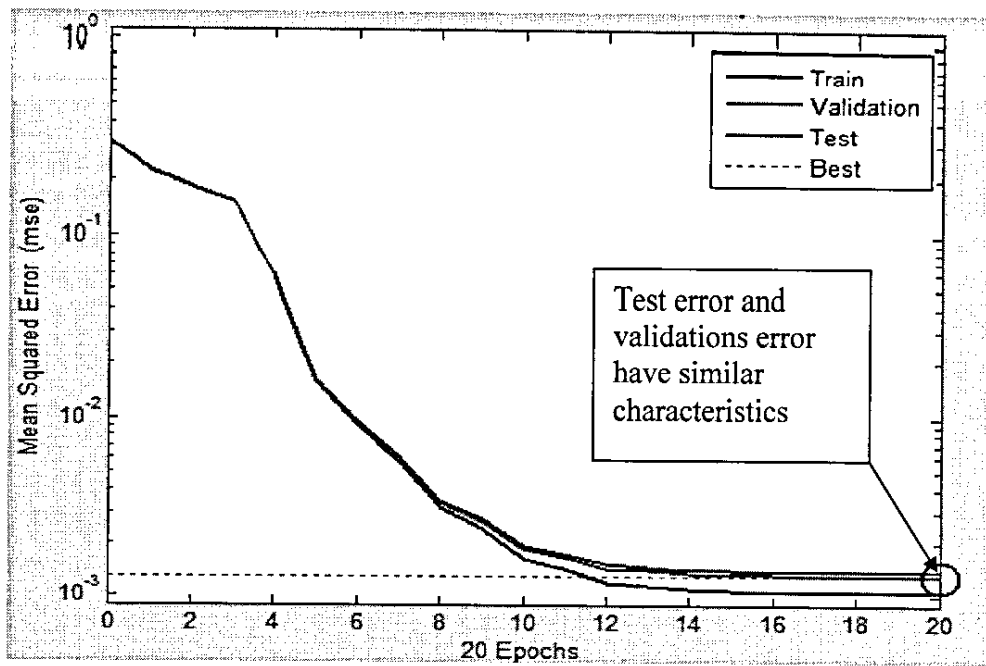


Figure 4.7 Box plot for the performance for the feed-forward neural network shows Test and validation errors have almost similar characteristics.

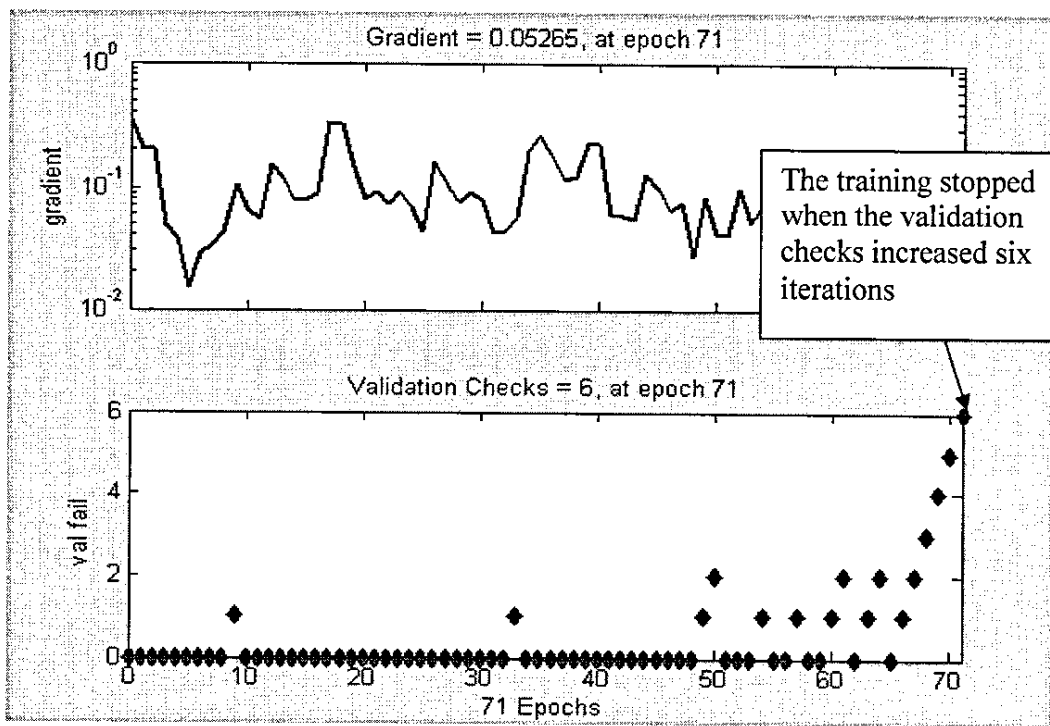


Figure 4.8 The validation and the gradient box for the feed-forward neural network. It shows that the training process stopped when the validation error increased six iterations

Figure 4.9 below shows the confusion matrices. The confusion matrix shows types of errors occurred in the final trained as cells, where it shows correctly cases classified and misclassified. The blue cell at the bottom shows the overall accuracies. The results for all three data sets (training, validation, and testing) show the efficiency of the recognition process.

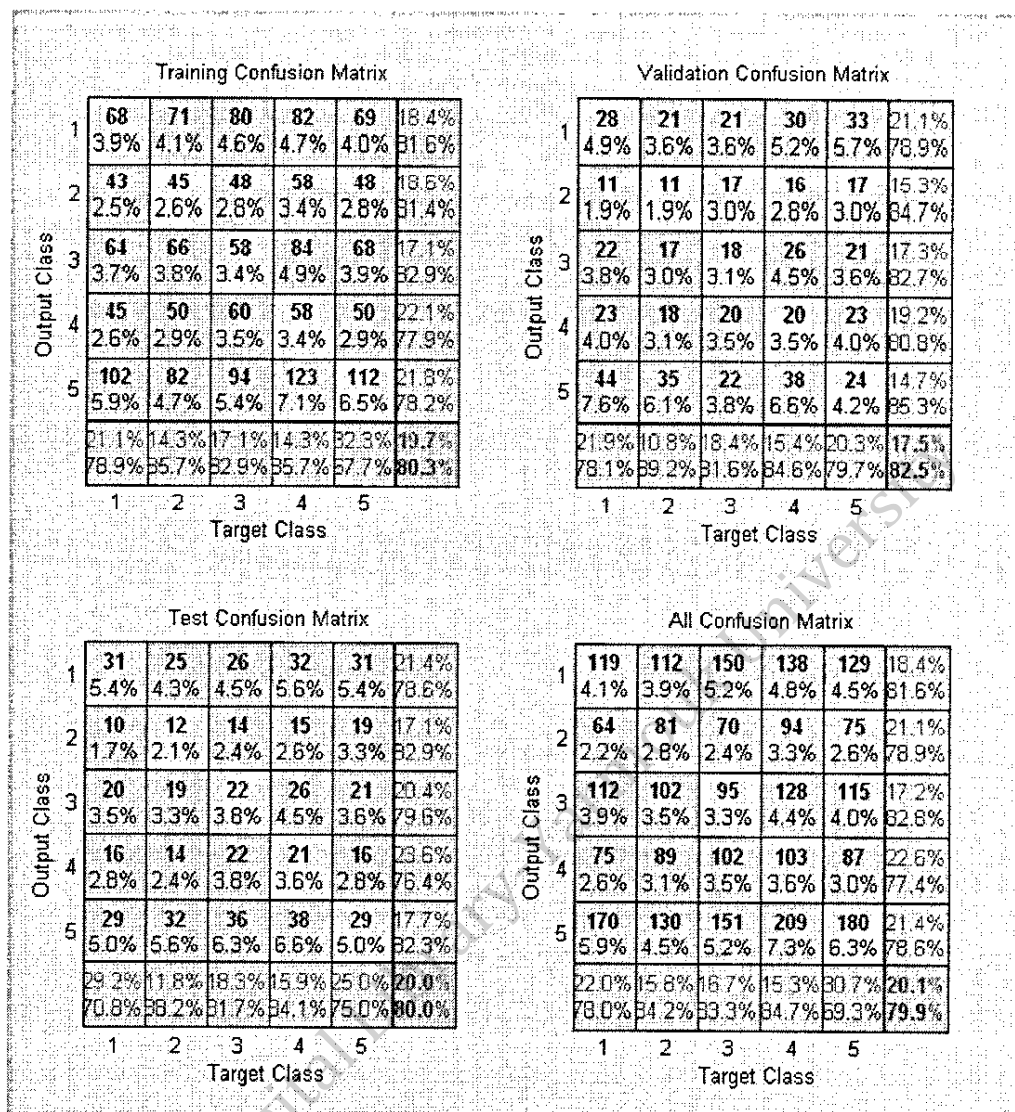


Figure 4.9 All confusion matrices, training, test and validation. They show an average percentage for each test in this sample.

4.4.2 Receiver Operating Characteristic (ROC).

To evaluate the performance of the classifier (learning) algorithm and check its reliability, there are several well accepted methods such as cross validation and Receiver Operating Characteristic (ROC). Figure 4.10 below shows the Receiver Operating Characteristic (ROC) curve. The colored lines in each axis represent the ROC curves for a test problem with 3 classes as an example. The ROC curve is a plot of the true positive rate (sensitivity) versus the false positive rate ($1 - \text{specificity}$) [41].

Sensitivity and specificity are expressions used to describe the performance of tests. These two measures are closely related to the concepts of type I and type II errors. These values are determined based on true positives, false positives, true negatives, and false

negatives as will be seen later. Sensitivity measures the proportion of actual positives which are correctly identified, while specificity measures the proportion of negatives which are correctly identified.

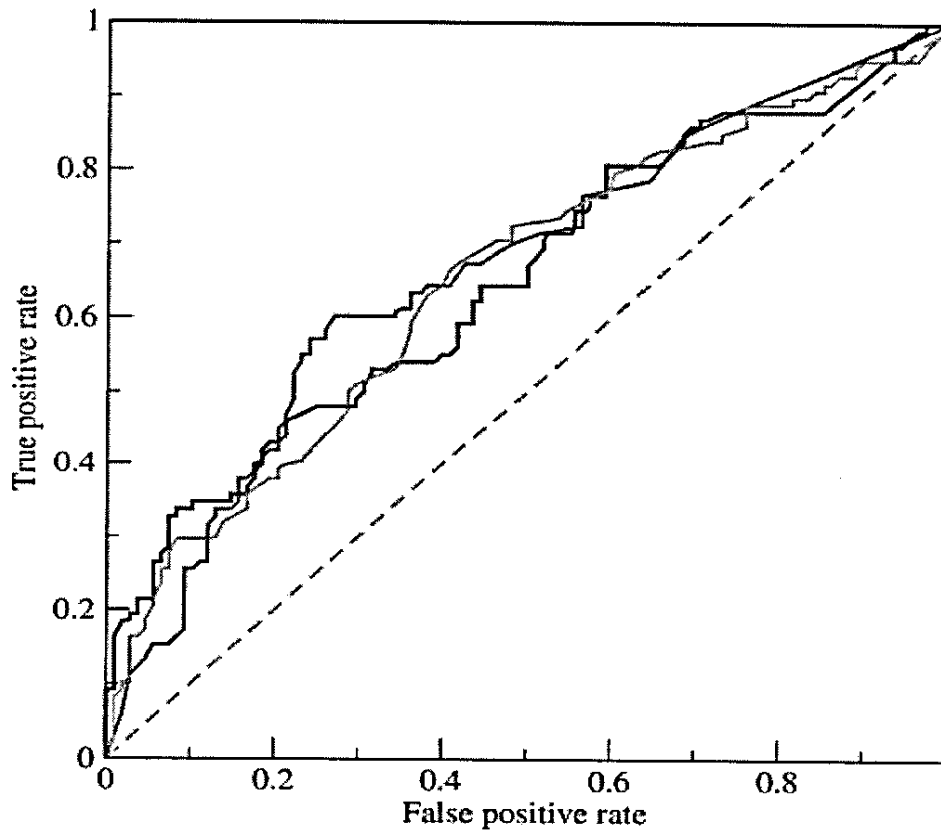


Figure 4.10 The Receiver Operating Characteristic (ROC) curve for 3 classes as an example. The more that these curves concave up, the better is the recognition process [43].

4.5 The Output of The Neural Network

From what has been mentioned above, and after the neural network has run in the previous algorithms, the output codes were written to give an outcome results that determined the following:

- The existence of the zinc finger.
- The zinc number inside a specific protein.
- The order of the zinc inside the protein.
- The class type of the zinc finger.
- The location of the zinc finger.
- The number of its repetition.

Figure 4.11 below shows some of these results with a test to see the right and wrong detection for some zinc finger proteins shown in Figures 4.12 and 4.13. While all the classified ones are shown in appendix I and test for sample of 400 zincs shown in appendix II

Zinc number is z001_01 has class type of 1: C4	Located from	1 24	Repaired	16 times
Zinc number is z001_02 has class type of 1: C4	Located from	21 41	Repaired	1 times
Zinc number is z001_03 has class type of 1: C4	Located from	38 61	Repaired	16 times
Zinc number is z001_04 has class type of 1: C4	Located from	95 118	Repaired	16 times
Zinc number is z001_05 has class type of 5: Otherwise	Located from	626 646	Repaired	30 times
Zinc number is z002_01 has class type of 1: C4	Located from	63 81	Repaired	1 times
Zinc number is z002_02 has class type of 1: C4	Located from	78 101	Repaired	16 times
Zinc number is z002_03 has class type of 2: CC-HC	Located from	366 386	Repaired	20 times
Zinc number is z002_04 has class type of 5: Otherwise	Located from	382 420	Repaired	3 times
Zinc number is z002_05 has class type of 1: C4	Located from	759 782	Repaired	16 times
Zinc number is z002_06 has class type of 2: CC-HC	Located from	779 797	Repaired	2 times
Zinc number is z003_01 has class type of 1: C4	Located from	1 24	Repaired	16 times
Zinc number is z003_02 has class type of 1: C4	Located from	21 31	Repaired	1 times
Zinc number is z003_03 has class type of 1: C4	Located from	28 51	Repaired	16 times
Zinc number is z003_04 has class type of 1: C4	Located from	48 68	Repaired	1 times
Zinc number is z003_05 has class type of 1: C4	Located from	108 124	Repaired	1 times
Zinc number is z003_06 has class type of 2: CC-HC	Located from	121 141	Repaired	20 times
Zinc number is z003_07 has class type of 5: Otherwise	Located from	137 196	Repaired	1 times
Zinc number is z003_08 has class type of 1: C4	Located from	193 216	Repaired	16 times
Zinc number is z003_09 has class type of 1: C4	Located from	213 245	Repaired	1 times
Zinc number is z003_10 has class type of 2: CC-HC	Located from	242 262	Repaired	20 times
Zinc number is z003_11 has class type of 4: CH-CC	Located from	349 370	Repaired	13 times
Zinc number is z003_12 has class type of 5: Otherwise	Located from	365 378	Repaired	5 times
Zinc number is z003_13 has class type of 5: Otherwise	Located from	378 397	Repaired	5 times
Zinc number is z003_14 has class type of 5: Otherwise	Located from	393 411	Repaired	14 times
Zinc number is z003_15 has class type of 4: CH-CC	Located from	408 429	Repaired	2 times
Zinc number is z003_16 has class type of 4: CH-CC	Located from	475 496	Repaired	13 times
Zinc number is z003_17 has class type of 5: Otherwise	Located from	491 504	Repaired	5 times
Zinc number is z003_18 has class type of 5: Otherwise	Located from	504 523	Repaired	5 times

Figure 4.11 The output of the neural network. It shows some of the results that were detected in this algorithm including their class type, location and number of repetition.

