Optimization Based Clustering and Classification Algorithms in Analysis of Microarray Gene Expression Data Sets

Karim Mardaneh

This thesis is submitted in total fulfilment of the requirement
for the degree of Doctor of Philosophy

School of Information Technology and Mathematical Sciences
University of Ballarat
PO Box 663
University Drive, Mount Helen
Ballarat, Victoria, Australia 3353

Submitted in November 2007

Abstract

Bioinformatics and computational biology are relatively new areas that involve the use of different techniques including computer science, informatics, biochemistry, applied math and etc., to solve biological problems.

In recent years the development of new molecular genetics technologies, such as DNA microarrays led to the simultaneous measurement of expression levels of thousands and even tens of thousands of genes. Microarray gene expression technology has facilitated the study of genomic structure and investigation of biological systems. Numerical output of this technology is shown as microarray gene expression data sets. These data sets contain a very large number of genes and a relatively small number of samples and their precise analysis requires a robust and suitable computer software. Due to this, only a few existing algorithms are applicable to them, so more efficient methods for solving clustering, gene selection and classification problems of gene expression data sets are required and those methods need to be computationally applicable and less expensive. The aim of this thesis is to develop new algorithms for solving clustering, gene selection and data classification problems on gene expression data sets.

Clustering in gene expression data sets is a challenging problem. The increasing use of DNA microarray-based tumour gene expression profiles for cancer diagnosis requires more efficient methods to solve clustering problems of these profiles. Different algorithms for clustering of genes have been proposed, however few algorithms can be applied to the clustering of samples. k-means algorithm, among very few clustering algorithms is applicable to microarray gene expression data sets, however these are not efficient for solving clustering problems when the number of genes is thousands and this algorithm is very sensitive to the choice of a starting point. Additionally, when the number of clusters is relatively large, this algorithm gives local minima which can differ significantly from the global solution. Over the last several years different approaches have been proposed to improve global

search properties of k-means algorithm. One of them is the global k-means algorithm, however this algorithm is not efficient when data are sparse. In this thesis we developed a new version of the global k-means algorithm, *the modified global k-means algorithm* which is effective for solving clustering problems in gene expression data sets.

In a microarray gene expression data set, in many cases only a small fraction of genes are informative whereas most of them are non-informative and make noise. Therefore the development of gene selection algorithms that allow us to remove as many non-informative genes as possible is very important. In this thesis we developed a new *overlapping gene selection algorithm*. This algorithm is based on calculating overlaps of different genes. It considerably reduces the number of genes and is efficient in finding a subset of informative genes.

Over the last decade different approaches have been proposed to solve supervised data classification problems in gene expression data sets. In this thesis we developed a new approach which is based on the so-called max-min separability and is compared with the other approaches. The *max-min separability algorithm* is an equivalent of piecewise linear separability. An incremental algorithm is presented to compute piecewise linear functions separating two sets. This algorithm is applied along with a special gene selection algorithm.

In this thesis, all new algorithms have been tested on 10 publicly available gene expression data sets and our numerical results demonstrate the efficiency of the new algorithms that were developed in the framework of this research.

Statement of Authorship

Except where explicit reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis by which I have qualified for or been awarded another degree or diploma. No other person's work has been relied upon or used without due acknowledgment in the main text and bibliography of the thesis.

Signed: _____

Dated: _____

Karim Mardaneh
Candidate

Acknowledgements

First and foremost, I would like to thank my associate supervisor Professor Alex Rubinov for accepting me as one of his students and for all his supports. I was always stunned by his humility and other spiritual virtues. I am fortunate that I had the opportunity to know him and work with him. Thank you Alex for your care and support. I will always pray for the happiness and progress of your soul.

Words are not sufficient to express my gratitude to my principal supervisor Dr. Adil Bagirov. His respect and care towards my work, ideas, and achievement gave me courage to continue my research especially during the difficult period of my study. He always made himself available for discussion about my project and willing to answer my repeated questions.

This project would not have been started and completed without the assistance of the head of school, Professor Sid Morris. I thank Sid for his continuous support of postgraduate students.

Thank to my friends at the research centre DAFIK and specially Shahnaz Kouhbor who all the time helped me patiently through writing stage of my research.

Finally, my heartfelt appreciation goes to my mother, father, brothers and sister, for their courage, loving support, guidance, and understanding during the last three years of hard work. They have always stood behind me, encouraged me, and supported me to achieve my goals. I am always thankful for all you have done for me.

Dedication

To

my mother Maliheh for her love and understanding.

To

my father Hassan for his love and support.

Karim Mardaneh

Table of contents

List of Figures

List of Tables

List of Abbreviations

BLAST Basic local alignment search tool

cDNA Complementary DNA

DNA Deoxyribonucleotide

E.coli Escherichia coli

HGP Human Genome Project

mRNA Messenger RNA

nt Nucleotide

PCR Polymerase chain reaction

PDB Protein Data Bank

RNA Ribonucleotide

rRNA Ribosomal RNA

SAGE Serial analysis of gene expression

SBH Sequencing by hybridization

SNP Single nucleotide polymorphism

tRNA Transfer RNA

List of Publications

Journal paper

1. A. Bagirov, K. Mardaneh, *Modified global k-means Algorithm for Clustering in Gene Expression Data Sets*, In M. Boden and T. Bailey, editors, *Proceedings of the AI 2006 Workshop on Intelligent Systems of Bioinformatics WISB–2006*, Volume 73, pages 23–28. Australian Computer Society Inc, Nov 2006.

2. A. Bagirov, K. Mardaneh, *Gene selection using hyperboxes*, Submitted to "Bioinformatics". Oxford Journals. Oxford University Press.

3. A. Bagirov, K. Mardaneh, *New classification algorithms for gene expression datasets based on piecewise linear separation*, Submitted to "Current Bioinformatics".

Introduction

In recent years the development of new molecular genetics technologies, such as DNA microarrays led to the simultaneous measurement of expression levels of tens of thousands of genes. This event opened the possibility of obtaining data sets of molecular information to represent many systems of biomedical and clinical interest. DNA microarray allows to measure expression levels of thousands of genes which results in investigation of biological systems. This technology facilitates the study of genomic structure, function, and interaction related with expression levels of thousands of genes.

Microarrays can be used to find the genes with different expression levels under different experimental conditions, find genes with correlated expression patterns that show functional relationship, and classify and predict subtypes of samples with gene profiling. By identifying genes and their expression, biological systems can be investigated that will help us to understand life processes and prevent harmful diseases. For this reason, DNA microarray-based tumour gene expression profiles are increasingly used for cancer diagnosis.

Although measurement of thousands of gene expression simultaneously is very efficient, success of microarray technology depends on the precision of the measurement, effectiveness of computational tools and statistical modelling. Microarray experiments raise questions in areas like image processing, clustering, machine learning, discriminant analysis, principal component analysis, multidimensional scaling, analysis of variance models, random effects models, multiplicative models, multiple testing, models with measurement errors, models to handle missing values, mixture models, Bayesian methods and sample size and power determination [30, 83].

The increasing use of DNA microarray-based tumour gene expression profiles for cancer diagnosis requires efficient methods with high accuracy for solving clustering, gene selection and classification problems in gene expression data sets. They can help a researcher

to discover hidden relationships between tumours and genes, to discover hidden sources of different tumours, to increase the understanding of difference between normal and disease states, to identify the informative genes and to classify cancer tumours.

The aim of this study is to develop new algorithms for solving clustering, gene selection and supervised data classification problems on gene expression data sets.

Clustering in gene expression data sets is a challenging problem. Clustering deals with the problems of organisation of a collection of patterns into clusters based on similarity. It is also known as the unsupervised classification of patterns. Different algorithms for clustering of genes have been proposed.

Some of the algorithms for solving clustering problems are agglomerative and divisive hierarchical clustering algorithms, heuristics like k-means algorithms and their variations (h-means, j-means etc.), mathematical programming techniques including dynamic programming, branch and bound, cutting plane, interior point methods, the variable neighbourhood search algorithm and metaheuristics like simulated annealing, tabu search, genetic algorithms that have been applied to solve it [1, 33, 44, 48, 59–61, 115, 120, 123, 134].

However due to the large number of genes and relatively small number of samples, which leads to the sparsity of data, only a few algorithms can be applied for the clustering of samples.

The k-means algorithm is known to be very fast for solving clustering problems on large data sets, however this algorithm is very sensitive to the choice of a starting point. Another drawback of this algorithm is that when the number of clusters is relatively large it gives local minima which can be significantly different from the global one. Over the last several years different approaches have been proposed to improve global search properties of k-means algorithm and its performance on large data sets. One of them is the global k-means algorithm. The global k-means algorithm proposed in [89] is a significant improvement over the k-means algorithm. This algorithm calculates clusters incrementally and the results of numerical experiments presented, show that this algorithm locates a better solution than the k-means algorithm. However, results also show that a drawback of the global k-means algorithm is that this algorithm is not efficient in finding clusters in sparse data sets.

In this research a new version of the global k-means algorithm for solving clustering problems is developed, the modified global k-means algorithm, which is effective for solving clustering problems in gene expression data sets. This algorithm calculates clusters

incrementally and has better global search properties. Computational results are presented using gene expression data sets.

The **gene selection approach** deals with the problems of selection of the most informative genes in a data set. Gene selection is an important step for tumour classification in gene expression data sets. There are many reasons for the selection of a minimal subset of genes. Gene expression data sets contain thousands and even tens of thousands of genes. Many of them generate noise and do not provide any information about cancer tumours. Only very few genes are informative in distinguishing different types of cancer tumours and normal tissues. Therefore the identification of the most informative genes is very important. Identification of the informative genes is beneficial in that those genes may reveal insights into the biological process [69].

Large numbers of genes increase computational complexity and many classifiers cannot deal with such a huge number of genes, which leads to the loss of valuable information. In general, in gene expression data sets the number of genes is two or even three orders of magnitude more than the number of tumours, that is we have too sparse a set of points in very high dimensional space. This circumstance worsens the generalisation capabilities of many classification algorithms. The gene selection algorithms may allow to find a subset of genes which might help clarify how cancer is developing [69].

Therefore the development of gene selection algorithms that allow us to remove as many non-informative genes as possible is very important. Different algorithms can be used for finding the subset of most informative genes [3], [40, 132].

In this research a new gene selection algorithm for gene expression data sets is developed. This algorithm essentially uses the overlaps for gene expression between different classes. The proposed algorithm is compared with two other gene selection algorithms, using results of numerical experiments. Results show that the algorithm works more efficiently than the other algorithms for finding the most informative genes.

Supervised classification of new cancer tumours is very important, however it cannot be done efficiently in whole gene expression data sets due to the very large number of genes and relatively small number of tumour samples. Therefore most conventional classification algorithms cannot be directly applied to these data sets. Over the last decade different approaches have been proposed to solve supervised data classification problems in gene expression data sets [7, 12, 27]. In this thesis, a new classification algorithm based

on a combination of max-min separability and overlapping gene selection algorithms is developed.

The concept of max-min separability was introduced in [13]. In this approach two sets are separated using a piecewise linear function. Because a piecewise linear can be represented as a max-min of linear functions, such a separability is called max-min separability. Max-min separability is an equivalent of piecewise linear separability. The max-min separability algorithm is applied to solve supervised data classification problems in microarray gene expression data sets.

This thesis consists of 4 chapters. Chapter 1 presents an overview of molecular biology including cells, chromosomes, DNA, RNA, amino acids, proteins and genes followed by genetic engineering. In regards to microarray, its history, types, technology and applications are discussed, followed by microarray data analysis and microarray gene expression.

Chapter 2 presents the main concepts of data mining, in particular clustering. In this chapter k-means and global k-means algorithms are described first and the modified global k-means algorithm that is developed through this research is discussed later. We demonstrate the numerical results of application of the algorithm over 10 microarray gene expression data sets and the conclusion part concludes the chapter.

Chapter 3 presents another main concept of data mining which is gene selection. We describe some gene selection algorithms such as one-dimensional overlaps, multi-dimensional overlaps and gene selection with multi-dimensional overlaps. The new gene selection algorithm that is developed through this research is discussed as well. This algorithm uses the overlaps for gene expression between different classes. We demonstrate the numerical results of application of the algorithm over 10 microarray gene expression data sets and the conclusion summarises the content of the chapter.

Chapter 4 presents supervised classification which is another main concept of data mining. We will describe supervised microarray data analysis and max-min separability concept. Later in the chapter we will discuss the new classification algorithm that is developed through this research. This algorithm is an incremental algorithm for computing a piecewise linear function separating two sets. We demonstrate the numerical results of application of the algorithm over 10 microarray gene expression data sets and the conclusion summarises the contect of the chapter.

The conclusion section discusses the possibility of the future optimization based re-

search for clustering, gene selection and classification of the microarray gene expression data sets and outlines the contribution made by this thesis.

Chapter 1

Background and Literature Review

Microarray is an ordered array of microscopic elements on a planar substrate that allows the specific binding of *genes* or gene products. Understanding microarray and *microarray gene expression*, requires some knowledge in biochemistry, and for this reason we will briefly consider some aspects of *bio-molecules* such as cells, chromosomes, *deoxyribonucleotide (DNA)* within the chapter. Gene expression is the cellular process by which genetic information flows from gene to *messenger ribonucleic acid (mRNA)* to protein. The necessity of creating huge databases brought some disciplines together and *bioinformatics* was born. This chapter starts with definitions of the main concepts of bioinformatics that are relevant to this research. Later some definitions and descriptions of bio-molecules will be presented followed by microarray, microarray analysis and microarray gene expression.

1.1 Bioinformatics

The purpose of this section is to provide a brief overview of the main concepts of bioinformatics that are relevant to this research.

Bioinformatics as a discipline tries to predict biological functions using only *sequence data*. Most predictions in bioinformatics are made by comparing the unknown sequence against the biological knowledge base [4].

In bioinformatics, scientists of biological and computational sciences and contributors from other disciplines work together to understand the biological processes [5]. Some activities of bioinformatics includes the following:

1. **Creation and maintenance of databases**

   The size and complexity of the data have led to the creation of relational databases to store and organise the data. At the moment, *DNA sequences* comprise the majority of these data. A DNA sequence or genetic sequence is a succession of letters representing the primary structure of a real or hypothetical *DNA* molecule or strand with the capacity to carry information. *Gen Bank*, SWISS-PROT and *PDB* are examples of databases that have been created to store the data.

2. **Analysis of sequence information**

   Analysis of sequence information might include methods for finding genes in the DNA sequence of different organisms, clustering sequences into the clusters of similar sequences, aligning similar genes and proteins, and examining the relationships [5]. In parallel with the development of large sequence databases, tools such as *Basic Local Alignment Search Tool (BLAST)* are used to search, view, and analyse the data in databases. BLAST can be used for comparing primary *biological sequence* information, such as *amino-acid sequences* of different proteins or the nucleotides of DNA sequences.

3. **Prediction of three-dimensional structure**

   Information gathered from molecules is being used to obtain a three-dimensional structure of proteins and other large molecules.

4. **Expression analysis**

   Expression analysis involves pattern analysis of *gene expression data* using data mining tools.

5. **Modelling dynamic life processes**

   As a key challenge, bioinformatics aims to develop ways of putting together the information gathered from all the diverse areas of research to facilitate understanding of the fundamental life processes.

1.1.1 Databases in bioinformatics

There are primary and secondary types of databases in bioinformatics [4]. Primary databases include original biological data like databases of DNA sequence which determines the primary nuclear sequence of a DNA molecule. Secondary databases try to add some more information to the primary databases and make them more useful for particular applications.

1.1.2 Pattern discovery

In the microarray literature data are considered as a *gene expression matrix* [5]. A $G \times p$ matrix, $X = \{x_{gi}\}$ whose G rows and p columns represent the G genes and p samples. In some experiments the p samples might represent p tissue types, cell lines, times, patients, treatments, experimental conditions, and so on. The values x_{gi} that comprise the gene expression matrix could be either the measured gene expression level for the gth gene in the ith sample, or the log of the ratio of the normalized gene expression level for the gth gene in the ith sample. If analysis wants to identify groups of genes with similar regulatory mechanisms, the columns will be considered as the variables and the rows as the observations. If the analysis wants to classify the samples according to the gene expression profiles, the rows are considered as variables and the columns as observations.

1.2 Cells

Understanding bio-molecules is crucial in bioinformatics. This section briefly describes one of the bio-molecular elements which is *cell* and its structure.

Every living thing is made up of cells [131]. A cell is a single unit or compartment, enclosed by a border, wall or membrane. Each of us is made up of 100 trillion cells. Cells carry a copy of a 'master plan'. Genes build cells and organise them to form a body. Genes in turn are made up of DNA. DNA is the biopolymeric molecule that constitutes the genetic blueprint of virtually every organism in the biosphere. Human beings are one of millions of species living on Earth. Each member of every species inherits genes from its parents which are needed to build it and make it an individual. Cells consist of many components that cooperate to make a cell work. The *nucleus* is a cell's control centre. *Mitochondria* is a

cell's membrane and it provides the energy the cell needs. Every cell has a structure called endoplasmic reticulum that makes essential substances.

1.2.1 Nucleotide and its structure

Nucleotides are biochemical building blocks that make up DNA and ribonucleic RNA [112]. Since DNA and RNA carry the genetic information, nucleotides are considered as the most important biochemical building blocks of a cell. Nucleotides have three biochemical components: base, sugar and phosphate. These three components are connected by bonds to form a nucleotide.

