DNA BASE-CALLING

A THESIS IN ELECTRICAL ENGINEERING
Master of Science in Electrical Engineering

Presented to the faculty of the American University of Sharjah
College of Engineering
in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by

OMNIYAH GUL MOHAMMED

B.S. 2008

Sharjah, UAE
May 2010

© 2010

OMNIYAH GUL MOHAMMED

ALL RIGHTS RESERVED

We approve the thesis of Omniyah Gul Mohammed — Date of signature

_____ _____

Dr. Khaled Assaleh
Associate Professor, Dept. of Electrical Engineering
Thesis Advisor

_____ _____

Dr. Ghaleb A. Husseini
Associate Professor, Dept. of Chemical Engineering
Thesis Advisor

_____ _____

Dr. Amin Majdalawieh
Assistant Professor, Dept. of Biology and Chemistry
Thesis Advisor

_____ _____

Dr. Hassan Al-Nashash
Professor, Dept. of Electrical Engineering
Graduate Commnittee

_____ _____

Dr. Abdul Salam Jarrah
Assistant Professor, Dept. of Mathematics & Statistics
Graduate Commnittee

_____ _____

Dr. Mohamed El-Tarhuni
Associate Professor and Head of Dept.
Dept. of Electrical Engineering
Coordinator, Electrical Engineering Graduate Program

_____ _____

Dr. Hany El-Kadi
Associate Dean, College of Engineering

_____ _____

Mr. Kevin Lewis Mitchell
Director, Graduate & Undergraduate Programs

DNA BASE-CALLING

Omniyah Gul Mohammed, Candidate for the Master of Science Degree

American University of Sharjah, 2010

ABSTRACT

The human genome sequence, consisting of approximately three billion bases, was decoded in the last decade as a result of the Human Genome Project. Information derived from the genomic sequences of different species is expected to contribute massively to advances in various fields, such as medicine, forensics and agriculture. The ability to decipher the genetic material is of huge importance to researchers trying to improve the diagnosis of genetic diseases, improve drug design to target specific genes, detect bacteria that may pollute air or water, explore species ancestry, etc. The impact of DNA sequencing in various fields has created a need to efficiently automate the mapping of signals obtained from sequencing machines to their corresponding sequence of bases, a process referred to as *DNA base-calling*.

This thesis attempts to solve the problem of base-calling by using *pattern recognition*, the act of classifying raw data based on prior or statistical information extracted from the data into various classes. In this thesis, two new frameworks are proposed using Artificial Neural Networks (ANN) and Polynomial Classifiers (PC) to model electropherogram traces. Data is obtained from the Sorenson Molecular Genealogy Foundation (SMGF) and the National Center for Biotechnology Information (NCBI) trace archive. Pre-processing, which includes de-correlation, de-convolution and normalization, needs to be implemented to minimize or eliminate data imperfections that are primarily attributed to the nature of chemical reactions

involved in DNA sequencing. Discriminative features that characterize chromatogram traces are subsequently extracted and subjected to the classifiers to categorize the events to their respective classes: A, C, T or G. The models are trained such that they are not restricted to the type of organism the chromatogram belongs to or to the chemistry involved in obtaining the chromatogram.

The base-calling accuracy achieved is compared with the existing standards, PHRED and ABI KB base-caller in terms of deletion, insertion and substitution errors. Experimental evidence indicates that the models implemented achieve a higher base-calling accuracy when compared to PHRED and a comparable performance when compared to ABI. The results obtained demonstrate the potential of the proposed models for efficient and accurate DNA base-calling.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF ABBREVIATIONS

A	-	Adenine
ABI	-	Applied Biosystems Incorporated
ANN	-	Artificial Neural Network
BLAST	-	Basic Local Alignment Search Tool
C	-	Cytosine
DNA	-	Deoxyribonucleic Acid
G	-	Guanine
GRC	-	Genome Reference Consortium
NCBI	-	National Center for Biotechnology Information
PC	-	Polynomial Classifier
PCR	-	Polymerase Chain Reaction
PHRED	-	Phil's Read Editor
SMGF	-	Sorenson Molecular Genealogy Foundation
T	-	Thymine

ACKNOWLEDGEMENTS

Praise be to Allah, the most Gracious and Merciful, for providing me the strength and patience to complete this thesis successfully. I would like to express my sincere gratitude and appreciation to several people who have been instrumental, either directly or indirectly, to make this work possible.

Through these lines and humble words, I would like to thank my advisors Dr. Khaled Assaleh, Dr. Ghaleb A. Husseini and Dr. Amin Majdalawieh who generously gave me a lot of their time, experience, wisdom, knowledge and more importantly advice.

It is difficult to overstate my gratitude to my mentor, Dr. Khaled Assaleh, who introduced me to the area of pattern recognition and taught me the ABC's of the field. Throughout these two years, he provided encouragement, sound advice, valuable teaching and lots of great ideas. He taught me how to push myself to the maximum and was always ready to discuss any of my absurd ideas. I would have been lost without him.

Special thanks go to my co-advisor, Dr. Ghaleb A. Husseini, for helping me in the challenging biological research that lies behind this thesis. His persistence in finding data was the reason for making this accomplishment a possibility. Dr. Ghaleb's enthusiastic and joyful face always helped in reducing the tension in the air in those weekly meetings following one of my shocking research updates. He taught me how to be a better writer and had confidence in me when I doubted myself. He was always there to meet and talk about everything and anything and to proof read my papers.

Furthermore, I would like to thank my second co-advisor, Dr. Amin Majdalawieh, for accepting our late invitation to help in this research. His biological

point of view and insightful comments proved to be a golden addition to this thesis. His support, dedication and tendency to ask good questions to help me think through my problems were an invaluable asset.

Besides my advisors, I would like to thank Dr. Scott Woodwards, from Sorenson Molecular Genealogy Foundation (Salt Lake City, Utah, USA), for providing the experimental data sets that were used in this thesis. Special thanks to my thesis examiners, Dr. Hassan Al-Nashash and Dr. Abdul Salam Jarrah for their time, effort and valuable comments. Of course, nothing would be possible if I did not have great teachers in the college of Electrical Engineering who put a lot of effort to make these years memorable. I thank them for that.

Surely, I could have never seen the completion of this thesis without the constant support of my friends. Special thanks to my best friend and room-mate, Yara Fayyad, who had played a pivotal role in supporting me from the beginning. Her cheerfulness in those long nights in the lab, her provocations throughout the day and her ability to get me out of depression with chocolate made these years memorable and unforgettable. I would like to thank also Yasmin Adel for her friendship and support. Her ganging up with Yara against me used to always succeed in lowering down our stress.

Last but not the least, no words can express my indebtedness to my family without whom none of this would have even been possible. To my brother, Mohammed Ishaque, for urging me to join the master's program from the beginning. To my sister, Badria Mohammed, for reminding me that this research is just a *thesis* and that the entire world is not revolving around it. To mom, thanks for always being there, for being so understanding, so patient with my negligence and so loving. Lastly and most importantly, to my father, who devoted his entire life to making this a possibility. Without you dad, I would not be what I am today. This is dedicated to you, dear father, from your forever grateful daughter.

To my family, two small words for all your love and your support which a million words would be too short to express:

Thank you

CHAPTER 1

INTRODUCTION

Until the last decade, the entire human genome sequence was not known. A massive research over the last few years resulted in deciphering the three billion constituents of the human genome. It is believed that significant future scientific achievements will be related to the analysis of the vast amount of information that is enclosed in the human genome, and genomics of other organisms. The impact of DNA sequencing in diverse areas, such as medicine and agriculture, has created a need to efficiently automate the mapping of sequencing signals, referred to as *electropherograms*, to their corresponding sequence of bases, a process known as *base-calling*. Figure 1.1 illustrates a portion of an electropherogram consisting of four sequencing signals corresponding to each of the four bases contained in a DNA signal: Adenine (A), Cytosine (C), Thymine (T) and Guanine (G). Base-calling involves translating Figure 1.1 to a string of A, T, C and G sequence - GTCTTTTGTTGTCTACCAACAA.

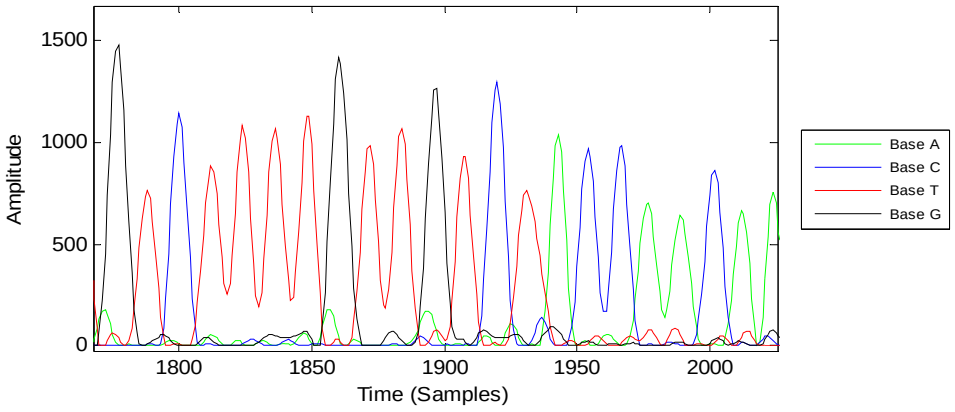


Figure 1.1. A sample electropherogram.

Prior to explaining the method adopted to solve the problem of base-calling, it is important to understand the structure and functionality of a DNA molecule. Section 1.1 of this chapter describes the shape, constituents and significance of a DNA molecule. A brief overview of DNA sequencing is then provided in section 1.2. The motivation and the importance of base-calling is presented in the following section. Section 1.4 proceeds to discuss the objectives of this research and highlights the contributions achieved in this thesis. Finally, the organization of the thesis is outlined in section 1.5.

1.1 What is DNA?

The very basic unit of every living organism's genome is a single DNA molecule which contains all the information needed for building and maintaining its existence. The DNA is found in each cell in the form of paired chromosomes. A typical human cell contains 23 pairs of chromosomes and each chromosome consists of a large number of genes which represents the basic unit of heredity.

DNA stands for *Deoxyribonucleic Acid*. It is mostly located in the cell nucleus, hence referred to as the nuclear DNA. A sequence made up of four chemical bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T) represent the coded information stored inside a DNA molecule (Figure 1.2). *Homo sapiens* genomic sequence code consists of about three billion bases that is unique to each individual. By learning how to decode this sequence, researchers hope to learn more about how to treat of genetic diseases, trace ancestry and solve crimes.

A DNA molecule has a double helical structure consisting of two intertwined chains made up of complementary nucleotide strands. Each nucleotide consists of a phosphate group, a deoxyribose sugar molecule and one of the four nitrogenous bases: A, G, C or T. Consecutive nucleotide chains are held together by bonds between the sugar molecule of one nucleotide and the phosphate group of the successive nucleotide. The repeated nucleotide units, hence, form the *backbone* of the DNA molecule. The two intertwined chains are then held together by weak hydrogen bonds formed between bases on the complementary nucleotides chains, such that A bonds with T while C pairs up with G, stabilizing the DNA molecule. The ends of the DNA

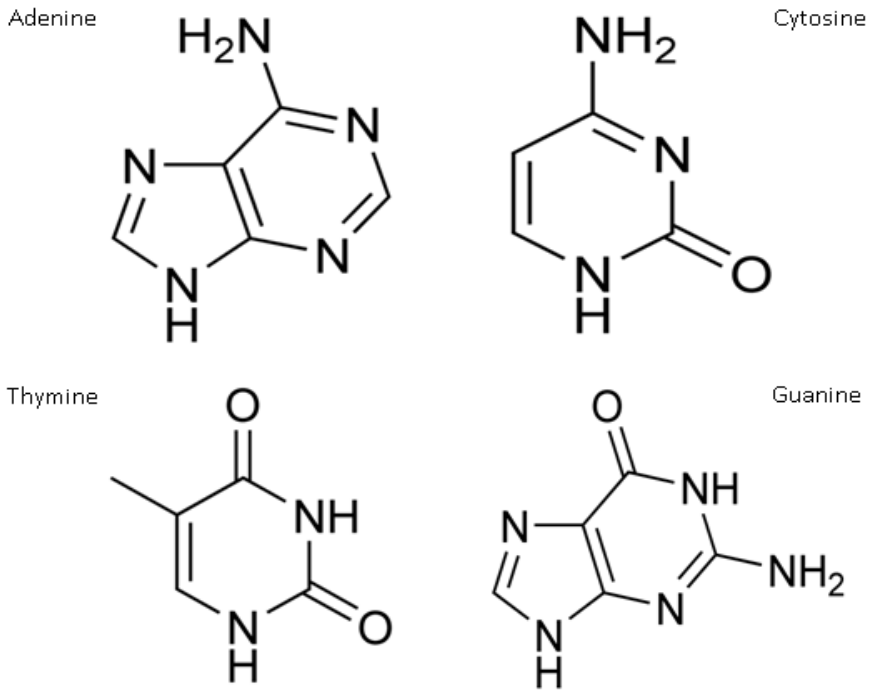


Figure 1.2. DNA's Four Bases: Adenine, Guanine, Cytosine and Thymine [1].

strands are known as the Five Prime (5') and the Three Prime (3') ends. The 5' end refers to the end of the strand which terminates with a phosphate group while the 3' end terminates the strand with a hydroxyl group. It should be noted that all the information held by one DNA strand is duplicated in the second strand since the two DNA helical strands are complementary to each other. Figure 1.3 illustrates the double-helical structure of a DNA molecule and the formation of base pairs between the complementary nucleotide strands. Notice that three hydrogen bonds are needed to form a GC pair, while A and T require only two hydrogen bonds to form a base-pair. Hence, DNA molecules having a larger number of GC base-pairs than AT pairs are more stable due to the stronger interactions between the DNA's complementary strands. Thus, to break the hydrogen bonds and obtain single DNA strands, a DNA molecule consisting of a larger number of GC pairs should be exposed to very high temperatures than a molecule with lower amount of GC pairs. Figure 1.4 shows the formation of base pairs between A and T and between C and G [2].

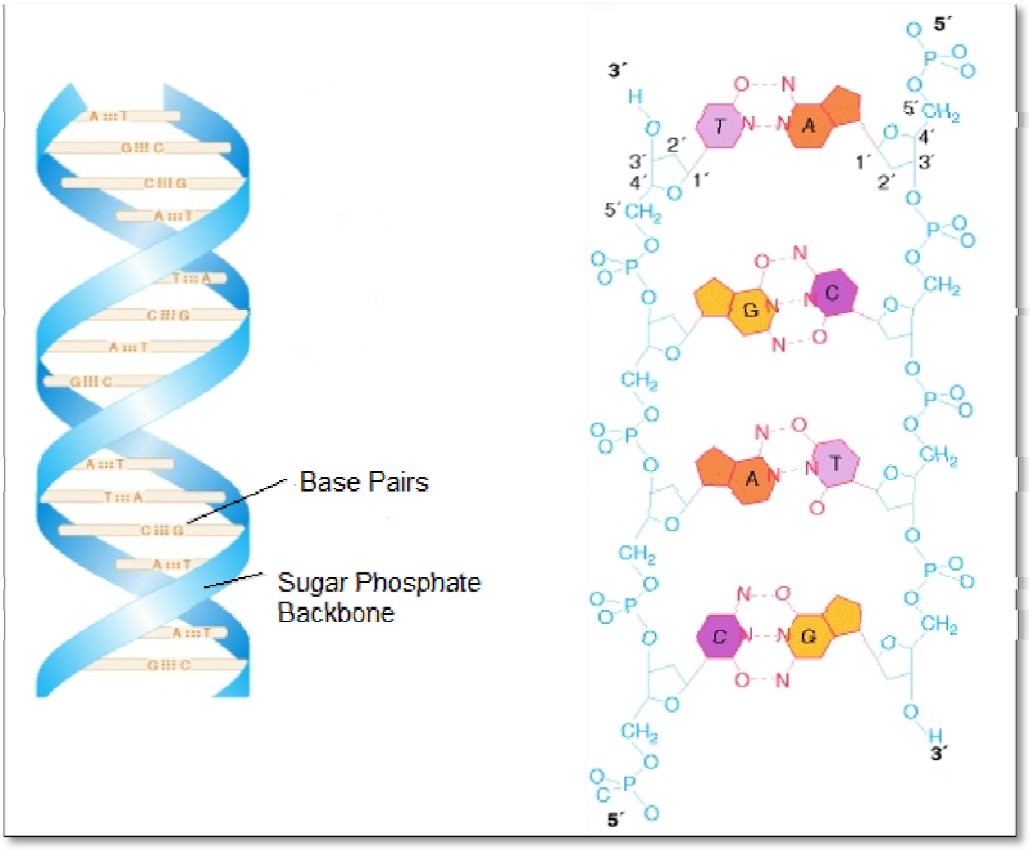


Figure 1.3. Double Helical Structure of a DNA molecule [3].

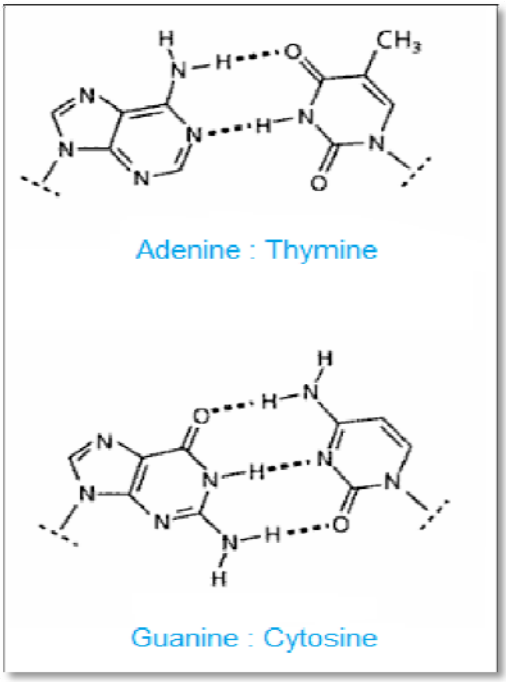


Figure 1.4. Base Pairing between G and C, A and T [4].

1.2 DNA Sequencing

DNA sequencing is the process of determining the ordered sequence of nucleotide bases in a DNA molecule. To determine the function of specific genes in a chromosome, the sequence of nucleotides that forms a gene is decoded using DNA sequencing. Knowledge of DNA sequences can be used for biological research and in applied fields such as DNA forensics and evolutionary studies.

DNA sequencing went through several phases of development prior to the advancement of rapid sequencing methods. In 1976, Maxam and Gilbert developed a sequencing method based on chemical modification of the DNA molecule which breaks a terminally labeled DNA template partially at each base. The reaction of Dimethyl Sulphate, piperidine, formic acid, hydrazine and sodium chloride individually or in combinations causes the cleavage of the four bases. The lengths of the labeled fragments then identify the positions of each base. Maxam and Gilbert's technique allowed the sequencing of at least 100 bases from the point of labeling [5].