1													Mean	Variance	Class	Detection
2	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
3	1.0000	0.2500	0.2500	1.0000	0.4333	0.4667	0.2667	0.2667	1.0000	0.3833	0.1667	1.0000	0.5403	0.1222	1= C4	✓
4	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
5	1.0000	0.2500	0.2500	1.0000	0.2667	0.4667	0.1667	0.2833	1.0000	0.3833	0.1667	1.0000	0.5194	0.1326	1= C4	✓
6	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
7	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
8	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
9	1.0000	0.2667	0.3833	0.1000	1.0000	0.2333	0.1333	0.2667	1.0000	0.3833	0.1667	1.0000	0.4944	0.1466	1= C4	✓
10	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
11	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
12	-1.0000	0.2000	0.4000	0.1000	1.0000	0.4000	0.3500	1.0000	0.2000	0.1333	0.4000	1.0000	0.3486	0.2956	5= Otherwise	✓
13	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
14	1.0000	0.2500	0.2500	1.0000	0.4500	0.1833	0.2500	1.0000	0.3333	0.1667	0.4000	1.0000	0.3569	0.2903	2= CC-HC	✓
15	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
16	1.0000	0.2500	0.2500	1.0000	0.4333	0.1833	0.1833	0.1500	1.0000	0.3833	0.1667	1.0000	0.5000	0.1434	1= C4	✓
17	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
18	1.0000	0.2500	0.2500	1.0000	0.3833	0.2500	0.2833	0.2833	1.0000	0.1000	0.2500	1.0000	0.5042	0.1379	1= C4	✓
19	1.0000	0.2333	0.4500	1.0000	0.2833	0.4667	0.2167	0.2333	1.0000	0.1667	0.3833	1.0000	0.5361	0.1256	1= C4	✓
20	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
21	-1.0000	0.2000	0.4000	0.1000	1.0000	0.2333	0.4000	0.3833	1.0000	0.3833	0.1667	1.0000	0.3556	0.2936	5= Otherwise	✓
22	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
23	1.0000	0.2500	0.2500	1.0000	0.4333	0.1833	0.2167	0.2333	1.0000	0.1667	0.3833	1.0000	0.5097	0.1368	1= C4	✓
24	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
25	1.0000	0.1667	0.4000	1.0000	0.1667	0.2000	1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.1653	0.3885	5= Otherwise	✓
26	-1.0000	0.2167	0.2667	0.1000	0.4667	1.0000	0.1833	0.3500	0.4500	0.1667	1.0000	-1.0000	0.0167	0.4299	5= Otherwise	✓
27	-1.0000	0.4000	1.0000	0.4333	0.1000	0.4000	0.2167	1.0000	0.4333	0.1000	0.2500	-1.0000	0.0278	0.4378	5= Otherwise	✓
28	-1.0000	0.4333	0.1000	0.2500	1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
29	1.0000	0.1667	0.4000	1.0000	0.1500	0.2167	1.0000	0.3333	0.1500	0.4667	0.2333	-1.0000	0.1764	0.3900	5= Otherwise	✓

Figure 4.12 Test to see the right and wrong detection for 29 zinc finger proteins which shows that they are all correct.

33	-1.0000	0.4333	0.1000	0.2500	1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
34	1.0000	0.1833	0.4500	1.0000	0.3833	0.4667	0.2333	0.2333	1.0000	0.1000	0.2500	1.0000	0.5250	0.1341	1= C4	✓
35	1.0000	0.1833	0.4500	1.0000	0.3500	0.4500	0.1667	0.1667	1.0000	0.3833	0.1667	1.0000	0.5264	0.1333	1= C4	✓
36	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
37	1.0000	0.1667	0.4000	1.0000	0.1667	0.2000	0.4000	0.4000	1.0000	0.1667	0.3833	1.0000	0.5236	0.1328	1= C4	✓
38	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
39	-1.0000	0.2000	0.4000	0.1000	1.0000	0.4000	1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.0125	0.4285	5= Otherwise	✓
40	-1.0000	0.2167	0.2667	0.1000	0.4667	1.0000	0.1833	0.3500	0.4500	0.1667	1.0000	-1.0000	0.0167	0.4299	5= Otherwise	✓
41	-1.0000	0.4000	1.0000	0.4333	0.1000	0.4000	0.2167	1.0000	0.4333	0.1000	0.2500	-1.0000	0.0278	0.4378	5= Otherwise	✓
42	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
43	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
44	-1.0000	0.2167	0.2833	0.2333	-1.0000	0.1000	1.0000	0.1000	1.0000	0.3833	0.1667	1.0000	0.0403	0.4870	5= Otherwise	✓
45	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
46	1.0000	0.2500	0.2500	1.0000	0.1000	0.1833	0.2500	0.1500	0.4000	0.2333	1.0000	-1.0000	0.3181	0.2958	4= CH-CC	✗
47	1.0000	-1.0000	0.2000	0.1667	0.4000	0.1833	0.4000	0.1833	1.0000	0.1667	0.3833	1.0000	0.3403	0.2946	2= CC-HC	✓
48	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.3500	0.2167	1.0000	0.1000	0.2500	1.0000	0.4972	0.1473	1= C4	✓
49	1.0000	0.2000	0.4000	0.1000	1.0000	0.1333	0.1833	0.2167	0.4333	1.0000	0.1833	1.0000	0.3208	0.2995	2= CC-HC	✓
50	1.0000	0.1833	0.4500	1.0000	0.2667	0.4000	0.2667	1.0000	0.4333	0.2167	0.1000	-1.0000	0.1931	0.3942	5= Otherwise	✓
51	1.0000	0.1333	0.2167	1.0000	0.3833	0.1833	0.2667	1.0000	0.2167	0.2833	0.2333	1.0000	0.4931	0.1437	1= C4	✓
52	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
53	-1.0000	0.2167	0.2833	0.2333	-1.0000	0.1000	0.1833	0.1000	0.2500	-1.0000	0.1667	1.0000	-0.0389	0.3916	5= Otherwise	✓
54	1.0000	0.2333	0.4500	1.0000	0.2667	0.4000	0.2667	1.0000	0.4333	0.2167	0.2167	-1.0000	0.2069	0.3934	5= Otherwise	✓
55	1.0000	0.2667	0.4333	1.0000	0.4000	0.4333	0.2167	1.0000	0.4333	0.1000	0.2500	-1.0000	0.0444	0.4446	5= Otherwise	✓
56	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
57	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
58	-1.0000	0.2000	0.4000	0.1000	1.0000	0.1333	0.4000	0.2333	0.2167	0.1667	1.0000	-1.0000	0.1542	0.3847	5= Otherwise	✓
59	1.0000	-1.0000	0.2667	0.3500	0.4500	1.0000	0.2667	0.3500	0.4500	1.0000	0.2833	-1.0000	0.2847	0.4454	3= CC-CH	✗
60	-1.0000	0.2667	0.3500	0.4500	1.0000	0.2833	1.0000	0.1000	0.2000	0.1667	0.4000	-1.0000	0.0181	0.4282	5= Otherwise	✓
61	-1.0000	0.1000	0.2000	0.1667	1.0000	0.4000	0.2667	0.3500	0.4000	1.0000	0.2667	1.0000	0.1792	0.3888	5= Otherwise	✓

Figure 4.13 Test to see the right and wrong detection for another zinc finger proteins. It shows that there are two wrong classes.

Because the designing of zinc finger proteins is an important technology in science of clinical applications; such as the information about the occurrence, location, number of repetition of zinc fingers, etc, also percentages of types of zinc fingers and percentages of amino acids are also important in the developing field of molecular genome engineering designers.

For a sample of data that has 235 different proteins; we can also use it to find the percentage of every type of the 4 types of zinc fingers (C4, CC-HC, CC-CH, and CH-CC) with respect to the others. Figures 4.14 to 4.16 show these percentages in the testing, training and validation sets

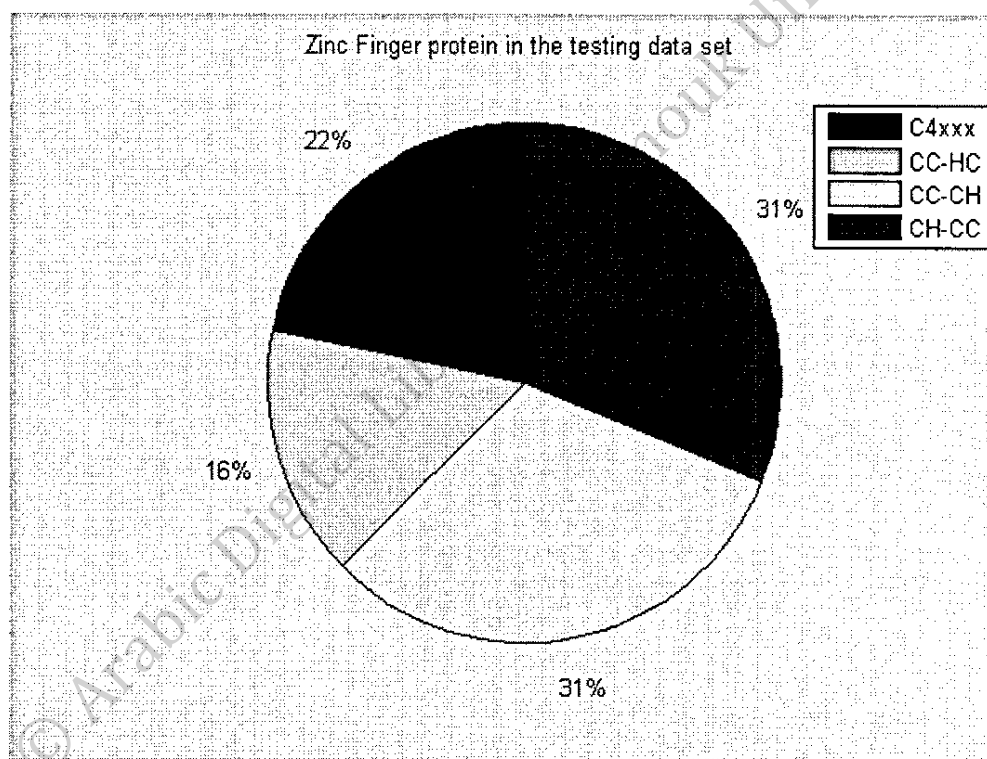


Figure 4.14 Percentage of each type of the 4 types of zinc fingers in the testing sets

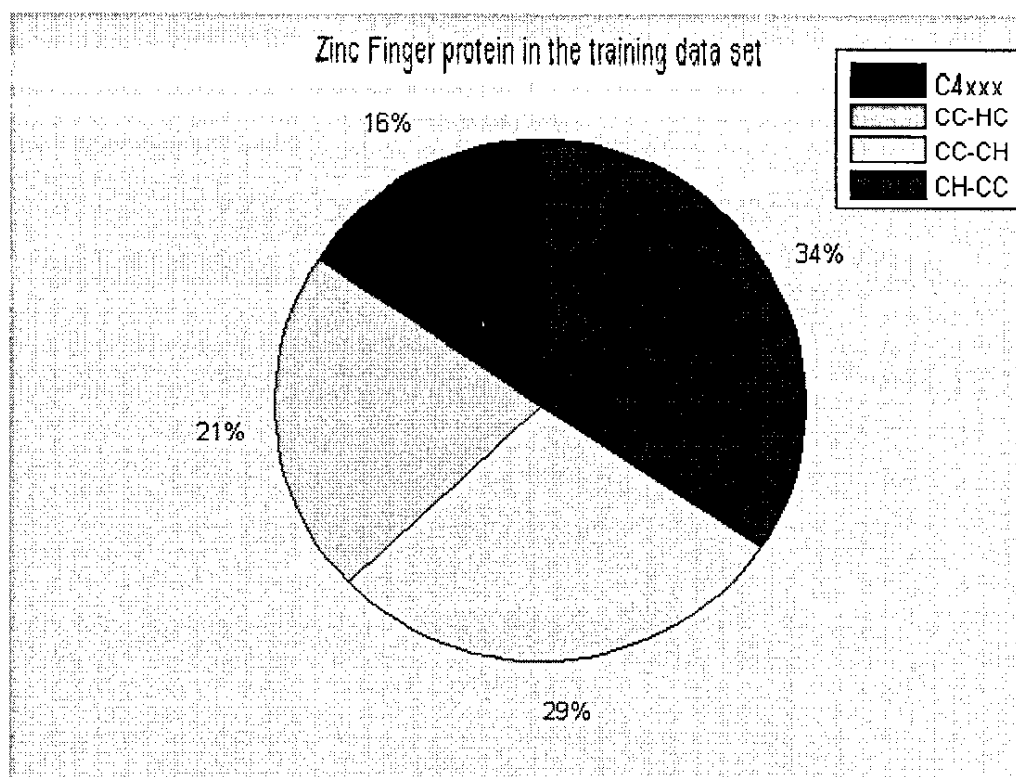


Figure 4.15 Percentage of each type of the 4 types of zinc fingers in the training sets