DNA nucleotides contain one of four different bases: adenine (A), guanine (G), cytosine (C), thymine (T). RNA nucleotides contain the same bases except thymine which is replaced by uracil (U). In other words, DNA and RNA bases are similar except that DNA contains A,G,C,T and RNA contains A,G,C and U. Chemically T and U are the same except that T has a methyl group and U has a hydrogen atom in one of the ring positions.

These five bases are divided into two chemical groups. A and G are known as purines and C, T and U are known as pyrimidines. The purines include two chemical rings and pyrimidines include single ring structure. Formation of a double helix occurs through interactions between purines and pyrimidines. A,G,C,T and U are known as bases and contain multiple nitrogen (N) atoms in the rings and are called nitrogenous base because nitrogen is present in their heterocycle. A and G contain five nitrogen atoms, C contains three and T and U contain two nitrogen atoms. See Fig 1.1. [112].

Each nucleotide has a five-carbon sugar part which is known as a pentose molecule. This name comes from the presence of five carbon atoms in the ring structure. The DNA sugar is called deoxyribose because it lacks an oxygen atom (deoxy) at the position that contains the OH group in the ribose RNA sugar.

RNA and DNA nucleotides contain a single phosphate group, which is known as nucleotide monophosphate. The nucleotide, present in enzymatically synthesized DNA or RNA, contains three phosphates and is known as a nucleotide triphosphate. Nucleotides are often written as GMP for guanosine monophosphate or dCTP for deoxycytidine triphosphate. A complete DNA nucleotide includes a deoxyribose molecule a phosphate molecule, and one of the four bases of A,G,C and T. A complete RNA nucleotide includes a phosphate, a ribose and one of the four bases of A,G,C and U. Because DNA and RNA are

located in the nucleus and a nucleotide has an acidic nature, the term nucleic acid is used as a term to describe DNA and RNA.

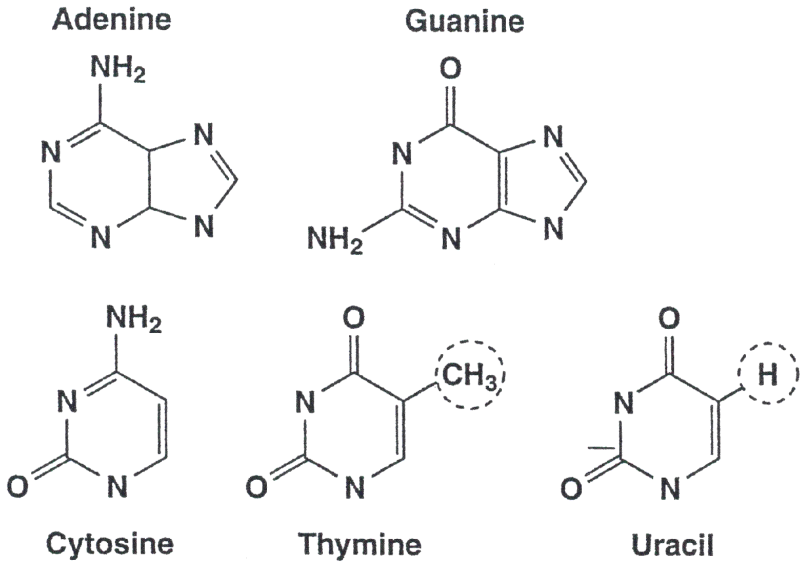

Figure 1.1: Chemical structures for the nucleotide bases.

1.2.2 Cell division

Each cell has genes, but cells do not last forever. In cell division process, each cell splits to produce two identical cells. "Nuclear division, or mitosis, divides a cell's nucleus exactly, parcelling out identical packs of gene-carrying chromosomes to each new cell" [131]. Each new cell created by mitosis has the same genes as the old ones.

1.3 Chromosomes

Chromosome is another bio-molecular element that is a long, thread-like structure inside the cell. This section briefly describes chromosomes.

In the 1880s scientists discovered long, thread-like structures inside the cells, which they named chromosomes [131]. By the early 1900s they found that chromosomes carry genes, that control the features we inherit from our parents and they contain the instructions that a cell needs to function. Chromosomes are packed inside the nucleus of a cell. They are long, very thin threads. When a cell divides to make a new cell, the long threads shorten to make chromosomes.

In 1903, Walter Sutton discovered that most cells have two sets of chromosomes. In the two sets, chromosomes come in matching pairs. Matching chromosomes have the same genes at the same positions along their length. Each species has its precise chromosome number [131], e.g. a human cell contains 23 pair of chromosomes and each pair is composed of maternal and paternal chromosomes.

1.3.1 Chromosomal alterations in cancer

Cancerous conditions are usually related to chromosomal abnormalities, specially the deletion of some parts of the chromosome and translocation in which some parts of non-homologous chromosomes are exchanged [64]. Normally deletions are related to solid tumours and translocations could be observed in *leukaemia* and *lymphomas* in which the cancerous production of *leukocytes* occurs. Leukaemia is a cancer of the blood or bone marrow and lymphoma is a variety of cancer that originates in lymphocytes.

1.4 Prokaryotes and Eukaryotes

Prokaryote cells have a single chromosome including circular double-stranded DNA. *Eukaryote* cells posses a nucleus that is separated from the rest of the cell by a membrane, and contains the gene's genetic material. This section briefly explains Prokaryote and Eukaryote cells.

Prokaryote cells: These cells have a single chromosome [64]. *E.coli* is an example of a prokaryotes chromosome. E. coli is one of the main species of bacteria living in the lower intestines of mammals, known as 'gut flora'. If this chromosome were in a linear form it would be about 1 mm long, but due to supercoiling of the DNA it has a compact structure.

Eukaryote cells: The nucleus of a eukaryote cell is separated from the rest of the cell. Nuclear DNA is organised in linear chromosomes. Higher plants, amphibia and fish, humans and other mammals are examples of eukaryote. Many higher plants, amphibia and fish have a larger genome than humans and other mammals.

1.5 DNA

This section contains a brief explanation of DNA, its discovery and structure, DNA sequencing and its applications and cDNA.

In 1869 Johan Friedrich Miescher was studying white blood cells. He isolated the nuclei of these cells and whilst analysing them he discovered a new substance that he called nuclein. This substance was later called DNA [131]. In 1944 Oswald Avery proved that DNA, not proteins, carries genes.

1.5.1 Discovery of DNA

For decades, nobody knew what comprised the code. On 28th February 1953, Francis Crick and James Watson discovered the structure of DNA, and the double helix was born [131].

"Rosalind Franklin's senior colleague, Maurice Wilkins, showed Watson and Crick-without her knowledge- Franklin's X-ray diffraction photograph of DNA." [131]. They realised that it would help them to find the structure of DNA. Eventually they completed their model of DNA successfully. DNA has been carrier of the genetic information inside Earth's living things for nearly four billion years. Within 50 years between 1953 and 2003, many secrets of life were revealed. In this period, discovery of the structure of DNA, the material from which genes are made, took place. Once the structure of DNA was known scientists found out that it contains the library of genes that controls the cells that make our body.

1.5.2 DNA molecules

Macromolecules (DNA, proteins, polysaccharides) control most of the activities of life. DNA molecules store information about the structure of the macromolecules. Each cell contains a complete copy of its genetic material in the form of DNA molecules. DNA can be copied and passed on to cells through *replication* [83]. Replication is a cellular process by which DNA is copied from a DNA template to produce an exact copy of the genome.

Genetic information is encoded in DNA by a sequence of nucleotides. The carbon atoms in deoxyribose sugar group of a nucleotide have numbers followed by a prime ($1'$, $2'$, etc). In DNA the nucleotides are connected to each other through a link of the $5'$ hydroxyl phosphate group of one pentose ring of the deoxyribose sugar to the $3'$ OH group of the

next pentose ring. Each chain makes a polarity with a $5'$ end and a $3'$ end. Two nucleotide chains are held together by hydrogen bonds between nitrogenous bases.

Millions of nucleotides link to each other to make up a DNA. "The deoxyribose and phosphate groups form a 'back bone' on the outside" [131]. 46 chromosomes inside the nucleus of a cell contain two metres of DNA in total.

1.5.3 Structure of DNA

The genetic map of every organism is stored in the molecule known as DNA [112]. Nucleotides join together to make up linear DNA sequences. Nucleotide bonds are made up of the enzymatic joining of nucleotide triphosphates, where $3'$ hydroxyl group of one nucleotide is attached to the $5'$ phosphate group of another nucleotide. A short synthetic DNA chain will contain 10-100 nucleotides, a human gene 20,000, and a human chromosome about 100,000,000 nucleotides.

A DNA chain has a different chemical group on each end of the molecule which is known as chemical polarity. The "top" end of a DNA chain contains a terminal phosphate group located on the $5'$ carbon atom of deoxyribose and is known as the $5'$ end. The "bottom" end of a DNA chain contains a terminal hydroxyl group located on the $3'$ carbon atom of deoxyribose and this end is known as the $3'$ end of the chain. [112].

Genes, chromosomes, and other DNA molecules are double-stranded. In a DNA double helix, hydrogen bonds between complementary bases hold the two helices containing the sugar and phosphate moieties together tightly. DNA chains are anti-parallel, meaning that one strand runs in the $3'$ to $5'$ direction and the other strand runs in the $5'$ to $3'$ direction. See Fig 1.2. Base pairing happens between A and T and G and C, but no other combinations of bases [112]. DNA chains that bond to each other through A-T and G-C interaction are called complementary strands. A double helix with perfect A-T and G-C base pairs is perfect match, whereas a double helix that contains one or more mismatches in the base pairing weakens a double helix. The number of different DNA sequences that can be built with four nucleotide is 4^n , where n is the number of nucleotides in the DNA chain.

Number of DNA sequences= 4^n

A total of 16 (4^2) different nucleotide DNA sequences can be built from the four bases, and if a DNA chain contains only 20 nucleotides (4^{20}), more than one trillion (1×10^{12}) different sequences are possible.

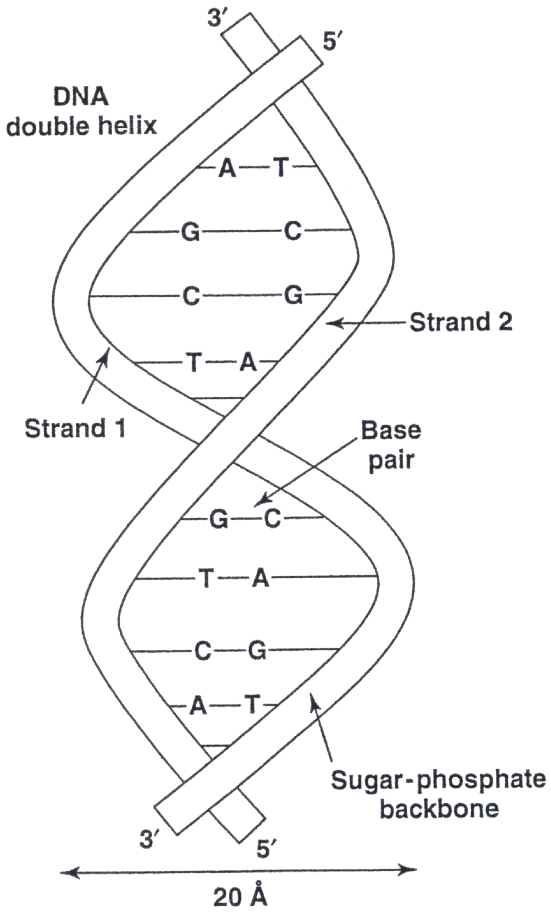

Figure 1.2: A DNA double helix.

1.5.4 DNA replication (copying DNA)

DNA is the only molecule in living things that can make a copy of itself. This happens just before mitosis in the process of cell division [131]. After cell division, the two new cells have duplicate sets of identical genes. The two strands of the DNA open up like a zip and then each unzipped piece of DNA functions like a template. Unattached nucleotides in the cell which contain one of the four bases of A,T,C,G, line up opposite their partner on the template. By linking bases to each other, a new back bone emerges and the new twin strands start to twist. This process continues along the DNA until two new double-stranded DNA are produced.

In this process when chromosomal DNA is opened to prepare a single-stranded template for replication, DNA polymerase copies both strands by storing polynucleotide chains in a $5'$ to $3'$ manner. [112]. DNA ligase closes the breaks in the newly formed chains, which produce two identical copies of DNA. See Fig 1.3.

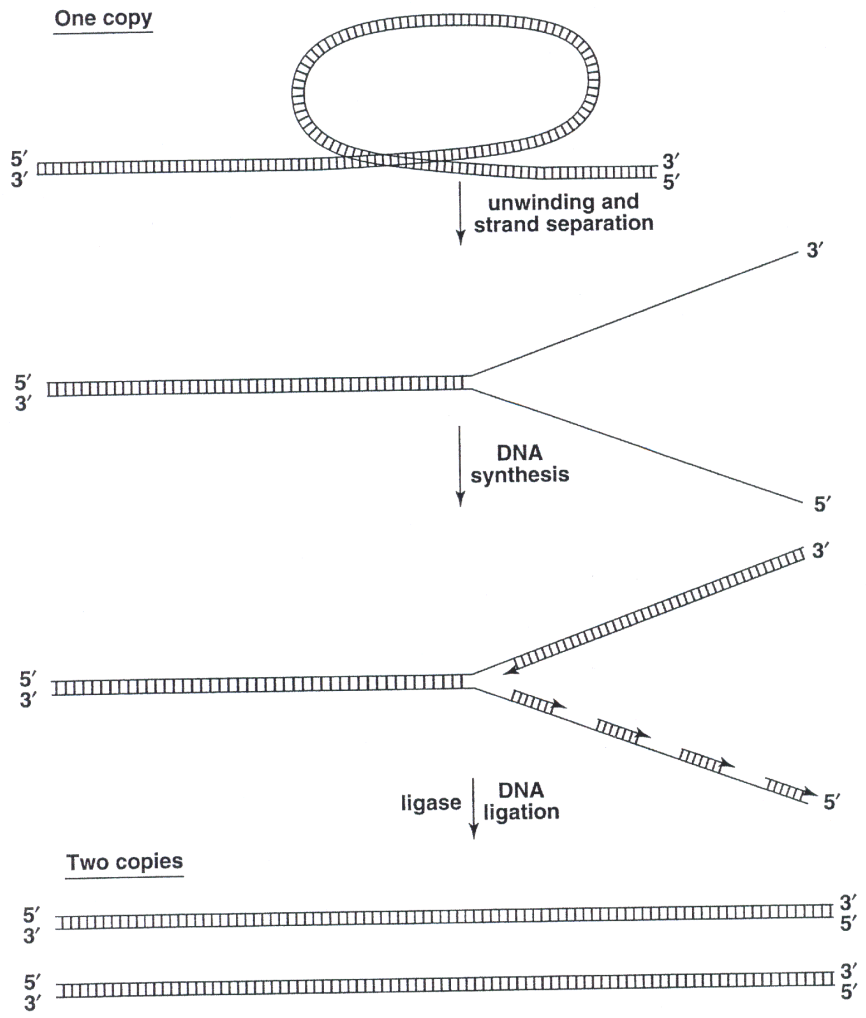

Figure 1.3: Chromosomal DNA is opened to prepare single-stranded template for replication.

1.5.5 DNA sequencing and its methods

Fredrick Sanger in the late 1940s determined the amino acid sequence of insulin, which is a protein that is used to treat diabetes [104]. The computational aspects of *protein sequencing* at that time were very similar to the computational aspects of modern *DNA sequencing*. In 1977, DNA sequencing technology was developed independently by Maxam and Gilbert at Harvard and Sanger and co-workers. Gilbert and Sanger shared the Nobel Prize in 1980 "for their contributions concerning the determination of base sequences in nucleic acids" [112].

Determination of all or a fragment of the nucleotide sequence of a DNA molecule is called DNA sequencing. Molecular biology revolution makes it possible to sequence DNA [4]. The main reasons for identifying the sequence of a DNA molecule are to predict its function and to facilitate manipulation of the molecule. *RNA sequencing* methods

were developed before DNA sequencing methods, however nowadays RNA is rarely sequenced and instead a *complementary DNA (cDNA)* copy is synthesized and sequenced. The sequence of this cDNA will show the sequence of the original RNA.

DNA information is encoded in the order of the bases (A,C,G,T). DNA sequencing determines the order or sequence of these bases in a given DNA molecule. Knowing the DNA sequence of a gene does not necessarily identify what that gene does. To do this requires much complementary information regarding the biological system that interprets the information.

DNA sequencing methods were invented in different places [104]. In 1974 Andrey Mirzabekov was visiting Walter Gilbert's lab and found a way to break DNA at A and G. Later Maxam and Gilbert found a method to break DNA at C and T and then they were able to sequence DNA.

There are two main DNA sequencing techniques (1) "Chain termination" known as "The Sanger method" and (2)"Chemical degradation" known as "Maxam and Gilbert's method" [4]. These two methods are briefly explained below.

Chain termination or Sanger method

A *primer* is an oligonucleotide that hybridizes to a complementary nucleic acid template and expedites enzymatic synthesis by providing a starting point for polymerase [64].

In this method fragments of a primer on a template of the DNA are obtained and DNA is enzymatically synthesized. The synthesis of new fragments is terminated randomly by a nucleotide that blocks further additions of nucleotides. For each of the four nucleotides, four reaction mixtures are set up.

Cells copy DNA, letter by letter, adding one base at a time [104]. Sanger found a method that replaced A,T,G,C by chemicals and this stopped a DNA's growth.

Chemical degradation or Maxam and Gilbert's method

This method is based on the base specific chemical degradation of a DNA molecule labelled at one end [4]. For DNA sequencing, single-stranded DNA is better than a double-stranded one. Although *polymerase chain reaction (PCR)* produces double-stranded DNA, it can be adapted to create a single-stranded one.