The most widely used technique for DNA sequencing is the Chain Termination method, also known as the Sanger method. The underlying principle behind the Sanger method is the separation of DNA molecules, differing in length by only a single nucleotide, into distinct bands by electrophoresis. The sequencing operation consists of the following steps:

1. A DNA strand is decomposed into smaller fragments using restriction enzymes.
2. The DNA fragments are then amplified using the *Polymerase Chain Reaction* (PCR) technique generating many copies of the DNA sequence. PCR consists of three main temperature changes (Figure 1.5):
    a) *Denaturation Step:* To break the hydrogen bonds between the base-pairs in the complementary DNA strands, the DNA template is heated to a temperature above 90 ºC resulting in two single DNA strands.
    b) *Annealing Step:* Large amount of primers are then added to the DNA single strands and the whole setup is allowed to cool to 50-60 ºC so that the DNA strands form hydrogen bonds with the primers due to their avail-

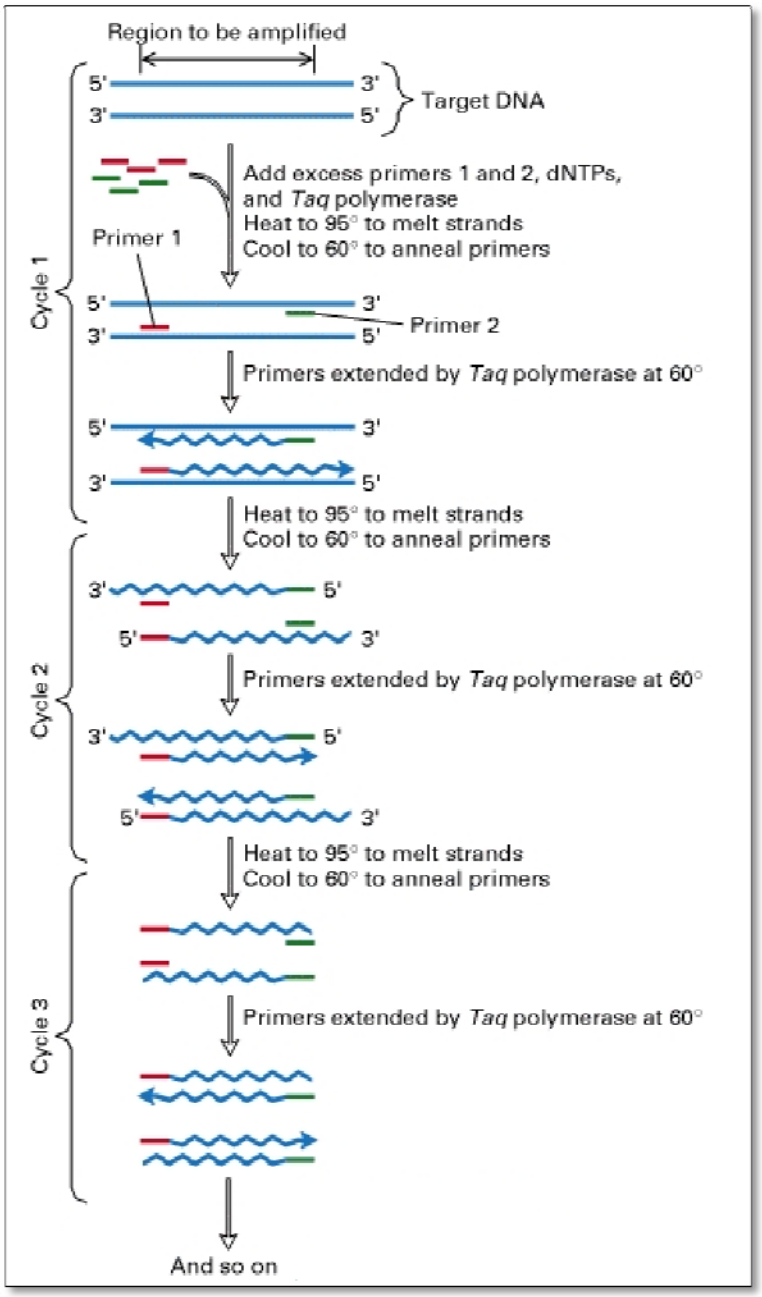


Figure 1.5: The different steps in PCR [6].

ability in excess.

c) *Primer Extension:* The setup is then heated to a temperature between 60-70 ºC and the DNA polymerase is added which binds to the primers and starts adding complementary bases to form a growing chain (i.e. on reading a G on the template strand, the polymerase adds a C and so on). Elongation of the DNA template, hence, occurs in the direction that the

primer faces and results in the production of a new double-stranded PCR product [7].

3. The single strand DNA template to be sequenced is divided into four separate sequencing reactions, each containing a primer attached to one end of the single strand DNA molecule. The primer serves as a starting point for DNA replication. To each reaction, all four of the standard deoxynucleotides (dATP, dGTP, dCTP, and dTTP) and a DNA polymerase are added. The DNA polymerase is capable of adding free nucleotides to the 3' end of the newly forming DNA strand causing elongation of the new strand in the 5'-3' direction. To each sequencing reaction, only one of the four di-deoxynucleotides (ddATP, ddGTP, ddCTP, and ddTTP) are added to be used as chain terminating nucleotides since they lack a hydroxyl group needed for bonding with the successive nucleotide. Hence, the di-deoxynucelotides terminates DNA strand elongation and results in various DNA fragments of different lengths [8].

4. Hence, the amplified DNA synthesizes DNA chains having the same start point but of varying lengths. Each of these synthesized DNA chains are terminated with one of four different radioactive fluorescent dyes (or marked with the fluorescent dye at the primer end) based on the terminating base in the chain.

5. *Electrophoresis* is then performed on the synthesized DNA fragments in which an electric field is used to separate DNA samples of different lengths. Electrophoresis is of two types:

    a) *Gel Electrophoresis:* DNA fragments are separated by size as they move through a gel matrix. In this technique, a slab of material called agarose is placed in a conducting buffer solution and is connected to positive and negative electrodes. The phosphate groups contained in the DNA molecule carry negatively charged oxygen resulting in an overall negatively charged DNA molecule. Hence, by turning on the power supply, the DNA molecules move at different rates, based on their radius, to the positive end (anode). Since the DNA stands differ by only one nucleotide in length and each strand is terminated using labeled fluorescent

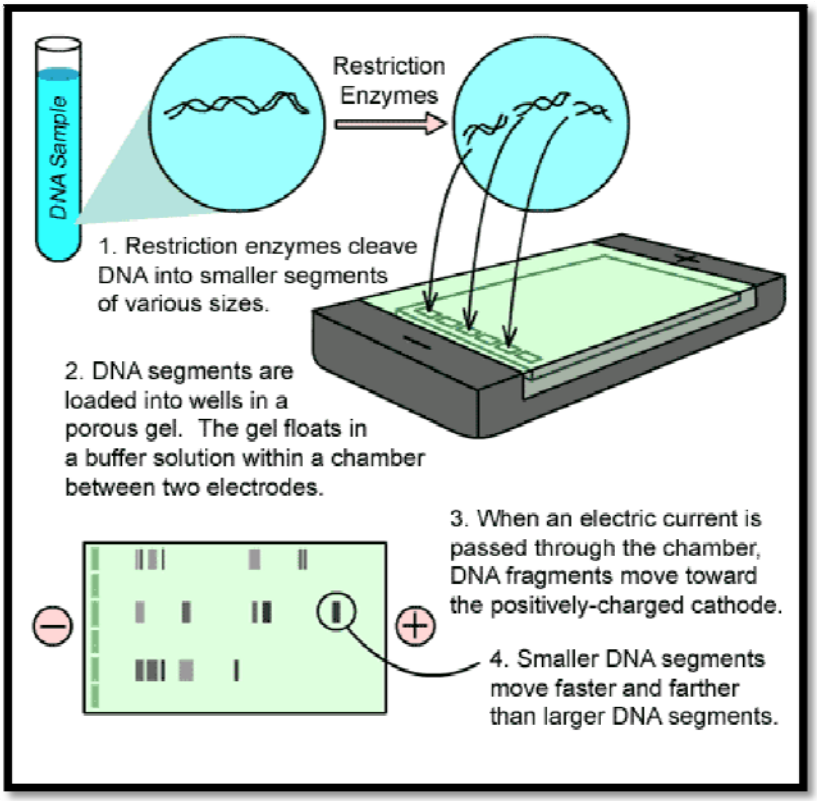


Figure 1.6: Gel Electrophoresis [9].

dyes each synthesized DNA strand reaches the anode at a different time. Thus, using the terminal fluorescent nucleotide, the DNA sequence can be recognized (Figure 1.6).

(b) *Capillary Electrophoresis*: It consists of a sample vial, a source vial, a destination vial, a high voltage power supply, an ultraviolet detector, electrodes and a capillary tube with an optical viewing window (Figure 1.7). Each of the source vial, destination vial and the capillary are filled with an aqueous buffer solution and the optical viewing window is aligned with the ultraviolet detector. Electrophoresis starts by placing one end of the capillary in the sample vial, which is filled with the DNA fragments to be sequenced, and then in the source vial. By applying a high voltage power supply to the cathode and the anode electrodes, an electric field is created which results in an electromotive force causing the analytes to migrate from the source to the destination at different rates. The analytes, hence, get separated and reach the destination vial at different times allowing for their detection. The detector system consists of a laser and a

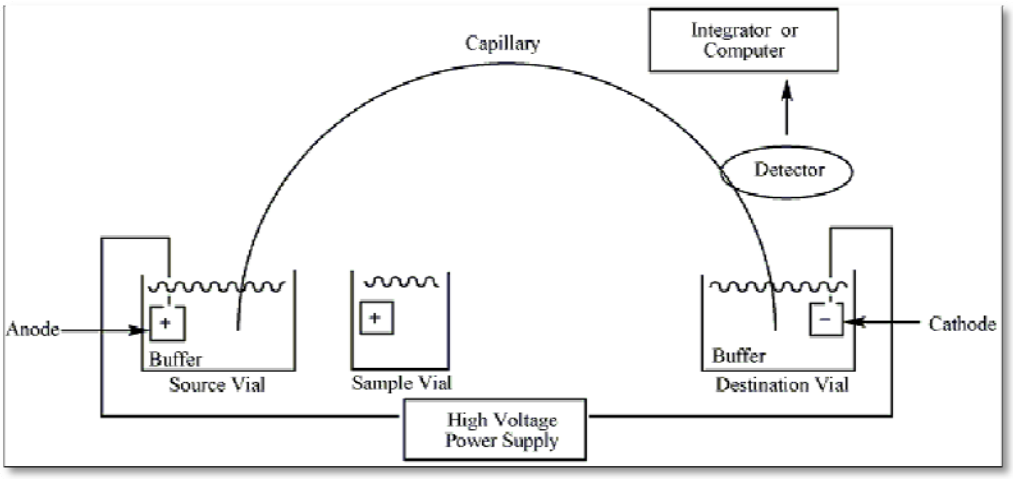


Figure 1.7: Capillary Electrophoresis System [10].

Either capillary electrophoresis or gel electrophoresis is performed on the synthesized DNA strands. The electrophoretic data output produced by the chain termination DNA sequencing method is then analyzed and DNA base-calling is performed. *DNA base-calling* is the process of identifying the ordered sequence of nucleotides by analyzing the electropherograms (Figure 1.1) obtained from electrophoresis. Figure 1.8 shows the main steps involved in the Sanger Method for DNA sequencing followed by gel electrophoresis in which the fragments are separated by size.

DNA sequencing via the chain termination method is more efficient than Maxam and Gilbert's method. Lower amounts of radioactive and toxic chemicals are needed in the chain termination method making it more appealing. However, it can only sequence short DNA fragments (300-1000 bases) in a single reaction. Hence, large scale sequencing methods were introduced which aim at sequencing long DNA fragments, such as whole chromosomes. Large DNA fragments are typically decomposed using restriction enzymes and the fragments are then cloned into a DNA vector inside the DNA of a virus or any other living organism. The organism is then

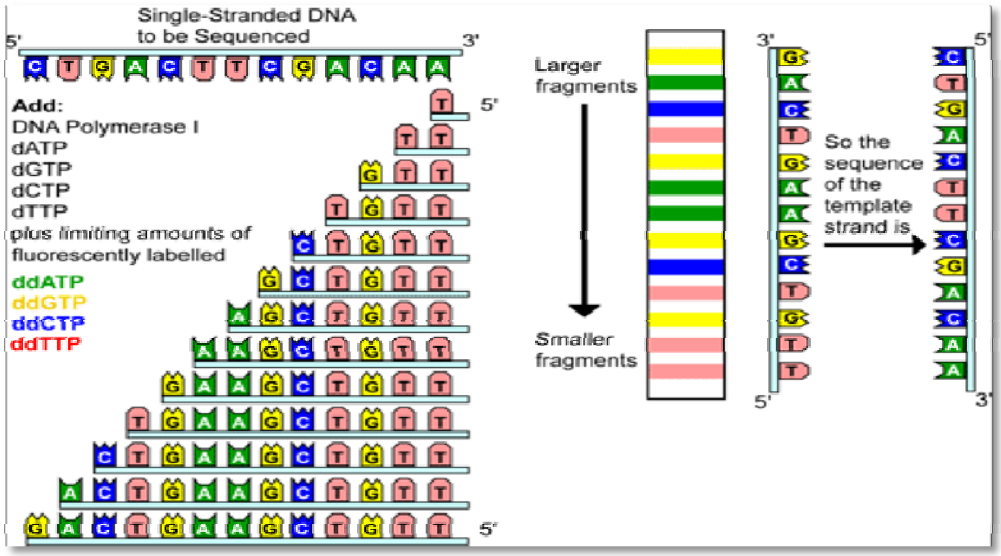


Figure 1.8: Chain Termination Main Steps for DNA Sequencing [11].

allowed to replicate itself, hence replicating the inserted DNA to produce large amounts of the DNA fragment. Short DNA fragments are then individually sequenced and assembled into one long sequence.

1.3 Motivation

The basic blueprint for a living organism is its DNA sequence. Information derived from the genomic sequence is likely to contribute enormously to medical advances such as more accurate diagnosis of genetic diseases, improved drug design to target specific genes causing certain diseases and gene therapy by replacement of defective genes. The ability to decode the genetic material is very important to researchers trying to improve the resistance of crops to parasites, detect bacteria that may pollute air or water, determine pedigree for seed or livestock breeds, explore species origin and ancestry, determine the cause of migration of different populations and various other evolutionary studies. DNA sequencing also has potential benefits in applied fields such as DNA forensics in which crime suspects can be identified by matching their DNA with evidence left at crime scenes, establish paternity, and identify crime and catastrophe victims.

1.4 Thesis Objectives and Contributions

Designing a highly efficient system for analyzing chromatogram traces obtained from the electrophoresis process would have widespread benefits both in biological researches and in applied fields such as DNA forensics. In this thesis, the last part of the whole DNA sequencing process will be focused on: *base-calling*. It is the process of translating electropherograms obtained from sequencing machines to an ordered sequence of letters representing the bases that compose the processed DNA samples. Currently available base-calling software suffer from at least one of the following limitations:

1. Ability of a base-caller to process chromatogram traces obtained from different sequencing technologies.
2. Ability to process efficiently long read lengths.
3. Minimum computational cost and time in base-calling.
4. Lack of theory supporting the principles upon which the base-calling software is built.
5. Fully automated and high performance software.

The main goal of this thesis is to design a robust and automated base-calling system that:

1. Adopts a pattern recognition approach to solve the problem.
2. Is not restricted to specific sequencing machines or chemistry.
3. Is not affected significantly in its performance by the noise inherent in the electropherograms.

In this thesis we developed a novel base caller utilizing two classifier models: Artificial Neural Network (ANNs) and Polynomial Classifiers (PCs) to achieve the previously mentioned objectives. We used data obtained from the Sorenson Molecular Genealogy Foundation (SMGF, Salt Lake City, Utah, USA) and from the National Center for Biotechnology Information (NCBI, Bethesda, Maryland, USA) trace archive to train, validate and test our pattern recognition models. The system was designed such that it is not restricted to one specific chemistry or sequencing

technology. The raw data was then subjected to the following series of operations before the DNA sequence is obtained:

1. Pre-processing: The pre-processing stage is designed to handle raw data produced by different sequencing technologies. It includes color correction by de-correlation, peak sharpening by de-convolution and resolve dynamic decay by windowed normalization.
2. Feature Extraction: Discriminative signal characteristics are identified to represent the patterns in the electropherograms. Features extracted are the positive gradient, amplitude and negative gradient of each sample point in the four base-signals.
3. Classification: Artificial Neural Networks (ANNs) and Polynomial Classifiers (PCs) are implemented as base-classifiers. Using the results obtained on subjecting the features extracted from the pre-processed data to each of the classifiers, base-call decisions are made and the overall performance of the classifier is determined.

The overall performance of the ANN and PC base-calling software proved to be better than that of the widely used PHRED base-caller and comparable to that of the ABI KB base-caller.

1.5 Thesis Outline

This thesis consists of six chapters which are organized as follows:

Chapter 2: Literature Review

Before the contributions of this work can be presented, a thorough literature review of DNA base identification algorithms is required. Chapter 2 describes the research carried out in this area in terms of pattern recognition models and overall efficiencies achieved. ABI and PHRED base-calling software are also introduced as they represent the most widely used DNA base-calling software both commercially and academically.

Chapter 3: Pattern Recognition

In this chapter, the general pattern recognition framework is explained to help the reader grasp the essence of the fundamentals upon which the thesis is built. The basic operations of the two classifiers implemented in this research are also described.

Chapter 4: Data Acquisition, Pre-processing and Feature Extraction

The source, properties and categorization of the data used for training and testing the classifier models are discussed in this chapter. Pre-processing the electropherograms conditions the signals for base-calling and involves color correction, peak sharpening and normalization. The chapter is then concluded by a discussion of the invariant features extracted from the traces to represent the inherent patterns.

Chapter 5: DNA Base-calling as a Pattern Recognition Problem

The DNA base-calling problem is tackled in this chapter from the pattern recognition point of view. The ANNs and PCs designed topologies are presented and the post-processing of the scores obtained on subjecting the classifiers to novel data is explained. The results acquired in terms of deletion, insertion and substitution errors are illustrated and analyzed.

Chapter 6: Conclusions and Future Work

In the last chapter of the thesis, a summary of the research findings is presented and future research work is suggested.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the related work in the area of base-calling is discussed. It is divided into two parts. Section 2.1 summarizes some of the methods adopted for the determination of DNA sequences while section 2.2 presents the two most widely used base-calling softwares: PHRED and ABI.

2.1 Proposed Base-Calling Methods

Giddings et al. [12] proposed an object oriented modular algorithm for the determination of DNA sequences. The system initially consists of a pre-processing stage which comprises of noise filtering using a dual Gaussian band pass filter, manual mobility shift correction, normalization and baseline correction. The base-calling involved identifying the peaks in the trace files for all four bases, determining the peaks that are most probable to represent a base by assigning a confidence value to each peak based on the peaks features: height, spacing and width. Peaks with low confidence values are discarded. The resulting sequence undergoes post processing which involves inserting bases in appropriate locations where no bases were called due to low signal strength and decreased resolution. Although Giddings et al.'s approach takes the advantage of object oriented programming techniques for modularity and extensibility, the proposed process is not automated for users who do not have enough background in the area.

In 1996, Berno [13] introduced a graph theoretic approach to perform base-calling by proposing an algorithm designed to be applicable to data from novel instruments. The approach initialized by low-pass filtering the data to remove the noise, followed by channel separation to eliminate crosstalk between the four channels. Berno then performed mobility shift corrections caused by the variance in the fluorescent tags weight attached to the bases. It was then followed by baseline removal and de-convolution to solve the problem of overlapping bases. A scoring function was then used to assess the confidence of occurrence of each peak and the sequence of peaks with the maximum score then selected for base-calling. On testing the algorithm, Berno observed that the system generated less insertion and mismatch errors compared to the ABI (Applied Biosystems Incorporated) sequencing software. However, it produced double the deletion errors than that of the ABI software. Although the proposed solution has a higher overall error rate, the proposed algorithm was observed to require no human intervention and requires minimal configuration for different sequencing conditions.

An automated base-calling algorithm called the Maximum Likelihood base-caller was presented by Brady et al. in 2000 [14]. The algorithm involved pre-processing as an initial step that consists of a soft caller and a hard caller. The *soft caller* is used to compute a set of tentative call amplitudes and their locations for each base producing a set of soft calls. The *hard caller* combines the tentative calls for all four bases and produces the final sequence estimate using a computationally expensive dynamic programming approach. On testing the proposed algorithm on 125 experimental datasets, the base-caller resulted in a 40% fewer errors than the ABI software (version 2.1.1), and its performance was observed to be comparable to that of the PHRED (Phil's Read Editor) base-caller.

M. Pereira et al. [15] proposed a statistical learning approach to solve the problem of DNA sequencing that enables the usage of prior knowledge to increase the accuracy of base-calling. The proposed solution used the Bayesian Expectation Maximization (EM) algorithm to perform base-calling. The pre-processing stage mainly included de-correlation, background subtraction and normalization. The feature vector (peaks, valleys and zero crossings) was then obtained from the pre-processed data. These features are used to segment the data into events which are fully characterized by four parameters: start-point, end-point, peak location and peak

height. The results obtained using the proposed approach was compared to ABI (version 2.1.1) and PHRED algorithms and was observed to have a comparatively lower error rate.