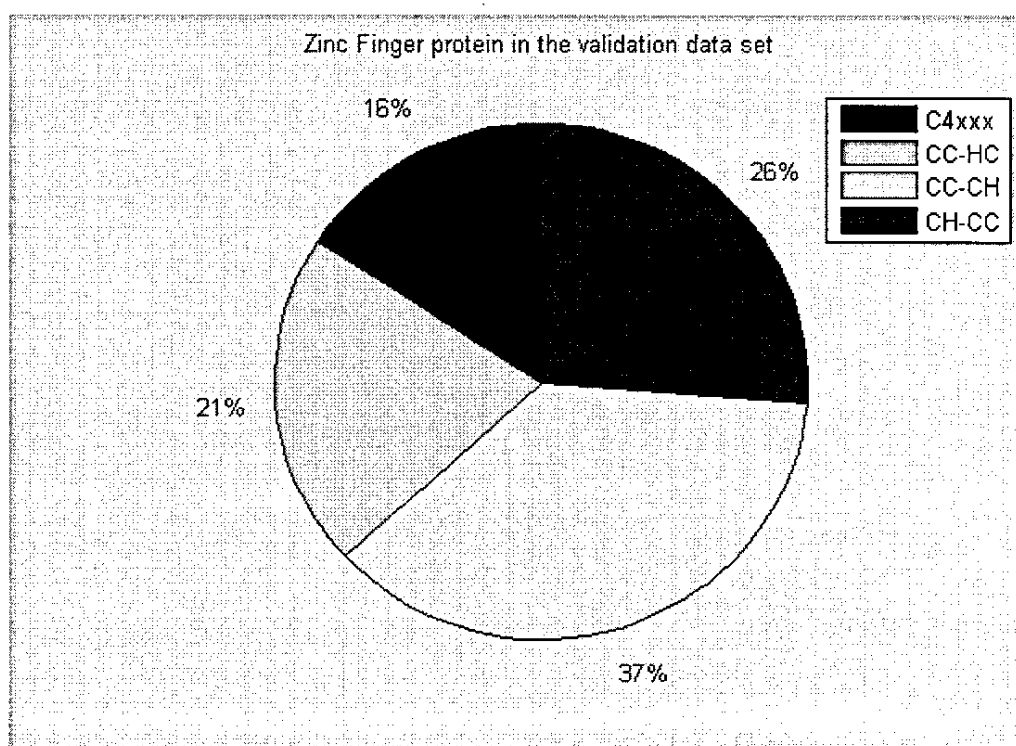


Figure 4.16 Percentage of each type of the 4 types of zinc fingers

The neural network method was applied to detect the percentage of each amino acid in the training sets. The following Figure 4.17 shows these percentages.

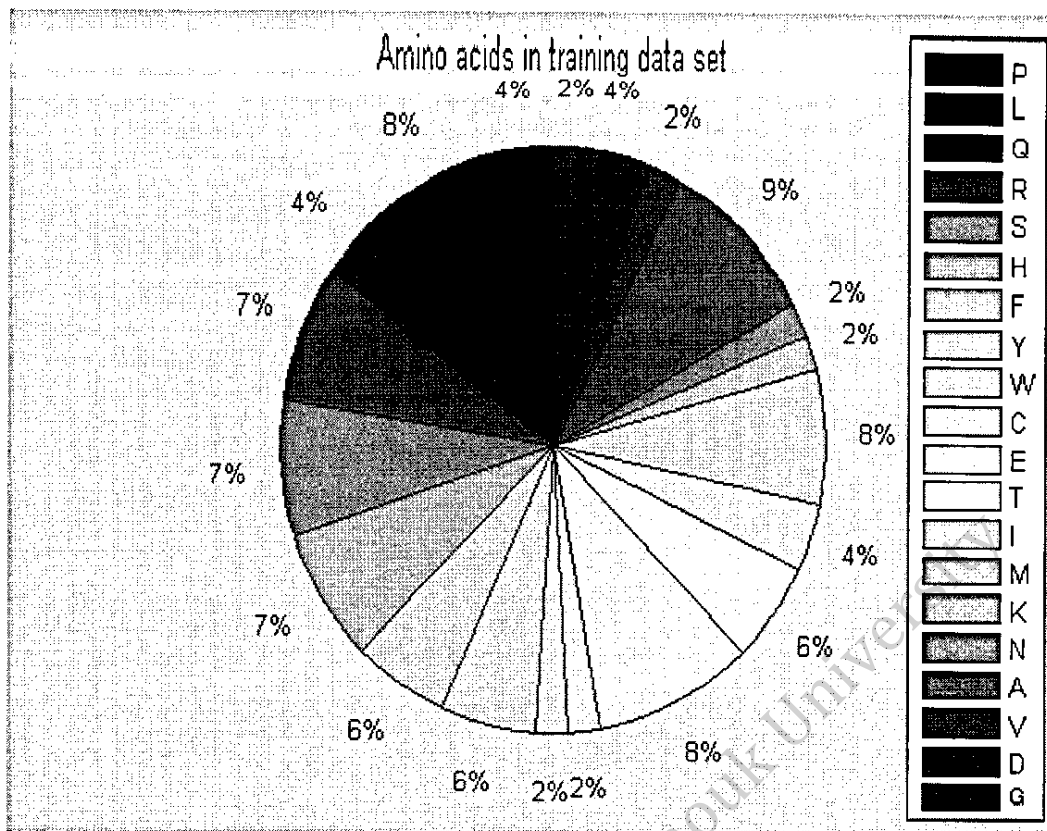


Figure 4.17 The percentages of each amino acid in the training sets.

The regression plot is another way that can be used to judge on the efficiency of the neural network. One can perform a linear regression plot between the network targets (T) with the corresponding inputs (P) or targets with the corresponding outputs, to find the regression (R-value). If $R=1$, then it means perfect correlation.

Figures 4.18 and 4.19 show the regression plot for network P with T and the regression plot for the output with T respectively. Figure 4.21 shows that for this sample P which is concentrated between $[-1,+1]$ and T which is concentrated between $[1,5]$ has good recognition process matching with regression value $R=0.99672$ for T with the corresponding outputs as shown in Figure 4.19.

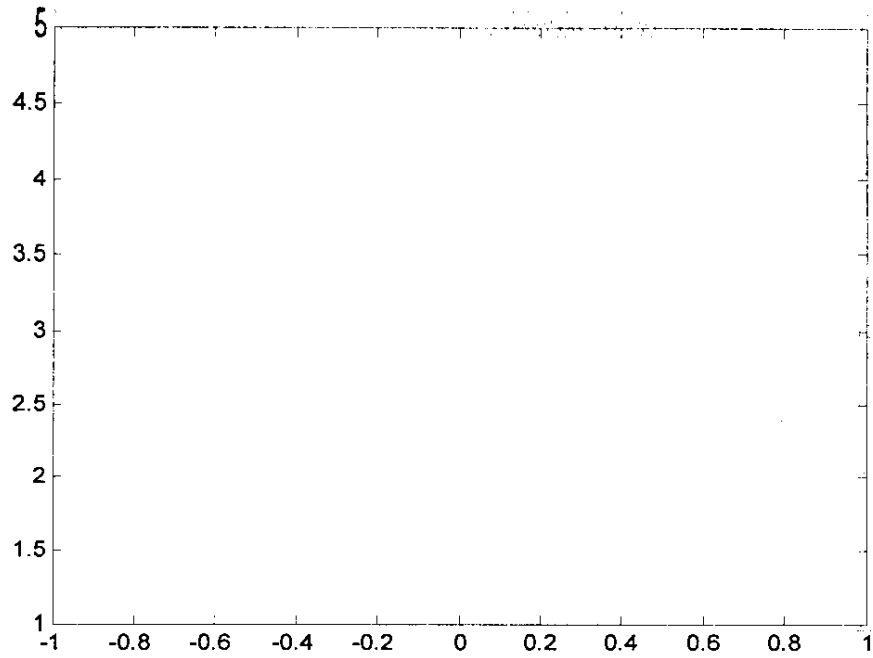


Figure 4.18 The regression plot for network P[-1,+1] with T[1,5] values.

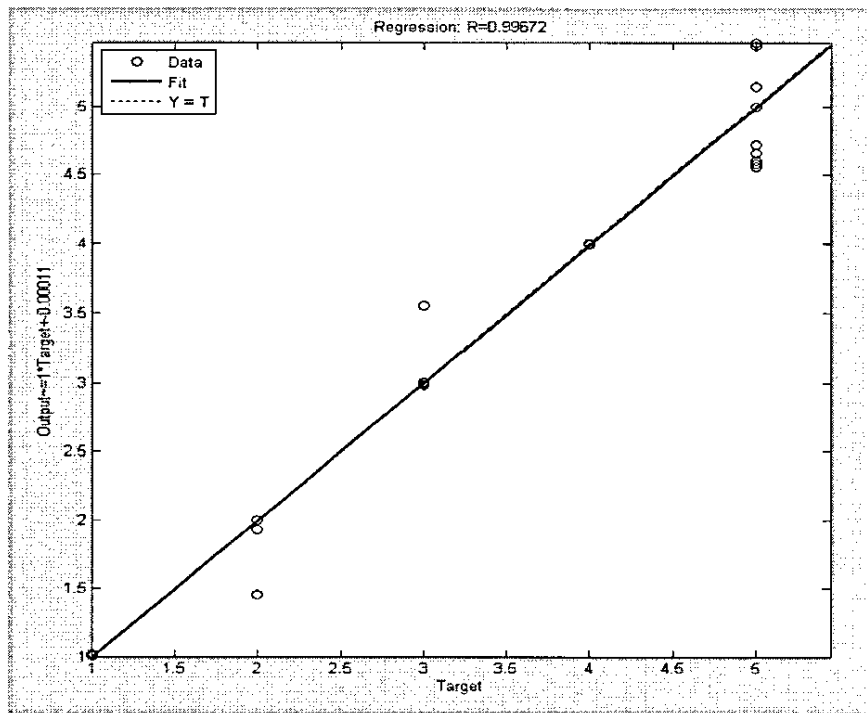


Figure 4.19 The regression plot for network output with T. It shows that for this sample the regression value $R=0.99672$.

More training of the neural network or more input vectors yields to best recognition.

Figure 4.20 shows that increasing the times of training process, leads to more accurate results. As seen in this figure, the regression plot of T with the corresponding outputs has a regression value $R=0.9993$

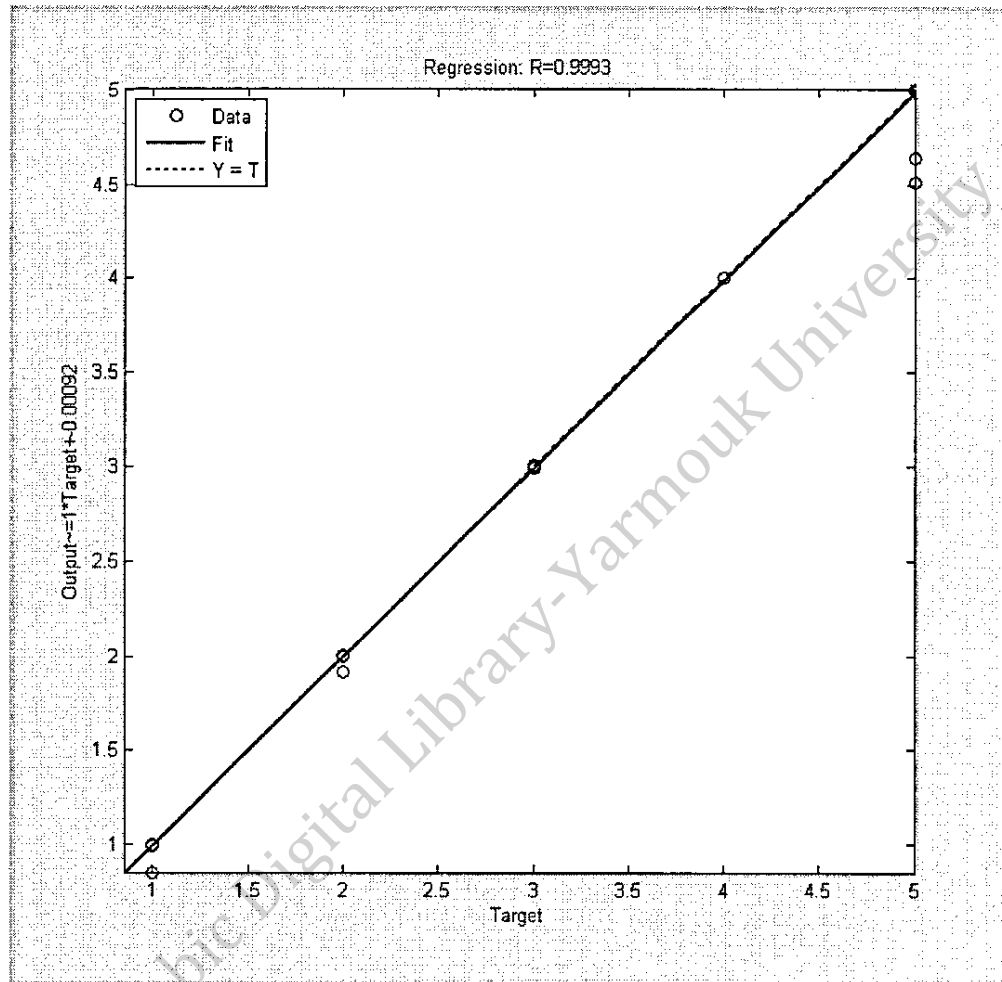


Figure 4.20 The regression plot for network output with T. It shows that more times of training process leads to more accurate results with regression value $R=0.9993$.

Results and Discussion

5.1 Introduction

This chapter shows the output results of this algorithm, which was implemented on each stage to reach the goals of this thesis. Each figure shows the output with an explanation for every one.

5.2 The Output of the Segmentation:

This method is very important for speeding up this algorithm. After converting the input data sequences to signals, the segmentation method has been used, which depends on partitioning the input to some vectors that have the most probability that they are zinc fingers. Figures 5.1 and 5.4 show the segmentation for some proteins, as an examples that they have some expected Zinc fingers at the beginning and the end for every one shown below of them.