Sequencing by hybridization and DNA array technology

Hybridization is the chemical process by which two complementary DNA or RNA strands zipper up to form a double-stranded molecule [104]. *Sequencing by hybridization (SBH)* was suggested in 1988. SBH includes building a miniature DNA array or DNA chips including thousands of DNA fragments attached to a surface. Each fragment has some information about unknown DNA fragment and when this information is combined it should sequence DNA fragments. In 1991 Fodor et al. created an approach that is based on light-directed polymer synthesis. Based on this method, a company known as *Affymetrix* built the first DNA array. DNA array is now one of the most important biotechnologies and it has revolutionised medical diagnostics and functional genomics. A *probe* is a labelled molecule in solution that reacts with a complementary target molecule on the substrate. A *target* is a molecule tethered to a microarray substrate that reacts with a complementary probe molecule in solution. Every probe p in a DNA array queries a target which is an unknown DNA fragment by answering the question of whether p hybridizes with this fragment.

1.5.6 Application of DNA sequencing

The main application of DNA sequencing is to re-sequence the fragments of DNA whose sequence is known or can be predicted [4]. The sequence might be required to check that a PCR product has the expected sequence or to determine the sequence of an *allelic* variant of a known sequence. Another application of DNA sequencing is clinical diagnosis of mutations responsible for genetic diseases. The sequence variation in the abnormal allele of many disease genes has been identified. It is much easier to examine the presence of these known alleles rather than determining the complete sequence of both copies of the gene from each patient.

1.5.7 Sequence variants

A change in the primary nucleotide sequence of DNA is known as a *sequence variant* [112]. There is a variety of sequence variant. *Single nucleotide polymorphism (SNP)* is a common sequence variant containing a one-base-pair change relative to the normal gene. *Insertion* is a mutation that results in the addition of one or more nucleotides to a DNA sequence and *deletion* is a mutation that results in the removal of one or more nucleotides from a DNA

sequence. A sequence variant produces a gene variant called an allele. A human gene that has 15 different variants would contain 15 different alleles of that gene. Mutation, insertion, and deletion are three main types of sequence variant. A *mutation* might happen during the life span of an organism and a *mutagen*, which is a chemical agent, is able to change the primary DNA sequence and cause mutation that leads to uncontrolled cell growth and cancer. Cancer-causing mutagen are called *carcinogen*.

1.5.8 Methods of investigation

DNA can be obtained for investigation through three main methods [64].

1. By enzymic synthesis on a RNA template by using the enzyme reverse transcriptase to obtain cDNA.

2. By chemical synthesis to create nucleotides. This can create nucleotide as short as 50-100 (nt).

3. By restricting endonucleases for hydrolytic divisions at specific sites. DNA produced in this way is combined with the DNA of an independent replication vector, which results in clones of cells. This method is called *DNA cloning*. The artificially produced DNA is called *recombinant DNA*, because it is a result of combination of two sources.

1.5.9 cDNA

Collections of cDNAs are useful for gene expression analysis. They only have the *exon* content of a gene, not *introns*. "Microarrays of cDNAs allow profiling of mRNA levels in hybridization-based assays such that the fluorescence intensity at each cDNA location provides a quantitative measure of the corresponding mRNA" [112]. Reverse transcriptase is a DNA polymerase that is used to synthesise cDNA molecules from mRNA.

1.6 RNA

RNA is one of the biomolecules in the microarray gene expression process. This section briefly explains RNA, its structure and types and sequencing history.

1.6.1 Structure of RNA

The genetic map is encoded by DNA and by using RNA as an intermediary so that information is converted to protein information [112]. Certain classes of RNA play a role in protein synthesis. In the same manner that deoxyribonucleotides are used to make up DNA, ribonucleotides are assembled to make RNA. Most RNA molecules have 70-10,000 ribonucleotides and, unlike DNA that is double-stranded, most RNAs are single-stranded.

RNA is chemically identical to DNA but due to presence of a $2'$ hydroxyl group on ribose, it is much less stable. The enzymatic instability of RNA allows the rapid turnover of RNA molecules and dynamic changes in gene expression.

1.6.2 Types of RNA

There are three main types of RNA known as messenger RNA (mRNA), *ribosomal RNA (rRNA)*, and *transfer RNA (tRNA)* [112]. These are single-stranded, however rRNA and tRNA form double-stranded via intra-molecular hydrogen bonding.

mRNA is an informational intermediate between gene (DNA) and protein. mRNA carries the genetic information from the nucleus (location of DNA) to the cytoplasm (location of protein synthesis) [112].

Each mRNA sequence is associated with a specific gene in DNA. Approximately 35,000 genes in the human genome correspond with 35,000 different mRNA sequences, each containing a specific string of condons specified by the DNA. mRNA has 1,000- 10,000 ribonucleotides and is read by tRNA to make protein. mRNA makes up about $1 - 5\%$ of the total RNA in the cell and tRNA and rRNA make up the other $95 - 99\%$.

tRNA molecules bind to amino acids and mRNA codon and facilitate protein synthesis from mRNA templates. Each tRNA contains about 75 ribonucleotides. tRNA molecules make up about $10 - 15\%$ of the total cellular RNA.

rRNA comprises a part of protein synthesis molecule which is called *ribosome*. Ribosome is the large cytoplasmic structure that facilitates protein synthesis. rRNA facilitates protein synthesis by providing a piece of molecular scaffold that binds mRNA and tRNA to the ribosome. Each type of rRNA may contain 100-5,000 ribonucleotides. rRNA comprises $75 - 85\%$ of the total cellular RNA.

1.6.3 RNA sequencing history and chemical difference of DNA and RNA

RNA was first sequenced in 1965 with "break-read the fragments-assemble" method [104]. Pevzner, Holley and collaborators at Cornell University took seven years to determine the sequence of 77 nucleotides in tRNA. Many years later DNA sequencing was carried out by transcribing DNA to RNA and then sequencing RNA [104]. The chemical difference between DNA and RNA is that the nucleotide of DNA has deoxyribose which has one more oxygen and the nucleotide of RNA has sugar ribose [83]. Moreover, RNA has uracil (U), instead of thymine (T).

1.7 Amino Acids

Amino acids are the building blocks that make up protein. This section includes a very brief description of amino acids.

1.7.1 Amino Acids

The biochemical building blocks that make up cellular protein are called amino acids [112], of which there are twenty types. Amino acids all have the same core structure but the side chain is unique for each. Amino acids are classified in several different classes according to the chemical characteristics of the side chains. The non-polar class includes amino acids with hydrophobic side chains, including alanine (ala), valine (val), leucine (leu), isoleucine (ile), methionine (met), phenylalanine (phe), proline (pro), and tryptophan (trp).

The polar amino acids have hydrophilic side chains and include glycine (gly), serine (ser), threonine (thr), cysteine (cys), asparagines (asn), glutamine (gln), and tyrosine (tyr). The charged amino acids are hydrophilic and have side chains that carry a positive or negative charge at neutral pH and include the negatively charged aspartate (asp), and glutamate (glu), and the positively charged lysine (lys), arginine (arg) and histidine (his).

1.8 Proteins

Proteins are made up of amino acids and it is possible for them to exist in a variety of structures and types. This section will outline the various structures of proteins as well as protein arrays.

Living organisms are composed of proteins [5] which perform life's basic functions. Structural proteins form part of the cellular structure. Enzymes catalyse biochemical reactions within a cell, regulatory proteins control the activity of other proteins or expression of genes, and transport proteins carry other molecules across membranes or around the body. The DNA sequence of a gene determines a protein's sequence of amino acid. Proteins belong to a class that is called polypeptides. The variety of proteins generated by a *genome* of an organism is called its *proteome* and study of protein structure and behaviour is called *proteomics*.

1.8.1 Proteins' structure

DNA makes up the genetic map, however proteins perform the functional instructions encoded by genes [112]. Amino acids link together into protein chains to make up proteins, in the same way that nucleotides link to make up DNA or RNA. A peptide bond is an amide linkage that links amino acids to each other. This connects the carboxyl group of the first amino acid to the amino group of the second amino acid. Series of peptide bonds link hundreds or thousands of amino acids to form a protein.

A total of 20 amino acids make up proteins, whereas four nucleotides make up DNA and RNA. The number of proteins that can be built from 20 amino acids is 20 to the nth power, where n is the number of amino acids in the protein chain.

The number of different proteins=20^n

400 dipeptides (20^2) can be made by 20 amino acids. If the polypeptide contains 10 amino acids, it is possible for more than 10 trillion (1×10^{13}) different proteins to be made. A cell can create enormous numbers of proteins from 20 amino acids.

> The cellular process of protein synthesis reads mRNA codons in a successive and non-overlapping manner, such that a protein sequence is co-linear with the DNA and mRNA. A DNA sequence containing three codons ($5'$ TTT CAC GGT $3'$) would specify an mRNA containing three codons ($5'$UUU CAC

GGU3′) and a three amino acid protein (NH_3^+) phe his gly (COO^-) [112].

The two ends of a protein molecule known as amino (NH_3^+) and carboxy (COO^-) termini, contain an amino and carboxy group, respectively. See Fig 1.4.

A specific cellular gene encodes each protein, therefore the approximately 35000 genes in the human genome encode about 35,000 different proteins.

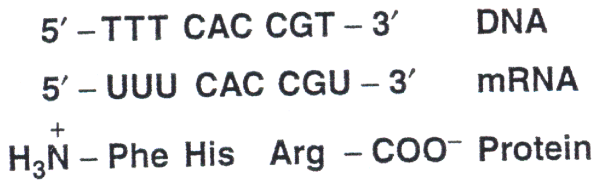

Figure 1.4: The order and identity of amino acids in proteins are the same as the codons specified in the mRNA and DNA.

1.8.2 Protein arrays

A protein array experiment can be designed similar to a DNA microarray experiment [5]. In this process a protein sample is obtained from a cell, labelled with dye and incubated with an array containing a large number of proteins bound on a glass slide. After removing any unbound protein sample, the array is scanned to measure the amount of bound sample protein. There are some differences between a protein array and a DNA microarray. The amino acid sequence and the three-dimensional structure into which the protein folds, are the main determinants of its function. Proteins cannot be printed on a two-dimensional surface to be studied as is done with DNA.

Some differences between protein arrays and DNA arrays are as follows [5]:

1. Protein arrays not only want to detect proteins but also want to measure the protein abundance, whereas most DNA microarrays want to see which genes are expressed or differentially expressed.

2. Methods like PCR, which is used in DNA microarray experiments, cannot be used in protein array experiments. It is possible to *amplify* the signal by three orders of magnitude using enzyme catalysers in single dye experiments. The detection level is important and a protein present in a sample with low concentration might not be detected.

3. Cross detection is an issue because some antigens might bind to more than one protein.

4. Protein population is much more diverse and involves many interactions, e.g. there are more than two thousand proteins in a human cell which are responsible for controlling gene expression.

Additionally, a protein array wants to study the functions of proteins. A typical experiment includes study of interaction between two proteins. The aim is to enable study of the functionality of many proteins at the same time in a single experiment.

There is an opportunity to use microarray technology to study proteins and their function [112]. This technology provides a platform for analysis of protein-protein interactions. Protein microarrays can be prepared from different sources like purified preparation, synthetic peptide and native cellular extracts. If proteins are removed from their original environment they can lose their shape.

1.9 Genes

This section briefly outlines gene's structure, gene conversion and prediction.

Our genes are what we inherit from our parents and pass on to our children, and they are the instruction set for life itself [131]. In 1861 Mendel, an Austrian monk, found that we inherit sets of instructions from each parent through a code. As body develops, it reads the code contained in the set of instructions. This set of instructions was later called genes. Mendel proved that characteristics remain separate when they are passed on from parents to the new generation. His work was unknown when he died in 1884, but scientists who were researching inheritance in the early 20th century discovered it.

Mendel's work became the basis of the science of genetics. Each set of chromosomes include between 30,000 and 40,000 genes. In every pair of chromosomes, genes appear in exactly the same order along the lengths of both chromosomes.

1.9.1 The structure of genes

Genes are continuous segments of DNA and are constructed from four nucleotide building blocks. Each gene encodes a specific mRNA and protein, and a gene is composed of DNA. An average human gene includes about 10,000 nucleotides *(nt)* and each nucleotide contains one of the four bases A,T,G and C.

Genes are made up of double-stranded DNA and they are measured by a unit that is called the base pair. In higher eukaryotes, genes contain exons and introns. An intron is a segment of a gene removed from the messenger RNA during processing and not represented in proteins, however an exon is not removed form the mRNA.

Fig 1.5 shows three hypothetical genes drawn to scale. The four exons (hatched boxes) in the human gene are separated by three introns (lines) and the yeast and bacterial genes each contain a single exon [112]. Exon is a segment of a gene that is copied to mRNA and maintained after mRNA processing. An intron is a segment of gene that is copied to mRNA and is removed from the mature mRNA before protein synthesis. Genes are illustrated from left to right. The left end of a gene is the $5'$ end, and the right end is the $3'$ end. A human gene might include 6 exons and 8 introns including 100-200 base pairs for exons and 1,000 base pairs for introns.

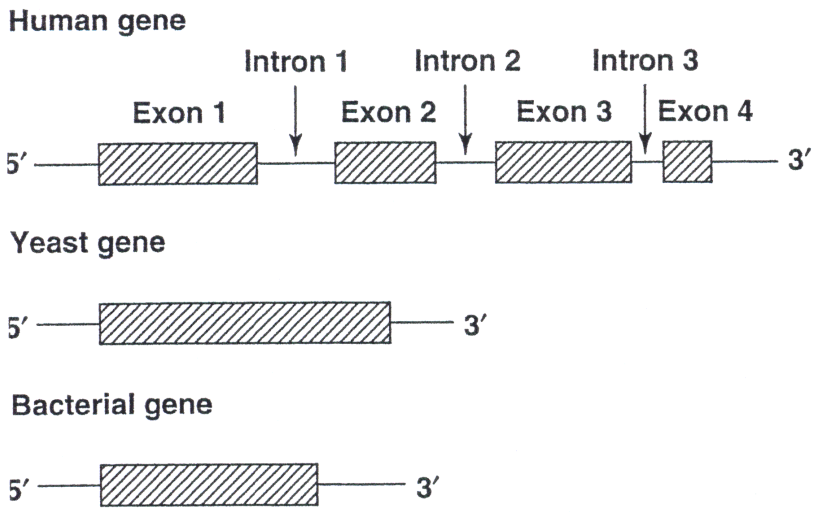

Figure 1.5: Three hypothetical genes. Exons (boxes) are separated by introns (lines).

Scientists have paid more attention to DNA sequences than to any other type.

1.9.2 Structure of prokaryotic and eukaryotic genes

"Both prokaryotic and eukaryotic genes have promoter sequences $5'$ - to the sites of initiation of transcription with a *TATA box*. This generally starts about 10 nucleotide (nt) $5'$ -to this site in prokaryotes and about 30 nt from this site in eukaryotes, though in yeast it is usually further away, at 40-100 nt" [64]. A *promoter* is an element that decides about the start site for RNA polymerase, which is an enzyme that makes mRNA from the DNA template. Many promoters have an AT-rich promoter sequence which is called TATA box.

Generally the more a TATA box differs from the normal sequence, the fewer mRNA transcripts will be made. As a result very low levels of the encoded protein will be made. Eukaryotic genes normally have a number of upstream sequences and trans-acting protein factors bind to them to regulate transcription. The untranslated sequences $5'$ - to the initiation *codon* is shorter in prokaryotic than in eukaryotic. Codon is any one of 64 three-nucleotide sequences or triplets in mRNA that specify one of the 20 amino acids used for protein synthesis.

The main difference between prokaryotic and eukaryotic genes is the presence of introns in the eukaryotic. This might be a sign of a very early evolutionary stage. Introns might have been eliminated from prokaryotic by pressures resulting in more compact genome.

1.9.3 Mutation and gene conversion

Human beings have 99.9% of genes in common [131]. It is the remaining 0.1% and its surrounding effects that make us individuals. Each chromosome in a pair of chromosomes is almost the identical image of the other. Alleles are genes that control particular features and they come in two or more versions. They produce difference between people such as eye colour. Alleles appear through mutation which can happen naturally or as a result of being exposed to radiation or harmful chemicals. Everyone has several mutations in their genes.

Changes in human DNA base sequences happen very slowly [64]. These changes are known as mutations. Organisms have the means to correct changes in DNA bases. Point mutation happens by chemical reactions that affect one of the functional groups on a single base in DNA. Highly reactive free radicals which are formed by ultra-violet radiation or X-rays can cause chemical changes in the bases of DNA and break DNA chains. The

enzymes responsible for repairing do not always act perfectly and sometimes make deletion or insertion of one or more nucleotides.

Gene conversion happens when two genes interact and as a result a part of the nucleotide sequence of one gene combines with the other. Both genes keep their location, however a non-reciprocal change happens to one of them. Gene conversion can occur between genes on different chromosomes or between genes in the same chromosome. The second situation is more likely, particularly if families of genes with repetitive and similar structure exist.

1.9.4 Gene prediction

In the 1960s it was discovered that a gene and its protein products have relations with the nucleotides in the gene and amino acids in the protein [104]. Overlapping genes and genes within genes were discovered in late 1960s when it was revealed that the computational problem of gene prediction is too complicated. Eukaryotic genomes are more complex than prokaryotic ones. Eukaryotes contain not only genes but also a large amount of DNA that does not code for any proteins. This is called "junk DNA". Most human genes are interrupted by junk DNA and are broken into exons.