An online Bayesian model for DNA sequencing was proposed by N. Haan and S. Godsill [16-17]. Using the Bayesian algorithm, the random nature of the DNA sequence was represented and the Monte Carlo Methods were adopted to perform the required filtering and model selection parameters. On testing the proposed algorithm and comparing its performance with that of the PHRED base-caller, it was observed that the Bayesian model was more accurate since PHRED does not model the peak shape and it uses a deterministic peak detection scheme making it more prone to base-calling errors.

In 2004, Boufounos et al. [18] presented several Hidden Markov Models to solve the problem of DNA base-calling. The state emission densities are modeled using Artificial Neural Network (ANN) and the overall model is trained using a modified Baum Welch re-estimation algorithm. The model used the rise, the apex and the fall of the corresponding peaks in the electropherogram to represent each state, and the trained model then used the Viterbi algorithm to perform base-calling. Although an HMM base-caller was found to perform better than the PHRED sequencing software in terms of insertions and substitutions, a significant number of deletion errors was observed using the base-caller. One main advantage of this model is that it does not assume a particular peak shape at the cost of requiring some initial training.

Thornley et al. [19] introduced a method to perform base-calling using machine learning by exploiting variation in peak heights. Thornley et al. proposed that difference in peak heights represents novel information which can be utilized for base-calling. The pre-processing stage involved carrying out skyline normalization and identifying the high quality regions of the traces. The data were then segmented and feature extraction was carried out. The feature vector consisted of bases, peak heights, peak height ratios, and peak spacing. The various features were then passed through neural network and classification tree classifiers, singly and in ensembles. The results obtained were found to be promising with lower error rates.

In 2006, Eltoukhy et al. [20] proposed to perform DNA base-calling by using Sequencing-by-synthesis methods such as Pyrosequencing. Given a test sequence and the expected noisy output DNA sequence, they proposed to determine the system parameters by finding the DNA sequence that minimizes the probability of decoding error. The pre-processing stage consisted of baseline correction and normalization. Iterative partial maximum likelihood sequence detection (MLSD) was applied to five pyrosequencing datasets. Of the two longest datasets, a total of 170 out of 208 bases and 205 out of 224 bases were observed to be correctly decoded while the other shorter datasets resulted in no errors on base-calling. Thus, the proposed algorithm has the advantage of being capable of recognizing more than 300 bases.

Liang et al. [21] introduced a Bayesian model for DNA base-calling using Hidden Markov Models (HMM). Markov Chain Monte Carlo Method (MCMC) is used to encode the prior biological knowledge into the base-calling algorithm and modeling the system using the training data. Liang et al. used the rise, apex, fall and the buffer state between bases to represent each individual state and the MCMC algorithm was then used to perform base-calling. On comparing the results of the proposed model with that of PHRED and Expectation Maximization (EM) base-callers, the Bayesian base-caller outperformed the other sequencing models and resulted in a lower total error rate compared to the other statistical base-callers.

Another approach to perform DNA base-calling was proposed by Thornely et al. [22] using Neuro-fuzzy classifiers. A Self-Adaptive Neuro-Fuzzy Inference System (SANFIS) classifier was chosen as a Neuro-fuzzy network due to its immunity from the problem of dimensionality. The classifier consisted of two stages. Initially, using four SANFIS classifiers, bases were attempted to be recognized. In case of failure to call a base, the second stage was implemented where a Neural Network is used as a classifier. On testing the model, accuracy rates of approximately 74%, 77%, 79% and 83% were obtained for bases T, A, C and G, respectively. Hence, a total accuracy rate of 68.77% was obtained.

Heuristic base-callers, devised as in [12-13], are not built on strong theoretical basis. They depend on a large number of parameters that needs to be optimized to a specific type of chemistry or to a certain type of sequencing technology. Statistical base-callers are either poorly tested or slow, due to the high computational complexity of the implemented algorithms [14-15]. In this thesis, a well established pattern

recognition framework is used to build our base-caller. The base-caller we present is not restricted to a specific chemistry or sequencing machine, and its performance is not affected by the noise inherent in the electropherograms or the length of the trace.

2.2 Popular Existing Base-Calling Software

The two most widely used base-calling software, commercially and academically, are the PHRED and the ABI KB base-callers.

2.2.1 PHRED base-caller

PHRED was developed in the early 1990s by Dr. Phil Green and Dr. Brent Ewing. It is the most widely used base-calling software due to its high accuracy. The procedure adopted by PHRED to determine the DNA sequence can be summarized into four main phases as follows [23]:

1. A series of evenly spaced peak locations are predicted.
2. Actual peaks and their areas relative to their neighbors are indentified.
3. Peaks identified in the second step are matched to their predicted peak locations.
4. An unmatched clear peak is called a base if: (a) it has the largest amplitude at that location compared to the other signals, (b) its size exceeds a certain threshold, and (c) the peak is surrounded by resolved peaks.

On calling the bases, PHRED assigns highly accurate quality scores to each base-call making them ideal tools to analyze the accuracy of DNA sequences [24].

2.2.2 Applied Biosystems (ABI) KB base-caller

The ABI base-caller was initially devised, based on mobility curves, to predict the spacing between consecutive peaks and indentify the base corresponding to the peak in successive intervals. The base-caller returns an N when it fails to assign a base due to noisy peaks or when it cannot find a peak. This results in a high rate of substitution errors, but it still remains the preferable base-caller. However, with the

advent of PHRED in the early 1990s, which has a higher accuracy rate [23], ABI improved their software and developed the KB base-caller. This new development of ABI incorporates quality values to every base-call similar to PHRED. It was calibrated using more than 20 million base-calls and tested using more that 10 million bases [25].

CHAPTER 3

PATTERN RECOGNITION

Base-calling is one of many problems that is studied in the field of *pattern recognition*. Pattern recognition, also referred to as *pattern classification*, is the act of classifying raw data, or *patterns*, into different categories or *classes*. A typical example is handwriting recognition in which the movement of a pen tip is captured as input and classified to sequences of letters or words. Another example is face recognition where the 'face' of a person needs to be recognized in an image.

The parameters of pattern recognition models are estimated usually by *training* some sample data. The trained model can then be used to classify novel data into its various classes. This method of training is called *supervised learning.* In cases when training data is not available and similar data needs to be clustered together, *unsupervised learning* is implemented [26, pp. 6-8].

In this chapter, section 3.1 discusses the general pattern recognition framework followed by some common considerations in the design of pattern recognition systems. A brief introduction to the statistical models, which are implemented in this thesis for DNA base-calling will then be presented. In particular, *Artificial Neural Networks (ANN)* and *Polynomial Classifier (PC)* will be discussed in sections 3.3 and 3.4 respectively.

3.1 General Pattern Recognition Framework

A typical pattern recognition system includes five main steps: sensing, pre-processing, feature extraction, classification and post processing. Figure 3.1 shows the general pattern recognition framework for any typical classification problem.

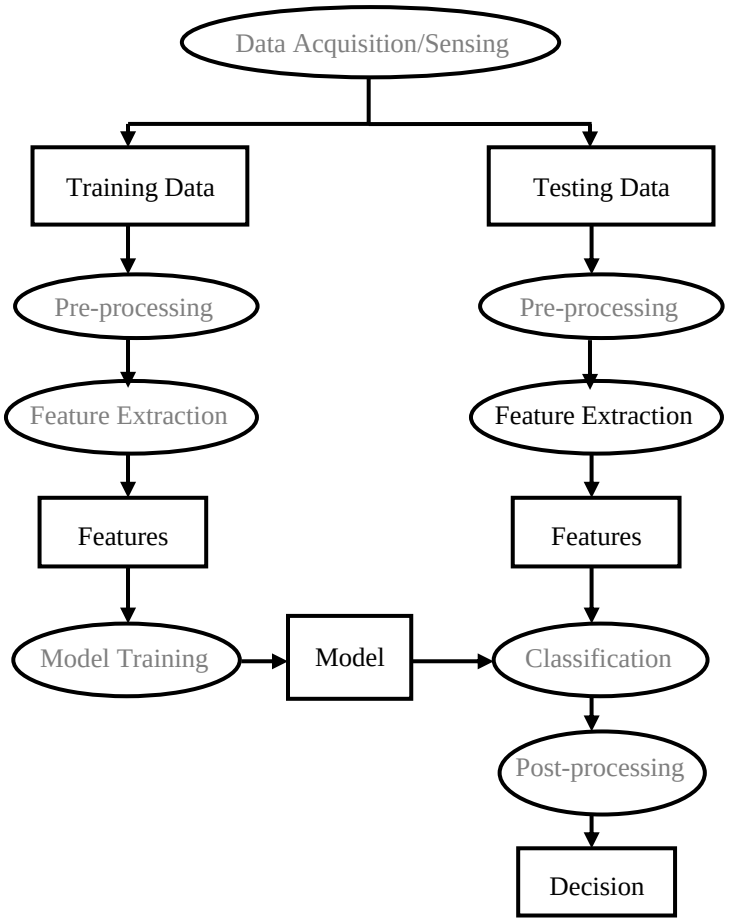


Figure 3.1. General Pattern Recognition Framework.

3.1.1 Data Acquisition/Sensing

Sufficient data to be used for training and testing the classifier model are obtained using appropriate sensing devices. For DNA base-calling, data are acquired from existing sequencing machines, such as ABI 3700, as chromatogram traces after a chosen DNA template has undergone chain termination, polymerase chain reaction (PCR) and electrophoresis. The collected data is divided into two sets: training and

testing. The training data set is used to estimate the chosen classifier model's parameters and the model's performance is then evaluated using the testing set.

3.1.2 Pre-processing

The pre-processing stage involves removal of noise from the acquired data and isolation of the inherent patterns from its background. For DNA base-calling, pre-processing may include wavelet decomposition, base line correction, mobility shift correction, outlier removal or normalization. The obtained patterns are then ready for feature extraction.

3.1.3 Feature Extraction

A *feature* represents the smallest data set that can be used for the classification and unique identification of each pattern. The purpose of feature extraction is to find a discriminative representation of the raw data and to realize that not all data points are useful for classification. For example a set of pixels representing a picture of a man could be reduced to a set of meaningful features such as shape of mouth, skin, or color.

3.1.4 Model Training

Implementing supervised learning, the model's parameters are estimated on mapping the features extracted from the raw training data to the known labels or classes. Parameters are estimated generally using an optimization technique, such as *Gradient Descent* or *Expectation Maximization,* aiming to minimize a related cost function.

3.1.5 Classification

In this stage, features extracted from novel data and learned models are used to assign each testing sample to a class. The performance of the trained model is then evaluated.

3.1.6 Post-processing

To improve the performance of the classifier, post-processing of the results obtained on testing the trained models on novel data is needed. Based on prior information, the classifier's output is modified to a more understandable and meaningful format.

3.2 Model Design Considerations

To design a highly efficient classifier as a pattern recognition system, several factors need to be taken into consideration. Correct recognition of input data increases based on the generalization of the training data, discriminative nature of the feature vector and optimal selection of the classification model.

3.2.1 Data Collection

Adequate training and testing data sets should be chosen; the larger the amount of data used, the better is the class representation. However, *over-fitting* needs to be prevented. Over-fitting refers to the tendency of a model to adapt itself to the minute details of a training data set due to the model's many parameters. This leads to poor generalization performance of the trained model to novel feature vectors [26, pp. 6-8]. To reduce over-fitting, a validation data set can be used to test the classification performance of a system. As training proceeds, performance of a model on a validation set is expected to increase. Once the performance starts decreasing, training should be stopped since the model starts over-fitting to the training data.

3.2.2 Feature Selection

Features that are chosen to represent a pattern should be discriminative, precise, concise and invariant to noise. On evaluating the model's performance, a high error rate indicates the features non-representativeness and needs to be changed.

3.2.3 Classifier Model Selection

An optimum model is one whose parameters are estimated once a related cost function is minimized such that it maps training data to a known output target. A model chosen to have a large parameter space requires a large duration of time to be trained and might result in over-fitting. On the other hand, a model with a smaller parameter space might result in a model with poor performance. Moreover, a classifier model is chosen based on the required performance and efficiency of the pattern recognition problem being tackled. Based on the available resources, the classifier model and its computational complexity are chosen. Note that a tradeoff between computational ease and performance is faced since a high performance system requires high computational complexity.

3.3 Artificial Neural Network

Artificial neural network (ANN) is a computational approach conceived as an imitation of the neural networks in the human brain. It is an adaptive system that changes its structure based on the information that it acquires from its environment in the learning stage. It consists of layers of simple processing units, called *neurons*, operating in parallel and connected through *weights* to solve specific problems [27]. ANN's quality as universal function estimators renders them attractive as pattern classifiers. ANN can be used to model any kind of data, linear or non-linear, which is advantageous. However, this property of ANN makes it prone to over-fitting. In this section, the neuron model is studied followed by an explanation of the general multi-layer feedforward neural network architecture.

3.3.1 Neuron Model:

A neuron is an *information-processing unit* that represents the basic building block of any ANN. It consists mainly of weights equal to the size of the data set, an adder to sum up the weighted inputs, and an activation function for limiting the output of the neuron. Figure 3.2 represents a neuron model [28].

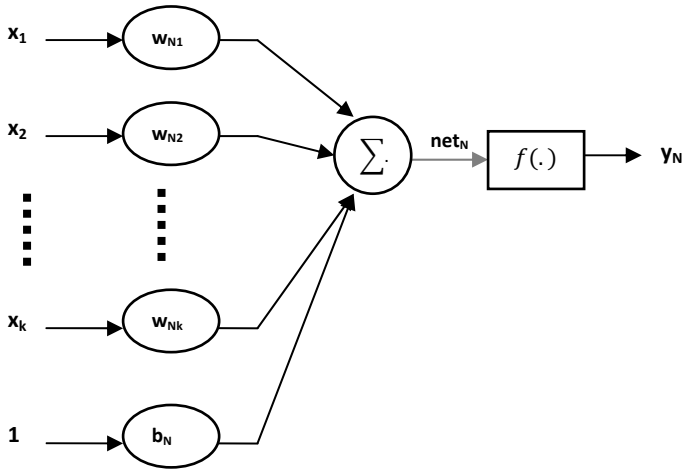


Figure 3.2. A General Neuron Model.

Representing the input data set, **X**, as,

$$\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \dots\dots\ \mathbf{x_k}] \tag{3.1}$$

And representing the N^{th} neuron set of weights, $\mathbf{w_N}$, as,

$$\mathbf{w_N} = [w_{N1}\ w_{N2}\ \dots\dots\ w_{Nk}]. \tag{3.2}$$

The operation of a neuron can be represented by the following pair of equations,

$$net_N = \left(\sum_{j=1}^{k} w_{Nj}x_j\right) + b_N \tag{3.3}$$

And

$$y_N = f(net_N) \tag{3.4}$$

Where b_N represents the bias unit, net_N is the adder's output, also referred to as the N^{th} neuron net activation, $f(.)$ represents the neuron's activation function, which is explained in the following section, and y_N is the N^{th} neuron's output.

3.3.2 Activation Functions

Activation functions, or transfer functions, are used to limit the output of a neuron to proper ranges. On implementing a single layer neural network, also known

as *perceptron*, a unit step activation function generally suffices. However, on using multi-layered neural networks, non-linear activation functions are chosen to increase the performance of the model [29]. Some popular types of activation functions are:

3.3.2.1. Unit Step Function

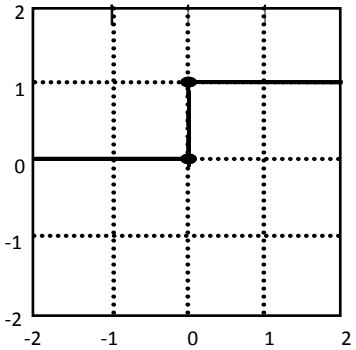


Figure 3.3. Unit Step Activation Function.

A unit step function is a discontinuous function which is zero for negative arguments and a one for positive arguments as shown in Figure 3.3. Hence, on passing a net activation value of less than zero through a unit step activation function, the output of the neuron would be a zero. On the other hand, a zero or positive net activation value would result in an output of one.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{3.5}$$

3.3.2.2. Linear Function

Linear functions are rarely used as transfer functions. They are mainly useful in applications where the entire range of numbers is needed as an output. On adopting a linear function (Figure 3.4) as an activation function, a non-modified output is obtained since the linear function is defined as follows [30],

$$f(x) = x \tag{3.6}$$

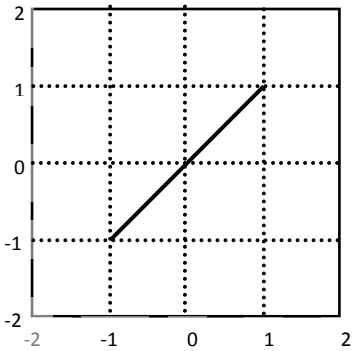


Figure 3.4. Linear Activation Function.

3.3.2.3. Hyperbolic Tangent Sigmoid Function

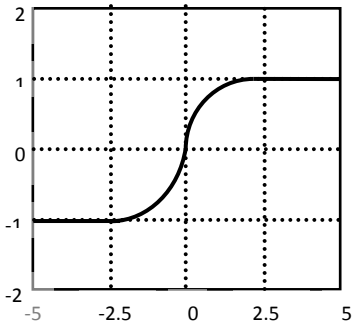


Figure 3.5. Sigmoid Activation Function.

A hyperbolic tangent sigmoid function used as a transfer function results in an output which varies continuously, though not linearly, as the input net activation value changes. The transfer function, as illustrated in Figure 3.5, returns both real positive and negative values and is differentiable, hence advantageous over a unit step activation function. A simple hyperbolic tangent sigmoid function is defined as follows [30],

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.7}$$

3.3.3 Feedforward Neural Networks

Figure 3.6 illustrates a typical feedforward neural network consisting of an input layer, a single hidden layer (The term *hidden* is used to indicate that this layer in the network is not seen directly by the user) and an output layer. Multiple neurons group together to form a layer and are connected to the neurons in the previous and next layers through biases and weights. The features extracted from the acquired data, represented as the input layer, constitute the input signals applied to the neurons comprised in the first hidden layer. Hence, the number of neurons in the input layer is equal to the dimensionality of the input feature vector. As a rule of thumb, a neural network with one hidden layer has the same expressive power as a network built from several hidden layers. Number of neurons in a hidden layer is generally chosen as twice that of the neurons in the input layer [31]. The outputs of the first hidden layer are then used as inputs to the second hidden layer, if a second hidden layer exists in the architecture, or to the output layer, in the case of absence of other hidden layers. The number of neurons in the output layer is equal to the number of classes representing the input data [28]. For e.g. for DNA base-calling, classification involves recognition of the four bases, A, C, T and G, referred to as classes. Hence, four neurons are used to form the output layer.

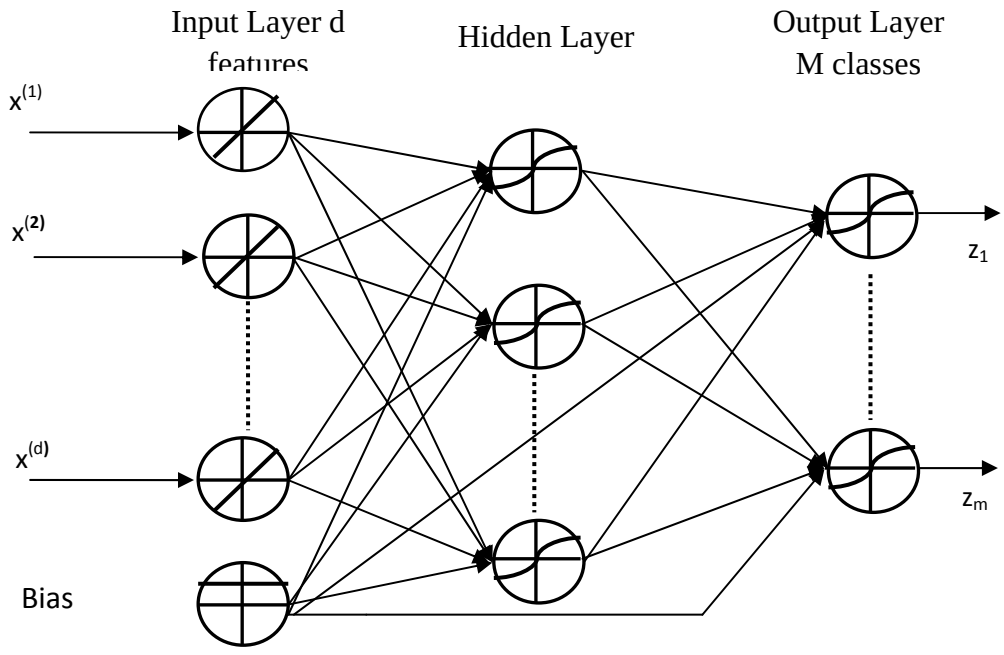


Figure 3.6. A Typical Feedforward Neural Network.