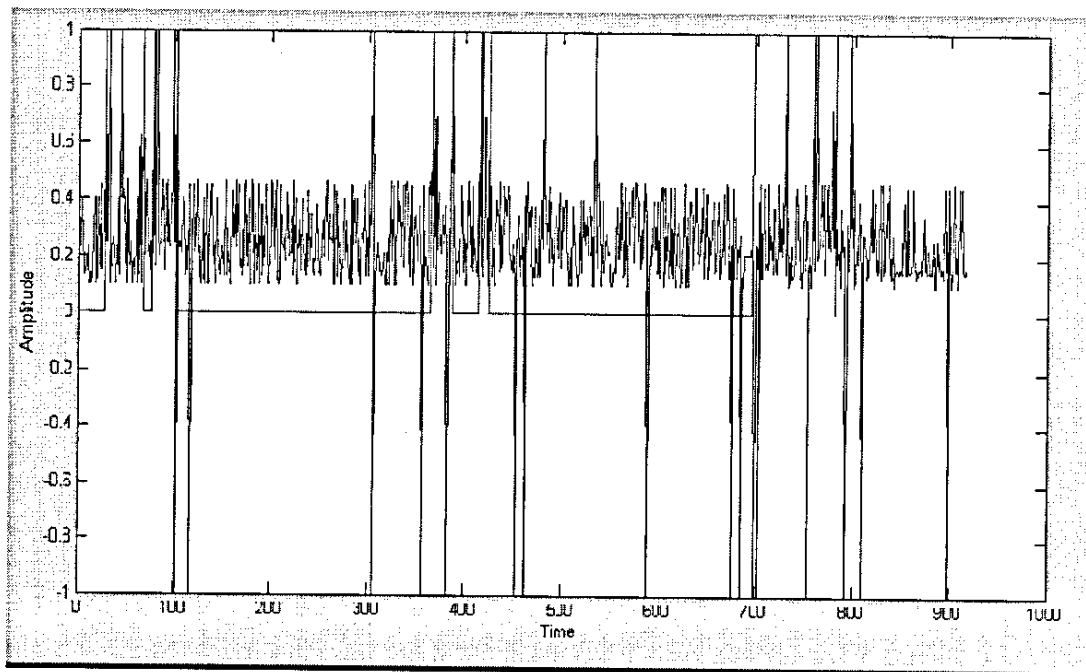


Figure 5.1 Segmentation method for one protein. It shows 6 expected zinc fingers with their locations at the bottom of the figure.

From Figure 5.1 one can see that this protein contains 6 possible zinc fingers. The first one is from 27 to 68 and the last one is from 758 to 783, while Figure 5.2 shows another protein that contains 8 possible zinc fingers.

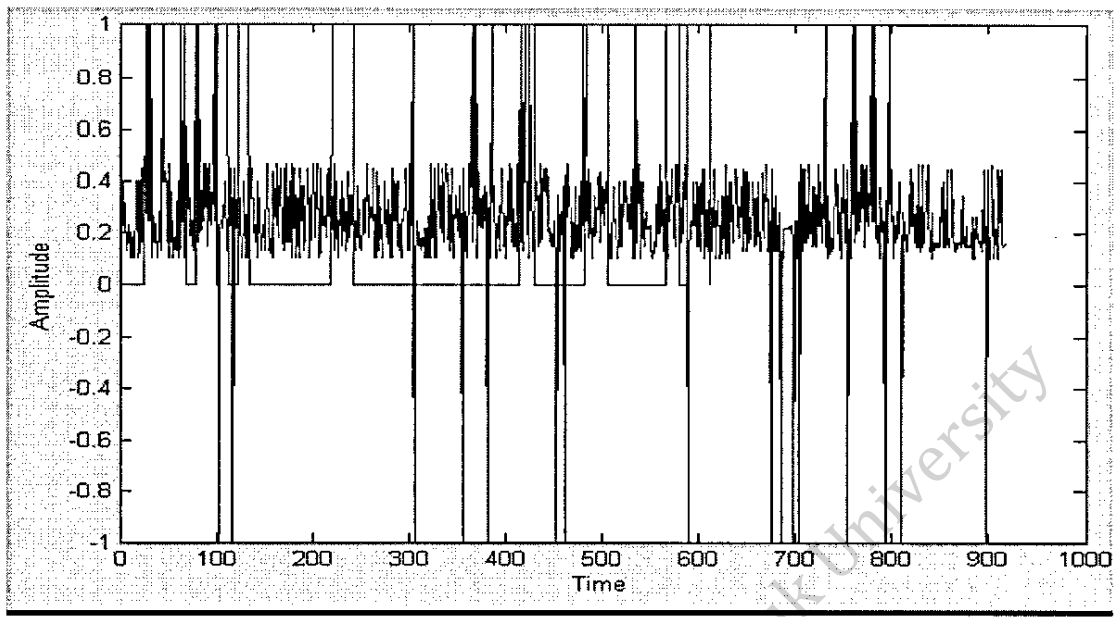
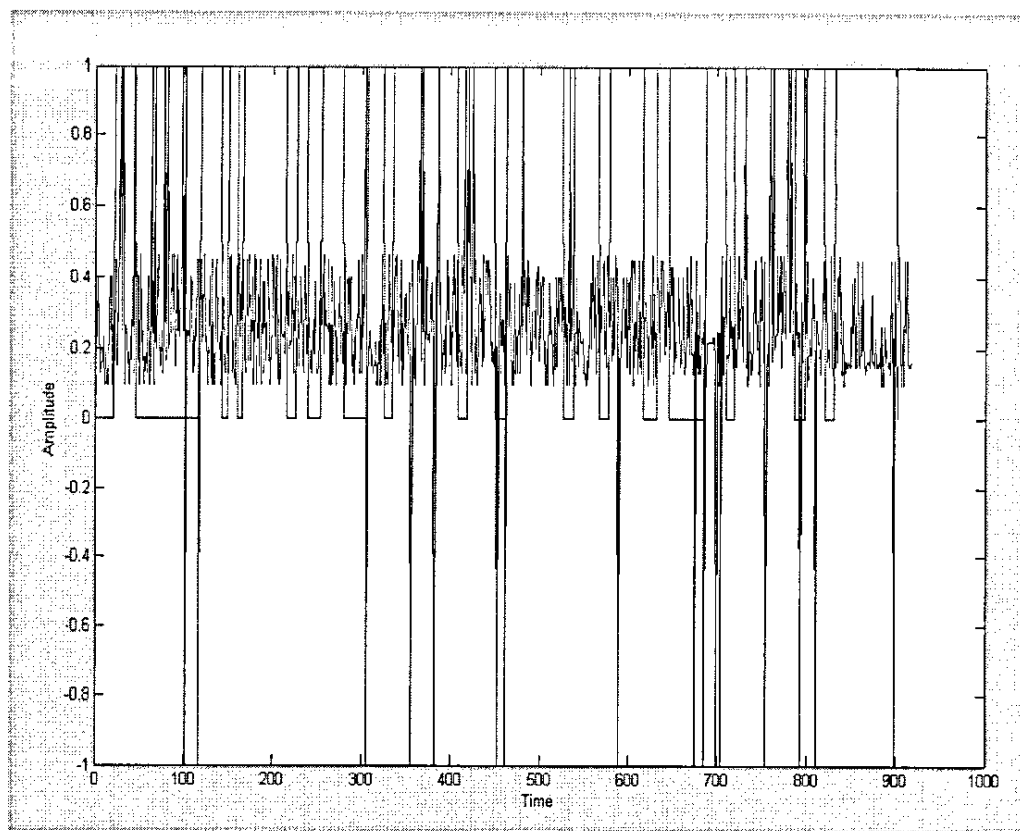


Figure 5.2 Segmentation method for another protein that shows 8 expected zinc fingers with their locations at the bottom of the figure.

This method has a great effect in this algorithm but it cannot determine the zinc fingers with very high percent, because it gives the most candidate zinc fingers, so some of them might not be zinc ones as you can see in figure 5.3.

Since this segmentation method has a great advantage in decreasing the time – it gives the output of the most candidate fingers signals only and not all the protein signals - it must be followed by a certain stage to confirm these fingers with high sensitivity, so it was the need for the neural network.



ans -

21	46
119	143
150	161
167	199
197	216
227	240
255	280
307	325
336	409
420	434
432	450
463	482
480	505
503	527
539	567
579	617
632	646
687	710
719	743
741	754
752	776
774	787
799	820
832	850
848	872
870	881
879	902

Figure 5.3 Segmentation method for long protein sequence that gives all the candidate zinc fingers. It shows the zinc fingers and the expected ones with their locations at the bottom of the figure.

5.3 The Output of The Neural Network:

5.3.1 Detection of Zinc Fingers

Here, the neural network algorithm has an excellent pattern recognition. The following Figure 5.4 shows some of the outcomes that determined the zinc finger results with an excellent percentage, followed by Figures 5.5 and 5.6 that represented a test check to see the right and wrong detection for some zinc finger proteins. The outcome results include:

- The existence of the zinc finger.
- The zinc number inside a specific protein.
- The order of the zinc inside the protein.
- The class type of the zinc finger.
- The location of the zinc finger.
- The number of its repetition.

FinalOutput =		
Zinc number is z001_01	has class type of 1=C4	Located from 1 24 Repeated 17 times
Zinc number is z001_02	has class type of 1=C4	Located from 21 41 Repeated 1 times
Zinc number is z001_03	has class type of 1=C4	Located from 38 61 Repeated 17 times
Zinc number is z001_04	has class type of 1=C4	Located from 58 74 Repeated 1 times
Zinc number is z001_05	has class type of 1=C4	Located from 71 94 Repeated 17 times
Zinc number is z001_06	has class type of 1=C4	Located from 119 142 Repeated 17 times
Zinc number is z001_07	has class type of 5= Otherwise	Located from 650 670 Repeated 30 times
Zinc number is z002_01	has class type of 1=C4	Located from 63 81 Repeated 1 times
Zinc number is z002_02	has class type of 1=C4	Located from 78 101 Repeated 17 times
Zinc number is z002_03	has class type of 2= CC-HC	Located from 366 386 Repeated 20 times
Zinc number is z002_04	has class type of 5= Otherwise	Located from 382 420 Repeated 3 times
Zinc number is z002_05	has class type of 1=C4	Located from 759 782 Repeated 17 times
Zinc number is z002_06	has class type of 2= CC-HC	Located from 779 797 Repeated 2 times
Zinc number is z003_01	has class type of 1=C4	Located from 1 24 Repeated 17 times
Zinc number is z003_02	has class type of 1=C4	Located from 21 31 Repeated 1 times
Zinc number is z003_03	has class type of 1=C4	Located from 28 51 Repeated 17 times
Zinc number is z003_04	has class type of 1=C4	Located from 48 68 Repeated 1 times
Zinc number is z003_05	has class type of 1=C4	Located from 108 124 Repeated 1 times
Zinc number is z003_06	has class type of 2= CC-HC	Located from 121 141 Repeated 20 times
Zinc number is z003_07	has class type of 5= Otherwise	Located from 137 196 Repeated 1 times
Zinc number is z003_08	has class type of 1=C4	Located from 193 216 Repeated 17 times
Zinc number is z003_09	has class type of 1=C4	Located from 213 245 Repeated 1 times
Zinc number is z003_10	has class type of 2= CC-HC	Located from 242 262 Repeated 20 times
Zinc number is z003_11	has class type of 4= CH-CC	Located from 349 370 Repeated 13 times
Zinc number is z003_12	has class type of 5= Otherwise	Located from 365 378 Repeated 5 times
Zinc number is z003_13	has class type of 5= Otherwise	Located from 378 397 Repeated 5 times
Zinc number is z003_14	has class type of 5= Otherwise	Located from 393 411 Repeated 14 times
Zinc number is z003_15	has class type of 5= Otherwise	Located from 408 429 Repeated 2 times

Figure 5.4 The output of the neural network. It shows some of the results that was detected in this algorithm, including their class types, location and number of repetition.

1													Mean	Variance	Class	Detection
2	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
3	1.0000	0.2500	0.2500	1.0000	0.4333	0.4667	0.2667	0.2667	1.0000	0.3833	0.1667	1.0000	0.5403	0.1222	1= C4	✓
4	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
5	1.0000	0.2500	0.2500	1.0000	0.2667	0.4667	0.1667	0.2833	1.0000	0.3833	0.1667	1.0000	0.5194	0.1326	1= C4	✓
6	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
7	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
8	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	-1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
9	1.0000	0.2667	0.3833	0.1000	1.0000	0.2333	0.1333	0.2667	1.0000	0.3833	0.1667	1.0000	0.4944	0.1466	1= C4	✓
10	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
11	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	-1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
12	-1.0000	0.2000	0.4000	0.1000	1.0000	0.4000	0.3500	1.0000	0.2000	0.1333	0.4000	1.0000	0.3486	0.2956	5= Otherwise	✓
13	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
14	1.0000	0.2500	0.2500	1.0000	0.4500	0.1833	0.2500	-1.0000	0.3333	0.1667	0.4000	1.0000	0.3569	0.2903	2= CC-HC	✓
15	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
16	1.0000	0.2500	0.2500	1.0000	0.4333	0.1833	0.1833	0.1500	1.0000	0.3833	0.1667	1.0000	0.5000	0.1434	1= C4	✓
17	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
18	1.0000	0.2500	0.2500	1.0000	0.3833	0.2500	0.2833	0.2833	1.0000	0.1000	0.2500	1.0000	0.5042	0.1379	1= C4	✓
19	1.0000	0.2333	0.4500	1.0000	0.2833	0.4667	0.2167	0.2333	1.0000	0.1667	0.3833	1.0000	0.5361	0.1256	1= C4	✓
20	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	-1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
21	-1.0000	0.2000	0.4000	0.1000	1.0000	0.2333	0.4000	0.3833	1.0000	0.3833	0.1667	1.0000	0.3556	0.2936	5= Otherwise	✓
22	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
23	1.0000	0.2500	0.2500	1.0000	0.4333	0.1833	0.2167	0.2333	1.0000	0.1667	0.3833	1.0000	0.5097	0.1368	1= C4	✓
24	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	-1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
25	1.0000	0.1667	0.4000	1.0000	0.1667	0.2000	-1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.1653	0.3885	5= Otherwise	✓
26	-1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.1833	0.3500	0.4500	0.1667	-1.0000	-1.0000	0.0167	0.4299	5= Otherwise	✓
27	-1.0000	0.4000	1.0000	0.4333	0.1000	0.4000	0.2167	1.0000	0.4333	0.1000	0.2500	-1.0000	0.0278	0.4378	5= Otherwise	✓
28	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
29	1.0000	0.1667	0.4000	1.0000	0.1500	0.2167	-1.0000	0.3333	0.1500	0.4667	0.2333	-1.0000	0.1764	0.3900	5= Otherwise	✓

Figures 5.5 Test to see the right and wrong detection for 29 zinc finger proteins. It shows that they are all correct.