In 1978 it was discovered that mammalian genes also have a split structure. When a new DNA fragment is sequenced, biologists try to find genes in this fragment. Knowing the location of a gene does not necessarily lead to the identification of the gene itself. In simple organisms like bacteria, genes are arranged in continuous DNA-like strings. In mammals, the situation is much more complicated. In a human gene that has roughly 2,000 letters, exons could be shuffled randomly into a section of DNA with a length of even a million letters. A typical human gene might have 10 or more exons, eg. the BRCA1 gene linked to breast cancer has 27 exons.

An analogy of this situation is having a magazine article that begins on page 1 then jumps to page 10 then goes to pages 41, 58,74,83,97 and so on with all pages of advertisement and other articles appearing between the pages of the article of the interest. Nobody knows yet why this happens. 97% of the human genome is advertising, which is called "junk DNA".

Prediction of a new gene in a new sequence is not easy. Many statistical methods determine which part of DNA is advertising and which part is the story. In terms of the magazine analogy, it is not expected when reading the human gene "story" to come across terms like

"for sale" or "telephone number" and so on. A combinatorial method for gene prediction uses templates of previously sequenced genes to recognize newly sequenced genes.

1.10 Genetic engineering and human genomes

Genetic code, genetic engineering and the Human Genome Project (HGP) are discussed in this section.

1.10.1 Genetic code

The *genetic code* is the cellular alphabet that specifies one of the 20 common cellular amino acids or stop codons from the 64 triplets in messenger RNA [112].

A codon or triplet includes three nucleotides that are read by cellular machinery to specify an amino acid. Four nucleotides might make 64 possible combinations (4^3).

The sequence of nucleic acid in DNA is important because it codes the sequence of amino acid in proteins [83]. The relationship between the sequence of DNA and the sequence of corresponding protein is called genetic code. The primary structure of a protein is a linear chain of building blocks called amino acids.

Genetic code is a cellular "conversion table" between codon and amino acid [112]. Of the 64 combinations, 61 are used to specify amino acids and the other 3 are known as stop codons which are genetic signals in the code that signal protein termination. [112]. See Table 1.1

1.10.2 Genetic engineering

Genetic engineering is a term for the process of manipulating genes, usually outside the organism's natural reproductive process [131].

Genetic engineering works on the organism's DNA. This happens by introducing a new gene from an organism of a totally different species. For this, scientists first select an organism with useful genes to create a desired feature. By using chemical 'scissors', they cut out the gene and insert it into the DNA of the other organism.

Recombinant DNA technology (or gene splicing) uses enzymes to cut and paste DNA into a "recombinant molecule" [112]. Since recombinant molecules spliced in the laboratories are identical clones of each other, it is also known as cloning.

Second Position (Middle)					
First Position (5′)	A	G	C	T	Third Position (3′)
A	lys	arg	thr	lle	A
	lys	arg	thr	met	G
	asn	ser	thr	lle	C
	asn	ser	thr	lle	T
G	glu	gly	ala	val	A
	glu	gly	ala	val	G
	asp	gly	ala	val	C
	asp	gly	ala	val	T
C	gln	arg	pro	leu	A
	gln	arg	pro	leu	G
	his	arg	pro	leu	C
	his	arg	pro	leu	T
T	stop	stop	ser	leu	A
	stop	trp	ser	leu	G
	tyr	cys	ser	phe	C
	try	cys	ser	phe	T

Table 1.1: The Genetic Code

1.10.3 Human Genome Project (HGP)

The genome defines the genetic construction of a cell or *genotype* [83]. Genotype is the genetic makeup of an organism, and the complete set of characteristics expressed by an organism is called its *phenotype*.

The genome of a cell consists of one or more molecules of DNA inside a chromosome. For bacteria the cell contains only one copy of the genetic material and is called haploid. Higher organisms have two copies and are called diploid.

Human genome project and its aims

The *human genome* is all the DNA in one set of chromosomes [131]. Fifty years after James Watson and Francis Crick found the structure of DNA in 1953, scientists researching the human genome succeeded in reading the sequence of bases in the DNA in human cells. A draft was published in 2000. They found the order of the“letters” A,T,C,G that compose the coded messages of genes. These coded messages determine how our body is assembled and how it works and shows if we are predisposed to suffer certain kind of diseases. The first aim of the *human genome project* is to find the precise sequence of bases in DNA that make up a genome. The second aim is to construct a complete map of genomes that will

show where the genes are located.

Human genome project importance

Sequencing of the human genome has been acclaimed as the greatest achievement in biology [101]. Completion of the human genome project is not the end of technology development related to DNA sequencing. There are many genomes to be sequenced and there are many individuals to be compared with the standard sequence.

The working draft of entire human genome, involved with the sequences of 85% to 90% of 3 billion DNA bases. This is the construction map of humans and has potential in the discovery of functional genes, distinguishing gene mutations responsible for diseases and development of methods and procedures for detection, treatment and prevention of variety of diseases. One of the objectives of human genome project is clinical use of genomic information. This might help pharmaceutical companies to produce medications compatible with the patient's genetic profile to increase efficiency and effectiveness, and to decrease the side effects.

1.10.4 Structural and functional genomics

A branch of biology that studies the structure and function of genes is called *genomics* [5]. Structural genomics is the application of sequencing technologies to create representative genome sequences for different organisms, specially humans.

Functional genomics is the study of the functions of genes to understand the behaviour of all the genes in a genome. Knowing the sequence of a gene does not mean that its function is known as well.

1.10.5 Genome analysis

Cytogenetics is the study of the structure of chromosomal material [101]. *Biophysics* is an interdisciplinary science that applies the theories and methods of physics to questions in biology. *Biochemistry* is the field of study that endeavours to understand the chemical basis of life by focusing on the study of DNA, RNA, proteins, and other biomolecules. *Molecular biology* is the study of biology at a molecular level. The field overlaps with other areas of biology and chemistry, particularly genetics and biochemistry. All these disciplines have

helped to develop genome analysis and have helped us to understand inheritance and the gene's role in an organism's phenotype. It can be said that a genome is a complete set of DNA instructions for a given organism organised into chromosomes. Genome sequencing can act as a base for analysis of transcription, gene regulation, chromosome structure, genetic pathologies and evolution.

1.11 Microarray

Microarrays may have a variety of types depending upon the samples they use and the technology upon which they are based on. They have a range of different applications. This section explores the concept of microarrays and microarray history, microarray structure and types, microarray technology and applications, microarray experiment process and nanoarrays.

1.11.1 Definition of Microarray

"Microarray is a new scientific word derived from the Greek word mikro (small) and the French word arayer (arranged)" [112].

A microarray includes an array of spots in columns and rows. An analytical device will be called a microarray if it is ordered, planar, microscopic and specific.

A Microarray is normally made of a planar and unbending substrate such as glass, plastic, or silicon, all of which are solids. Nitrocellulose and nylon filters, which were developed in 1970s and 1980s, are also considered to be solids.

1.11.2 History of Microarray

Schena and colleagues developed the microarrays at Stanford University in the 1990s [112]. The idea arose after a talk with one of his colleagues regarding the development of a new technology to study plant gene expression. He supported the idea of producing glass chips to study plant transcription factors. At the beginning it was difficult to answer the questions regarding the manufacturing of the chips, reading them, hybridizing cDNA on the glass, labelling the probes and so on. These problems were solved by Davis' laboratory and the Stanford Biochemical department. The research goal was to manufacture microarrays

containing plant gene sequences and measure plant gene expression by fluorescent labelling mRNA and hybridizing them to cDNA on the substrate. The fluorescent signals provide a quantitative measure of each gene on the substrate.

Three main approaches were considered for microarray manufacturing including photolithography, ink jetting and contact printing. Photolithography or optical lithography is a process used in semiconductor device fabrication to transfer a pattern from a photo-mask (also called reticle) to the surface of a substrate. The development of microarray technology was facilitated by cooperation of six major disciplines including biology, chemistry, physics, engineering, mathematics and computer science.

The first microarrays (printed in 1995) had 96 genes with 200-μm features, whereas the high-density microarrays manufactured in 2001 contained 30000 genes with 16μm features.

1.11.3 Microarray structure and function

A microarray is an array of microscopic elements on a substrate that allows binding of genes or gene products. Fig 1.6 shows that circles on the microarray contain target DNA molecules attached to a glass substrate (top inset). The single-stranded target molecules hybridize to fluorescent labelled probes in the solution (bottom inset). Spots glow with different intensities determined by the expression level of each gene. [112].

A microarray enables exploration of genomics. Fig 1.7 shows another figure of microarray analysis. In microarray analysis that uses mRNA as a probe, mRNA is extracted from cell and labelled with fluorescent and hybridized to the microarray containing gene sequences. Each spot on the chip links specifically with the labelled molecules in the solution and glows differently. Glowing intensities can be coded and then quantitative data can be obtained by determining the intensity level at each spot. A microarray enables scientists to examine the entire human genome in one experiment [112].

> Microarray elements are collections of target molecules that allow specific binding of probe molecules including genes and gene products and a typical printed DNA spot contains approximately one billion molecules attached to the glass substrate [112].

Targets can be extracted from whole genes or parts of genes and can appear as genomic DNA, cDNA, mRNA, protein, small molecules, tissues, or any type of molecules that allow

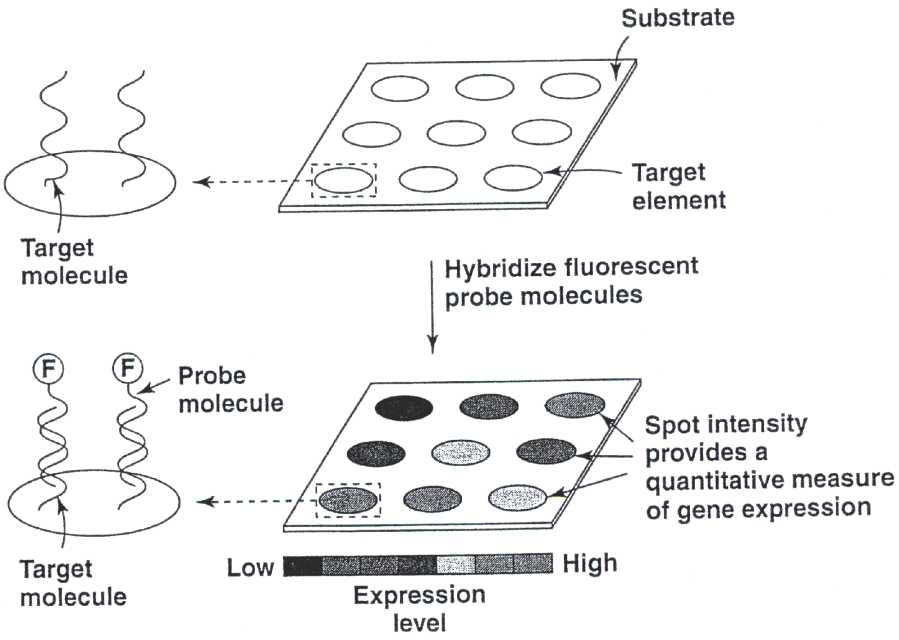

Figure 1.6: A microarray is an array of microscopic elements on a substrate that allows binding of genes or gene products.

quantitative gene analysis. Targets can be derived from a variety of sources such as cells and enzymatic reactions. Chemical synthesis provides an excellent source of target material that produces single-stranded oligonucleotide.

1.11.4 Types of microarray

Nucleic acid microarrays include DNA microarray and oligonucleotide microarrays. They contain DNA or RNA as the target material.

1. DNA microarrays

A DNA microarray consists of a solid surface, usually a microscope slide [122]. DNA molecules are bonded to the slides.

Classification of DNA microarrays is based on the type of probes they use on the array, their generation and immobilisation [100]. This will create cDNA arrays, oligonucleotide arrays and genomic arrays.

DNA microarray allows to measure expression of thousands of genes which results in understanding of complex biological systems [30]. Its high density and small size allow

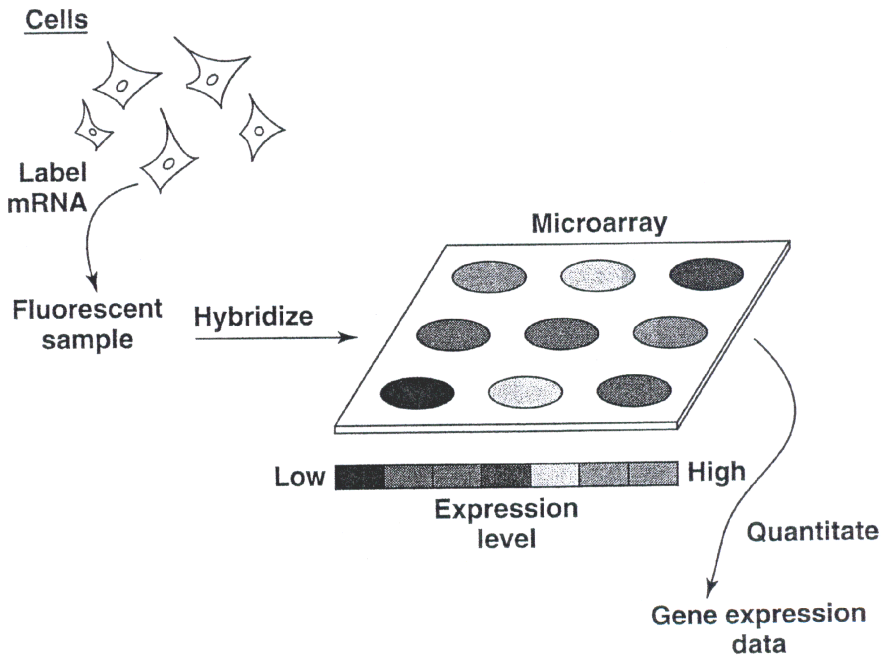

Figure 1.7: Microarrays are used to examine samples by fluorescently labelling messenger RNA (mRNA) from cells or tissues.

researchers to examine different samples in parallel. This technology facilitates the study of genomic structure, function and interaction related to the expression levels of thousands of genes.

Microarrays can be used to find the genes with different expression levels under different experimental conditions. This might include finding genes with correlated expression patterns that show a functional relationship, detecting target genes to create new research hypotheses, and classifying and predicting subtypes of samples with gene profiling.

Substrates to be used in microarrays must have some specific physical and chemical attributes [95]. A variety of techniques exist to attach probes to cDNA or oligonucleotides to substrates. Two main strategies include In-situ synthesis strategy (which is synthesis of oligonucleotides on the substrate) and attachment strategy (which is attachment of cDNA or presynthesized oligonucleotides to the substrate).

2. Oligonucleotide microarray

In this method many different, single-stranded DNA oligonucleotide of varied sequences are attached to a microarray [37]. A sample of single-stranded cellular DNA or RNA is added and allows the two single strands to form specific Watson-Crick base-pairs with

any matching oligonucleotide on the chip. If any DNA or RNA sample is labelled with a fluorescent dye, each cell-derived sample will give a pattern of fluorescent light signals on the biochip.

Oligonucleotide microarrays have been useful in measuring the amounts of mRNA present in any living cell. This enables the comparison of the relative transcription of DNA into RNA for thousands of different genes in biological samples. This comparison would be possible for example before and after giving a patient a therapeutic drug or before and after certain cells have turned cancerous.

1.11.5 Microarray technology

After genome sequencing, microarray technology has been used widely in genomic studies in biology and other corresponding sciences [83]. Microarray technology provides facilities to measure molecular biology, resulting in information for gene control, and the results of controlling gene transcriptions. Microarray technology facilitates the examination of DNA and RNA variations. Grouping genes with similar expression patterns helps to identify genes with the same function or genes that are likely to be co-regulated. By comparing gene expression in normal and abnormal cells, microarray technology can help to discover biological processes for cures.

The success of microarray technology depends on the precision of the measurement, using tools for data mining and statistical modelling. The strength of the microarray technology depends on data mining and analytical methods.

Genomics provide biologists with all the genes to be used to assemble life. Microarray technology provides high measurement in molecular biology which leads to information for the reconstruction of gene control networks.

Microarray technologies can be classified in two main groups: spotted microarray and In-situ microarrays.

1. Spotted cDNA and spotted oligonucleotide microarrays

In spotted cDNA and spotted oligonucleotide microarrays samples might be spotted onto glass slides in which laser fluorescence may be used to detect two-colour hybridization from two samples at once [28]. Alternatively they may be spotted cheaply onto filters, in which case radio-labelled material is used for hybridization of one sample at a time.

Spotted microarrays consist of a solid surface and nucleotide sequences are placed on them. Each spot represents a specific gene, an expressed sequence tag (a partial gene sequence which provides a tag for a gene); a clone (a population of identical DNA sequences) derived from cDNA libraries; or an oligonucleotide (a short sequence specifically syntyhesised for experiment). The spots act as probes against which target and reference mRNA is hybridized [28]. Probes are deposited on the array through a process called contact spotting or printing. The spotting machinery prints nucleotide spots on the array. In cDNA approach, DNA is prepared from cDNA clones.

Spotted cDNA experiments include four steps as follows [30]:

1. DNA clones with known sequences are spotted and immobilized onto a glass slide.

2. Pools of mRNA from tissue or cell population are transcribed to cDNA and labelled with one of the two fluorescence dyes (e.g.Cy3 "Green" and Cy5 "Red").

3. Two pools of mRNA are mixed and applied to a microarray with many spots. In this stage strands of cDNA will hybridize to the complementary sequences on the glass slide and any unhybridized cDNA will be washed off.

4. The microarray will be scanned and the red and green colours will be read by computer to measure the expression level of genes. The ratio of red to green signals is used for data analysis.