3.3.4 General Neural Network Model

A general neural network model can be defined as follows:

- Number of layers = L.
- Number of neurons in the N^{th} layer = k.
- Input matrix, **X**, composed of d −dimensional input feature vectors,

$$\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \dots\dots\ \mathbf{x_d}]^T \tag{3.8}$$

- Weight matrix,**W**, of size k × d of the N^{th} layer,

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,d} \\ w_{2,1} & w_{2,2} & \dots & w_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,1} & w_{k,2} & \cdots & w_{k,d} \end{bmatrix} \tag{3.9}$$

- Output vector of the N^{th} layer, **y**, can be defined as,

$$\mathbf{y} = [y_1\ \ y_2\ \dots\dots\ y_k]^T \tag{3.10}$$

The output vector can also be defined in terms of the activation function output,

$$\mathbf{y} = f(\mathbf{WX}) \tag{3.11}$$

3.3.5 Error Backpropagation

To solve a problem using neural networks, the network is trained using *error backpropagation* (based on gradient descent), where each input is mapped to its corresponding target output. The difference between the output of the modeled network and the desired target is determined to obtain the error vector, **e**, which represents the cost function that needs to be minimized. The error vector is defined as,

$$\mathbf{e} = \mathbf{y} - \mathbf{y_t} \tag{3.12}$$

where $\mathbf{y_t}$ represents the target output vector. The magnitude squared of the error vector can be defined as,

$$E = \frac{1}{2}\sum_{i=1}^{k}(y_i - y_{ti})^2 = \frac{1}{2}\sum_{i=1}^{k} {e_i}^2 \tag{3.13}$$

The derivative of the cost function is then used to update the weights after initializing them to random values. The following is then used to update the weights,

$$w^{(t+1)} = w^{(t)} - \eta\frac{\partial E}{\partial w} \tag{3.14}$$

where t represents the epoch or iteration number and η represents the step size or learning rate on which the backpropagation depends. The cost function, E, is expected to decrease during gradient descent. However, if the sum of the squared error oscillates then the chosen step size is large. If, on the other hand, the cost function was observed to decrease but at a very slow rate, then the chosen learning rate is small and needs to be increased.

Using the updated weights, the training process is repeated until convergence of the error is achieved and no significant change in the weights is observed.

ANNs have several advantages that make them an attractive tool for classification. They have the ability to learn complex mappings and adapt to various data. Though it is one of the simplest models to implement and does not require prior knowledge of the process being identified, ANNs need a large sample size for the model to be trained and requires a large computational effort to minimize over-fitting. Successful implementation of neural networks depends on proper selection of the number of layers, number of neurons in each layer, and the choice of activation functions. However, if the chosen topology results in a poor performance classifier then a new network structure needs to be selected.

3.4 Polynomial Classifier

Polynomial classifier, also known as higher order neural network, is a single layer neural network that adopts the polynomial terms of the pattern features as inputs. On considering a classifier model to be built using k: d −dimensional feature vectors, and assuming that the classes to be categorized are non-linearly separable, a

mapping function between each input and its respective class needs to be determined. The components of the model are defined as follows,

- Number of classes = m.
- Input matrix, **X**, composed of k input feature vectors d −dimensional each.

$$\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \dots\dots\ \mathbf{x_k}]^{\mathrm{T}} \tag{3.15}$$

$$\mathbf{x} = [x_1\ x_2\ \dots\dots\ x_d] \tag{3.16}$$

- Output vector, **y**, can be defined as,

$$\mathbf{y} = [y_1\ y_2\ \dots\dots y_k]^{\mathrm{T}} \tag{3.17}$$

- Target vector, $\mathbf{t_x}$, can be defined as,

$$\mathbf{t_{x_m}} = \left[t_{x_{m,1}}\ t_{x_{m,2}}\ \dots\dots t_{x_{m,k}}\right]^{\mathrm{T}} \tag{3.18}$$

- Non-linear functions (to define the general case) to describe the mapping between **x** and **y**, f_i:

$$f_i = [f_1\ f_2\ \dots\dots f_k], \quad i = 1,2\dots, m \tag{3.19}$$

The model output, **y**, can then be described by,

$$\mathbf{y} \equiv [f_1(\mathbf{x})\ f_2(\mathbf{x})\ \dots\ \ f_k(\mathbf{x})]^{\mathrm{T}} \tag{3.20}$$

Assuming that the classes are separable by a non-linear hyper-surface, $g(\mathbf{x}) = 0$, if a mapping function which converts the non-linearly separable categories to linear classes can be determined, non-linear $g(\mathbf{x})$ can be represented as a linear combination of the interpolation functions, f_i and system weights, **w** [26, pp. 156-158].

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{k} w_i f_i\,(\mathbf{x}) \tag{3.21}$$

Once the interpolation function is determined, the non-linear classification problem reduces to a simple linear classifier model problem where only the weights, the linear

model parameter, in the k-dimensional space of the input data need to be determined. One such known interpolation function, which is used in pattern recognition, is the *polynomial classifier*.

A K^{th} order polynomial classifier uses a K^{th} order polynomial expansion function to map a d-dimensional feature vector, $\mathbf{x}$, into a $O_{d,K}-$ dimensional vector space, $\mathbf{p}(\mathbf{x})$ to increase the probability of obtaining linearly separable categories or classes. $O_{d,K}$ is a function of both the dimensionality of the feature vector, d, and the order of the polynomial expansion, K, and can be expressed, for orders 1,2, 3 and 4, as shown in Table 3.1 [32].

Table 3.1. Lengths of the polynomial expansion vector.

Order, K	Polynomial Expansion Length, $O_{d,K}$
1	$O_{d,1} = d + 1$
2	$O_{d,2} = O_{d,1} + \sum_{l=1}^{d} k$
3	$O_{d,3} = O_{d,2} + d^2 + C(d,3)$
4	$O_{d,4} = O_{d,3} + d^2 + 2C(d,2) + 3C(d,3)+C(d,4)$

Where $C(d,l) = \binom{d}{l}$ is the number of distinct subsets of l elements made from a set of d elements. For example, let $\mathbf{x}$ be a 2-dimensional feature vector, i.e. $d = 2$, represented as $\mathbf{x} = [x_1\ x_2]$. The mapping of $\mathbf{x}$ to a higher dimensional space of order $K = 2$ produces

$$\mathbf{p}(\mathbf{x}) = [1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_1^2 \quad x_2^2\ ]^T \tag{3.22}$$

Similarly, the sequence of feature vectors, $\mathbf{X}$, is expanded into their polynomial basis terms,

$$\mathbf{M} = [\mathbf{p}(\mathbf{x_1}) \quad \mathbf{p}(\mathbf{x_2}) \quad \dots \quad \mathbf{p}(\mathbf{x_k})\,] \tag{3.23}$$

The hyper-surface, $g(\mathbf{x})$, can then be defined as follows [26, pp. 161-162],

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} w_{ij} x_i x_j + \sum_{i=1}^{d} w_{ii} x_i^2 \tag{3.24}$$

Thus, when $d = 2$,

$$g(\mathbf{x}) = \mathbf{w}\,\mathbf{p}(\mathbf{x}) + w_{\mathbf{0}} \tag{3.25}$$

$$\mathbf{w} = [w_1 \quad w_2 \quad w_{12} \quad w_{11} \quad w_{22}] \tag{3.26}$$

The polynomial classifier is then trained using $\mathbf{M}$ to determine the optimum set of weights, $\mathbf{w}_{\mathrm{opt_m}}$, that minimizes the difference between the model output and the desired target output such that,

$$\mathbf{w}_{\mathrm{opt_m}} = \min_{w} \left\| \mathbf{M}\,\mathbf{w} - \mathbf{t}_{\mathrm{x_m}} \right\|_2 \tag{3.27}$$

The weights can then be obtained explicitly by [33],

$$\mathbf{M}^{\mathrm{T}}\mathbf{M}\mathbf{w}_{\mathrm{opt_m}} = \mathbf{M}^{\mathbf{T}}\mathbf{t}_{\mathrm{x_m}} \tag{3.28}$$

$$\mathbf{w}_{\mathrm{opt_m}} = \mathbf{M}^{-\mathbf{1}}\mathbf{t}_{\mathrm{x_m}} \tag{3.29}$$

Using the parameters obtained from the training stage, an unknown feature vector, $\mathbf{z}$, is expanded to its polynomial terms, $\mathbf{p}(\mathbf{z})$, to test the trained model. The target vector, $\mathbf{t_z}$, is obtained as follows,

$$\mathbf{t_z} = \mathbf{w}_{\mathrm{opt}}\mathbf{p}(\mathbf{z}) \tag{3.30}$$

A general block diagram summarizing the training and testing of a polynomial classifier is shown in Figure 3.7. The simplicity and speed of PCs algorithm makes it an easy to implement classifier. However, the memory storage required and the model complexity increases as the order of the polynomial selected to make the data linearly seperable increases.

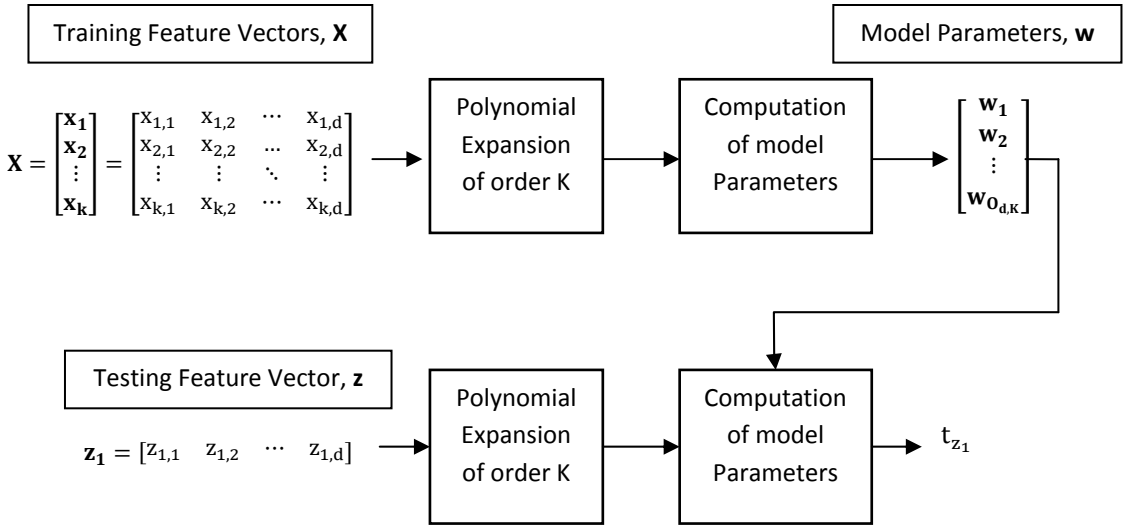


Figure 3.7. Training and Testing of a Polynomial Classifier.

CHAPTER 4

DATA ACQUISITION, PRE-PROCESSING AND FEATURE EXTRACTION

The solution adopted in this thesis to solve the problem of base-calling is based on designing a pattern recognition classifier. The success in the implementation of any approach depends on the effectiveness of the features extracted to represent a DNA pattern. Since the data acquired from the sequencing machines are noisy, a pre-processing stage is needed to achieve noise removal prior to efficient feature extraction.

In this chapter, the first three main components of any pattern recognition model: data acquisition (or sensing), pre-processing and feature extraction are discussed. Electropherogram traces acquired from sequencing machines, such as ABI 3730, are used as input data. The source and properties of the data used for training and testing the classifier models are discussed in the first section of this chapter. The details of the pre-processing stage are then presented in section 4.2. Pre-processing is needed to condition the signal for base-calling without losing useful information from the chromatogram traces. A block diagram of the pre-processing stage, which consists of: color correction, peak sharpening and normalization, is illustrated in Figure 4.1. The methods used to implement the various stages of pre-processing are explained in the subsequent section. The chapter is then concluded by a discussion of the features used to represent the pattern and the technique adopted to extract the chosen features from the chromatogram trace in section 4.3.

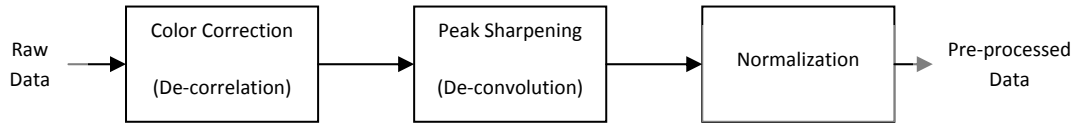


Figure 4.1. Pre-processing stage modules.

4.1 Data Acquisition/Sensing

One of the main considerations on designing a pattern recognition classifier is the presence of an adequate size of data to train and test the model. Performance of the classifier model increases as the amount of data used increases. The training data need to be chosen such that every possible case scenario is seen and learnt by the model. However, over-fitting needs to be prevented so that the generalization of the classifier model to novel data is not affected.

Chromatogram traces are needed to be labeled for training and testing the classifier model. However, acquiring labeled electropherogram traces proved to be difficult, expensive and labor intensive. Employing existing softwares, such as Codon Code Aligner and Bioedit, to label the data by using existing base-callers code, such as PHRED (Phil's Read Editor) or ABI (Applied Biosystems), results in an inaccurate benchmark since the model would be impacted by the adopted base-callers limitations. To avoid such a drawback in our model, the data, which are obtained from the Sorenson Molecular Genealogy Foundation (SMGF) and from the National Center for Biotechnology Information (NCBI) trace archive [34], are labeled using one of the following methods:

- Consensus Sequence: The DNA sequence obtained from the sequencing of overlapping fragments of a gene several times is referred to as the *consensus sequence*. Data provided by Dr. Scott Woodward from SMGF are supplemented with their respective consensus sequences. These sequences are used to label the chromatogram traces for accurate training of the classifier.
- NCBI BLAST: Traces obtained from the NCBI trace archive are initially labeled using commercially available PHRED base-calling software, CodonCode Aligner. PHRED is used since it demonstrates high accuracies when tested over a wide variety of sequencing methods and has proven to have a higher system performance compared to other existing base-callers

[24]. The NCBI Basic Local Alignment Search Tool (BLAST) is then run on each PHRED generated sequence to locate the corresponding consensus sequence for each DNA fragment being tested.

To evaluate the performance of the designed classifier models based on noise contamination, source and read length of the electropherograms, the traces obtained from the SMGF and from the NCBI trace archive are categorized into three main data sets:

1. Data set one: 6 electropherogram traces belonging to *Homo sapiens* chromosomes 5, 6, 11, 12 and 13 are obtained from the NCBI trace archive. All the traces consensus sequences are acquired from GRCh37, the most recent human assembly produced by the Genome Reference Consortium (GRC). Compared to the following data sets, the effect of noise and anomalies introduced during DNA sequencing is considerably higher. On average, the data set contains traces consisting of 600 to 700 bases.
2. Data set two: 11 electropherogram traces belonging to *Homo sapiens* mitochondrion D-loop chromosome, *Saccharomyces mikatae* (yeast) and *Drosophila melanogaster* (fruit fly) are obtained from SMGF and NCBI trace archive. The traces are noisy and consist of 675 to 775 bases on average.
3. Data set three: The data, which consists of 5 traces, are obtained from the SMGF. It belongs to *Homo sapiens* mitochondrion D-loop chromosome. The data, compared to the previous data sets, are contaminated with a low noise level in the first 500 bases of the trace. On average, the chromatogram traces consists of 700-800 bases.

For the analysis of classifier performance, the DNA sequence obtained on using PHRED and ABI base-callers on the above three data sets is needed. The following software are used to call bases using PHRED and ABI respectively:

- CodonCode Aligner

CodonCode Aligner is a powerful software used for assembling, aligning and editing DNA sequences. It performs base-calling of raw data using PHRED. Figure 4.2 illustrates a chromatogram trace executed in CodonCode Aligner. The bases seen on each peak are the result of the base-calling using PHRED algorithm. The software

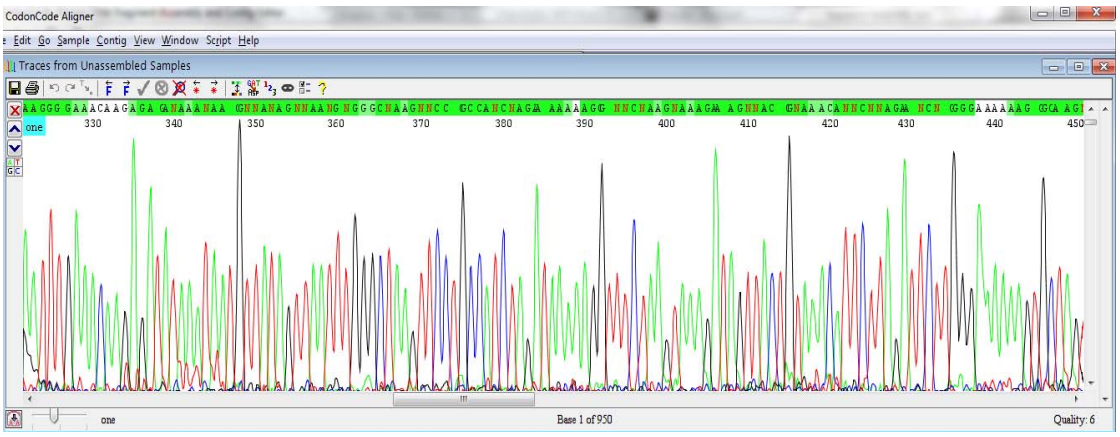


Figure 4.2. A Chromatogram Trace in CodonCode Aligner software.

Figure 4.3. A base view option of a chromatogram trace in CodonCode Aligner after base-calling using PHRED.

- BioEdit

It is an easy to use biological sequence aligner and editor designed for sequence manipulation. It is capable of reading and editing ABI files, complementing

the traces and their sequence and converting the trace to scf format among other functions. Figure 4.4 illustrates a chromatogram trace using Bioedit.

Figure 4.4. A segment of a chromatogram trace executed in BioEdit software.

4.2 Pre-processing

After the implementation of electrophoresis, the obtained DNA trace may be contaminated by noise introduced at various stages of sequencing. Noise contamination occurs as a result of the imperfections in the chemistry involved and the electronics of the electrophoresis. Noise superimposed on a DNA trace may appear in the form of overlapping spectra, presence of one or more large peaks at the beginning of the trace, a drift in the DC value of the signal, variations in the dynamic range and low peak resolution. The data chosen for both training and testing the designed models are hence subjected to several stages of pre-processing to condition the signals without losing useful information. It is to be noted, though, that the classifiers in this thesis were designed with the aim that minimum pre-processing is required to achieve a model with an appropriate performance. Hence, this stage involves three main processing functions: *color correction*, *peak sharpening* and *windowed normalization*.