33	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
34	1.0000	0.1833	0.4500	1.0000	0.3833	0.4667	0.2333	0.2333	1.0000	0.1000	0.2500	1.0000	0.5250	0.1341	1= C4	✓
35	1.0000	0.1833	0.4500	1.0000	0.3500	0.4500	0.1667	0.1667	1.0000	0.3833	0.1667	1.0000	0.5264	0.1333	1= C4	✓
36	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
37	1.0000	0.1667	0.4000	1.0000	0.1667	0.2000	0.4000	0.4000	1.0000	0.1667	0.3833	1.0000	0.5236	0.1328	1= C4	✓
38	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	-1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
39	-1.0000	0.2000	0.4000	0.1000	1.0000	0.4000	-1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.0125	0.4285	5= Otherwise	✓
40	-1.0000	0.2167	0.2667	0.1000	0.4667	-1.0000	0.1833	0.3500	0.4500	0.1667	-1.0000	-1.0000	0.0167	0.4299	5= Otherwise	✓
41	-1.0000	0.4000	1.0000	0.4333	0.1000	0.4000	0.2167	1.0000	0.4333	0.1000	0.2500	-1.0000	0.0278	0.4378	5= Otherwise	✓
42	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
43	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
44	-1.0000	0.2167	0.2833	0.2333	-1.0000	0.1000	-1.0000	0.1000	1.0000	0.3833	0.1667	1.0000	0.0403	0.4870	5= Otherwise	✓
45	1.0000	0.3833	0.1667	1.0000	0.2833	0.2000	0.3833	0.4667	1.0000	0.2500	0.2500	1.0000	0.5319	0.1263	1= C4	✓
46	1.0000	0.2500	0.2500	1.0000	0.1000	0.1833	0.2500	0.1500	0.4000	0.2333	-1.0000	-1.0000	0.3181	0.2958	4= CH-CC	✗
47	1.0000	-1.0000	0.2000	0.1667	0.4000	0.1833	0.4000	0.1833	1.0000	0.1667	0.3833	1.0000	0.3403	0.2946	2= CC-HC	✓
48	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.3500	0.2167	1.0000	0.1000	0.2500	1.0000	0.4972	0.1473	1= C4	✓
49	1.0000	0.2000	0.4000	0.1000	1.0000	0.1333	0.1833	0.2167	0.4333	-1.0000	0.1833	1.0000	0.3208	0.2995	2= CC-HC	✓
50	1.0000	0.1833	0.4500	1.0000	0.2667	0.4000	0.2667	1.0000	0.4333	0.2167	0.1000	-1.0000	0.1931	0.3942	5= Otherwise	✓
51	1.0000	0.1333	0.2167	1.0000	0.3833	0.1833	0.2667	1.0000	0.2167	0.2833	0.2333	1.0000	0.4931	0.1437	1= C4	✓
52	1.0000	0.1000	0.2500	1.0000	0.1333	0.2167	0.2667	1.0000	0.2167	0.2833	0.2333	-1.0000	0.1417	0.3772	5= Otherwise	✓
53	-1.0000	0.2167	0.2833	0.2333	-1.0000	0.1000	0.1833	0.1000	0.2500	-1.0000	0.1667	1.0000	-0.0389	0.3916	5= Otherwise	✓
54	1.0000	0.2333	0.4500	1.0000	0.2667	0.4000	0.2667	1.0000	0.4333	0.2167	0.2167	-1.0000	0.2069	0.3934	5= Otherwise	✓
55	1.0000	0.2667	0.4333	-1.0000	0.4000	0.4333	0.2167	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.0444	0.4446	5= Otherwise	✓
56	-1.0000	0.4333	0.1000	0.2500	-1.0000	0.3500	0.4667	0.2000	1.0000	0.1667	0.4000	1.0000	0.1972	0.3965	5= Otherwise	✓
57	1.0000	0.1667	0.3833	1.0000	0.4000	0.1000	0.2167	-1.0000	0.2000	0.4000	0.1000	1.0000	0.3306	0.2991	2= CC-HC	✓
58	-1.0000	0.2000	0.4000	0.1000	1.0000	0.1333	0.4000	0.2333	0.2167	0.1667	-1.0000	-1.0000	0.1542	0.3847	5= Otherwise	✓
59	1.0000	-1.0000	0.2667	0.3500	0.4500	1.0000	0.2667	0.3500	0.4500	1.0000	0.2833	-1.0000	0.2847	0.4454	3= CC-CH	✗
60	-1.0000	0.2667	0.3500	0.4500	1.0000	0.2833	-1.0000	0.1000	0.2000	0.1667	0.4000	-1.0000	0.0181	0.4282	5= Otherwise	✓
61	-1.0000	0.1000	0.2000	0.1667	0.4000	-1.0000	0.2667	0.3500	0.4000	1.0000	0.2667	1.0000	0.1792	0.3888	5= Otherwise	✓

Figures 5.6 Test to see the right and wrong detection for another zinc finger proteins. It shows that they are two wrong classes.

5.3.2 Detection the Percentage of the Four Class Types of the Zinc Fingers.

As mentioned previously, the designing of zinc finger proteins is an important technology in science for clinical applications; furthermore percentages of zinc finger types and percentages of amino acids also can be important in the developing field of molecular genome engineering for zinc finger designers.

For a sample of data for different proteins; this method found the percentage of every class type of the 4 types of zinc fingers (C4, CC-HC, CC-CH, and CH-CC) with respect to the others. Figures 5.7 to 5.9 show these percentages in testing, training and validation sets

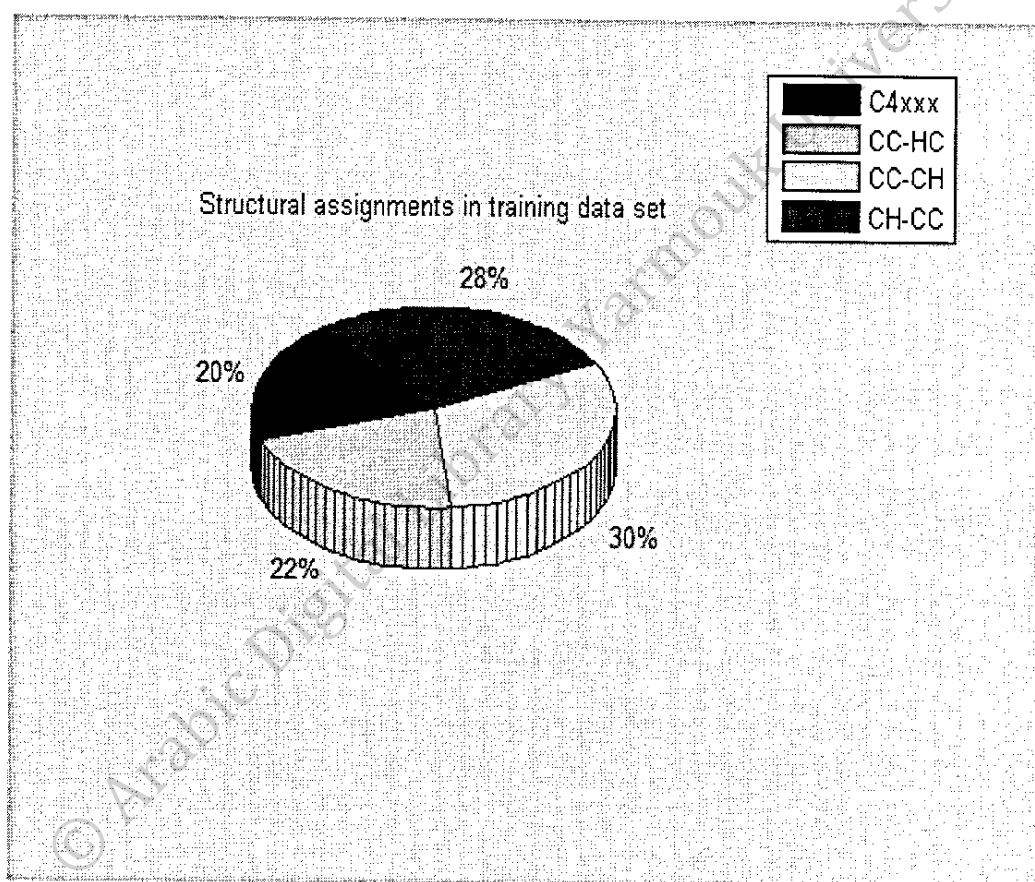


Figure 5.7 Percentage of each type of the 4 types of zinc fingers with respect to others in training sets of protein sequences.

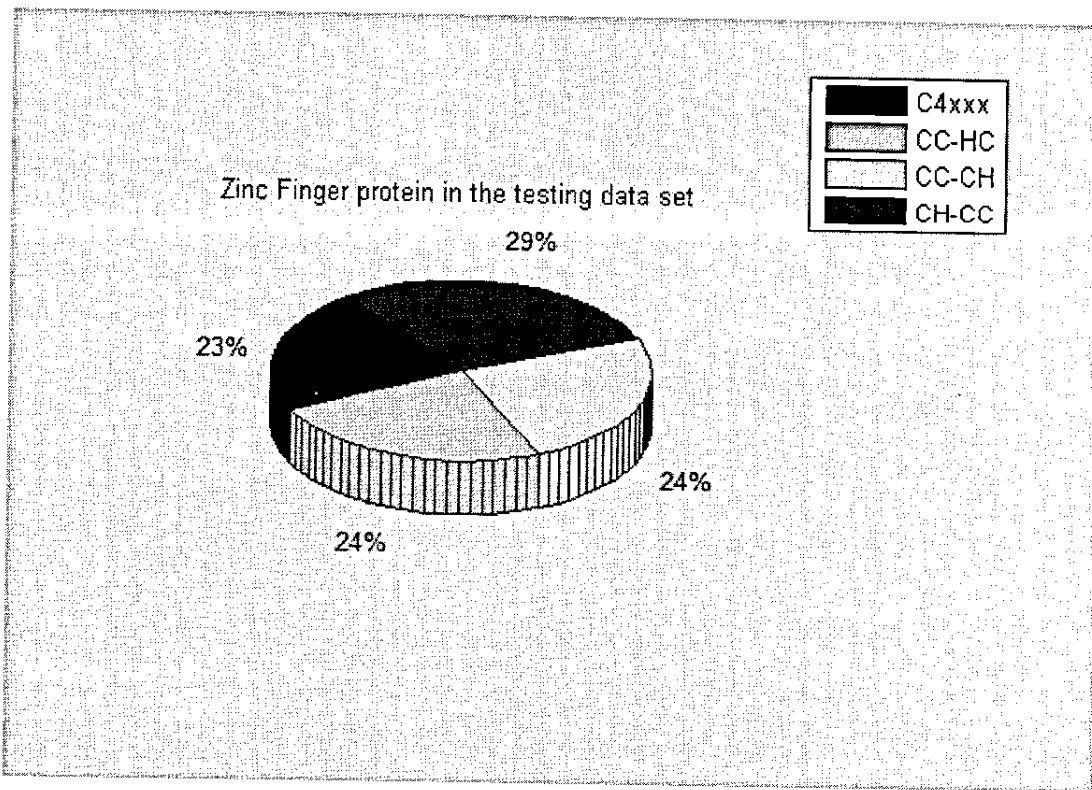


Figure 5.8 Percentage of each type of the 4 types of zinc fingers with respect to others in testing sets of protein sequences.

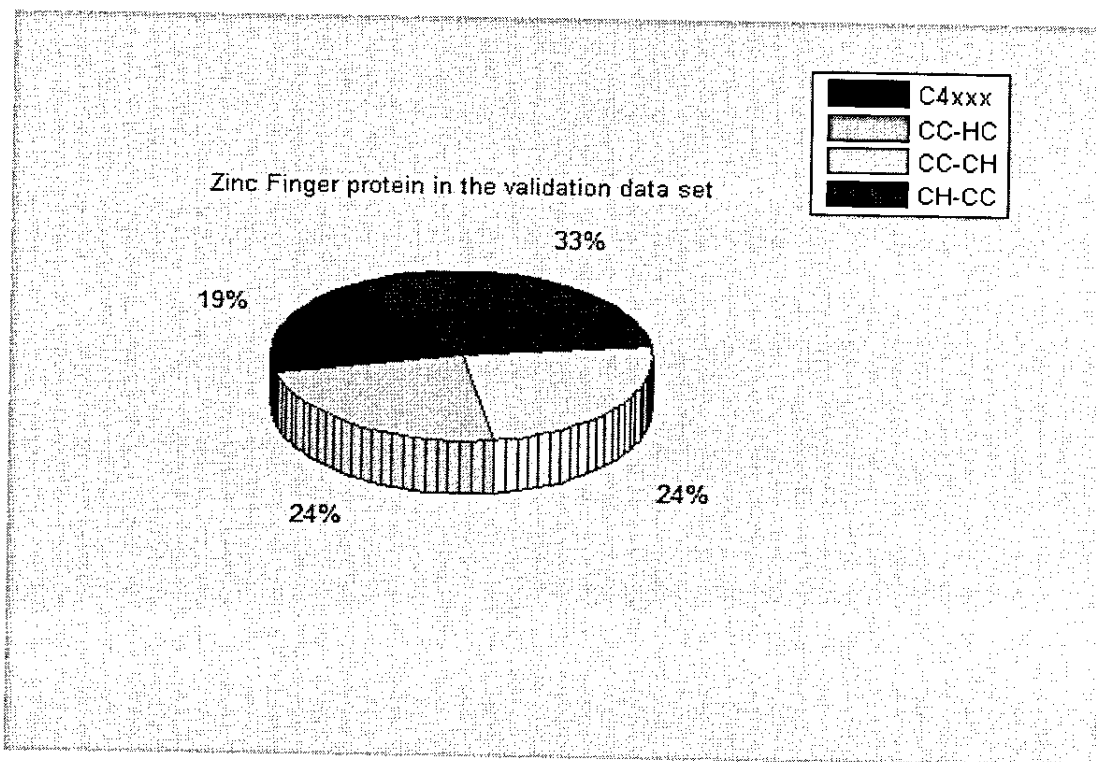


Figure 5.9 Percentage of each type of the 4 types of zinc fingers with respect to others in validation sets of protein sequences.