   Spotted oligonucleotide experiments are similar to spotted cDNA experiments, except that synthetic oligonucleotides are used as probes instead of cDNA [83].

2. In-situ synthesized oligonucleotide arrays

This method uses a combination of photolithography and solid phase oligonucleotide chemistry to synthesize short oligonucleotide probes directly on the solid support surface [83]. In this method the test and reference samples (treatment and control samples) are hybridized separately on different chips. Unlike the two previous methods, a test and a reference sample labelled with two different fluorescent dyes are simultaneously hybridized on the same array.

In in-situ synthesised oligonucleotide arrays method, oligos are built up base-by-base on the surface of the array instead of presynthesising oligonucleotides [122]. In-situ synthesised oligonucleotide arrays use *Affymetrix GeneChip Technology* for experiment [30].

The Affymetrix Gene chips is one of the most popular microarray platforms. This has some advantages over spotted arrays. The main advantage is start-up time.

Affymetrix chips are manufactured in a unique way and can be read by the special Affymetrix machine [28]. Affymetrix gene chips use oligonucleotide of 25 bases per probe. Affymetrix uses 22 probes per gene and up to 23,000 genes per chips. Each gene is identified by a collection of probes called a probe set. Multiple probe genes make up the gene (11 probe pairs per gene). Each probe pair consists of one probe called the perfect match and another called a mismatch. The perfect match has a sequence similar to the sequence of the gene of interest, however, a mismatch has a different sequence.

1.11.6 Microarrays based on the samples they use

In a classification, microarrays can be classified according to the molecule they utilise in experiments. The following classification is based on the type of samples that each of the microarrays might utilise including cDNA, oligonucleotide, tissue and protein [112].

1. cDNA microarray

The first microarray experiments were performed by complementary DNA microarrays. cDNA is a nucleic acid molecule derived from mRNA. cDNAs normally include 500 to 2,500 base pairs [112]. Approximately 65% of all the researches into microarrays includes cDNA microarray research. See Fig 1.8.

2. Oligonucleotide microarray

Oligonucleotides are single-stranded molecules including 15-70 nucleotides that are made by chemical synthesis. These targets provide a high specificity of binding and good signal strength in hybridization reactions. About 26% of all the published experiments have used oligonucleotides as target molecules in the hybridization process.

3. Tissue microarray

Tissue microarray contains parts of human tumour samples or other tissues of interest. This kind of microarray experiment comprises about 6.6% of all the published experiments. This microarray is more recent than nucleic acid microarrays.

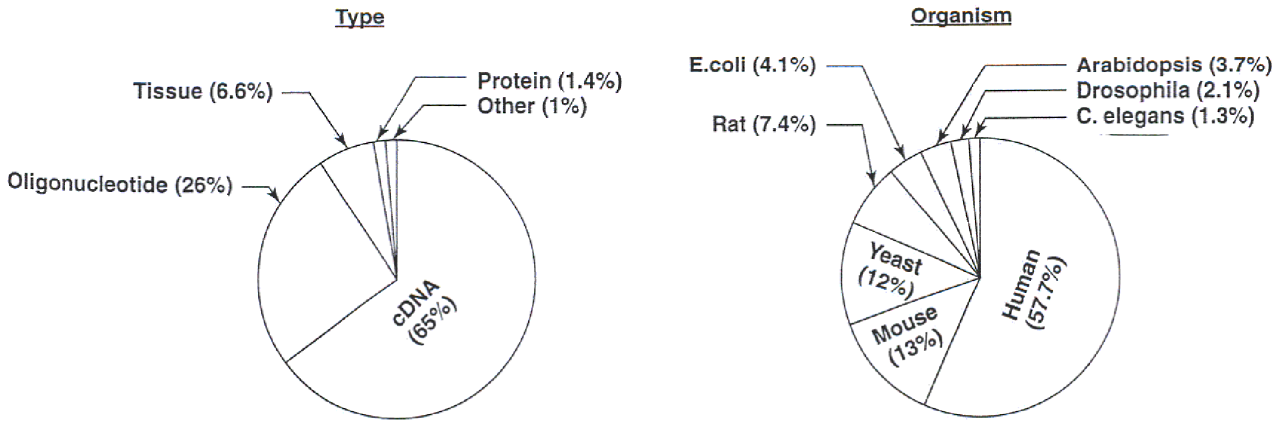

Figure 1.8: Microarray papers published since 1995, categorized according to target type and organism.

4. Protein microarray

Protein microarrays contain pure proteins or cell extracts. This kind of microarrays accounts for approximately 1.4% of all the published experiments. This microarray is more recent than nucleic acid microarrays.

1.11.7 Reasons for using arrays

There are three main reasons for using arrays. These are as follows [95]:

1. Arrays to identify patterns

The GeneChip array is an excellent "survey array" developed by Affymetrix. According to the company, the two arrays in the "Human Genome U133 Set" contain more than 1 million oligonucleotide features that enable expression observation of 39,000 varieties of 33,000 different human genes in a single sample. cDNA and oligonucleotide array of GeneChips are commonly used for expression observation of genes in diseased tissues or during treatment with a drug. Experiment with DNA arrays show that the vast majority of the genes are either not expressed or not affected by disease. Normally a pattern of gene expression or 'signature' is characterised that involves fewer than 50 genes.

2. Arrays to measure patterns

'Scan arrays' normally measure specific patterns. They are appropriate for diagnosis and drug discovery. In array-based diagnostic tests, tests occur at different sites like reference laboratories, hospital laboratories, and physicians offices. In drug discovery, after target validation, the results show how collections of chemical compounds can be tested to identify the compounds that are responsible for the desired effect.

3. Arrays for parallel processing

The experiment of adopted array formats can be found in combinatorial chemistry theory. Synthesis of chemical compound libraries has been performed in an array format. By using combinatorial chemistry, the photolithographic process which is used by Affymetrix to create its DNA chips is possible.

1.11.8 Microarray applications

Microarrays have a variety of applications. Although the first application was to monitor gene expression, these days an array of ordered bio-molecules on the chip to examine a sample biochemically is widely used [110]. In addition to gene expression, hybridization based arrays have been used for mutation detection, and other types of microarrays have been used for polymorphism analysis, mapping and evolutionary studies.

One application of microarray is in pharmaco-genomics, which is a new area in biomedicine and an interdisciplinary area involving pharmacology and genomics.

Schena has discussed some of the main microarray applications as the following [112]:

1. **Development**

   Microarrays can be used to build a database of gene expression levels resulting from the function of the cell and tissue type. This will be facilitated by examining gene expression patterns on a genomic scale. These databases can provide a profound understanding of the basic mechanisms that are responsible for controlling multicellular development, and clarify the pathological cellular events in terms of how human diseases start and progress.

2. Human disease

Genetics, diet, environment and presence of infectious agents are some of the complex set of factors that are responsible for the onset and progression of human diseases. Microarrays have a unique ability to detect each of these contributing factors. Cancer has accounted for 83% of published experiments on human diseases to date. Diabetes, cardiovascular disease, Alzheimer, stroke, AIDS, cystic fibrosis, Parkinson, autism, and anaemia are investigated using microarray analysis. As an example, by comparing gene expression patterns in brain tissues from normal individuals with those from Alzheimer patients, it should be possible to determine the genetic basis of this disease. All human illness can be studied by microarray analysis and the aim is to develop a treatment or cure for every human disease by 2050.

3. Genetic Screening and Diagnostics

Any small error in genetic code can lead to the production of faulty proteins that are not capable of functioning normally and so cause human disease. Many sequence variants that are responsible for disease are known. In a microarray screening process, patient samples are amplified by PCR, printed into microarrays and hybridized with synthetic nucleotides.

Microarrays are scanned for fluorescent intensity detection and data are represented in a two-colour image. This screening process allows normal, carrier, and diseased genotypes to be detected and distinguished. The acquisition of such information regarding genetic diseases should improve the quality of health care and reduce its costs.

Amaratunga et al. have described other applications for microarrays as the following [5]:

4. Complex diseases

There are some diseases that are caused by the combination of small genetic variations (polymorphisms). Coronary artery disease, multiple sclerosis, diabetes and

schizophrenia are complex diseases where genetic make-up plays an important role in causing disease.

5. **Tissue-specific Gene expression**

   Cells from different tissues serve different functions and the reason is as yet unknown. Since different proteins, particularly enzymes, control the biochemical reactions within a cell, a cell's functions are determined by which proteins are produced by the cell, and this in turn will depend on which genes are expressed by the cell. Microarray experiments can show which genes are expressed in which tissues. This can give scientists crucial information about mechanisms that are responsible for the functioning of cells and genes.

6. **Pharmacological agents**

   Expression levels of some genes are changed when the organism is exposed to external factors such as pharmacological agents in the environment. Microarray experiments can be used to identify genes that express differently when they are exposed to external agents.

7. **Plant breeding**

   Microarray experiments can be used to identify genes responsible for various traits of interest and determine the conditions under which these traits are expressed. This will enable scientists to create plants with a desired combination of traits.

8. **Environmental monitoring**

   It is important to assess how environmental stressors (such as combination of food, water, and air) might impact on the genome-level. Microarrays can compare and contrast gene expression patterns across affected and unaffected organisms.

1.11.9 Microarray experiment process

A microarray analysis cycle (see Fig 1.9) may include five basic steps as follows [112]:

1. **A biological question**

   A question must be formulated before starting any kind of microarray experiment. For example if the research topic is to understand the gene expression in different tissues, the research question could be formulated as "How do the patterns of gene expression compare in lung and bladder tissues?" Formulating the question will help in focusing the research, identifying potential pitfalls, selecting controls and streamlining data analysis and modelling.

2. **Sample preparation**

   This phase might include DNA and RNA extraction and purification, target synthesis, probe amplification and preparation, and microarray manufacture.

3. **Biochemical reaction**

   This is hybridization of the fluorescent sample with the microarray which leads to biochemical interactions of target and probe molecules. If the experiment uses protein microarrays, this phase will utilise protein-protein interaction rather than hybridization.

4. **Detection**

   In this stage a microarray image is created using a scanning or imaging instrument.

5. **Data analysis and modelling**

   In this stage the images are analysed and modelled. This might include quantitating data, calculating the ratios and clustering the results.

1.11.10 Nanoarrays

Microarray technology has advantages and disadvantages [95]. Microelectronics and nanoelectronics deal with electrons, however lithography prints features of tens of nanometres. The reasons that we need nanoarrays are as follows:

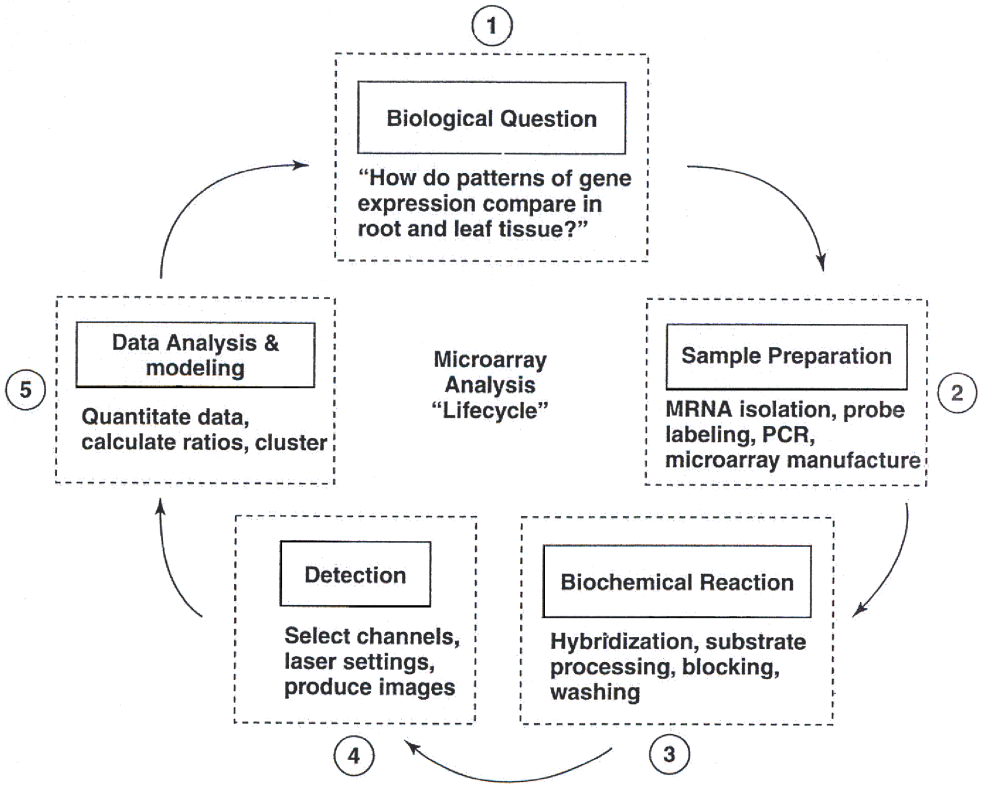

Figure 1.9: The five steps of microarray analysis cycle with specific examples of the experimental activities performed at each step.

1. **To decrease the size of the feature**

   Decreasing the size of features from one hand will increase the capability of microarrays as a result of the decrease in cost and on the other hand will increase the efficiency and reliability by increasing the number of bio-molecules in the same chip.

2. **The ability to deal with single biomolecules facilitates the imagination of different microarrays and nanoarrays**

   Current arrays are 'passive' and 'one use'. If there are 'active' and 'multiple-use' arrays then biomolecules will have different functionality, e.g., they will be calculable. Nowadays arrays are 'static' as biomolecules are probed on one location. Imagine 'dynamic' arrays in which biomolecules will be able to move across the array from side to side or circulatory, in order to perform different functions such as sensing, power generation and computation.

Normally microarrays spotted by robots contain spots as small as 100 microns with up to 10,000 different spots on a chip. In-situ synthesis is capable of producing up to 400,000 oligonucleotides on a chip using $20\mu m^2$ spots. Reducing the size of features from 20-200μm down to microns or sub-microns would massively increase the volume of genetic information that could be observed simultaneously on one chip. Achieving such high resolution will facilitate the study of binding and detection in arrays that are up to 10,000 times more complex. A decrease in feature size will provide assays in which a particular number of targets will be observable using a smaller volume of samples. This scale will facilitate high throughput and high-resolution screening tools to be developed.

1.12 Microarray analysis

The process of using microarrays for scientific exploration is called microarray analysis and even though this field has experienced a huge expansion since early 1990s, the general strategies and approaches remain the same [112]. This section will outline microarray data, processing of raw data and data analysis, statistical analysis, normalization and variance, empty and missing values and microarray analysis process.

1.12.1 Quantitative analysis

Quantitative analysis is the process of measuring the amount, number or intensity of molecules in a sample. Numerical output from microarrays enables quantitative analysis of microarray data. Accurate patient genotyping requires methods that are able to differentiate homozygotes from heterozygotes that only differ by 50% in gene concentration.

1.12.2 Microarray data

Microarray data contains two basic aspects: biological significance and statistical significance. The biological significance tells to what extent the expression of a gene is influenced by the conditions of the study [83]. The statistical significance quantifies how trustworthy the biological significance is. Because of the sources of variability in microarray experiments, the statistical analysis is vital for the interpretation of the phenomena under study.

Processing of raw data into gene expression data matrix and data analysis

Every major experiment might consist of two stages: data collection and information processing [38]. In microarray experiments, the data collection stage can be broken into five small stages. These stages will be array manufacturing, preparation of biological samples, extraction and labelling RNA, hybridization of the labelled extracts to the array, and scanning of the hybridized arrays.

The information processing stage might be broken into stages such as image quantitation (which is measuring the fluorescence intensity of spots in the array), data normalization and integration, gene expression, data mining and analysis and generating new hypotheses regarding the underlying biological processes. Digital images are considered to be raw data in microarray processing. These images are analysed and the intensity of each spot is measured. This is normally done by image analysing software. The main purpose of image analysis is to process the images, extract the data and tabulate them. The output from this process is called the spot quantitation matrix [38]. In this process images are quantified then normalized and combined. In the matrices rows normally represent measurements and columns represent genes. See Fig1.10.

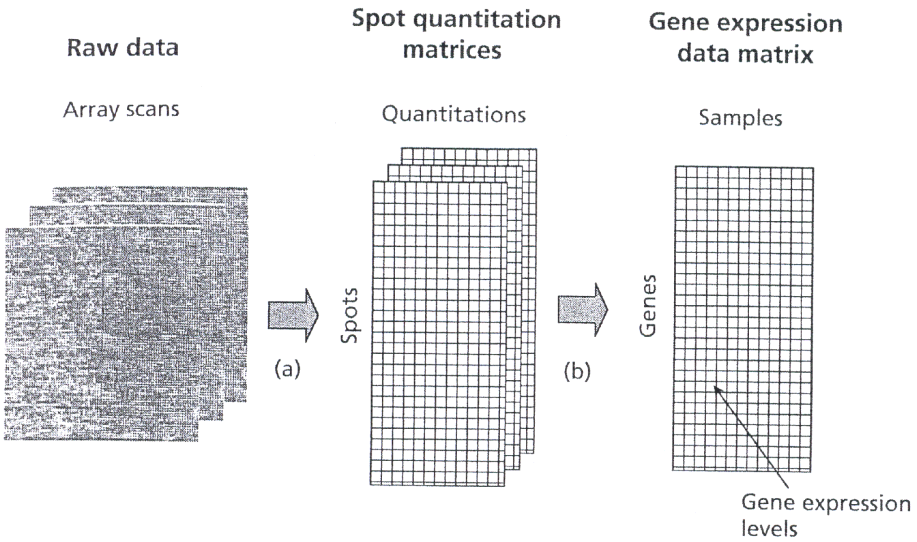

Figure 1.10: Processing of raw data into a gene expression matrix.