4.2.1 Color Correction

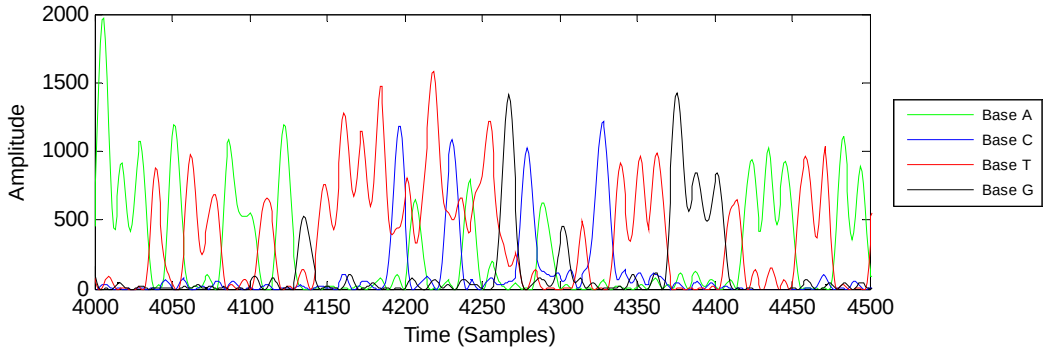
Cross talk refers to the detection of false peaks or peaks with erroneous excitation wavelengths as a result of variations in the signals present in the DNA sequencing channels. Cross talk interference distorts the signals and results in overlapping spectra which affects the performance of any base-caller. To minimize inter-lane cross talk chemically, the dyes as well as the laser wavelengths should be chosen sufficiently far apart to allow appropriate filtering. However, this needs to be optimized and implemented by the DNA sequencer manufacturers and is beyond the scope of this thesis. Instead, if the data is distorted due to cross talk, de-correlation is implemented to reduce the interference.

De-correlation, also known as color correction, is a linear operation which is implemented on raw data (Figure 4.5 (a)) to remove the cross-talk between the four lanes. A 4×4 cross correlation matrix, $\mathbf{M}$, is needed to obtain a color separated trace. Each column of the cross correlation matrix, also known as the mixing matrix, represents the relative signal intensity of each dye compared to the other dyes. However, $\mathbf{M}$ is not known initially and needs to be determined. One common practice uses the manufacturer's provided mixing matrix to implement the linear transformation. If the manufacturer is not known, the components of $\mathbf{M}$ can be determined by identifying a clear known peak in each lane of the raw data. For each of the identified peaks, the corresponding relative signal intensities are obtained and are placed as a single row in the matrix [12-13].

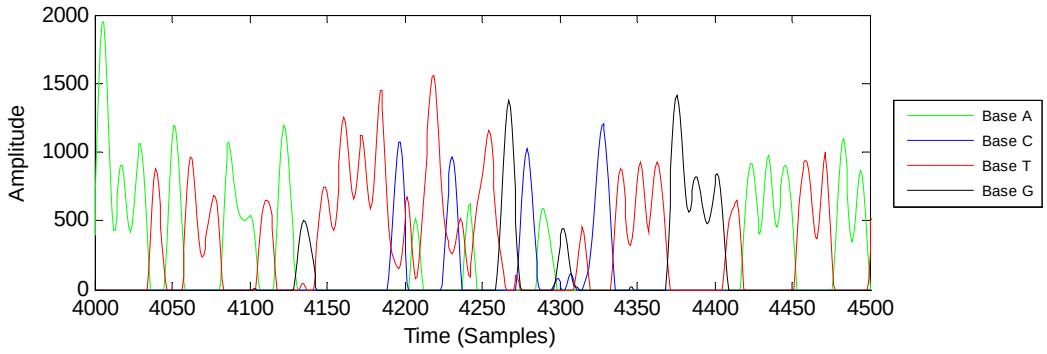
Since the data are acquired mainly from public databases, the matrix $\mathbf{M}$ provided by the respective manufacturers could not be found. Instead, the matrix $\mathbf{M}$ was initially estimated by the identification of four clear peaks in the chromatogram trace and the relative signal intensities were obtained. However, it was observed that the data did not achieve sufficient de-correlation. Hence, $\mathbf{M}$ was re-estimated by taking into consideration the entire trace, not only four clear peaks. $\mathbf{M}$ was determined using the MATLAB command "*corrcoef*" to obtain the correlation coefficients calculated from a raw input trace, $\mathbf{X_R}$, of size $n \times 4$, whose rows represent the observation samples and the columns are the bases (the variables). A linear transformation, using the matrix, $\mathbf{M}$, is then implemented to obtain the desired color corrected signal, $\mathbf{X_{cc}}$, as follows,

$$\mathbf{X_{cc}} = \mathbf{M}\mathbf{X_R} \tag{4.1}$$

Where $\mathbf{X_R} = [\mathbf{x}_{R,A}\ \mathbf{x}_{R,C}\ \mathbf{x}_{R,T}\ \mathbf{x}_{R,G}]$ and $\mathbf{X_{cc}} = [\mathbf{x}_{cc,A}\ \mathbf{x}_{cc,C}\ \mathbf{x}_{cc,T}\ \mathbf{x}_{cc,G}]$. Figure 4.5 (b) shows the trace data obtained after the implementation of the above de-correlation routine. Observe that the noisy background ripples are removed and the overlapping peaks have been either eliminated or their amplitude in most cases have been reduced as a result of color correction.



(a)

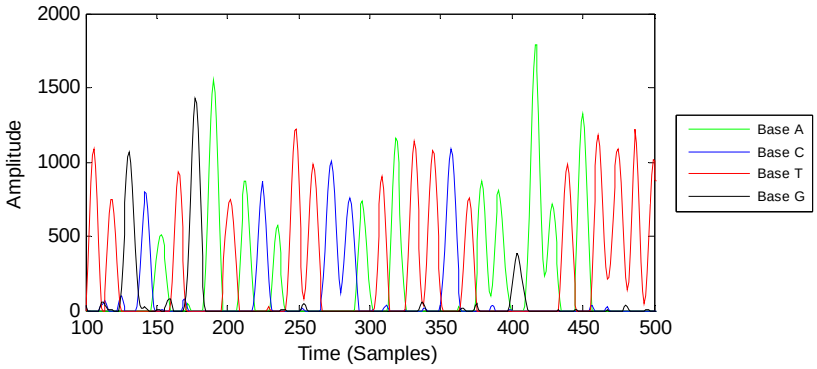


(b)

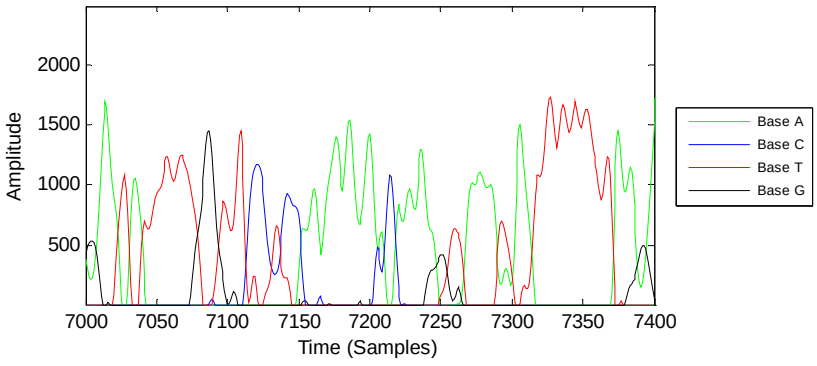
Figure 4.5. Part of a chromatogram trace (a) before and (b) after color correction.

4.2.2 Peak Sharpening

Ideally, each base in an electropherogram needs to be represented by a single peak. However, this rarely happens in reality. During electrophoresis, based on the length of the DNA fragments, the time needed for a fragment to reach the photo-detector depends on its length. Short DNA fragments travel faster than longer ones and hence are located in the early segments of a chromatogram trace. Typically, a certain range



(a)



(b)

Figure 4.6. (a) High resolution peaks at the initial parts of a trace and (b) Low resolutions peaks at the final parts of a trace.

Low resolution peaks results in inaccurate peak detection and hence, needs to be resolved. A non-linear iterative de-convolution algorithm [36] is implemented to recover the sharp base peaks. Chromatogram traces obtained from electrophoresis ideally represents a linear system. A high resolution trace, $\mathbf{x_{D,i}}$, is assumed to be a sparse pulse train corresponding to the occurrence of each base.

$$\mathbf{x_{D,i}} = \sum_{\mathbf{k}} \mathrm{a(k)p(n-k)} \tag{4.2}$$

Where k represents the peak positions, $\mathrm{p(n)}$ represents a very narrow pulse, and $\mathrm{a(k)}$ represents the amplitude of each pulse. The observed low resolution chromatogram trace, hence, can be obtained by the convolution of the desired trace with a point spread function which represents a blurring effect. The low resolution trace, $\mathbf{x_{cc,i}}$, can be obtained, mathematically, by the convolution of the high resolution trace, $\mathbf{x_{D,i}}$, with a point spread function, $\mathbf{h}$. That is,

$$\mathbf{x_{cc,i}} = \mathbf{x_{D,i}} \otimes \mathbf{h} \tag{4.3}$$

Thus, to reconstruct the high resolution trace, iterative de-convolution is adopted. The following outlines the general procedure to obtain the de-convolved DNA trace:

- Color corrected data, $\mathbf{x_{cc,i}}$, of size $\mathrm{n} \times 1$ is treated as the observed signal. Note that i represents the four bases: A, C, T and G.
- $\mathbf{x_{cc,i}}$ is initially normalized by its maximum observation to obtain $\mathbf{x_{CN,i}}$.

$$\mathbf{x_{CN,i}} = \frac{\mathbf{x_{cc,i}}}{\max(\mathbf{x_{cc,i}})} \qquad \text{for i} = \mathrm{A, C, T \text{ and } G} \tag{4.4}$$

- A normalized point spread Gaussian function, $\mathbf{h}$.
- De-convoluted data, $\mathbf{x_{D,i}}$, of size $\mathrm{n} \times 1$ is used to represent the desired signal.
- Initializing our first iteration, $\mathrm{y} = 0$, to obtain $\mathbf{x_{D,i_y}}$ as follows,

$$\mathbf{x_0} = \mathbf{x_{CN,i}} \tag{4.5}$$

$$\mathbf{x_{D,i_0}} = \mathbf{x_0} \tag{4.6}$$

- The initial assumption is convoluted with the point spread function and $\mathbf{x_{D,i}}$ is updated as follows,

$$\mathbf{x_{D,i_{y+1}}} = \mathbf{F}\,\mathbf{x_{D,i_y}} = \mathbf{x_{D,i_y}} + \lambda\left(\mathbf{x_{CN,i}} - \mathbf{h}\otimes \mathbf{x_{D,i_y}}\right) \qquad (4.7)$$

Where **F** is an operator and λ is the relaxation constant. When y is sufficiently large, $\mathbf{x_{D,i_{y+1}}}$ converges to the underlying pulse train,

$$\lim_{y\to\infty} \mathbf{x_{D,i_y}} = \mathbf{x_{D,i}} \qquad (4.8)$$

By performing iterative de-convolution, peak sharpening and enhancement of signal quality are achieved. Figure 4.7 (a) shows part of a chromatogram trace prior to de-convolution, while Figure 4.7 (b) shows the same trace after de-convolution. By comparing the two figures, the low resolution peaks sharpened to a higher resolution as a result of de-convolution can be observed.

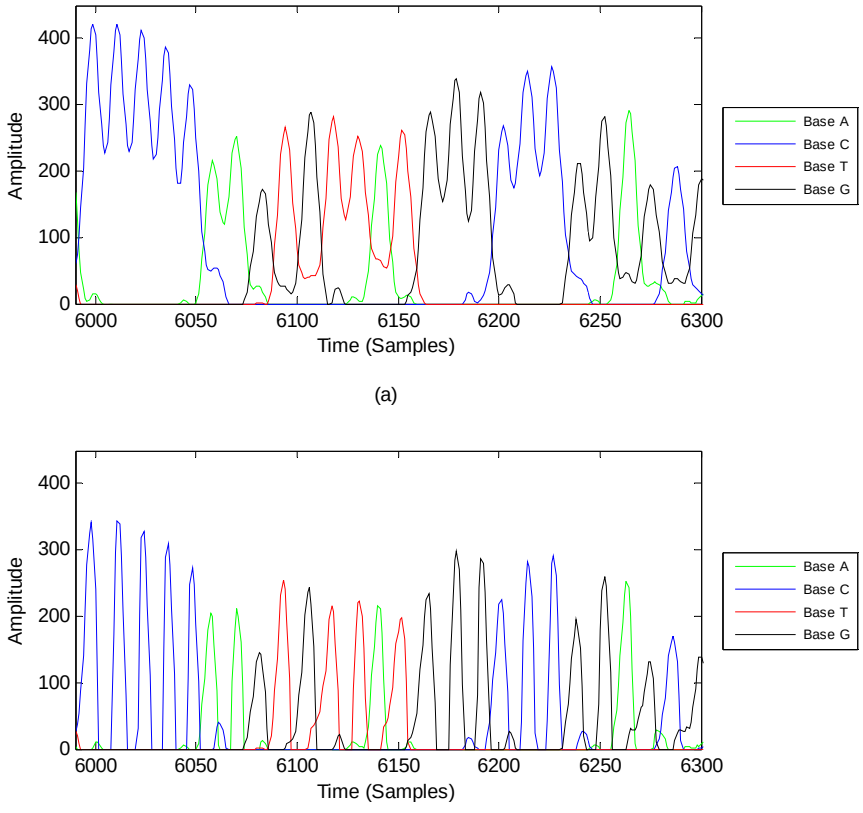


Figure 4.7. Chromatogram trace (a) before and (b) after de-convolution.

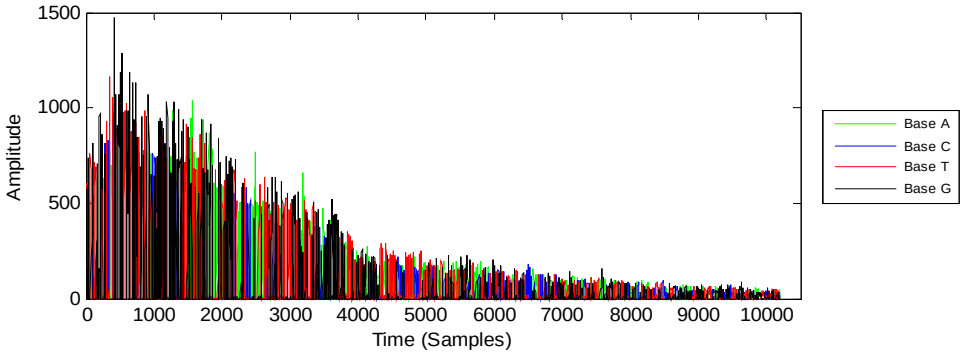
4.2.3 Normalization

The amplitude of observation points in a chromatogram trace is observed to decay with time due to several factors including chemical and laser source imperfections, system configuration and variations in detector sensitivity. Due to the difference in the dynamic range of a trace, it is vital to normalize the signals before base-calling is initiated. Normalization can be achieved using many different techniques. Giddings et al. [12] proposed segmentation of the observation points into consecutive windows. A scaling factor was then determined such that the segmented data are normalized to the [0, 1] range. Another method adopted, in [35], involves also the segmentation of the observation points into windows. However, for each window the average peak height is calculated and the segmented data are normalized according to it.

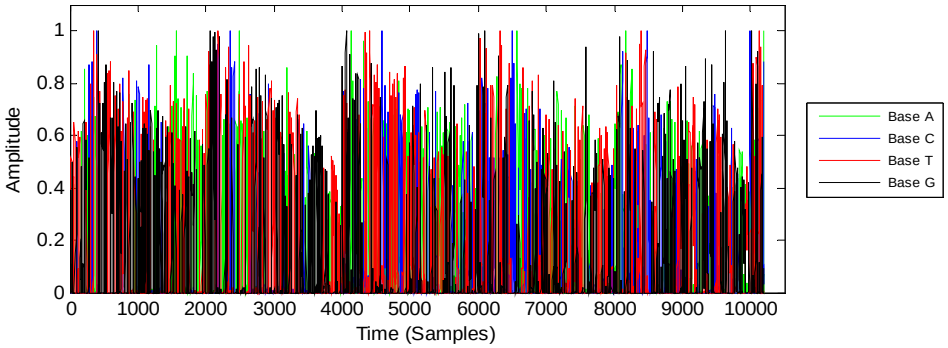
In this thesis, a simple normalization technique is adopted to obtain a sequence of sufficiently normalized observation points. The outline of the implemented method is as follows:

- A color corrected high resolution trace, $\mathbf{x_{D,i}}$, of size $n \times 1$ is used as input data.
- $\mathbf{x_{D,i}}$ is divided into non-overlapping consecutive windows of 2000 sample points in width. Choosing a window size of lesser than 2000 samples increased computational complexity and no increase in the performance of the classifier was observed. On the other hand, a window size of more than 2000 samples decreased the performance of the classifiers. Hence, an optimum window size of 2000 samples was chosen.
- The data in each segment is then normalized by its maximum amplitude.
- The above is repeated for each of the four lanes: A, C, T and G.

Figure 4.8 (a) illustrates the trace signals before applying normalization while Figure 4.8 (b) shows the same trace after normalization. The decay in the amplitude is evident in Figure 4.8 (a) while the uniformity of the signal height after normalization is seen in Figure 4.8 (b).



(a)



(b)

Figure 4.8. A Chromatogram trace (a) before and (b) after normalization.

4.3 Feature Extraction

Feature extraction is the heart of any classifier model design. Feature extraction is the main stage which has a vital affect on the performance of a pattern recognition model. Feature extraction is the process in which discriminative and unique functions are extracted from the pre-processed data to characterize the chromatogram trace. These features are then used to train and test the designed classifier model.

In our approach, the features chosen to represent each observation point in a chromatogram trace are as follows:

$$\mathbf{F} = [\mathbf{F_A}\ \ \mathbf{F_C}\ \ \mathbf{F_T}\ \ \mathbf{F_G}] \tag{4.9}$$

Where,

$$\mathbf{F_i} = [\mathbf{g^-}\ \mathbf{x}\ \mathbf{g^+}] \qquad \text{for i} = \text{A, C, T and G} \tag{4.10}$$

And:

- $\mathbf{F_i}$ represents the set of feature vectors of base i. It is a matrix of size $n \times 3$ where n is the number of sample points in the chromatogram trace.
- $\mathbf{x}$ is a vector of size $n \times 1$. It represents the signal strength of base i at each observation point,

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n] \tag{4.11}$$

- $\mathbf{g}^-$ is a vector of size $n \times 1$. It consists of the gradient values for each observation point,

$$\mathbf{g}^- = [g_1^- \quad g_2^- \quad \dots \quad g_n^-] \tag{4.12}$$

  It is calculated using the signal strength of the prior three sample points for each observation.

$$g_z^- = \frac{x_z - x_{z-3}}{3} \quad \text{for } z = 4,5,\dots,n-3 \tag{4.13}$$

- $\mathbf{g}^+$ is a vector of size $n \times 1$. It consists of the gradient values for each observation point,

$$\mathbf{g}^+ = [g_1^+ \quad g_2^+ \quad \dots \quad g_n^+] \tag{4.14}$$

  It is calculated using the signal strength of the subsequent three sample points for each observation.

$$g_z^+ = \frac{x_{z+3} - x_z}{3} \quad \text{for } z = 4,5,\dots,n-3 \tag{4.15}$$

From the chromatogram trace, it is observed that the positive ascent of a peak is defined by a minimum of 3 sample points before reaching the apex. Similarly, the negative ascent of a peak is represented by a minimum of 3 sample points before reaching a subsequent valley or positive slope. Hence, for the calculation of positive (4.15) and negative (4.13) gradient values, 3 samples points are used.

CHAPTER 5

DNA BASE-CALLING AS A PATTERN RECOGNITION PROBLEM

As discussed earlier in chapter 3, any typical pattern recognition system involves five main steps as demonstrated in Figure 5.1. The method we adopted to acquire and pre-process the data needed to design our DNA base-caller has been explained in chapter 4. Using the features extracted in the previous chapter, the DNA base-calling problem can now be tackled. In this chapter, the implementation of our base-calling method and its results are discussed. In section 5.1, the classifiers topologies that fit the requirement of the problem will be developed. The post-processing of the scores obtained will then be explained in section 5.2. Section 5.3 presents the results obtained on training and testing the ANN and PC model on each of the three different acquired data sets (explained in section 4.1). A new training technique, leave-one-out method, which we adopted to create a large sample of training data, will also be briefly discussed in section 5.3. Finally, section 5.4 presents a summary of the conclusions we reach on observing the performance of our base-callers.

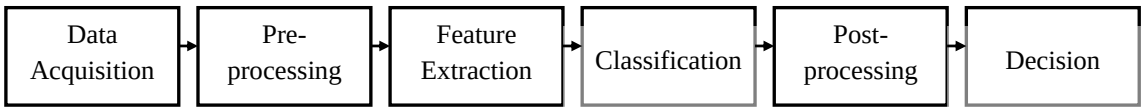


Figure 5.1. A Typical Pattern Recognition Framework.