5.3.3 Detection of Amino Acids

The neural network method was applied successfully in this algorithm. First, it has been applied on four different types of zinc fingers which are: C4, CC-CH, CC-HC and CH-CC. One of these results for this method was detecting the percentage of each amino acid in the test, validation and training sets. The following figures 5.10 to 5.12 show these results. As you can see, these percentages differ from each other because the samples used in testing, validation and training are not the same samples.

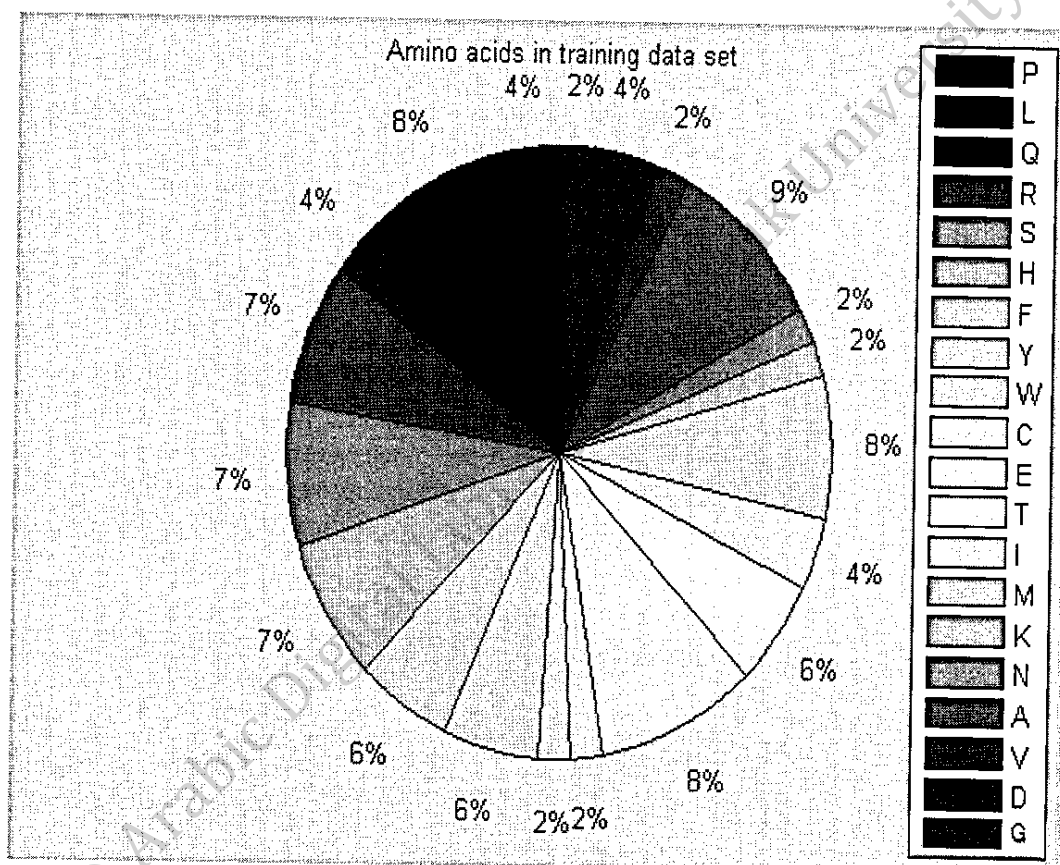


Figure 5.10 Percentage of each amino acid with respect to the others in all proteins in the training sets.

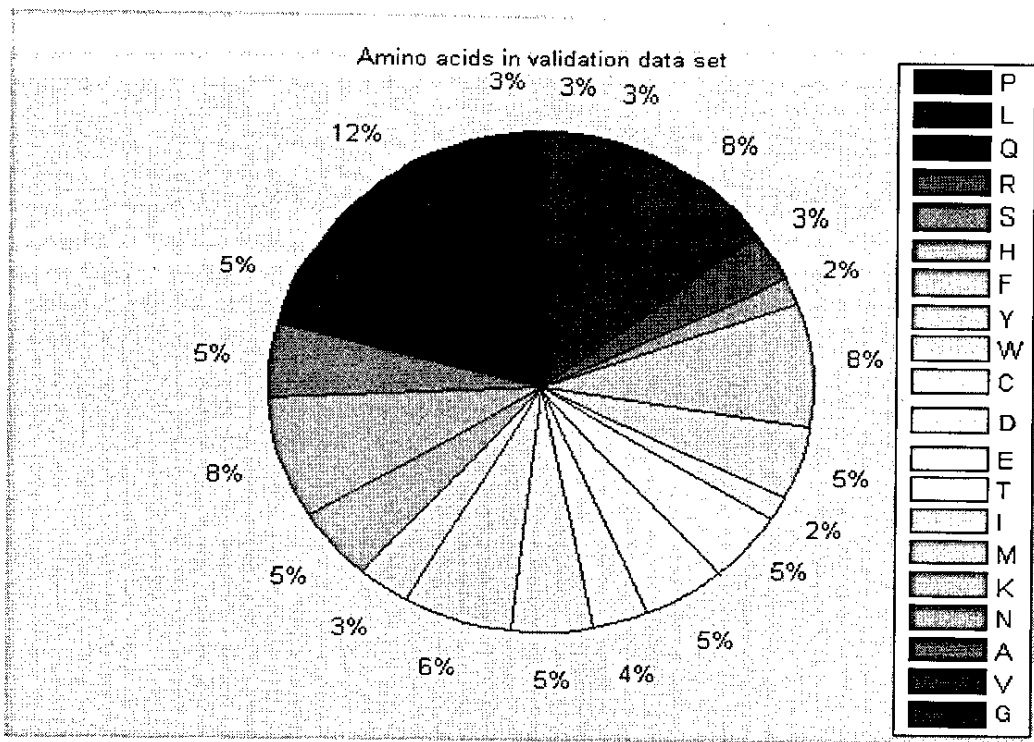


Figure 5.11 Percentage of each amino acid with respect to the others in all proteins in the validation sets.

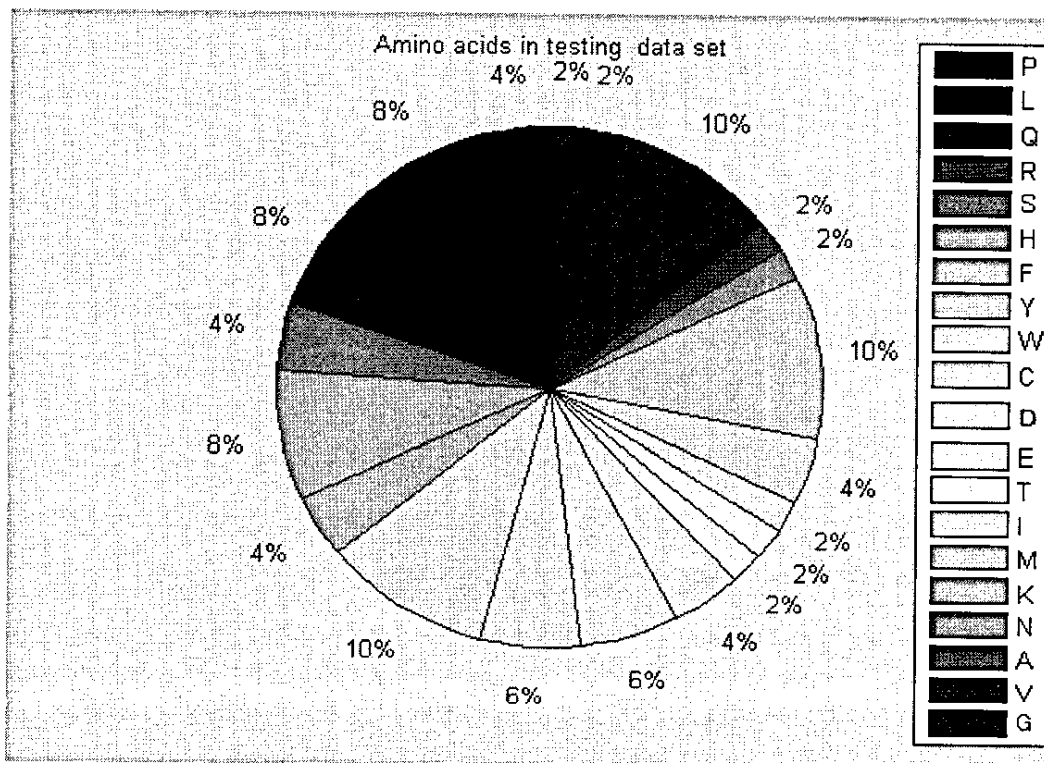


Figure 5.12 Percentage of each amino acid with respect to the others in all proteins in the testing sets.

5.4 Statistical Calculations

5.4.1 Sensitivity and Specificity

Sensitivity and Specificity can be calculated from the positive and negative values as shown in Table 5.1 below. These calculated values were used on sample =400. All results were shown in Table 5.2.

No.of Samples		<i>Positive</i>	<i>Negative</i>	Predictive values
Test	<i>Positive</i>	True Positive	False Positive (Type I error)	→ Positive predictive value
	<i>Negative</i>	False Negative (Type II error)	True Negative	→ Negative predictive value
		↓ Sensitivity	↓ Specificity	

Table 5.1 The positive and negative test values.

Sample=400		<i>Positive</i>	<i>Negative</i>	Predictive values
Test	<i>Positive</i>	TP = 145	FP = 12	→ Positive predictive value = TP / (TP + FP) = 145 / (145 + 12) = 145 / 157 = 92.236%
	<i>Negative</i>	FN = 5	TN = 238	→ Negative predictive value = TN / (FN + TN) = 238 / (5 + 238) = 238 / 243 = 97.9%
		↓ Sensitivity = TP / (TP + FN) = 145 / (145 + 5) = 145 / 150 = 96.67%	↓ Specificity = TN / (FP + TN) = 238 / (12 + 238) = 238 / 250 = 95.2%	

Table 5.2 The calculated results for sensitivity and specificity.

5.4.2 More calculations (α and β)

Table 5.3 below shows more calculation values for α , β , Likelihood ratio positive and negative values.

Sample=400		
Test	False positive rate (α)=	$FP / (FP + TN) = 12 / (12 + 238) = 1 - \text{specificity}$ $= 8.47\%$
	False negative rate (β) =	$FN / (TP + FN) = 5 / (145 + 5) = 1 - \text{sensitivity}$ $= 3.33\%$
	Power =	Sensitivity = $1 - \beta = 96.67\%$
	Likelihood ratio positive =	Sensitivity / (1 - specificity) = $96.67\% / (1 - 91.53\%) = 11.41$
	Likelihood ratio negative =	(1 - sensitivity) / specificity = $(1 - 96.67\%) / 91.53\% = 0.036$

Table 5.3 α and β calculations.

As can be seen from previous tables, there are few numbers of false positive and large numbers of true positive, so the positive predictive test value (PPV = 92.236%) is in itself excellent for detection the zinc fingers

However, detecting up 96.67% of all zinc fingers (the sensitivity) is an excellent ratio also. Since, a negative result is very good at emphasis that a sample does not have zinc fingers (NPV = 97.9%) this test correctly identifies 91.53% of those who do not have zinc fingers (the specificity).

5.4.3 Time of Detecting the location and number of Zinc Fingers.

Using the neural network in this algorithm was not only accurate but also very fast method. The fast and accurate detection of the location and number of zinc fingers is shown in Table 5.4 below. These values for the performance of the neural network - after running in training, testing and validation process – were taken from the output of the neural network which is shown in Figures 5.13 to 5.15.

Type of Zinc Finger	Number of Iterations In training	Time of Training	Training Performance	Training Gradient
C4	45	0.00:01	0.0127	0.0427
CC-HC	Number of Iterations In Testing	Time of Testing	Testing Performance	Testing Gradient
CC-CH	25	0.00:00	0.147	0.309
	Number of Iterations In Validation	Time of Validation	Validation Performance	Validation Gradient
CH-CC	37	0.00:01	0.44	0.0530

Table 5.4 The performance of the neural network and how fast and accurate was the detection of location and number of zinc fingers.