Generation of the spot quantitation matrices is an intermediate stage. The data must be transformed and organised in a *gene expression data matrix*. Normally after generation of a gene expression data matrix, data analysing and mining begin. The simplest way of analysing gene expression data is to identify the genes that are expressed differently in two

given samples. Data analysis starts with the hypothesis that there might be biologically relevant patterns to be discovered in the data. For example, there might be genes with different expression patterns that allow samples to be classified differently. Reverse engineering of gene regulatory networks is one of the approaches used in data analysis. This is based on the hypothesis that genes with similar expressions under different conditions might have been regulated by the same mechanism.

A gene expression database includes three major parts: the gene expression data matrix, gene annotation, and sample annotation. The data are meaningful only in the context of the underlying biology, so gene annotation and sample annotation are important [38]. There are many-to-many relationships between genes in the gene expression matrix and samples or features on the array, so it is crucial to have a detailed description of each of the features on the array.

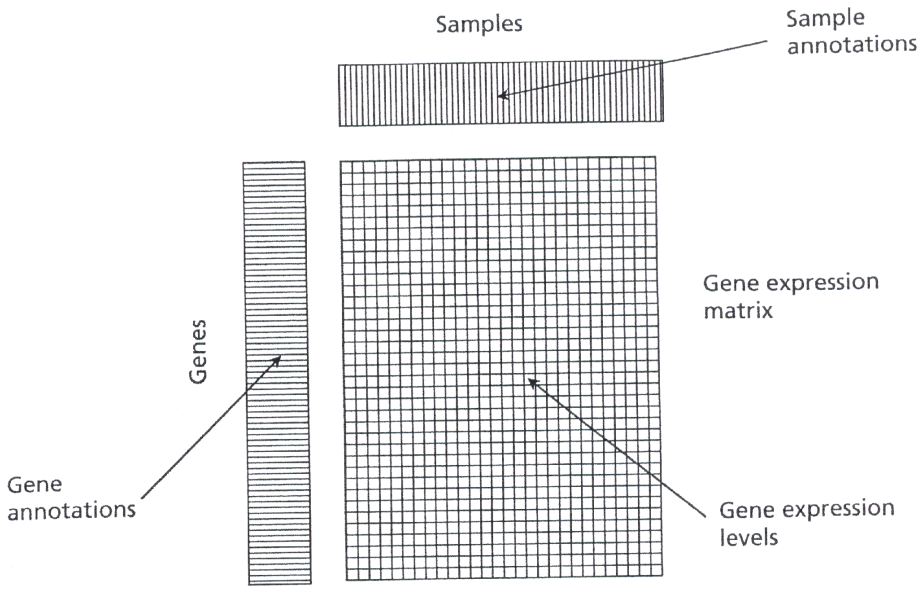

Figure 1.11: Three parts of a gene expression data matrix include gene expression data matrix, gene annotation and sample annotation.

In a gene expression data matrix, rows represent genes and columns represent experimental conditions or samples or features. See Fig1.11. The values at each position of the matrix show the expression level of a particular gene under a particular experimental condition. These values are called gene expression levels.

Rows of values in the matrix are gene expression profiles and the columns are sample expression profiles. Gene and sample annotations add some more biological information to

the matrix. Gene annotation may include gene names and sequence information, location in the genome, description of the functional roles for known genes. Sample annotation may include the information about the part of the organism from which the sample was taken or which cell type was used. The gene expression matrix together with the annotation is called an annotated gene expression data matrix.

1.12.3 Statistical analysis

Statistical stages for data analysis include data standardization and normalization, statistical testing and exploratory analysis and interpretation of the biological functions. [30].

Moreover the corresponding stages are:

1. **Data standardization and normalization**

   The raw data obtained by the computer after scanning the images are noisy and messy. After pre-processing of the images, normalization and transformation are needed.

2. **Statistical testing and exploratory analysis**

   Following this, statistical testing and/or data mining are applied. This might include preliminary analysis and a scatter-plot of the data (which help to detect unusual genes/arrays and systematic variances), hypothesis testing, exploratory analysis (like clustering analysis), classification and prediction.

3. **Interpretation of the biological functions**

   Statistical analysis usually ends up with lists of genes of interest, which must be interpreted biologically.

There are several analytical methods including: methods based on P value adjustment, Bonferroni correction, Sidak single-step procedure, Holm's step-down method, Duncan's procedure and modified Duncan's procedure, Sidak step-down and modified Sidak step-down procedures.

1.12.4 Normalization

The main goal of analysing the microarray is to identify the genes and gene groups that are different because their expression patterns are functions of biological differences [30]. This would be easy to do if the gene expression measurement were accurate and consistent, however it is subject to technical errors, linear and non-linear biases, and biological variances.

The purpose of normalization is to minimise the extraneous variation in the measured gene expression levels of hybridized mRNA samples so that biological differences (differential expression) are easily distinguished [83].

In a quantitative estimation, normalization is used to measure the systematic errors due to imperfection of equipment. Considering experimental processes, potential sources of systematic error include [28]:

1. **Sample preparation:**

   The processes of mRNA extraction, reverse transcription, cDNA amplification and labelling can affect the sample.

2. **Variability in hybridization:**

   Temperature, uneven hybridization and DNA quantity on the array can cause systematic error.

3. **Spatial effects:**

   Pin geometry and print tip problems can play a significant role in spotted microarrays.

4. **Scanner setting:**

   If parameters are subject to change, they can lead to bias.

5. **Experimenter bias:**

   Results from hybridization carried out by the same experimenter often cluster together. This is one source of systematic variation.

1.12.5 Variance

There are two sources of variance: technological variation and biological variation [30]. Technological variation can be divided into systematic and measurement errors. Systematic errors refer to sources of difference resulting from procedural variation (e.g. sample preparation, RNA extraction, etc). This can be reduced by becoming familiar with the corresponding steps of each process. Measurement error refers to errors resulting from the limitations of the tools used (e.g. printer, amplifier, etc). Biological variation refers to the difference in the subjects being examined.

1.12.6 Empty and missing values

Empty values have no corresponding values [30]. If there are missing values they show that some values were not captured. Missing values are common in a microarray study and can be caused due to deletion in data entry, equipment malfunction, scanning resolution, and dust or scratches on microarray glass surface. Missing values can affect negatively many analysis methods and can lead to false assumptions or conclusions about the biological process.

There are alternative strategies that can deal with the treatment of missing values [28].

The first is to remove the affected expression profile (gene or array profile) from the matrix altogether. This is a very radical approach and many useful values could be lost in the presence of some missing values.

The second is just to ignore the problem and leave the matrix as it is. This approach is acceptable if the proportion of missing values is within an acceptable tolerance.

The third way is to substitute a reasonable value for the missing values. In this method an average value is assigned in place of the missing value. Another approach is to use a level representing balanced expression (red-green ratio equals 1) in place of a missing value.

Imputation methods replace missing values by estimates derived from the observed data, and convert an incomplete data set into a complete one [83]. Some examples of imputation

methods include row average, k-nearest neighbours method, regression estimate method and principal component method.

1.12.7 Microarray analysis process

The process of microarray analysis includes: identifying scientific aims or tasks, designing experiments, making arrays, hybridizing or scanning spots, processing images, deriving data matrix and pre-processing the matrix [28].

1.13 Microarray gene expression

In this section microarray gene expression (or the so-called "fundamental dogma of molecular biology") will be explained. Additionally, this section includes a discussion of the microarray gene expression matrix and gene expression analysis.

Gene expression is the process by which mRNA and protein are synthesised from the DNA template of each gene [83].

Through the process of fundamental dogma of molecular biology, genetic information is carried from DNA into RNA and from RNA into protein. See Fig 1.12 [112].

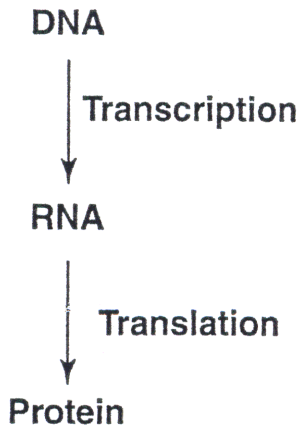

Figure 1.12: Fundamental dogma of molecular biology.

The paired-bases sequence allows DNA to encode information and replicate it by using strands. A cell's genome contains information that is necessary to synthesise (construct) proteins, and for proteins to perform all the functions that a cell needs. In the structure of a cell's genome, there is a mechanism for self-replication and transforming gene information to protein. This happens through transcription and translation. In a cellular process, cells

express their genes, so expressed genes under different physiological conditions provide important information about gene function [112].

1.13.1 Transcription process

The first stage of making protein is transcribing the information in the DNA of the genes into single-stranded RNA. The synthesis of RNA from DNA is called transcription, because this process is similar to the process of copying written words. The DNA is transcribed into RNA and the RNA is called a transcript. Through transcription the RNA copy of one strand of DNA is produced [83].

> ”When a gene is copied, first the stretch of DNA containing that gene ’unzips’ just like it does during replication. Then, free RNA nucleotides line up with the unzipped section, pairing up with matching bases on one of the DNA strands. The bases link up to form an RNA strand called messenger RNA or mRNA. Once it has copied the message contained on the gene, the mRNA molecule travels through a hole in the nuclear membrane and into the cytoplasm, ready for the next stage in the process” [131].

Transcription happens in genes where their DNA sequences are composed of coding sequences, non coding sequences and regulatory elements. Coding sequences (exons) specify protein information and non coding sequences (introns) are removed or do not have coding information. Regulatory elements are short DNA sequences of 10-100 base pairs and they control the expression of genes. As Fig 1.13 shows, the transcription process (wavy arrow), which results in the synthesis of single-stranded cellular mRNA, is mediated by specific DNA sequences or regulatory elements located adjacent to cellular genes. An enhancer (solid rectangle) modulates the efficiency of transcription, and a promoter (thick line) provides a start site for the RNA polymerase enzyme. [112].

There are varieties of regulatory elements including promoters and enhancers and they are located near the genes that they regulate. A promoter is an element that decides about the start site for RNA polymerase, which is an enzyme that makes mRNA from the DNA template. Many promoters have an AT-rich promoter sequence, which is called a TATA box.

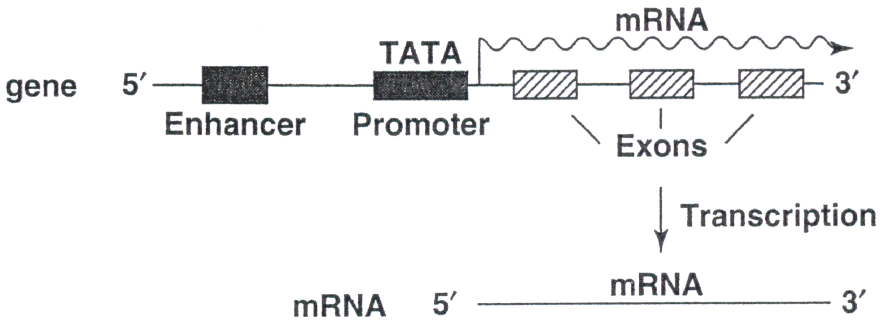

Figure 1.13: The transcription process is mediated by specific DNA sequences or regulatory elements.

An *enhancer* is an element that alters a promoter's efficiency by increasing or decreasing the rate of transcription. An increase in the rate of transcription is called *activation* (or up-regulation) and a decrease in the rate of transcription is called *repression* (or down-regulation). Cellular proteins known as transcription factors mediate the activity of promoters, enhancers and other gene regulatory elements [112]. They bind to specific nucleotide sequences within regulatory elements and modulate transcription by a variety of mechanisms. See Fig1.14.

1.13.2 mRNA processing

During the transcription of cellular genes, mRNA molecules are synthesised from DNA templates [112]. mRNA processing occurs after transcription through which mRNAs are edited. See Fig1.15. In mRNA processing, an unprocessed mRNA with exons and introns undergoes capping, which adds a single G residue to the $5'$ end and increases mRNA stability. Splicing removes introns from the mRNA to create a functional coding sequence. Polyadenylation results in the addition of a poly A tail to the $5'$ end, which increases mRNA stability and the efficiency of protein synthesis (translation). The process of cap addition, poly A addition and intron removal are known as capping, polyadenylation and splicing respectively.

1.13.3 Translation

Translation is a process in which proteins are synthesised according to the RNA information [83]. The process of reading mRNA sequence and converting it into amino acid is similar

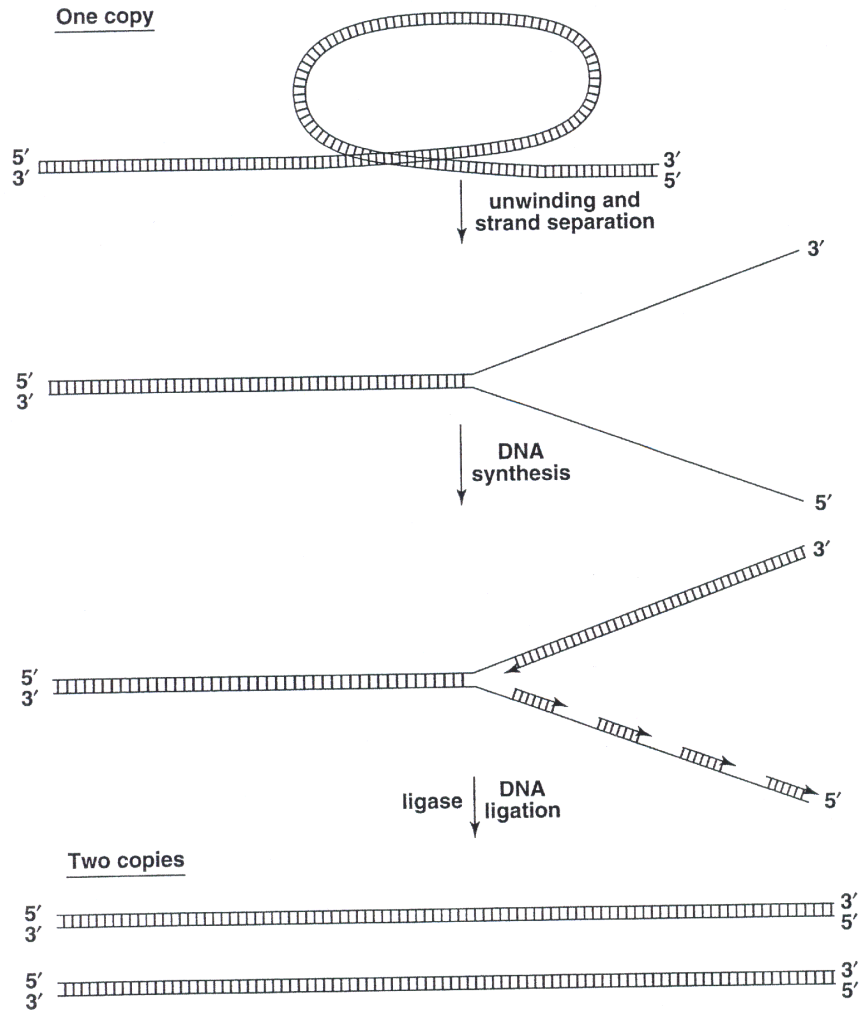

Figure 1.14: Transcription of cellular genes is regulated by activators (oval) and repressors (diamond), functioning through enhancer regulatory elements (solid rectangles).

to translating from one language to another and this is why it is called translation. The four-letter alphabet of the genes is translated into the 20 amino acid alphabet of proteins in ribosome. Translation of genetic code into a protein is achieved with the help of tRNA.

In the process of gene expression, RNA provides mRNA, tRNA, and rRNA. cDNA is complementary to a given mRNA and is made by reverse transcription. Reverse transcription allows mRNA to be retrieved as cDNA. Existence of mRNA and cDNA shows that the information in either type of nucleic acid is convertible.

Like messages in the gene, mRNA's instructions are composed of words such as ACC or UAC (called codons) and each include three letters or bases. mRNA's message is decoded according to the genetic code. By using genetic code, the cell translates the language of

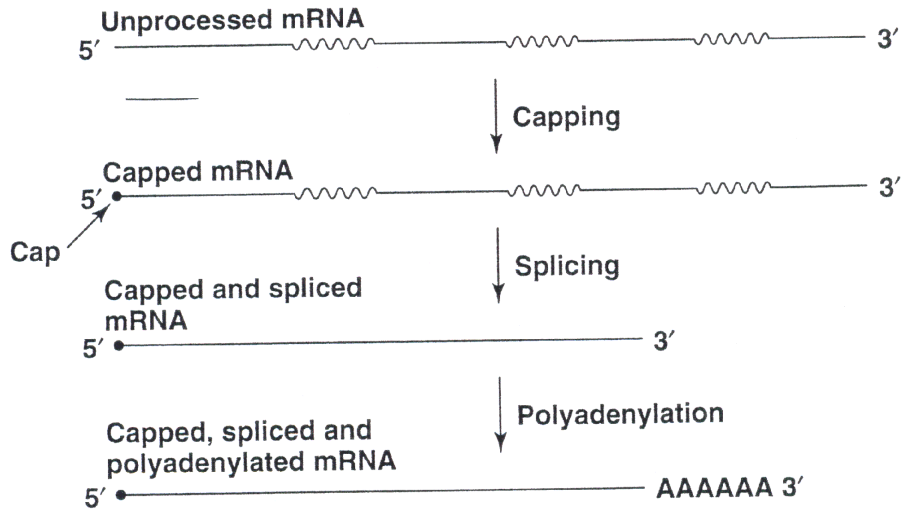

Figure 1.15: mRNA processing. An unprocessed mRNA undergoes capping, splicing and polyadenylation.