5.1 Classification

In this thesis, two pattern recognition models are used to solve the base-calling problem: Artificial Neural Networks (ANNs) and Polynomial Classifiers (PCs). This section explains how each model's framework was designed and the notations used to describe the input data, output data and the parameters chosen.

5.1.1 ANN Model Topology

The neural network adopted is a single hidden layer feedforward model that updates its weights according to the backpropagation algorithm explained earlier in section 2.3.4. The chromatogram trace pattern is represented using the features extracted from each sample point, to form the input data matrix, **X**, where,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_1} \\ \vdots \\ \mathbf{x_n} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \tag{5.1}$$

X, is an $n \times d$ matrix made up of d-dimensional feature vectors, where d is 12 since each base signal is represented by three features. Hence, to represent the entire chromatogram trace, which consists of n sample points, made up of four base signals, each feature vector consists of 12 features. For training an ANN model, a target matrix, **T**, is needed to label the classes for each input feature vector, **x**. Each chromatogram trace consists of 5 classes, i.e. $m = 5$, as follows:

- A class to represent the presence of base A.
- A class to represent the presence of base C.
- A class to represent the presence of base T.
- A class to represent the presence of base G.
- A class to represent the absence of all the above four classes.

The columns of the $n \times 5$ target matrix, **T**, represents the base signals A, C, T and G, respectively, and the last column represents the absence of all the bases, also referred to as N.

$$\mathbf{T} = \begin{bmatrix} \mathbf{t_1} \\ \mathbf{t_2} \\ \vdots \\ \mathbf{t_n} \end{bmatrix} = \begin{bmatrix} t_{1,A} & t_{1,C} & t_{1,T} & t_{1,G} & t_{1,N} \\ t_{2,A} & t_{2,C} & t_{2,T} & t_{2,G} & t_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n,A} & t_{n,C} & t_{n,T} & t_{n,G} & t_{n,N} \end{bmatrix} \tag{5.2}$$

The values assigned to the elements of **T** are as follows:

- $(t_{z,i=B}) = 1$ while $(t_{z,i\neq B} \text{ and } t_{z,N}) = 0$ for sample point z and $B \in \{A, C, T, G\}$, if base i has a positive feature g_z^-, indicating a positive slope for the three sample points prior to z, and a negative feature g_z^+, indicating a negative slope for the three sample points subsequent to z .
- $(t_{z,N}) = 0.05$ while $(t_{z,i}) = 0$ for sample point z and $B \in \{A, C, T, G\}$ if the above condition is not satisfied. The prior probabilities of the presence of a base and the absence of a base are imbalanced due to the large availability of class N in a chromatogram trace compared to the other bases. Since it is difficult to balance the amount of data belonging to each class, the weight given for class N is reduced by assigning it a target value of 0.05 [33].

Using the Neural Network toolbox, found in MATLAB R2009b, the neural network model was trained and tested using a single hidden layer, an output layer consisting of five neurons to represent each of the five previously mentioned classes and hyperbolic tangent sigmoid transfer functions. To avoid the problem of over-fitting, the acquired data are divided into three sets for training, validation and testing. The validation data set is used to stop the training process when further training will only optimize the performance of the model on the training set at the expense of its ability to generalize.

5.1.2 Polynomial Classifier

Polynomial classifiers (PCs) represent non-linear system identifications providing an efficient method to describe non-linear input/output relationships. Its parameters are estimated using least squares to minimize the error between the model output and the desired target output. On employing a 2nd order polynomial expansion,

the data was still observed to be non-linearly separable. Hence a 3rd order polynomial classifier using a 3rd order polynomial expansion function is implemented in this thesis. The 12 features extracted from the $n-$sample point chromatogram trace to represent the observed pattern involved is used to form the data matrix, **X**, where,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_1} \\ \vdots \\ \mathbf{x_n} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,12} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,12} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,12} \end{bmatrix} \tag{5.3}$$

The polynomial expansion function is then used to map each $12-$dimensional feature vector into a higher dimensional vector space, $\mathbf{p}(\mathbf{x})$. Since a 3rd order polynomial classifier is used, the length of the interpolated feature vector, obtained by using Table 3.1, is

$$O_{d,4} = O_{d,2} + d^2 + C(d,3) \tag{5.4}$$

Where $C(d,l) = \binom{d}{l}$ is the possible number of distinct subsets of l elements made from a set of d elements. The higher order monomials, can then be stacked together to form a set of feature vectors, **M**, which is used for training the polynomial classifier model and obtaining its parameters. Moreover, similar to ANNs, a target matrix, **T**, is needed to label the classes: A, C, T, G and N, for each input feature vector, **x**. The target matrix is identical to that generated for ANNs. The trained model is then tested using novel data from the three data sets and a set of scores are obtained which undergoes post-processing to attain the final DNA sequence.

5.2 Post-processing

Through the training of ANN and PC models, a set of optimum weights is obtained. The trained models are then tested on novel data to determine the performance of each. A set of scores is obtained, based on which, the class of each sample point is decided. Post-processing involves the following steps:

- Features resulting in negative scores are equated to zero. Negative scores represent the absence of a peak in that specific base signal.

Adjacent peaks, irrespective of the bases, need to be separated by a minimum of 6 sample points. If two neighboring peaks are separated by less than 6 sample points, the peak with the highest score is called while the other peak is classified as an N.

- After observing a histogram of the scores obtained, a specific threshold is chosen below which a sample point is classified as an N.

The Bioinformatics toolbox in MATLAB is then used to align the DNA sequence that was obtained using the trained model and the reference sequence. A $3 \times n$ character array is attained where the two sequences in first and third rows, while symbols representing their optimal alignment are found in the second row. The "|" symbol indicates matched bases, the "-" indicates the absence of a base and the ":" indicates a mismatch. Figure 5.2 illustrates a sample of aligned sequences.

Using the alignment, the performance of the trained model can be measured. The performance measure we used is based on the three types of errors that can occur in DNA base-calling: *deletion errors, insertion errors* and *substitution errors*. A *deletion error* occurs when a base is not called by the base-caller where there should have been one. For example, when the actual base sequence is ATCGT and the base-caller calls ACGT, a deletion error occurs. An insertion error occurs when a base is called by the base-caller where there should have been none. For example, when the actual base sequence is ATCGT and the base-caller calls ACTCGT, an insertion error occurs. A substitution error occurs when the called base is different from that of the actual sequence. For example, when the base sequence is ATCGT and the base-caller calls AACGT, a substitution error occurs. The bases called are not only compared to the actual reference sequence but also to the DNA sequences obtained using PHRED and ABI base-callers to validate the performance of our classifiers.

Figure 5.2. A sample aligned sequence.

5.3 Results and Discussion

As explained in section 4.1, data acquired were divided into three sets based on the extent of noise contamination, source of the data: NCBI or SMGF, the organism the trace belongs to, and the read length. The method adopted to train and test each data set's ANN and PC model is as follows:

5.3.1 Data set one

The data set consists of six chromatogram traces obtained from the NCBI trace archive. It belongs to *Homo sapiens's* chromosomes 5, 6, 11, 12 and 13. The reference sequence, which is used to generate the target matrix for training the models, was acquired from the GRCh37, the most recent human assembly produced by the Genome Reference Consortium (GRC). The data chosen in this set were contaminated with a high level of noise so that the performance of the model to

various anomalies could be observed. On average, the data set contains traces consisting of 600 to 700 bases.

In light of the limitation in the number of bases, *round robin strategy* is used in training and testing the proposed models to increase the statistical significance of the results. The available traces are divided into k disjoint sets, such that k models are trained using the data in the $k-1$ sets and tested on the remaining non-trained data set. In the case where k is equal to the number of traces in the data set, i.e. $k=6$, *leave-one-out method* is implemented (Figure 5.3), i.e. out of the six available traces, traces 2 to 6 are used for training while the first trace is used for testing. The next round uses traces 1, 3 to 6 for training and the second trace is spared for testing, and the cycle repeats itself k times [37].

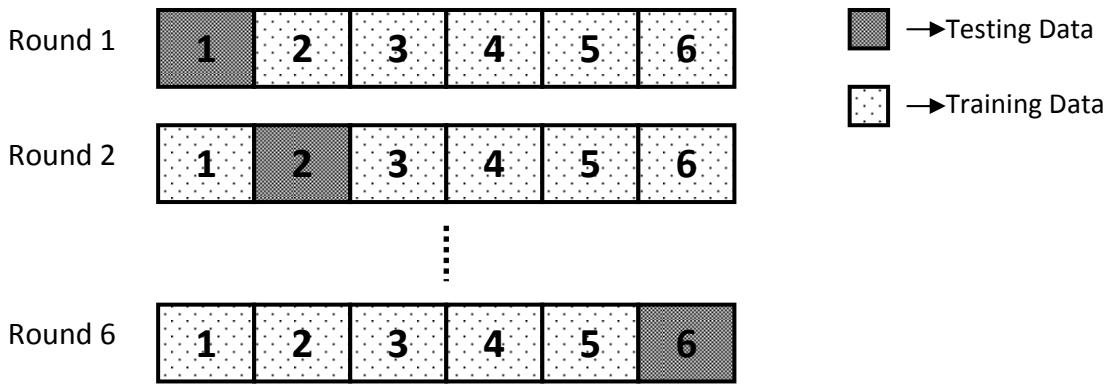


Figure 5.3. Assignment of the testing data set using leave-one-out method.

5.3.1.1. Evaluation of performance of ANNs

The performance of the trained model in terms of correct bases obtained, deletion errors, insertion errors and substitution errors is shown in Table 5.1 and Table 5.2. Table 5.1 compares the performance of the proposed ANN model to that of PHRED while Table 5.2 compares it to that of ABI. As observed from the tables, on average, an accuracy of 97.39% is achieved on testing the trained ANN model. After a close examination of the errors, a lower substitution error is observed for all tested data compared to PHRED. The misclassifications are generally observed after crossing 500 bases which might be a result of the data being of a low resolution in the

However, on comparing the proposed model to ABI, the performance of the ABI base-caller is higher than that of ANN. Being a heavily optimized software, ABI made a lower total error specially in terms of deletion error. The model was trained using only six chromatogram traces; hence, it was only a quick implementation to provide enough evidence for the validity of the method. Figure 5.4 illustrates the different types of errors as a function of read length for the trained ANN, PHRED and ABI base-callers for trace 5.

Table 5.1. Performance measure of the trained ANN compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	PHRED	ANN	PHRED	ANN	PHRED	ANN	PHRED
Trace 1 - 639 Bases	97.65	61.19	1.56	18.47	0.16	0.78	0.63	19.56
Trace 2 - 623 Bases	96.15	71.43	1.12	4.01	1.61	0.32	1.12	24.24
Trace 3 - 632 Bases	97.63	97.63	0.79	0.16	0.95	0.16	0.63	2.06
Trace 4 - 632 Bases	97.47	98.73	1.11	0	0.16	0	1.27	1.27
Trace 5 - 722 Bases	97.23	97.51	0.83	0.14	0.83	0.69	1.11	1.66
Trace 6 - 722 Bases	98.20	88.92	0.28	8.73	0.83	0.69	0.69	1.66

Table 5.2. Performance measure of the trained ANN compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	ABI	ANN	ABI	ANN	ABI	ANN	ABI
Trace 1 - 639 Bases	97.65	99.21	1.56	0.16	0.16	0.63	0.63	0
Trace 2 - 623 Bases	96.15	97.75	1.12	0.16	1.61	0.16	1.12	1.93
Trace 3 - 632 Bases	97.63	99.68	0.79	0	0.95	0	0.63	0.32
Trace 4 - 632 Bases	97.47	99.37	1.11	0.16	0.16	0.32	1.27	0.16
Trace 5 - 722 Bases	97.23	98.75	0.83	0.55	0.83	0.14	1.11	0.55
Trace 6 -722 Bases	98.20	99.45	0.28	0.14	0.83	0.28	0.69	0.14

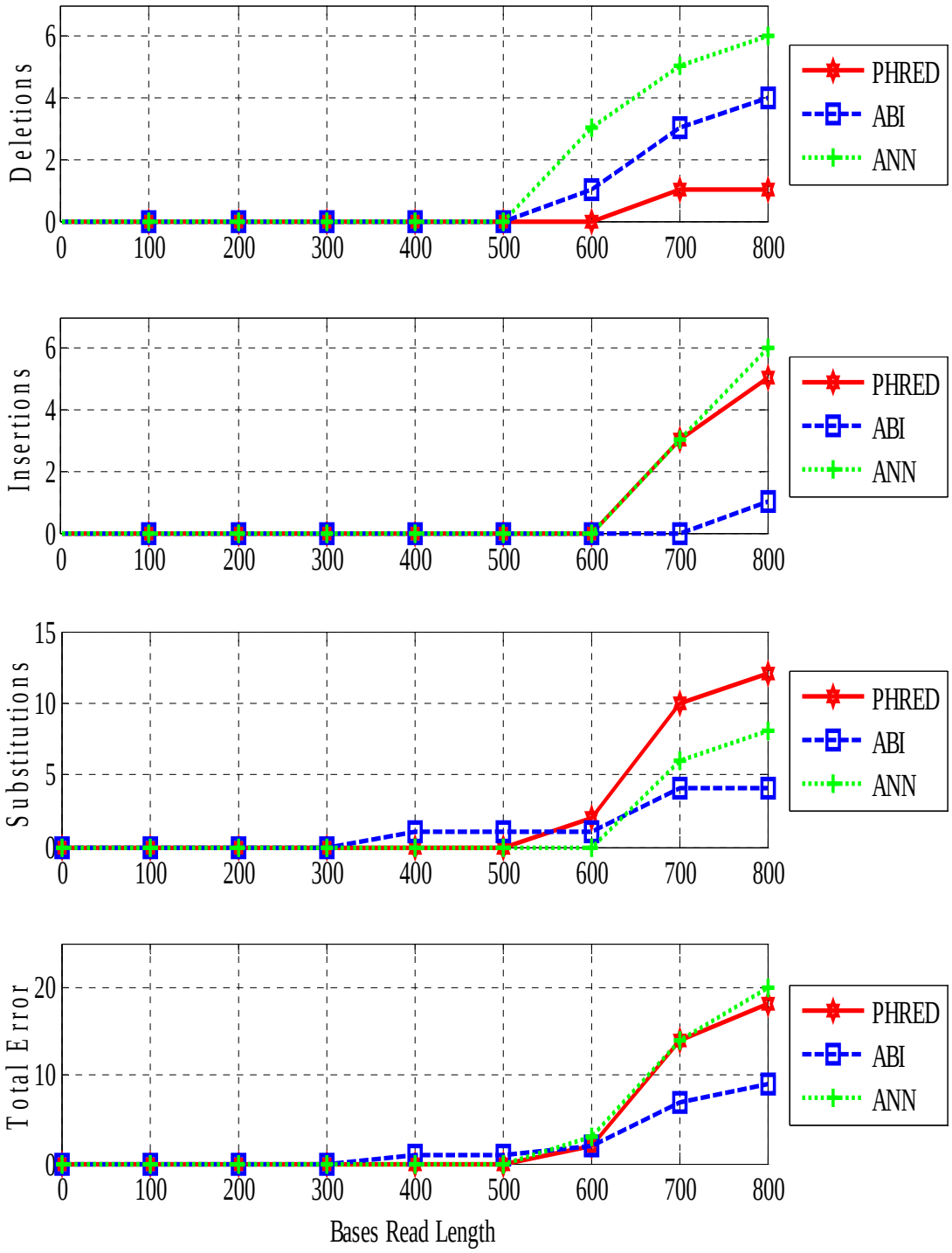


Figure 5.4. The performance of PHRED, ABI and the proposed ANN as a function of read length.

5.3.1.2. Evaluation of performance of PCs

Table 5.3 and Table 5.4 illustrate the error rates attained on testing the trained polynomial classifier on data set one. An overall average accuracy of 97.9% is achieved. The substitution error of PC model compared to that of PHRED is significantly lower. As apparent from the tables, PC overcomes PHRED in its performance in most of the testing cases. However, the performance of PC is observed to be comparable to that of ABI. The errors observed in read lengths lower than 500 bases is negligible when testing the PC model and comparing its results to that of ANN. Moreover, the overall efficiency attained by PC is higher than that attained by ANN. Figure 5.5 shows the performance of PC model in comparison to that of PHRED and ABI for electropherogram number 5.

Table 5.3. Performance measure of the trained PC compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	PHRED	PC	PHRED	PC	PHRED	PC	PHRED
Trace 1 - 639 Bases	98.6	61.19	0.63	18.47	0	0.78	0.78	19.56
Trace 2 - 623 Bases	95.83	71.43	0.64	4.01	1.12	0.32	2.41	24.24
Trace 3 - 632 Bases	98.1	97.63	0.79	0.16	0.32	0.16	0.79	2.06
Trace 4 - 632 Bases	98.26	98.73	0.47	0	0.32	0	0.95	1.27
Trace 5 - 722 Bases	98.06	97.51	0.83	0.14	0.42	0.69	0.69	1.66
Trace 6 - 722 Bases	98.34	88.92	0.28	8.73	0.55	0.69	0.83	1.66

Table 5.4. Performance measure of the trained PC compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	ABI	PC	ABI	PC	ABI	PC	ABI
Trace 1 - 639 Bases	98.6	99.21	0.63	0.16	0	0.63	0.78	0
Trace 2 - 623 Bases	95.83	97.75	0.64	0.16	1.12	0.16	2.41	1.93
Trace 3 - 632 Bases	98.1	99.68	0.79	0	0.32	0	0.79	0.32
Trace 4 - 632 Bases	98.26	99.37	0.47	0.16	0.32	0.32	0.95	0.16
Trace 5 - 722 Bases	98.06	98.75	0.83	0.55	0.42	0.14	0.69	0.55
Trace 6 -722 Bases	98.34	99.45	0.28	0.14	0.55	0.28	0.83	0.14

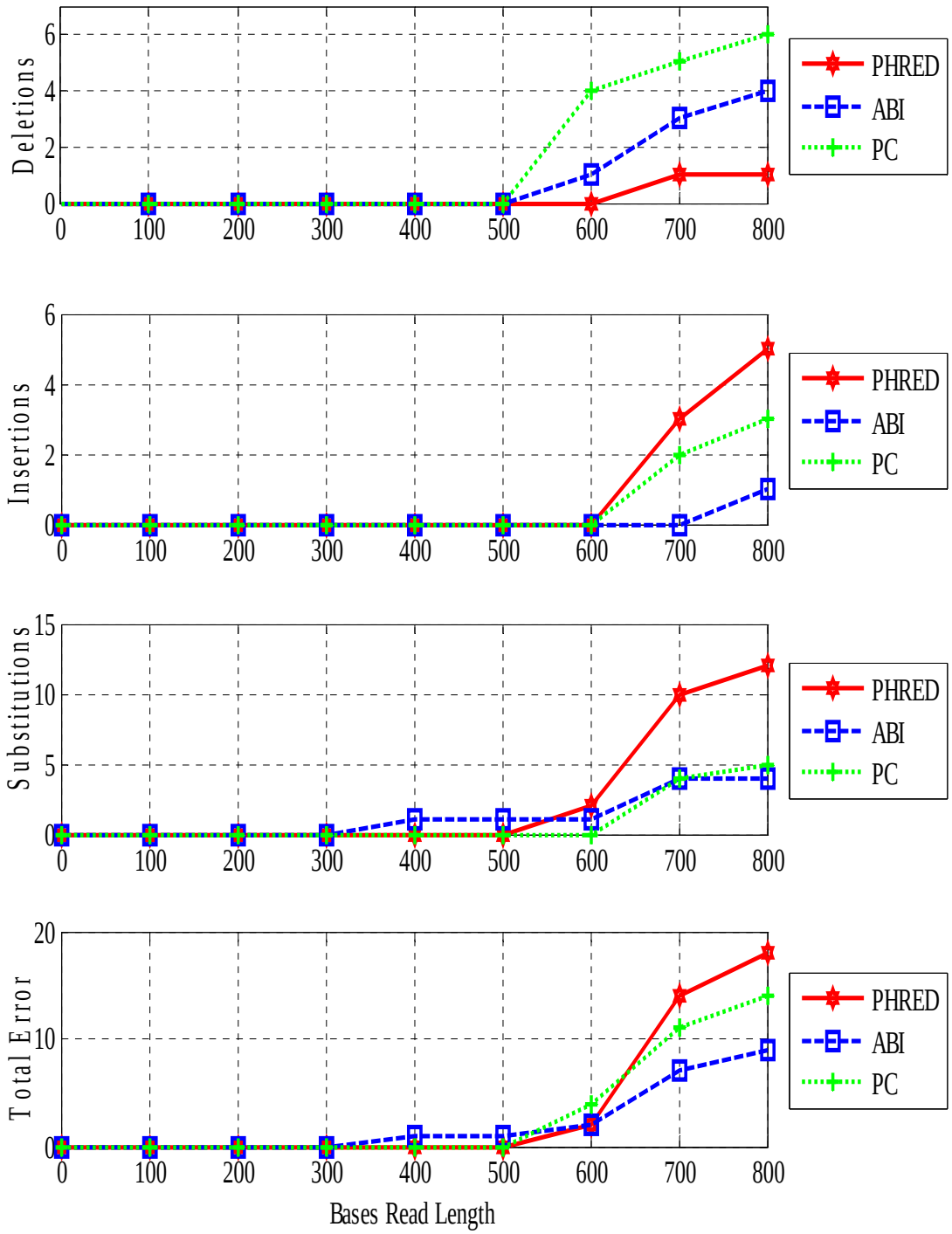


Figure 5.5. The performance of PHRED, ABI and the proposed PC as a function of read length.