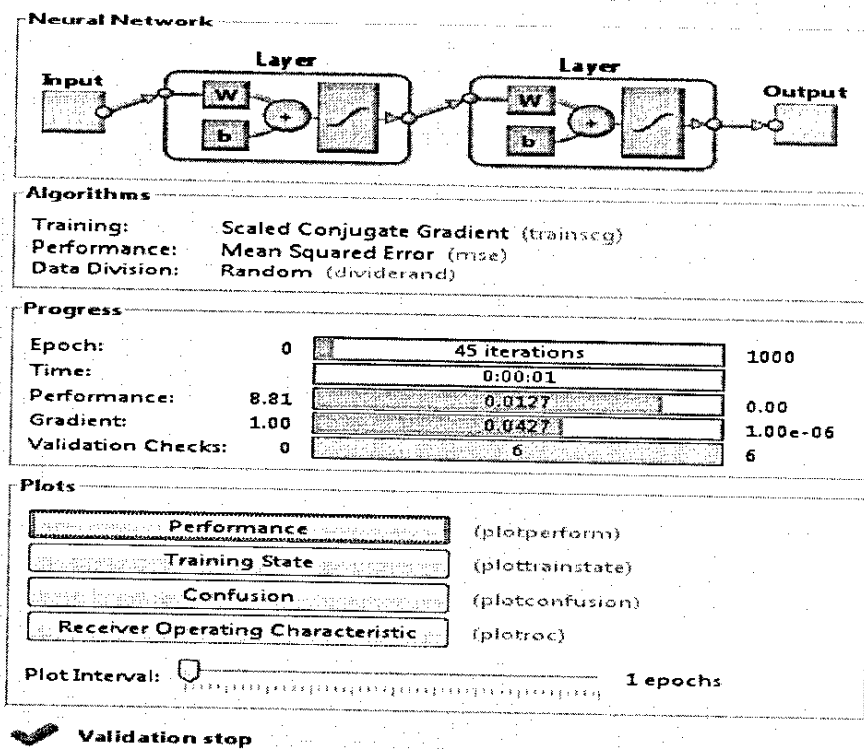


Figure 5.13 The performance of the training of neural network

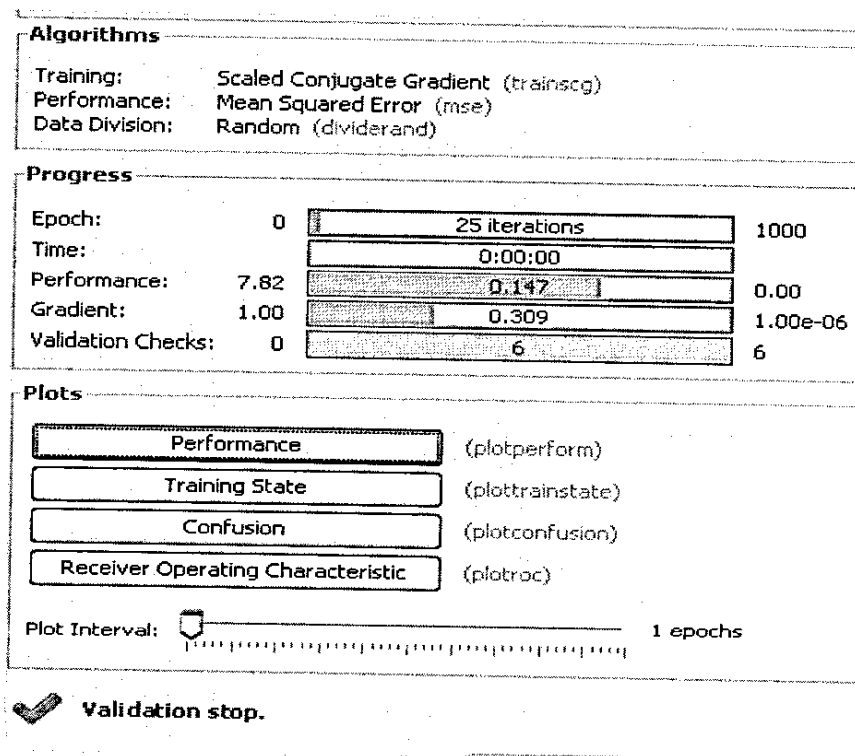


Figure 5.14 The performance of the testing of neural network

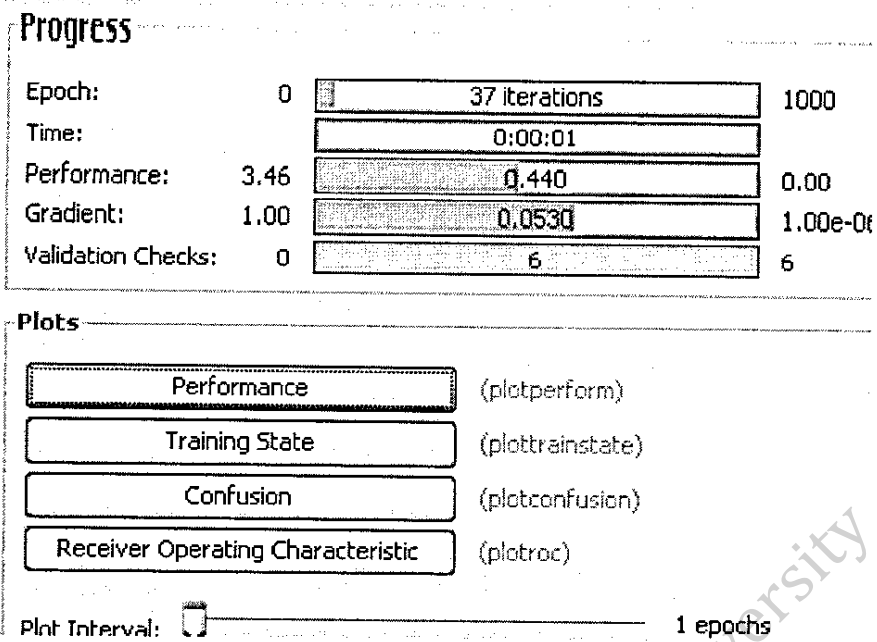


Figure 5.15 The performance of the validation of neural network

Best recognition means more training of the neural network or more input vectors. Figure 5.16 shows that increasing the times of training process, leads to more accurate results. Here, the regression plot of T with the corresponding outputs has a regression value $R=0.9993$.

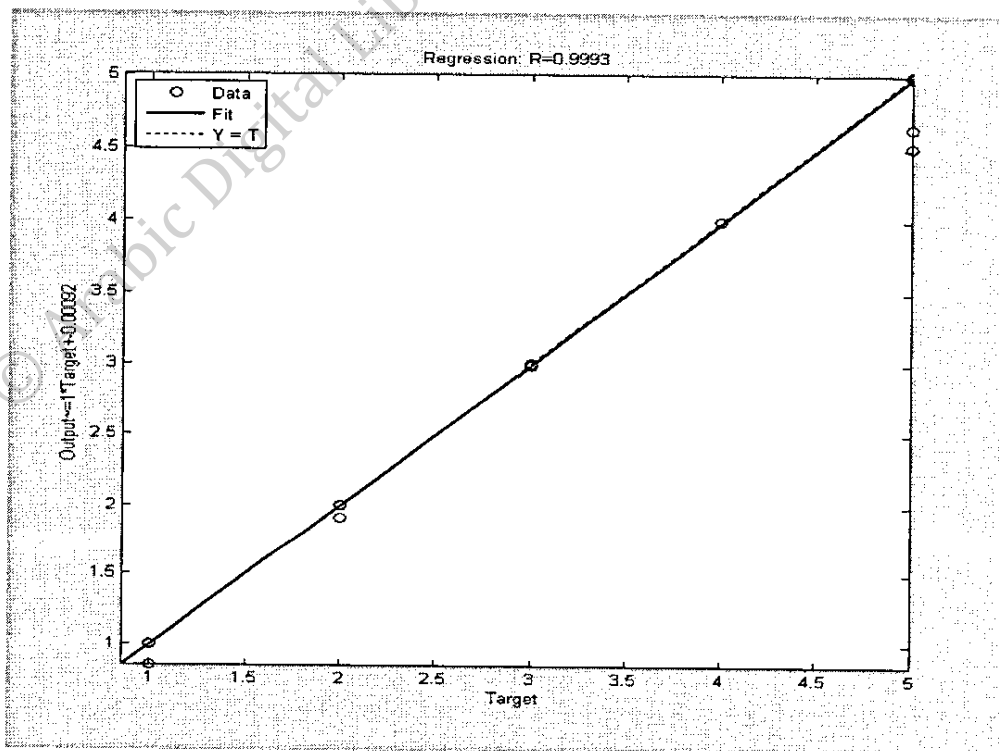


Figure 5.16 The regression plot for network output with T. It shows that more times of training process leads to more accurate results with regression value $R=0.9993$.

5.4.4 Confusion Matrix and Receiver Operating Characteristic (ROC).

A confusion matrix contains information about the actual and predicted classifications used in a classification algorithm. Performance of an algorithm is evaluated using the data in this matrix. The confusion matrix contains values like the correct negative prediction numbers, incorrect positive prediction numbers etc.

The following Figure 5.16 shows the confusion matrix for the 5 classes classifier with an overall percentage average for all tests shown in blue color.

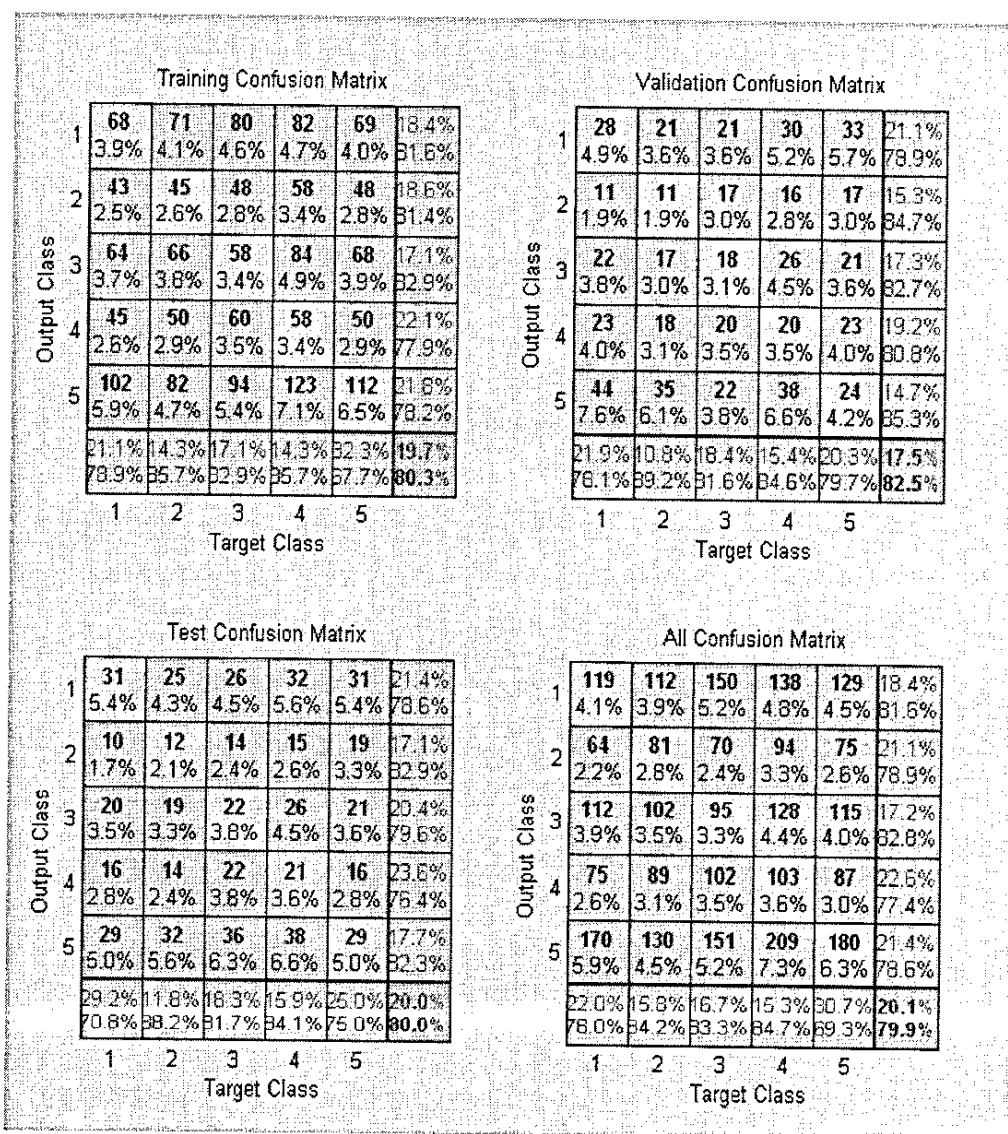


Figure 5.17 The confusion matrix for the 5 classes with an overall percentage average for all tests shown in blue color.

The ROC graph is a plot with the false positive rate on the X-axis and the true positive rate on the Y-axis. ROC graph is used to examine the performance of classifiers. The more concaving up of this curve the more perfect classifier. It means that it has classified more positive cases and negative cases correctly. Also the false positive rate is minimum and the true positive rate maximum. Figure 5.17 below shows the ROC curve for a small test sample; meanwhile Figure 5.18 shows how good increasing the number of the sample.

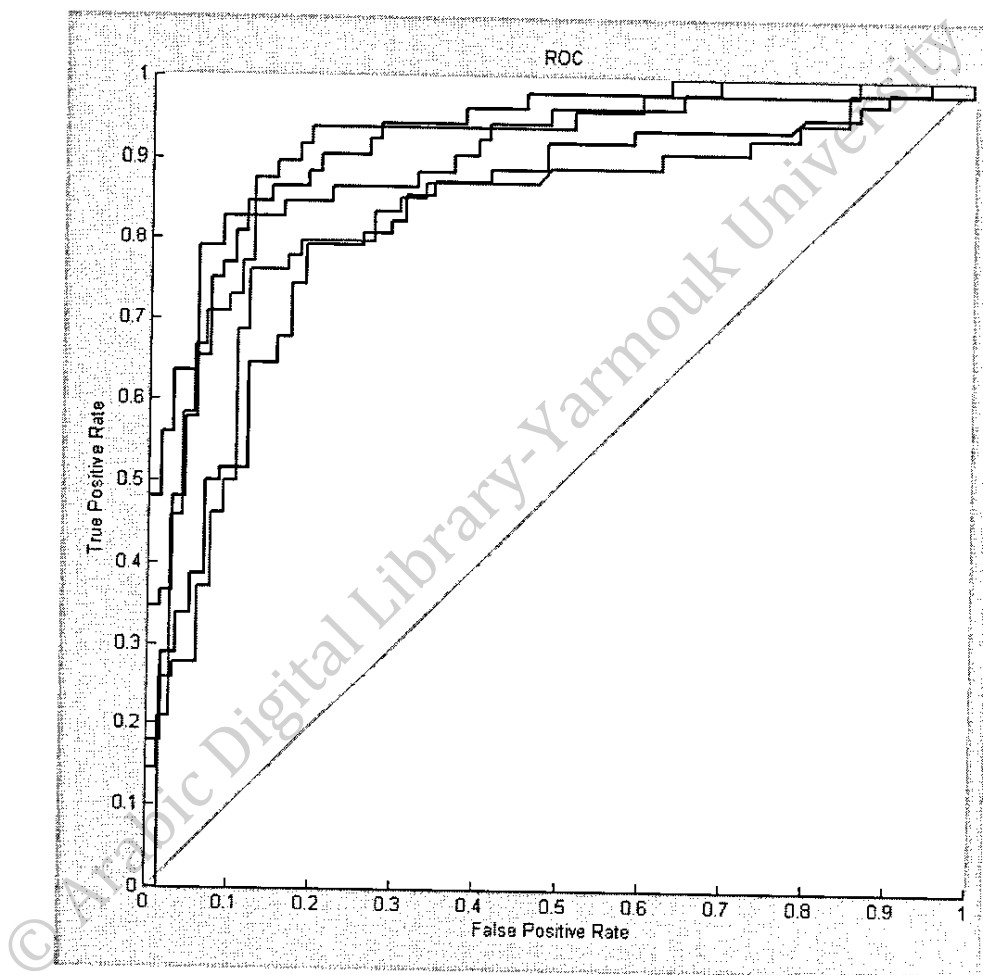


Figure 5.18 The ROC curve for a small test sample. It shows low sensitivity.