DNA written in words or codons to the language of proteins written in amino acids [131].

Translation is the synthesis of a polypeptide chain from processed mRNA. Translation happens on cytoplasmic structures which are called ribosomes [112].

In the translation process, a ribosome attaches itself to the capped ($5'$) end of the mRNA and moves in a $5'$ to $3'$ manner until it meets a start codon (AUG) and translation starts immediately. Methionine (MET) is added as the first amino acid. The ribosome coordinates interactions between the mRNA and tRNA molecules. For each codon one amino acid is added to the polypeptide chain. Translation terminates when the ribosome meets a stop codon (UAA, UGA OR UAG) in the mRNA sequence for which no tRNA exists and causes the release of a fully synthesised polypeptide (wavy line). See Fig1.16.

1.13.4 mRNA and protein abundance

Proteins are the last product of a gene expression process and proteins synthesised by a cell's genome are called proteome [28].

Measuring real gene expression means measuring the abundance of proteins. DNA microarray experiments measure the abundance of mRNA but not protein abundance. A specific gene (genomic DNA sequence) always produces the same amino acid sequence of the related protein which folds to assume its native state. Measuring mRNA abundance will give accurate information about protein abundance. It will also reveal the primary structure

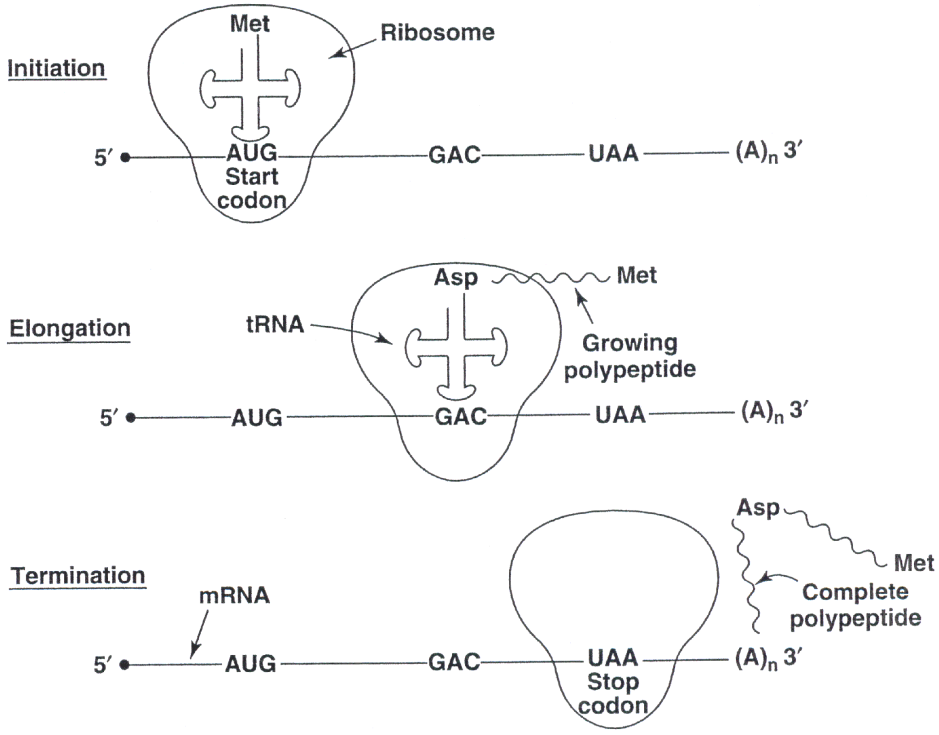

Figure 1.16: Translation including initiation, elongation and termination stages.

of the proteins related to the measured mRNA. There are three ways by which proteins are formed and the genome might be subject to alterations.

Firstly, genomic DNA might change due to a replication machinery error that causes it to copy damaged DNA.

Secondly, difference might occur due to differences in promoter selection, because genes may have different promoters. It is also possible that one mRNA will be edited before translation, or one base in mRNA will be replaced by another base, altering one amino acid. There is no connection between the amount of mRNA and the amount of protein translated from it. There is a correlation between mRNA abundance and protein abundance, but the control of rates of translation can be quite different.

Finally, structure-modifying alterations might occur after translation. DNA gene microarrays actually measure mRNA transcript abundances. DNA microarray studies are preferred to protein expression and modification studies, which are still very expensive and need high-level techniques.

1.13.5 Microarray gene expression matrix

Microarray gene expression matrix measures the expression of many genes with a number of conditions and represents this in a table [30]. This table shows the function of each gene in the genome measuring gene expression levels at different stages, tissues or different conditions [72]. Rows in this table would correspond to genes and columns would show different variables such as tissues, treatments and so on. Each position in the table will represent values showing the expression level of a particular gene in a particular sample. This table is known as microarray gene expression matrix (GEM). Thus GEM=g X n, in which g is number of genes whose expression is measured in each n array.

A database of gene expression matrices composed of different microarray experiments will help to understand gene regulation, the genetic mechanism of diseases and reactions of cells to drug treatments. This can help to predict gene function for genes whose functions are unknown (and this prediction would be based on the expression similarity to the known genes); to identify which genes are important in diseases or cellular process; to discover how cells respond to various compounds; and to learn gene regulation by studying groups of co-regulated genes. By showing the status quo of gene expression levels of known genes, microarrays are transforming a black box of a cell to a transparent box.

"Microarrays measure the relative or absolute mRNA abundance indirectly by measuring the intensity of the fluorescence of the labelled mRNA bound to spots on the array" [72]. The intensities of each fluorescent dye is measured on a separate channel, and the raw data produced by microarrays are monochrome images for each channel. ScanAlyze is the most popular software package for image analysis from spotted arrays. It is important to know not only the value of the measurement, but also the standard error for each data point.

1.13.6 Gene expression outliers

In a gene expression matrix we come across certain values that are called *outliers* [30]. Outliers are measurements that are inconsistent, compared with the other members of the same matrix. Outliers can affect some analysis tools by their presence. These values can be created by experimental handling and can account for up to 15% of the variation in a microarray experiment.

1.13.7 Gene expression analysis

In a DNA microarray experiment, a probe of one DNA strand that matches a particular mRNA in a cell is used to measure the concentration of mRNA in the cell [77]. Concentration of a particular mRNA is a result of expression of its corresponding gene. This application is normally called expression analysis. If different probes matching all mRNAs in a cell are used, a snapshot of the total mRNA pool of a living cell or tissue can be obtained. This is called the expression profile, because it reflects the expression of every single measured gene at that moment. This can sometimes be used to show the expression of a simple gene over different conditions. Expression analysis can be performed by a method that is called *serial analysis of gene expression (SAGE)*. SAGE uses traditional DNA sequencing to identify and count the number of mRNAs in a cell.

With microarray technology, large sets of gene expression data can be created. These are called gene expression profiles (or transcriptional profiles) and gathering is called profiling. Transcriptional profiling can be either sequencing-based or hybridization-based.

1.13.8 Gene expression profile

This term explains the expression value for a single gene across many samples or experimental conditions, and for many genes under a single condition or sample [28].

One gene over multiple samples

A *gene profile* is a gene expression profile that describes expression values for a single gene across many samples or conditions.

Many genes over one samples

An *array profile* is a gene expression profile that explains the expression values for many genes under a single condition or sample.

$$E = (X_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ . & . & . & . \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{pmatrix}$$

where x_{ij} denotes the expression level of sample j for gene i, such that $j = 1,\dots M$, and $i = 1,\dots N$.

1.13.9 Measuring and reporting expression

Estimation of the expression for each gene is the starting point for each analysis [38]. In analysis we try to see the expression based on measured hybridization intensities. Hybridization is looking for the relative RNA representation from which expression can be inferred.

In a gene expression measurement process, each gene is represented by some features on the array for which fluorescence intensities are measured. From these measurements, we attempt to determine the expression level of a gene. In the array a range of diverse information such as selected samples, collection condition, RNA extraction and labelling conditions, hybridization conditions and others could be tracked. Ultimately what is measured is RNA representation, not expression, and each step of the process can affect the final result. See Fig1.17. Hybridization measurement and report method can have a crucial effect on the conclusion of an experiment. [38].

After identifying the features, microarray image analysis software measures the intensities in each channel for each pixel that comprises the image of each feature and reports a variety of statistics. These normally include the total intensity for each feature and some statistical facts. The main goal of each microarray is to identify the expression difference

	C1	C2
A	2	3
B	3	4
C	4	2

Table 1.2: Gene expression matrix of three genes under two conditions. The gene expression measurements are in arbitrary units.

for each gene. There are some methods for measuring the fluorescent intensity for the arrayed features. Most microarrays use either the background-subtracted median or total intensities as the statistic representing each feature.

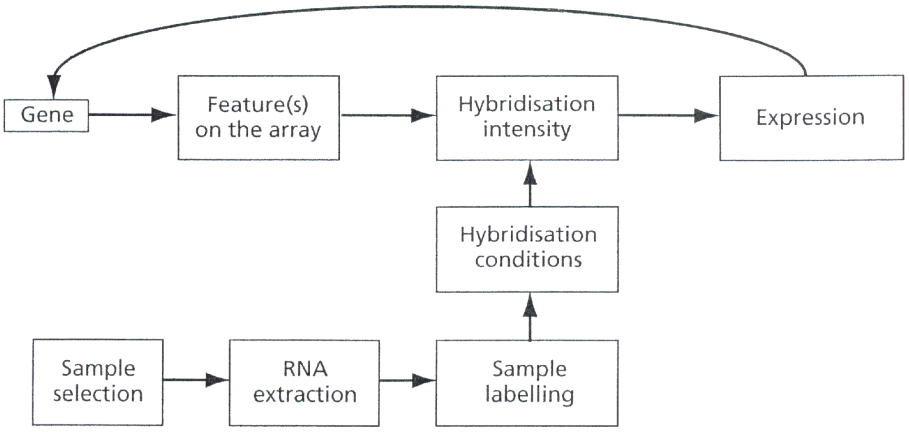

Figure 1.17: A gene expression measurement diagram.

1.13.10 Expression data as a vector space

In a data matrix, we can talk about gene space and sample space [38]. Each gene may be considered as a point in m-dimensional space where m is the number of samples and each sample can be assumed as a point in an n-dimensional space where n is the number of genes. Here we give an example of three genes A, B, C and two conditions $C1$ and $C2$. ($m = 2$ and $n = 3$). See Table 1.2 [38].

This can be visualised either as 3 two-dimensional vectors in the condition space or 2 three-dimensional vectors in the gene space. See Fig1.18 [38].

In a multidimensional space each point defines a vector. Representing genes and samples as vectors allows the use of linear algebra and data analysis methods.

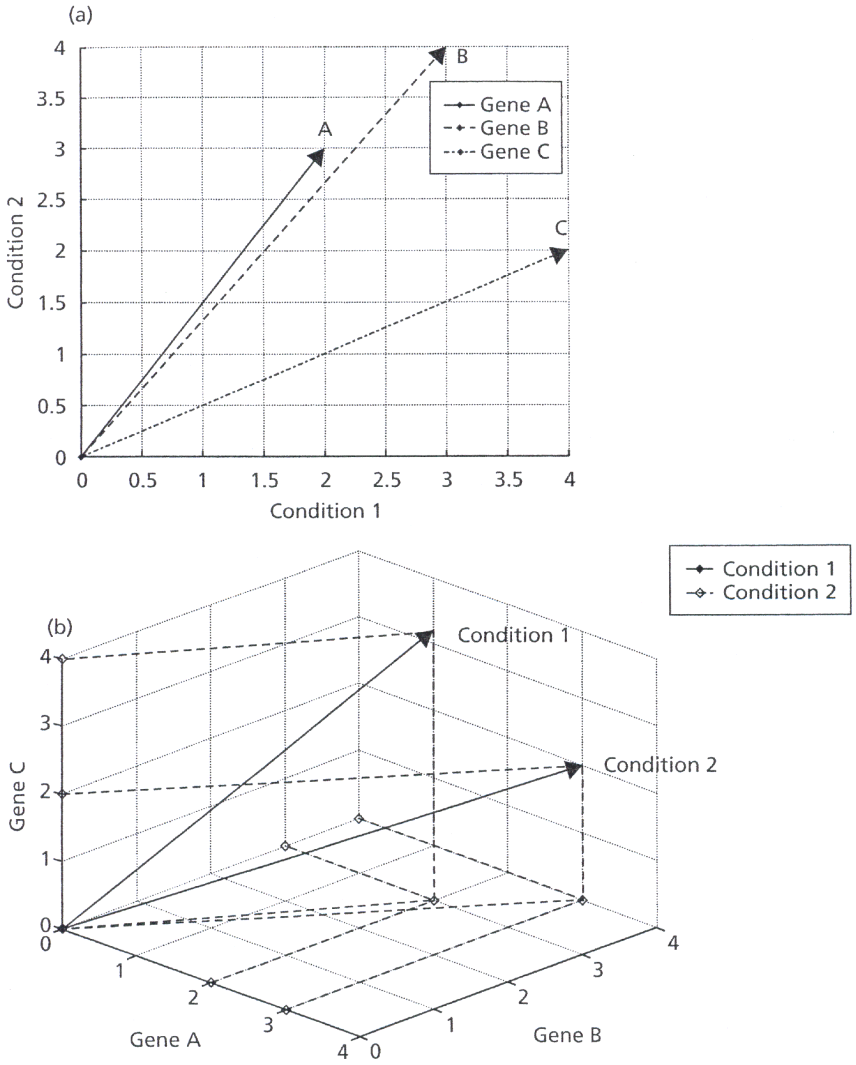

Figure 1.18: Visualizing genes in condition space (a) and conditions in genes space (b) for the gene expression matrix of Table1.2

Let X be the gene expression matrix with m columns and n rows. Let X_{ij} be the

expression value in the i_{th} row and the j_{th} column. i.e.

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nm} \end{pmatrix}$$

As an example the expression of the three genes under two conditions is represented by the 3×2 matrix.

$$\begin{pmatrix} 2 & 3 \\ 3 & 4 \\ 4 & 2 \end{pmatrix}$$ [4] [21] [74] [28] [131] [5] [112] [64] [104] [83] [98] [101] [37] [76] [100]

[110] [30] [122] [95] [38] [72] [77] [19] [137] [15] [92] [25] [26] [35] [129] [126] [12] [13] [132] [88] [79] [80] [109]

[**?**] [**?**] [**?**] [52] [**?**] [68] [75] [**?**] [**?**] [119] [120] [**?**] [67] [1] [7] [9] [10] [11] [**?**] [16] [17] [14] [27] [31] [32] [**?**] [33] [47] [48] [**?**] [44] [50] [58] [59] [60] [61] [62] [70] [78] [89] [**?**] [**?**] [115] [123] [117] [71] [91] [134] [135]

[3] [**?**] [**?**] [20] [29] [**?**] [40] [46] [55] [57] [65] [63] [102] [103] [107] [54] [69] [18] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [2] [**?**] [**?**] [81] [125] [66]

[8] [39] [49] [56] [73] [82] [84] [85] [86] [87] [116] [118] [124] [133] [136] [138] [139] [23] [41] [93] [34] [108] [106] [127] [**?**] [**?**] [**?**] [43] [51] [113] [114] [111] [94] [36] [45] [22] [96] [97] [6] [] [105] [121] [42] [99] [90] [128] [53]

Chapter 2

Clustering in gene expression data sets

In this chapter we develop a new algorithm to solve cluster analysis problems in gene expression data sets. We start with an explanation of the main concepts of data mining.

2.1 Introduction

2.1.1 Data mining

Data mining is the process of automatically searching large volumes of data for patterns. In a data mining process, analytical tools are used to correlate any kind of data under survey and to present them in a meaningful way. If there are any hypotheses, statistical analysis might be applied to examine them [83]. Data mining has developed to facilitate the identification of useful information within data reservoirs, and it involves the application of discovery algorithm to the data [74]. Data mining aims may include prediction, classification and description.

Knowledge discovery in databases (KDD) is the process of extracting models and patterns from large databases [109]. The terms KDD and data mining (DM) have often been used interchangeably, however, strictly speaking KDD is the umbrella of the mining process and DM is only a step in KDD. One main objective of KDD is to simplify the underlying model in the data for use and make it understandable for a decision maker.

Steps of Knowledge Discovery in Databases (KDD)

Sarker et al. [109] argued that the literature contains many descriptions of the steps involved in KDD. However, they described 13 steps in KDD as the following:

1. **Problem definition and determining the mining task:** This step identifies if the mining task should take place or not and, if so, the aims of the mining task.

2. **Data description and selection:** In this step suitable data fields and files are selected.

3. **Data conversion:** The database file system suitable for the mining process is identified and the data are converted from the original format to the selected one.

4. **Data cleaning:** Reduces or removes noise and errors in the data.

5. **Data transformation:** The logical relation between probable existing tables is reflected in a single table that contains all the information necessary for the mining process.

6. **Data reduction and projection:** In projecting the data, information is condensed into a smaller number of attributes.

7. **Domain-specific data pre-processing:** A set of operations using domain-specific knowledge that makes the attributes valid from the domain point of view.

8. **Feature selection:** Identifies a subset of features that significantly contribute to the discrimination or prediction problem.

9. **Choosing the mining algorithm:** Chooses one or more computational techniques that are efficient.

10. **Algorithm-specific data pre-processing:** Does not alter the database and a view of the data would be available to each algorithm with the pre-processing taking place on the view level.

11. **Applying the main algorithm:** Includes the application of one or more computational techniques that are efficient and can produce particular patterns or models over the data.