5.3.2 Data set two

To test the applicability of our approach on various organisms; traces from three species: *Homo sapiens* mitochondrion D-loop chromosome, *Saccharomyces mikatae* (yeast) and *Drosophila melanogaster* (fruit fly), were included in data set two. The data set consists of 11 chromatogram traces obtained from the NCBI trace archive and SMGF. From the database, 8 traces are used for training while 3 traces were chosen for testing the trained model. It should be noted that the training set did not include any trace from *Drosophila melanogaster* (fruit fly) so that the generalization of the model to new organisms could be tested. On an average, the electropherograms consist of 650 to 750 bases per trace.

5.3.2.1. Evaluation of performance of ANNs

The performance of the ANN model trained using the training set is measured in terms of correctly called bases obtained, insertion errors, deletion errors and substitution errors. Table 5.5 and Table 5.6 compare the performance of the trained ANN model to that of PHRED and ABI, respectively.

As observed from the tables, on average, an accuracy of 98.27% is achieved on testing the trained ANN model using data obtained from the three different organisms. The performance of the model did not change significantly although the organisms, the source of the data and the amount of noise varied among the traces. It should be noted that by varying the source of chromatogram traces, dynamic decay along the trace varies since different sources use different concentrations of chemicals in DNA sequencing. Moreover, even though the model did not encounter data obtained from *Drosophila melanogaster* (fruit fly) while training, the total performance of ANN when tested on this novel species is better than the ABI base-caller proving the generalization of our model. A close examination of the errors in the other two traces shows that ANN performance is comparable to that of both ABI and PHRED base-callers. The performance of the model in terms of insertion and substitution errors is satisfactory, but its effectiveness in terms of deletion errors requires improvement. Figure 5.6 illustrates the different types of errors as a function of read length for PHRED, ABI and ANN base-callers, for the trace that belongs to *Drosophoila melanogaster* (fruit fly).

Table 5.5. Performance measure of the trained ANN compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	PHRED	ANN	PHRED	ANN	PHRED	ANN	PHRED
Homo sapiens - 639 Bases	97.81	61.19	0.47	18.47	0.94	0.78	0.78	19.56
Yeast - 674 Bases	99.41	99.41	0.59	0.15	0	0.15	0	0.30
Fruit Fly - 744Bases	97.58	99.33	2.15	0.67	0	0	0.27	0

Table 5.6. Performance measure of the trained ANN compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	ABI	ANN	ABI	ANN	ABI	ANN	ABI
Homo sapiens - 639 Bases	97.81	99.21	0.47	0.16	0.94	0.63	0.78	0
Yeast - 674 Bases	99.41	99.41	0.59	0.15	0	0.15	0	0.30
Fruit Fly - 744 Bases	97.58	97.45	2.15	1.75	0	0.27	0.27	0.54

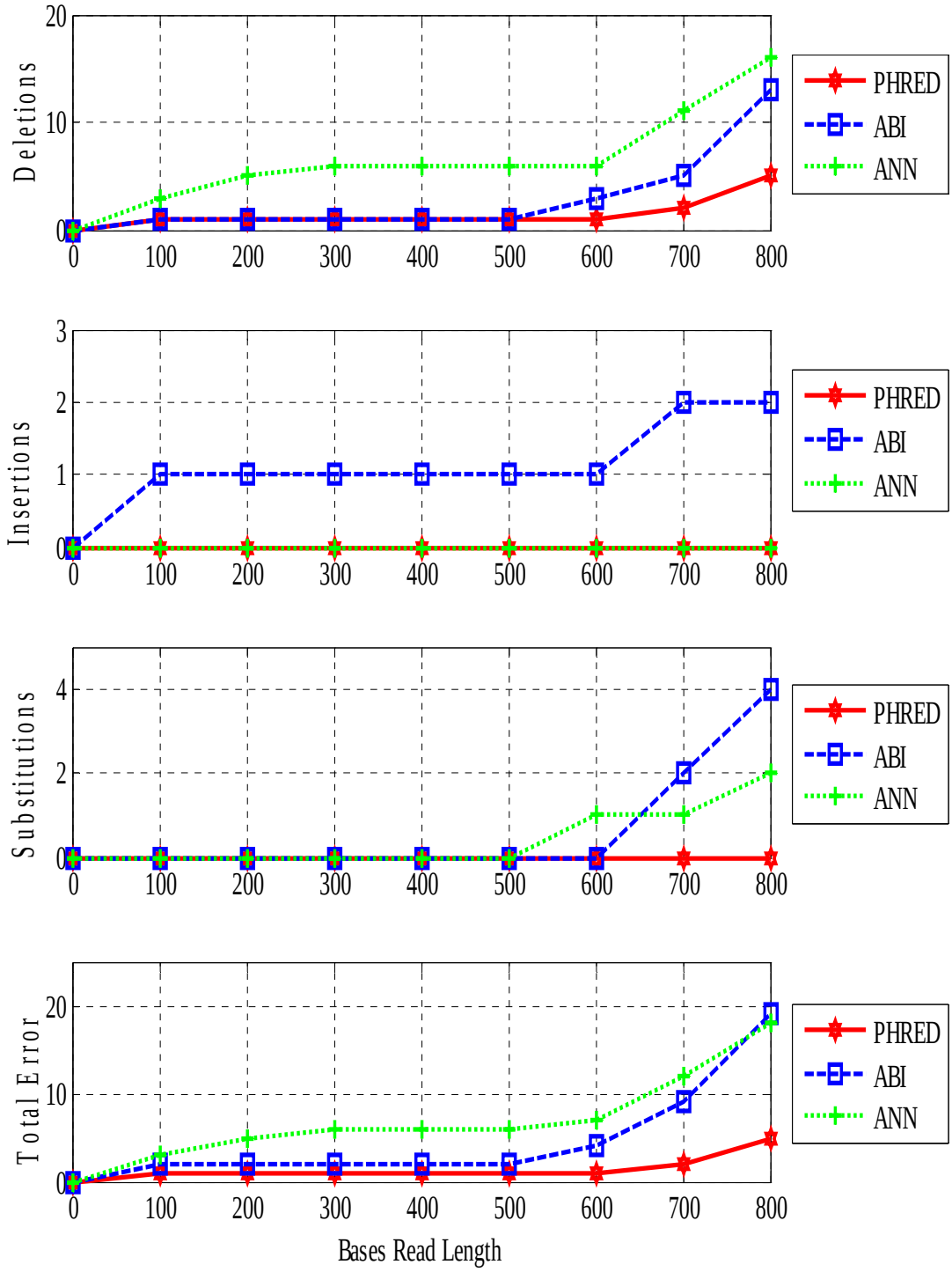


Figure 5.6. The performance of PHRED, ABI and the proposed ANN as a function of read length.

5.3.2.2. Evaluation of performance of PCs

The performance of the trained PC model compared to PHRED and ABI base-callers in terms of deletion, insertion and substitution errors is illustrated in Table 5.7 and Table 5.8, respectively. As observed from the tables, on average, an accuracy of 98.53% is achieved through testing the trained model with novel data. A close examination of the performance achieved, PCs are observed to attain comparable classification rates to that of the other two base-callers. The performance of the model remained consistent irrespective of the specie the electropherogram belongs to. The trained model's generalization to unseen data was observed to be better than that of ANN as evident from the PC's higher recognition rate in base-calling *Drosophila melanogaster* (fruit fly).

Figure 5.7 exemplifies the three types of errors used as a measure of the performance of the base-callers: PHRED, ABI and PC for the trace that belongs to *Drosophoila melanogaster* (fruit fly).

Table 5.7. Performance measure of the trained PC compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	PHRED	PC	PHRED	PC	PHRED	PC	PHRED
Homo sapiens - 639 Bases	98.12	61.19	0.63	18.47	0.94	0.78	0.31	19.56
Yeast - 674 Bases	98.81	99.41	0.59	0.15	0.59	0.15	0	0.30
Fruit Fly - 744Bases	98.66	99.33	0.81	0.67	0.27	0	0.27	0

Table 5.8. Performance measure of the trained PC compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	ABI	PC	ABI	PC	ABI	PC	ABI
Homo sapiens - 639 Bases	98.12	99.21	0.63	0.16	0.94	0.63	0.31	0
Yeast - 674 Bases	98.81	99.41	0.59	0.15	0.59	0.15	0	0.30
Fruit Fly - 744 Bases	98.66	97.45	0.81	1.75	0.27	0.27	0.27	0.54

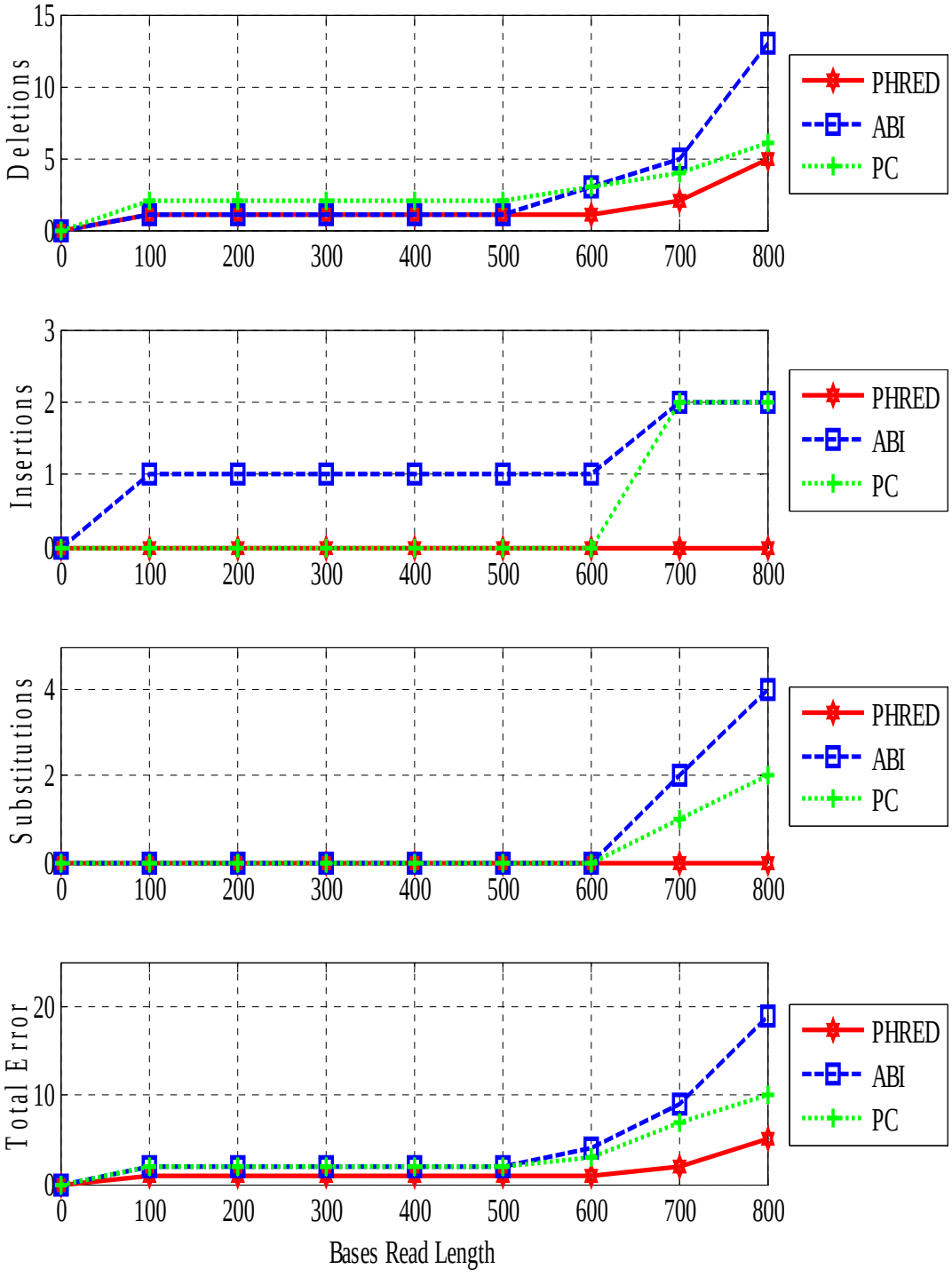


Figure 5.7. The performance of PHRED, ABI and the proposed PC as a function of read length.

5.3.3 Data set three

One of the main difficulties faced in this thesis is acquiring noisy traces with more than 750 bases so that a more representative performance measure of the model can be obtained. NCBI trace archive does provide long traces but since the noise contamination of the chromatogram traces beyond 600 bases is high, it was practically impossible to determine the consensus sequences of such traces. To demonstrate the potential of the model when tested using long traces, traces along with their reference sequence were attained from SMGF. The chromatogram traces belonged to *Homo sapiens* mitochondrion D-loop chromosome. The ANN and PC model obtained on training the entire first data set was used and its efficiency was measured using five different chromatogram traces consisting of 750-850 bases on an average.

5.3.3.1. Evaluation of performance of ANNs

On average, an accuracy of 99.55% is achieved on testing the neural network model trained using data set one. This can be observed from Table 5.9 and Table 5.10 which compare the performance of ANN model to that of PHRED and ABI base-callers, respectively. ABI achieves the highest base recognition when judged against PHRED and the proposed ANN. However, the performance of the neural network model was comparable to the existing base-callers irrespective of the length of the data. A better system can be developed by solving the comparatively high rate of deletion errors that are obtained using ANN. Figure 5.8 illustrates the different types of errors as a function of read length for the trained ANN model, PHRED and ABI base-callers for the third trace contained in the data set.

Table 5.9. Performance measure of the trained ANN compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	PHRED	ANN	PHRED	ANN	PHRED	ANN	PHRED
Trace 1 - 759 Bases	99.87	100	0.13	0	0	0	0	0
Trace 2 - 882 Bases	99.66	99.89	0.23	0	0	0	0.11	0.11
Trace 3 - 866 Bases	99.31	99.53	0.46	0.12	0.23	0	0	0.35
Trace 4 - 740 Bases	99.19	99.73	0.68	0.27	0.14	0	0	0
Trace 5 - 710 Bases	99.72	100	0.14	0	0	0	0.14	0

Table 5.10. Performance measure of the trained ANN compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	ANN	ABI	ANN	ABI	ANN	ABI	ANN	ABI
Trace 1 - 759 Bases	99.87	100	0.13	0	0	0	0	0
Trace 2 - 882 Bases	99.66	100	0.23	0	0	0	0.11	0
Trace 3 - 866 Bases	99.31	100	0.46	0	0.23	0	0	0
Trace 4 - 740 Bases	99.19	99.86	0.68	0.14	0.14	0	0	0
Trace 5 - 710 Bases	99.72	100	0.14	0	0	0	0.14	0

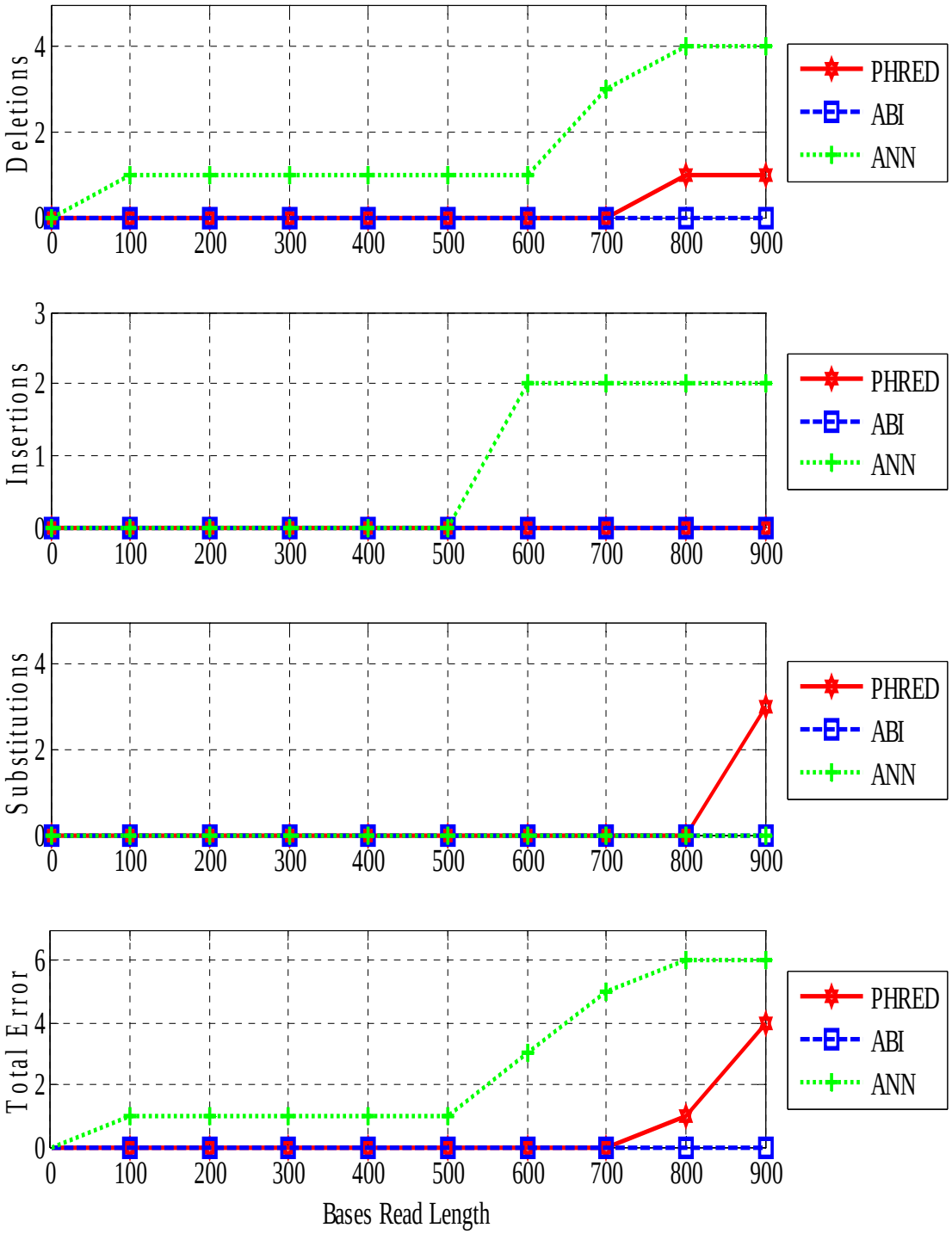


Figure 5.8. The performance of PHRED, ABI and the proposed ANN as a function of read length.