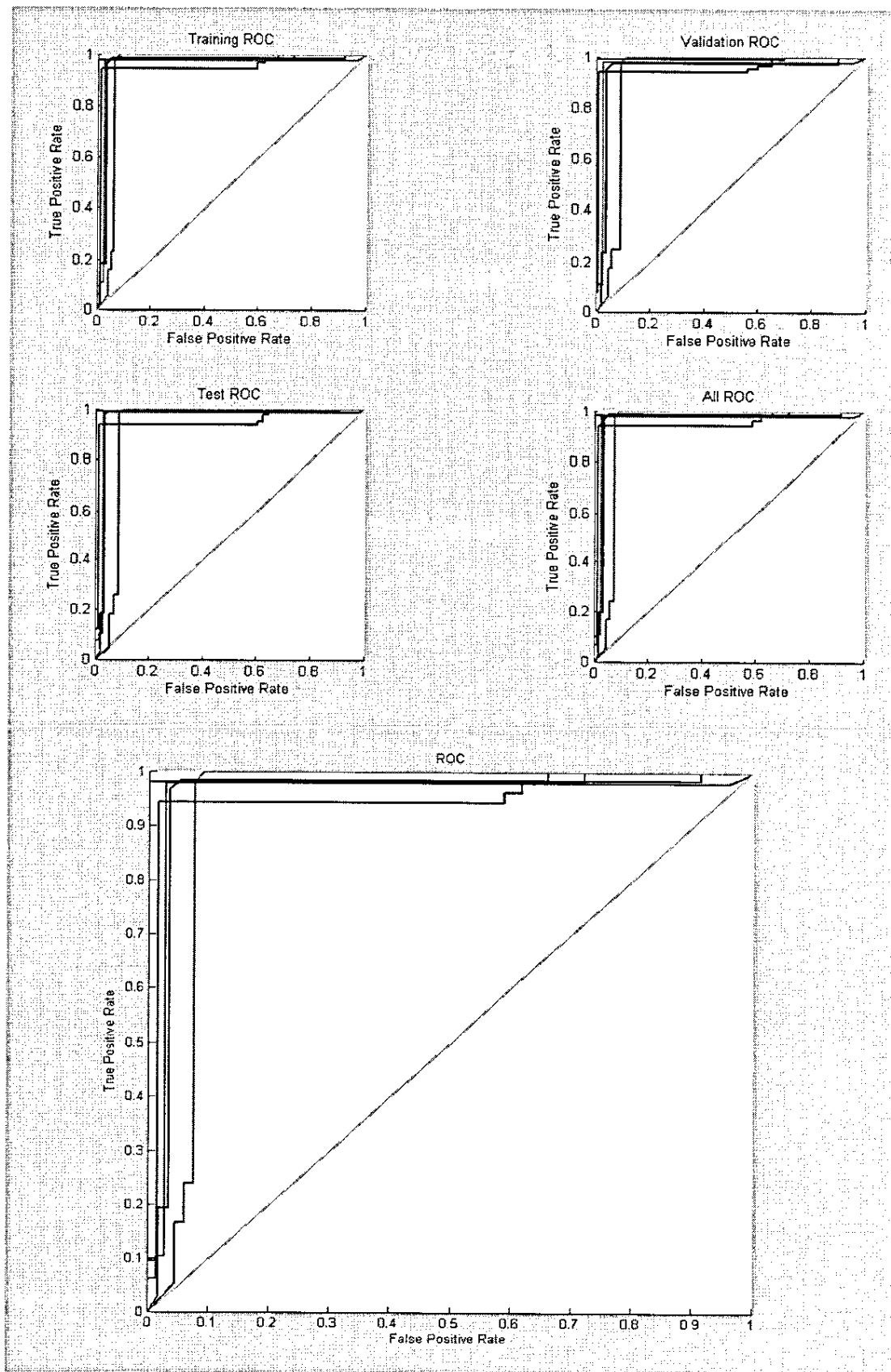


Figure 5.19 Increasing the number of the sample is more perfect classifier. It shows high sensitivity.

5.5 Thesis Outcomes

- ✓ The result outcomes of this study was, detecting the intended zinc finger protein in the protein sequences and its class type, with fast and accurate method.
- ✓ As a second outcome, this study has detected the location of the zinc finger in all of the specific proteins.
- ✓ Thirdly, this work is detecting the number of occurrences of the zinc finger in these sequences.
- ✓ Finally, hopefully coming with a new method that will have not only fast and accurate but also gives what is called sufficient information which is needed for the zinc finger designer that he may come out with a new theorem.

منهجية جديدة وسريعة لاكتشاف وتحديد مكان سلسلة إصبغ الزنك ألبروتيني

إعداد: عمر علي الحواري

إشراف الدكتور

عوض الزين

المشرف المساعد الدكتور

خالد البطاينة

إن مشكلة تحديد مكان وعدد إصبغ الزنك ألبروتيني في سلسلة البروتينات مهم جدا حيث أن هذا البروتين يقوم بدور مهم في عملية الربط والاستساخ التي تحدث خلال العمليات البيولوجية .

كما تشاهد من هذه الدراسة، أن هناك عددا " هائلا" من البروتينات والتي تزداد في كل وقت. لقد تم في بداية هذه الدراسة طرح موضوع كيفية التعامل من خلال هذه المنهجية بطريقة تحوي هذه السلسلة من البروتينات بتحويلها إلى إشاره ثم معالجة هذه الإشارة ودراسة النتائج المستخلصة منها عن طريق تمييزها و اشتقاقها ثم الرسم الكونتوري والطيف الترددي لها .

وبعد ذلك تم تطبيق طريقة الشبكة العصبية الصناعية على عينه من البروتينات تحتوي أربعة أنواع مختلفة من الزنك ألبروتيني ، عن طريق تجزئة ألعينه المستخدمة واستخراج المصفوفة المميزه منها ثم تغذيتها للشبكة العصبية والتي تضم مراحل التدريب ، التحقق والاختبار مع وجود الزنك ألبروتيني داخل هذه العينة ثم تطبيقها على عينه أخرى لاستخراج النتائج. تم استخلاص وتحديد مكان وجود وعدد إصبغ الزنك ألبروتيني للعينة المستخدمة ضمن زمن قياسي.

References:

- [1] R. J. Broker, "Genetic analysis and principles," McGraw, Hill International, 3rd edition.2009.
- [2] J. Pevsner, " Bioinformatics and Functional Genomics," Wiley-Blackwell , 2nd edition 2009, pp. 64-70.
- [3] J.Michel and C.Notredame, "Bioinformatics For Dummies" Wiley Publishing, Inc, 2nd edition 2007, pp. 35-180.
- [4] P. P. Vaidyanathan, " Genomics and Proteomics: A Signal Processor's Tour," IEEE circuits and systems magazine, 2004.
- [5] S.E. Krishna , I.Majumdar and N.V. Grishin "Survey and Summery of Structural classification of zinc fingers," Oxford university , January 2003.
- [6] R. M. Gordley and J. D. Smith, " Evolution of Programmable Zinc Finger-recompenses with Activity in Human Cells," Journal of molecular and biology,2007.
- [7] Jeffrey G. Mandelland and Carlos F. Barbas, "Zinc Finger Tools: custom DNA-binding domains for transcription factors nucleases" Nucleic Acids Research, IIIW516–W523 2006, Vol. 34.
- [8] Jayakanthan. M, Muthukumaran. J, Chandrasekar. S, Punetha A., Chawla K. and D. Sundar "Zif-Base: a data base of zinc finger proteins and associated resources", from the World Wide Web: <http://web.iitd.ac.in/~sundar/zifbase> current 2010.
- [9] A.R. Nordin, and D. Iman, "A Guided Dynamic Programming Approach for Searching a Set of Similar DNA Sequences," University of Malaysia Terengganu, IEEE 2009.

- [10] B.C.H. Chang and S. K. Halgamuge, "Fuzzy Sequence Pattern Matching in Zinc Finger Domain Proteins," Mechatronics Research Group, Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria, Australia IEEE 2001.
- [11] L. Wentian, P. B.Galva , F. D. Haghghi and I. Grosse, "Applications of recursive segmentation to the analysis of DNA sequences," Journal of molecular and biology, 2002.
- [12] D. Bertrand and O. Gascuel, "Topological Rearrangements and Local Search Method for Tandem Duplication trees," ACM transactions on computational biology and bioinformatics, vol. 2, no. 1, January-march IEEE-2005.
- [13] G. Rambally, "A Visualization Approach to Motif Discovery in DNA Sequences," Department of Computer Science Prairie View A&M University Prairie View, IEEE-2007.
- [14] F. Chin and H. Leung, "DNA Motif Representation with Nucleotide dependency," ACM transactions on computational biology and bioinformatics, vol. 5, no. 1, IEEE 2008.
- [15] L. Xikui and Li Yan, "Some Notes on 2-D Graphical Representation of DNA Sequence," College of Information & Engineering, Shandong University of Science and Technology, Qingdao China . Journal of chemical Information and modling.2008.
- [16] National Center for Biotechnology Information: Basic Local Alignment Search Tool (BLAST), from the World Wide Web: <http://blast.ncbi.nlm.nih.gov/Blast.cgi> .
- [17] Alfred V. Aho and Margaret J. Corasick Bell, "String Matching: An Aid to Bibliographic Search," Laboratories communication ACM of Volume 18.1975

- [18] A. Auda and H. Raafat, "An Automatic Text Reader Using Neural Networks Gasser," Computer science Department, University of Regina, Regina, Saskatchewan, Canada. IEEE 1993.
- [19] P. A. Mitkas, S. P. Sastry and T. W. Plymell, "Text Search with an Acoustic Charge Transport Device," Department of Electrical Engineering Colorado State University. IEEE 1995.
- [20] H.C. Leet and F. Ercalt, "RMESH Algorithms For Parallel String Matching,". IEEE1997.
- [21] H. Fujisawa, "Forty years of research in character and document recognition-an industrial perspective," pattern recognition vol. 41, issue 8. 2008.
- [22] Juan V. Lorenzo-Ginori, Aníbal Rodríguez-Fuentes, Ricardo Grau Ábalo and Robercy Sánchez Rodríguez, "Digital Signal Processing in the Analysis of Genomic Sequences" Current Bioinformatics, 4, pp.28-40 Bentham Science Publishers Ltd, 2009.
- [23] L. Howard Holley and Martin Karplus "Protein secondary structure prediction with a neural network". Proc. Nati. Acad. Sci. USA Vol. 86, pp. 152-156, January 1989
Biophysics
- [24] Steve Fairchild, Ruth Pachter, and Ronald Perrin, "Protein Structure Analysis and Prediction,".The mathematica journal , 1995 Miller Freeman Publications
- [25] Gilbert White and William Seffens, "Using a neural network to back translate amino acid sequences". EJB Electronic Journal of Biotechnology Vol.1 No.3,. Universidad Católica de Valparaíso – Chile. December 15, 1998.

- [26] Sitanshu Sekhar Sahu' and Ganapati Panda, "a new approach for identification of hot spots in proteins using s-transform filtering". Department of Electronics and Communication Engineering National Institute of Technology,Rourkela,India, 2009.
- [27] Dariusz Plewczynski, Lukasz Slabinski, Krzysztof Ginalski and Leszek Rychlewski "Prediction of signal peptides in protein sequences by neural networks" Centre of Mathematical and Computational Modelling, Warsaw University, Acta Biochimica polonica Poland; Vol. 55 No. 2/2008, pp.261–267.
- [28] X. Xiao , J.D.Zhen and W. Ling, "Using Cellular Automata Images to Predict Protein Structural Classes,". IEEE 2007.
- [29] W. Zhong and L. X. Xiao, "Using Grey Model GM(2,1) and Pseudo Amino Acid Composition to Predict Protein Sub cellular Location,". IEEE 2008.
- [30] K. Deergha Rao, and M. N. S. Swamy," and Analysis of Genomics Proteomics Using DSP Techniques". Transactions on circuits and systems ,Vol. 55, NO. 1, IEEE 2008.
- [31] Molecular Biology Web Books, from the World Wide Web:
<http://www.webbooks.com/MoBio/Free/Ch4F2.htm> current 2010.
- [32] V. Buzenac, R. Settineri, M. Najim, and J. Paty. " EOG segmentation using fast algorithms". Engineering in Medicine and Biology Society.IEEE-EMBS,2005.
- [33] Pan J. and W. Tompkins." A real time QRS detection algorithm". Trans Biomed Eng BME-32,230-236.IEEE 1985.
- [34] Willis J. Tompkins." Biomedical digital signal processing", Prentice Hall USA.March 2nd 1993

- [35] Juan V. Lorenzo-Ginori , Aníbal Rodríguez-Fuentes, Ricardo Grau Ábalo and Robersy Sánchez Rodríguez, "Digital Signal Processing in the Analysis of Genomic Sequences". Current Bioinformatics, 2009, 4, 28-40
- [36] Matin Akay." Time frequency and wavelets in biomedical signal processing".IEEE Press Series in biomedical engineering .1996 pp.3-60
- [37] Ivan Selesnick, "The short time Fourier transform", from the World Wide Web: <http://cnx.org/content/m10570/latest/> current 2010.
- [38] Hiam H. AL-Quran, " Electro-Oculogram Pattern Recognition and Gaze Angle Prediction". M.sc.Thesis Yarmouk University./2008.
- [39] R. Nelson, S.Foo and M. Weatherspoon, "Using Hidden Markov Modeling in DNA Sequencing,". IEEE2008.
- [40] S.N. Sivanadam and M. Paulraj," Introduction to artificial neural networks".Vikas publishing house PVT LTD, New Delhi / 2003 pp.1-23
- [41] John A. Swets, "Signal detection theory and ROC analysis in psychology and diagnostics : collected papers" BBN Corporation and Haward Medical School, Lawrance Elrabaum Associates, Puplishers. Mahawa, New Jersey, 1996
- [42] Bioinformatics Toolbox, from the World Wide Web: <http://www.mathworks.com/products/bioinfo/> 1994-2010 The MathWorks, Inc.
- [43] The receiver operating characteristics, from the World Wide Web: http://en.wikipedia.org/wiki/receiver_operating_characteristics