5.3.3.2. Evaluation of performance of PCs

To verify that the performance of the proposed model does not get affected as the read length increases beyond 650-750 bases, data set three was used. The PC model trained using data set one was used and an average accuracy of 99.5% was achieved as evident in Table 5.11 and Table 5.12. Table 5.11 represents the performance of the model compared to that of the PHRED base-caller while Table 5.12 compares to the ABI base-caller. The misclassifications were generally observed to occur beyond 700 bases, below which the PC model resulted in a maximum of one erroneous call. The performance of the model was observed to be comparable to the existing base-callers. Figure 5.9 demonstrates the performance of the polynomial classifier model compared to that of PHRED and ABI in terms of the various types of errors for the third electropherogram.

Table 5.11. Performance measure of the trained PC compared to PHRED.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	PHRED	PC	PHRED	PC	PHRED	PC	PHRED
Trace 1 - 759 Bases	99.74	100	0.13	0	0.13	0	0	0
Trace 2 - 882 Bases	99.55	99.89	0.23	0	0.11	0	0.11	0.11
Trace 3 - 866 Bases	99.53	99.53	0.35	0.12	0.12	0	0	0.35
Trace 4 - 740 Bases	98.78	99.73	0.68	0.27	0.27	0	0.27	0
Trace 5 - 710 Bases	99.86	100	0.14	0	0	0	0	0

Table 5.12. Performance measure of the trained PC compared to ABI.

Chromatogram Traces	Correct Recognition (%)		Deletion Errors (%)		Insertion Errors (%)		Substitution Errors (%)	
	PC	ABI	PC	ABI	PC	ABI	PC	ABI
Trace 1 - 759 Bases	99.74	100	0.13	0	0.13	0	0	0
Trace 2 - 882 Bases	99.55	100	0.23	0	0.11	0	0.11	0
Trace 3 - 866 Bases	99.53	100	0.35	0	0.12	0	0	0
Trace 4 - 740 Bases	98.78	99.86	0.68	0.14	0.27	0	0.27	0
Trace 5 - 710 Bases	99.86	100	0.14	0	0	0	0	0

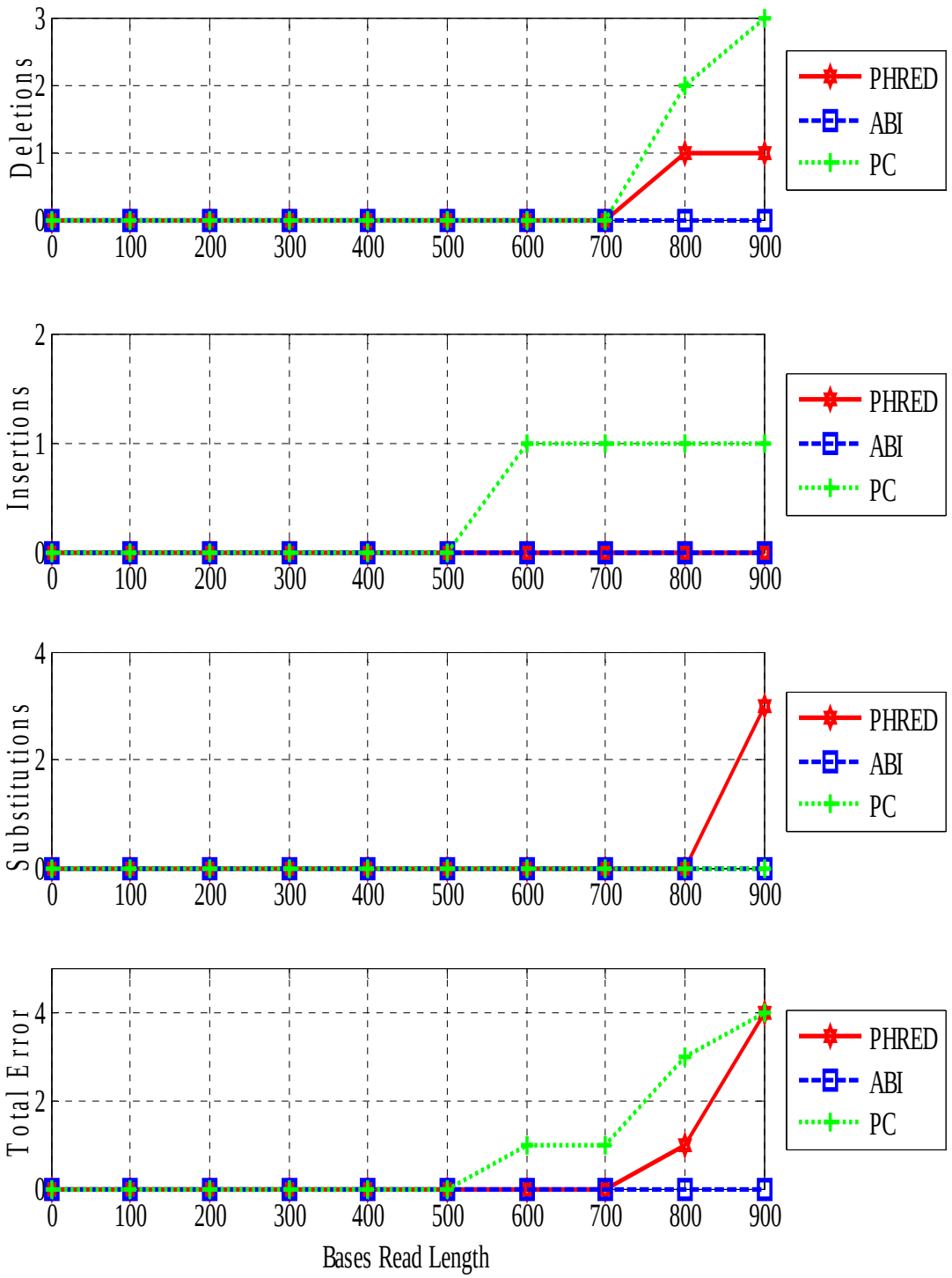


Figure 5.9. The performance of PHRED, ABI and the proposed PC as a function of read length.

5.4 Conclusions

PHRED is currently the most widely used base-caller software due to its high base-calling accuracy which exceeds that of ABI [23]. The ABI base-calling software was improved by developing the KB base-caller which incorporates base-specific quality scores similar to PHRED. ABI KB was calibrated using more than 20 million base-calls and tested on more than 10 million bases [38]. Hence justifying the high accuracy of ABI compared to the proposed models and PHRED. However, it should be noted that PHRED results in high error in some traces which already has their quality scores assigned. In such cases, PHRED makes obvious errors in perfectly clear sequences as it is apparent in Figure 5.10 which demonstrates the first trace in data set one. As observed in base number 210, an “N” is called although it clearly is a “T”.

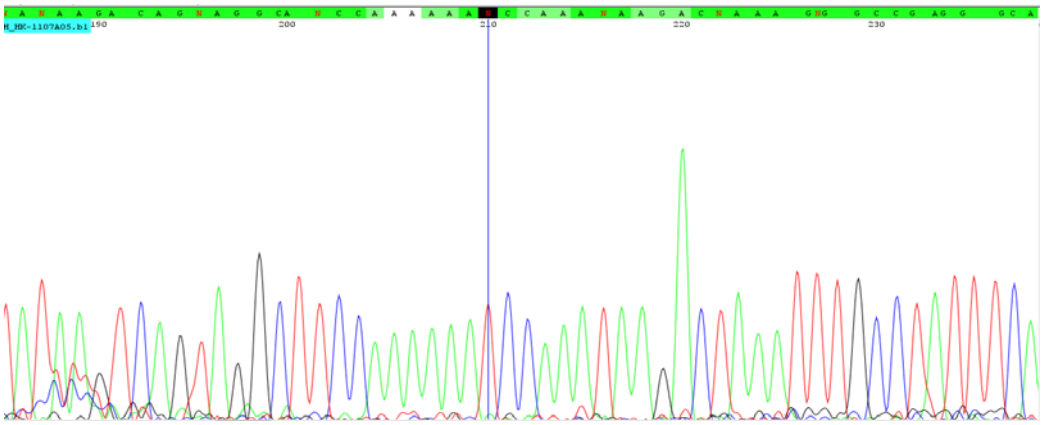


Figure 5.10. Trace one of data set one illustrating the base-calling errors in PHRED.

ABI and PHRED base-callers are both based on predicting the spacing between consecutive peaks. Moreover, in the event of failure to call a base, an “N’ is assigned. This is not so in the proposed models which does not depend on the spacing between adjacent peaks that varies dynamically as we progress through the trace. In addition, the models were designed to not assign an “N” to a peak. The base with the highest score is assigned to a peak irrespective of the noise. Moreover, the ANN and PC models have not been trained and tested in this thesis using thousands of chromatogram traces. In fact, discrete number of traces were utilized and a performance that exceeds the accuracy of PHRED and comparable to ABI was obtained. This indicates the huge potential of the proposed methods.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The problem of base-calling has not been completely solved; there is still room for improvement. In this chapter, some directions for potential future research to improve the performance of the proposed base-caller are suggested in section 6.1. Furthermore, section 6.2 summarizes the contributions of this thesis which attempts to solve the problem of base-calling by implementing a pattern recognition framework.

6.1 Future Work

Although the objective of the thesis, which is - to devise a base-caller using a pattern recognition framework - has been achieved, the subject has by no means been exhausted. Further investigation in several areas should be addressed to obtain a robust, automated, high performance base-caller.

- Data Acquisition: For a more accurate and generalized model, a larger data set is needed for training and testing the models.
- Pre-processing: To ensure that useful information is not eliminated from the chromatograms, most pre-processing operations could be removed. The models could be trained to perform the needed pre-processing internally hence, improving the performance of the base-caller.

- Feature Selection: The features used for the designed models achieved a high recognition rate. Other deterministic features could be extracted to increase the classification performance of the tested models. Moreover, in the polynomial classifier, feature selection of the higher order features could be implemented to reduce computational complexity and improve classification.
- Design a model that does not take into consideration each sample point. This would reduce the computational complexity of the base-caller.

6.2 Conclusions

Efficiently deciphering the human genome through DNA sequencing has been anticipated widely for the contribution it will make in a range of applications such as understanding the causation of genetic diseases and human evolution. However, the relatively high cost of the chemistry involved in DNA sequencing results in high operational cost in genome research centers. This fact has motivated the research on improving the accuracy of base reads in chromatograms with low signal to noise ratios so that re-sequencing of the required DNA fragment is not needed, thereby, reducing sequencing expenses.

The main objective of this thesis is to solve the problem of efficient base-calling by designing a system that:

1. Adopts a sound pattern recognition model,
2. Is capable of processing chromatogram traces that were obtained from different sequencing machines using different chemistries, and
3. Is not affected significantly by the noise inherent in the electropherograms.

The three objectives have been achieved. The data acquired from the NCBI trace archive and from SMGF were subjected to simple pre-processing operations to condition the signals without losing useful information. Pre-processing involved color correction by de-correlation, peak sharpening by de-convolution and compensation of dynamic decay by windowed normalization (Chapter 4). The pre-processed data were then subjected to feature extraction in which discriminative features are extracted to

characterize the chromatogram trace (Chapter 4). These features are then used to train and test the designed classifier models.

The models, ANN and PC, were designed such that they were not restricted to specific types of chemistries and were capable of recognizing bases up to 890 bases in long read DNA fragments deteriorated with noise. This was tested by subjecting the classifiers to three data sets to observe each model's performance when subjected to (Chapter 5):

1. Noisy chromatogram traces consisting of an average of 600-700 bases.
2. Traces belonging to three different organisms: *Homo sapiens* mitochondrion D-loop chromosome, *Saccharomyces mikatae* (yeast) and *Drosophila melanogaster* (fruit fly).
3. Electropherograms consisting of 750-850 bases on average.

In the three cases above, an overall average of 98.4% and 98.64% are achieved by ANN and PC respectively indicating the flexibility of the designed topologies. Moreover, the performance of the classifiers was compared to the currently most widely used base-calling software: ABI and PHRED in terms of deletion, insertion and substitution errors. Both proposed models achieved a higher accuracy than PHRED and a comparable performance to that of ABI. However, ABI and PHRED base-callers were designed using thousands of chromatogram traces while the models designed in this thesis used a discrete number of traces for its training and testing. This indicates the potential of the proposed classifiers as base-callers.

The proposed method has the potential of solving the problem of base-calling such that the final software developed is independent of the source of the data and the amount of noise inherent in the data. However, more work is needed to further investigate the different areas involved in the problem to build a highly efficient system that would not only solve the problem of base-calling but also encourages the usage of pattern recognition models as a tool in the field of biology.

BIBLIOGRAPHY

[1] A. G. Moat, J. W. Foster and M. P. Spector, *Microbial Physiology*, 4th ed. New York: Wiley-Liss, 2002, pp. 545-547.

[2] A. J. F. Griffiths, S. R. Wessler, R. C. Lewontin, W. M. Gelbart, D. T. Suzuki and J. H. Miller, *An Introduction to Genetic Analysis*, 8th ed. New York: W.H. Freeman, 2005, pp. 2-5.

[3] A. J. F. Griffiths, W. M. Gelbart, J. H. Miller and R. C. Lewontin, *Modern Genetic Analysis*, New York: W. H. Freeman & Co., 1999.

[4] A. D. Bates and A. Maxwell, *DNA Topology*, 2nd ed. New York: Oxford University Press, 2005, pp. 2.

[5] M. Maxam and W. Gilbert, "A New Method for Sequencing DNA," in *Proceedings of the National Academy of Sciences of the United States of America*, 1977, vol. 74, no. 2, pp. 560-564.

[6] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore and J. E. Darnell, *Molecular Cell Biology*, New York: W. H. Freeman & Co., 1999.

[7] M. J. Mcpherson, P. Quirke and G. R. Taylor, *PCR1: A Practical Approach*, UK: IRL Press, 1991, pp. 1-11.

[8] F. Sanger, S. Nicklen and A. Coulson, "DNA Sequencing with Chain Terminating Inhibitors," in *Proceedings of the National Academy of Science*, 1977, vol. 74, pp. 5463-5467.

[9] "The Science and Practice of Testing for Huntington's disease," Jan. 30, 2004 [Online]. Available: http://hopes.stanford.edu/diagnsis/gentest/f_s02gelelect.gif. [Accessed: 18th February, 2009].

[10] "Capillary Electrophoresis," in Wikipedia Encyclopedia, Dec. 2, 2004 [Online]. Available: http://en.wikipedia.org/wiki/Capillary_electrophoresis. [Accessed: 18th February, 2009].

[11] "DNA Sequencing," in *The Internet Encyclopedia of Science*, Jan. 5, 2008 [Online]. Available: http://www.daviddarling.info/images/chain_termination_ sequencing.gif. [Accessed: 15th April, 2009].

[12] M. Giddings, R. Brumley, M. Haker and L. Smith, "An Adaptive, Object Oriented Strategy for Base-calling in DNA Sequence Analysis," *Nucleic Acids Research*, vol. 21, no. 19, pp. 4530-4540, 1993.

[13] A. Berno, "A Graph Theoretic Approach to the Analysis of DNA Sequencing Data," *Genome Research*, vol. 6, no. 2, pp. 80-91, 1996.

[14] D. Brady, M. Kocic, A. Miller and B. Karger, "Maximum Likelihood Base-Calling for DNA sequencing," *IEEE Journal of Biomedical Engineering*, vol. 47, no. 9, pp. 1271-1280, 2000.

[15] M. Pereira, L. Andrade, S. El-Difrawy, B. Karger and E. Manolakos, "Statistical Learning Formulation of the DNA Base-calling Problem and its Solution Using a Bayesian EM framework," *Discrete Applied Mathematics*, vol. 104, no. 1-3, pp. 229-258, 2000.

[16] N. Haan and S. Godsill, "Sequential Methods for DNA Sequencing," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, vol. 2, pp. 1045-1048.

[17] N. Haan and S. Godsill, "Bayesian Models for DNA Sequencing," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002, vol. 4, pp. IV-4040-IV-4023.

[18] P. Boufounos, S. El-Difrawy and D. Ehrlich, “Base-calling Using Hidden Markov Models,” *Journal of the Franklin Institute*, vol. 341, pp. 23-36, 2004.

[19] D. Thornley and S. Petridis, "Machine Learning in Base-calling-Decoding Trace Peak Behavior," *Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1-8, 2006.

[20] H. Eltoukhy and A. Gamal, "Modeling and Base-Calling for DNA Sequencing-By-Synthesis," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, vol. 2, pp. II-II.

[21] K. Liang, X. Wang and D. Anastassiou, "Bayesian Base-calling for DNA Sequence Analysis Using Hidden Markov Models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 3, pp. 430-440, 2007.

[22] D. Thornley and S. Petridis, "Decoding Trace Peak Behavior-A Neuro-Fuzzy Approach," *IEEE International Fuzzy Systems Conference, Fuzz-IEEE*, 2007, pp. 1-6.

[23] B. Ewing, L. Hillier, M. C. Wendle and P. Green, "Base-calling of Automated Sequencer traces using PHRED I. Accuracy Assessment," *Genome Research*, vol. 8, pp. 175-185, 1998.

[24] P. Richterich, "Estimation of Errors in Raw DNA Sequences: A Validation Study," *Letter in Genome Research*, vol. 8, pp. 251-259, 1998.

[25] W. Brockman, P. Alvarez, S. Young, M. Garber, G. Giannoukos, W. L. Lee, C. Russ, E. S. Lander, C. Nusbaum and D. B. Jaffe, "Quality Scores and SNP Detection in Sequencing-by-Synthesis Systems," *Genome Research*, vol. 18, pp. 763-770, 2008.

[26] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. San Diego: Academic Press, 2006.

[27] T. Klassen and M. I. Heywood, "Towards the On-line Recognition of Arabic Characters," in *Proceedings of the International Joint Conference on Neural Networks, Hawaii*, 2002, vol. 2, pp. 1900-1905.

[28] S. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. New Jersey: Prentice Hall, 2009, pp. 10-22.

[29] J. P. Marques de Sa, *Pattern Recognition: Concepts, Methods, and Applications*, New York: Springer, 2001, pp. 155-159.

[30] J. Heaton, *Introduction to Neural Networks for Java*, 2nd ed. New York: Heaton Research, 2008, pp. 151-156.

[31] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley and sons, 2000.

[32] M. Khasawneh, K. Assaleh, W. Sweidan and M. Haddad, "The Application of Polynomial Discriminant Function Classifiers to Isolated Arabic Speech Recognition," *International Joint Conference on Neural Networks, Budapest, Hungary*, 2004.

[33] W. M. Campbell, K. Assaleh and C. C. Broun, "Speaker Recognition with Polynomial Classifiers," *IEEE Transactions in Speech and Audio Processing*, vol. 10, pp. 205-212, 2004.

[34] "Trace Archive," in *National Center for Biotechnology Information* [Online]. Available: http://www.ncbi.nlm.nih.gov/Traces/trace.cgi#.

[35] S. A. El-Difrawy, "A Soft Computing System for Accurate DNA Base-calling," Ph.D. dissertation, Northeastern University, 2003.

[36] X-P. Zhang and D. Allison, "Iterative Deconvolution for Automatic Base-calling of the DNA Electrophoresis time series," *Workshop on Genomic Signal Processing and Statistics*, Raleigh, NC, 2002.

[37] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*, Washington : The International Society for Optical Engineering, 2005, pp. 11.

[38] W. Brockman, P. Alvarez, S. Young, M. Garber, G. Giannoukos, W. L. Lee, C. Russ, E. S. Lander, C. Nusbaum and D. B. Jaffe, "Quality Scores and SNP Detection in Sequencing-by-Synthesis Systems," *Genome Research*, vol. 18, pp. 763-770, 2008.

VITA

Omniyah Gul Mohammed was born on May 08, 1986, in Abu Dhabi, United Arab Emirates (U.A.E). She was educated in private schools and graduated from Islamia English School in 2004. For the next four years, she did her undergraduate studies at the American University of Sharjah, U.A.E and graduated magna cum laude in 2008 with Bachelor of Science degree in Electrical Engineering. Ms. Mohammed joined the Master's program in Electrical Engineering at the American University of Sharjah and received a graduate teaching assistantship throughout her study. She was awarded the Master of Science degree in Electrical Engineering in 2010.