12. **Analysing and refining the results:** This is to analyse the outcomes and refine them.

Data Mining Process

In practice the data mining process includes the following steps [74]:

1. **State the problem and formulate the hypothesis:** The experience domain needs to be identified to enable the development of a meaningful problem statement. In this step the researcher specifies a set of variables with unknown dependency or variables with only a general idea about the dependency of them. This step uses the application domain and a data mining model.

2. **Collect the data:** This step refers to how the data are generated and collected. There are two possibilities. The first one is when a modeller controls the data generation process. This is called a designed experiment. The other possibility is when the modeller does not have any control over data generation, which is known as the observational approach or random data generation.

3. **Pre-processing the data:** This includes the following tasks:

    3.1. **Outlier detection (and removal):** Outliers are unusual data values. The researcher may choose to detect and remove outliers or to develop modelling methods that are insensitive to outliers.

    3.2. **Scaling, encoding, and selecting features:** In this stage variable scaling or data encoding might be used. It is recommended that the features are scaled to bring them to the same weight for further analysis.

    3.3. **Estimate the model:** Selection and implementation of an appropriate data mining technique occurs.

    3.4. **Interpret the model and draw conclusions:** Data mining models are used for decision making, so they need to be interpretable. In order to obtain accurate results, modern data-mining methods using high-dimensional models might be utilised.

2.2 Cluster analysis problems

Cluster analysis is one of the important data mining tasks. Clustering (or cluster analysis) aims to partition a set of objects (genes or samples) into the groups that are relatively similar [100]. This means that objects in the same group will be more similar than the objects in different groups.

Clustering is also called unsupervised classification of the patterns [19]. Cluster analysis is involved with the problem of organizing a collection of patterns into clusters based on similarity. Different similarity measures can be used in cluster analysis and the squared Euclidean distance is one of the most widely used similarity measures:

$$\| x - y \|^2 = \sum_{i=1}^{n} (x_i - y_i)^2.$$

Another similarity measure can be absolute or city block metric, which is defined as

$$\|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i|$$

Euclidean distance will be used as a similarity measure. Each cluster is identified by its centre (or centroid). In cluster analysis, we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A = \{a^1, \ldots, a^m\}, \text{ where } a^i \in \mathbb{R}^n,\ i = 1, \ldots, m.$$

There are different types of clustering. Here we consider the hard unconstrained partition clustering problem, that is the distribution of the points of the set A into a given number k of disjoint subsets $A^j,\ j = 1, \ldots, k$ with respect to predefined criteria such that:

1) $A^j \neq \emptyset,\ j = 1, \ldots, k$;

2) $A^j \bigcap A^l = \emptyset,\ j, l = 1, \ldots, k,\ j \neq l$;

3) $A = \bigcup_{j=1}^{k} A^j$.

4) no constraints are imposed on the clusters $A^j,\ j = 1, \ldots, k$.

The sets $A^j,\ j = 1, \ldots, k$ are called clusters. We assume that each cluster A^j can be identified by its centre (or centroid) $x^j \in \mathbb{R}^n,\ j = 1, \ldots, k$. Then the clustering problem

can be reduced to the following optimization problem [31, 119]:

$$\text{minimize } \psi(x,w) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} w_{ij}\|x^j - a^i\|^2 \tag{2.1}$$

subject to

$$x = (x^1,\ldots,x^k) \in \mathbb{R}^{n\times k}, \tag{2.2}$$

$$\sum_{j=1}^{k} w_{ij} = 1,\ i = 1,\ldots,m, \tag{2.3}$$

and

$$w_{ij} = 0 \text{ or } 1,\ i = 1,\ldots,m,\ j = 1,\ldots,k \tag{2.4}$$

where w_{ij} is the association weight of pattern a^i with cluster j, given by

$$w_{i,j} = \begin{cases} 1 & \text{if pattern } a^i \text{ is allocated to cluster} \\ 0 & \text{otherwise} \end{cases}$$

and

$$x^j = \frac{\sum_{i=1}^{m} w_{ij}a^i}{\sum_{i=1}^{m} w_{ij}},\ \ j = 1,\ldots,k.$$

Here $\|\cdot\|$ is a Euclidean norm and w is an $m \times k$ matrix. The problem (2.4) is also known as the minimum sum-of-squares clustering problem. Different algorithms have been proposed to solve the clustering problem. Jain et al. [70] provided a survey of the most existing algorithms. Among clustering algorithms we can mention here heuristics like k-means algorithms and their variations (h-means, j-means etc.), mathematical programming techniques (including dynamic programming, branch and bound, cutting plane, interior point methods), the variable neighbourhood search algorithm and metaheuristics like simulated annealing, tabu search and genetic algorithms [1, 33, 44, 48, 50, 58–61, 78, 115, 119, 123]. Since the number of genes in gene expression data sets are very large, most of these algorithms cannot be applied to the clustering of samples in such data sets.

The problem (2.1)–(2.4) is a global optimization problem and the objective function ψ in this problem has many local minima. However clustering algorithms based on global optimization techniques are not applicable to even relatively large data sets. Algorithms that are applicable to such data sets can locate only local minima of the function ψ and, as the number of clusters increases, these local minima can differ significantly from the global

solutions. Another difficulty is that the number of clusters, as a rule, is not known a priori. Over the last several years different incremental algorithms have been proposed to address these difficulties. Results of numerical experiments show that an incremental approach allows one, as a rule, to locate a local solution close to the global one. Consequently it can produce a better cluster structure of a data set. Bagirov et al. [19] developed an incremental algorithm based on nonsmooth optimization approaches to clustering. The global k-means algorithm was developed in [89]. The incremental approach is also discussed in [62].

Here a new version of the global k-means algorithm for solving clustering problems in gene expression data sets is being proposed. In this algorithm a starting point for the k-th cluster center is computed by minimizing the so-called auxiliary cluster function.

2.2.1 Clustering algorithms

Clustering in gene expression data sets is a challenging problem. One can consider two types of clustering in gene expression data sets: clustering of genes and clustering of samples.

Clustering of genes

Most of methods were designed to solve gene clustering problems. In unsupervised methods, current knowledge regarding the functional role of different genes is not considered [30]. Hence, unsupervised microarray data analysis introduces a process in which the system shows existing gene categories and ignores an imposed structure. The system uses a data set to find regularities, patterns or groups.

It is assumed that each gene belongs to a category that is associated with a function or co-regulation. In this case it is expected that unsupervised analysis will introduce a new explanation regarding gene expression association that has not been evident previously.

Clustering invokes unsupervised methods that can be used to determine if the elements of a gene expression matrix belongs to a special group. It is assumed that similar expression levels must indicate the same biological function or co-regulation. Clustering helps to determine the function of the unknown genes. In the clustering process, expression values are grouped according to the distance function.

The members of each gene expression cluster are similar to other members in the same cluster, but they are different from the members in the other clusters. The first step in

clustering is describing similarity and dissimilarity by a distance function.

Different algorithms for clustering of genes have been proposed [47,91,134,135]. Some of the main techniques are described in further detail below.

Hierarchical clustering Hierarchical clustering is used to identify genes with similar profiles and thus similar functions [30]. In the clustering process, each gene expression value is expressed as coordinates, which represent the distance from the other genes, by using pair-wise similarity measures. Hierarchical clustering is divided into two sub groups according to the criterion of dissimilarity (divisive) and similarity (agglomerative). A divisive approach (top-down) starts with all gene expression values in a single cluster and starts splitting until a criterion is met. An agglomerative approach (bottom-up) begins with each gene expression value in different (singleton) sets, and merges the clusters until a criterion is met.

The result of hierarchical clustering is a tree-shaped graph called a dendrogram. This represents a visual summary of the clustering process. A dendrogram is a colour-coded graph in which each gene expression value is a leaf. Red indicates an increase in gene expression levels and green indicates a decrease in gene expression levels. The intensity of the colour is a measure of the difference between other values and clusters. The length of the horizontal line that connects two clusters (nodes) shows the relative closeness.

Bi-clustering (or two way clustering) is a technique that is capable of clustering genes and microarray subsets simultaneously. Hierarchical clustering employs different methods [30] including: single-linkage method, complete-linkage method, average-linkage clustering, centroid-linkage method, median-linkage clustering, and Ward's clustering method.

Partitional clustering Partitional clustering divides gene expression values g into k groups until each group presents a cluster and $k \leq g$ [30]. This process has two requirements.

Firstly, each cluster must contain at least one gene expression value. Secondly, each gene expression value must belong to a cluster.

k-means clustering This is one of the most popular unsupervised methods applied to microarray data [30]. It is said that k-means methods give better results in microarray data sets where the clusters have similar gene expression values and they are expected to be compact and therefore have similar biological functions. In microarray data analysis, the k-means algorithm represents each gene expression value as a point. The algorithm identifies

k-points (seed points) and assumes them as centroids. For a pass through a data set, k-points are assumed to be fixed at any iteration. In the next iteration, the remaining points are assigned to the nearest k-points so as to minimize the sum of the distance between seed points and all the other points.

Fuzzy clustering In 'fuzzyfication', numbers such as gene expression levels are changed to qualitative descriptors [30]. The difference between fuzzy k-clustering and standard k-clustering is that fuzzy k-clustering assumes each gene point as a member of each cluster with certain degree. This allows a fuzzy k-means algorithm to identify overlapping groups of genes and identify the role of a gene in different pathways.

Clustering of samples

VizCluster technique for sample clustering Zhang et al. [137] present the VizCluster technique, which is a visualization approach to cluster analysis. The aim of clustering and classification is to find out the pattern or structure of data sets. Visualizing these patterns or structures can help in exploratory data analysis. This technique uses graphical visualization methods to show the data structure or underlying data pattern. Using both high-dimensional scatterplot and parallel coordinate plots helps to produce a non-linear projection and changes n-dimensional vectors into two dimensional points.

Zhang et al. have developed two approaches:

1. Supervised maximum entropy approach, which uses pre-known classes of samples as a training set, then applies the maximum entropy model to generate the optimal pattern model which can be used on new samples.

2. Unsupervised interrelated two-way clustering method, which dynamically uses the relationship between the groups of genes and samples while clustering through both gene-dimension and sample-dimension to identify important genes and classify samples simultaneously.

VizCluster supports three types of data analysis including cluster/class discovery in both supervised and unsupervised analysis, class prediction and class assessment. Here the goal is the classification of samples in gene expression data.

Due to the large number of genes only a few algorithms can be applied to the clustering of samples [13]. As the number of clusters increases the number of variables in the cluster-

ing problem increases drastically and most clustering algorithms become inefficient. The k-means algorithm and its different variations are among those algorithms which are still applicable to the clustering of samples in gene expression data sets. However as the number of clusters increase, the k-means algorithms in general converge only to local minima and these local minima may be significantly different from the global solutions. Recently the global k-means algorithm has been proposed to improve global search properties of k-means algorithms [89].

In their work Bagirov et al. [15] propose a new clustering algorithm which is based on methods of non-smooth optimization. In this algorithm, clusters are calculated incrementally. The algorithm calculates as many clusters as exist in a data set, with respect to a given tolerance.

2.2.2 k-means and the global k-means algorithms

In this section we give a brief description of the k-means and the global k-means algorithms.

The k-means algorithm proceeds as follows:

Algorithm 1. The k-means algorithm.

Step 1. Choose a seed solution consisting of k centres (not necessarily belonging to A);

Step 2. Allocate data points $a^i \in A$ to its closest centre and obtain k-partition of A;

Step 3. Recompute centres for this new partition and go to step 2 until no more data points change cluster.

The effectiveness of this algorithm is highly dependent upon the starting point. It converges only to a local solution, which can differ significantly from the global solution in many large data sets.

The global k-means algorithm proposed in [89] computes clusters successively. At the first iteration of this algorithm, the centroid of the set A is computed and in order to compute the k-partition at the k-th iteration this algorithm uses centres of $k-1$ clusters from the previous iteration. The global k-means algorithm for the computation of $q \leq m$ clusters in a data set A can be described as follows.

Algorithm 2. The global k-means algorithm.

Step 1. (Initialization) Compute the centroid x^1 of the set A:

$$x^1 = \frac{1}{m}\sum_{i=1}^{m} a^i, \;\; a^i \in A, \; i = 1, \ldots, m$$

and set $k = 1$.

Step 2. Set $k = k + 1$ and consider the centres $x^1, x^2, \ldots, x^{k-1}$ from the previous iteration.

Step 3. Consider each point a of A as a starting point for the k-th cluster centre, thus obtaining m initial solutions with k points $(x^1, \ldots, x^{k-1}, a)$; apply k-means algorithm to each of them; keep the best k-partition obtained and its centres $x^1, x^2, \ldots, x^k$.

Step 4. (Stopping criterion) If $k = q$ then stop, otherwise go to Step 2.

This version of the algorithm is not applicable for clustering in middle-sized or large data sets. Two procedures were introduced to reduce its complexity [89]. We mention here only one of them because the second procedure is applicable to low-dimensional data sets. Let d^i_{k-1} be a squared distance between $a^i \in A$ and the closest cluster centre among the $k-1$ cluster centres obtained so far. For each $a^i \in A$ we calculate the following:

$$r_i = \sum_{j=1}^{m} \min\{0, \|a^i - a^j\|^2 - d^j_{k-1}\}$$

and we take the data point $a^l \in A$ for which

$$l = \arg \min_{i=1,\ldots,m} r_i$$

as a starting point for the k-th cluster centre. Then k-means algorithm is applied starting from the point $x^1, x^2, \ldots, x^{k-1}, a^l$ to find k cluster centres. We used this procedure in our numerical experiments.

It should be noted that the k-means algorithm and its variants tend to produce only spherical clusters and they are not always appropriate for solving clustering problems. However applying k-means algorithms, we assume that clusters in a data set can be approximated by n-dimensional balls.

2.2.3 Computation of starting points

The clustering problem (2.1)–(2.4) can be reformulated in terms of nonsmooth, nonconvex optimization as follows [15, 17]:

$$\text{minimize } f(x) \tag{2.5}$$

subject to

$$x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \tag{2.6}$$

where

$$f(x^1, \ldots, x^k) = \frac{1}{m} \sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|^2. \tag{2.7}$$

We call f a *cluster function*. If $k > 1$, the function f is nonconvex and nonsmooth. The number of variables in problem (2.1)–(2.4) is $(m + n) \times k$ whereas in problem (2.5)–(2.6) this number is only $n \times k$ and the number of variables does not depend on the number of instances. It should be noted that in many real-world data sets, the number of instances m is substantially greater than the number of features n. On the other hand, in the hard clustering problems the coefficients w_{ij} are integer, that is the problem (2.1)–(2.4) contains both integer and continuous variables. In the nonsmooth optimization formulation of the clustering, all problem variables are continuous. All these circumstances can be considered as advantages of the nonsmooth optimization formulation (2.5)–(2.6) of the clustering problem.

Let us consider the problem of finding k-th cluster centre assuming that the centres $x^1, \ldots, x^{k-1}$ for $k - 1$ clusters are known. Then we introduce the following function:

$$\bar{f}^k(y) = \frac{1}{m} \sum_{i=1}^{m} \min \left\{ d_{k-1}^i, \|y - a^i\|^2 \right\} \tag{2.8}$$

where $y \in \mathbb{R}^n$ stands for k-th cluster centre and

$$d_{k-1}^i = \min \left\{ \|x^1 - a^i\|^2, \ldots, \|x^{k-1} - a^i\|^2 \right\}.$$

The function $\bar{f}^k$ is called an *auxiliary cluster function*. It has only n variables.

Consider the set

$$\overline{D} = \left\{ y \in \mathbb{R}^n : \|y - a^i\|^2 \geq d_{k-1}^i \right\}.$$

$\bar{D}$ is the set where the distance between any point y and any data point $a^i \in A$ is no less than the distance between this data point and its cluster centre. We also consider the following set

$$D_0 = \mathbb{R}^n \setminus \overline{D} \equiv \{y \in \mathbb{R}^n :$$

$$\exists I \subset \{1, \ldots, m\},\ I \neq \emptyset : \|y - a^i\| < d^i_{k-1}\ \ \forall i \in I\}.$$

The function $\bar{f}^k$ is a constant on the set $\overline{D}$ and its value in this set is

$$\bar{f}^k(y) = d_0 \equiv \sum_{i=1}^{m} d^i_{k-1},\ \ \forall y \in \overline{D}.$$

It is clear that $x^j \in \overline{D}$ for all $j = 1, \ldots, k-1$ and $a^i \in D_0$ for all $a^i \in A,\ \ a^i \neq x^j,\ j = 1, \ldots, k-1$. It is also clear that $\bar{f}^k(y) < d_0$ for all $y \in D_0$.

Any point $y \in D_0$ can be taken as a starting point for the k-th cluster centre. The function $\bar{f}^k$ is a nonconvex function with many local minima and one can assume that the global minimum of this function would be a good candidate as the starting point for the k-th cluster centre. However it is not always possible to find the global minimum of $\bar{f}^k$ in a reasonable time. Therefore we propose an algorithm for finding a local minimum of the function $\bar{f}^k$.

For any $y \in D_0$ we consider the following sets:

$$S_1(y) = \left\{a^i \in A : \|y - a^i\|^2 = d^i_{k-1}\right\},$$

$$S_2(y) = \left\{a^i \in A : \|y - a^i\|^2 < d^i_{k-1}\right\},$$

$$S_3(y) = \left\{a^i \in A : \|y - a^i\|^2 > d^i_{k-1}\right\}.$$

The set $S_2(y) \neq \emptyset$ for any $y \in D_0$.

The following algorithm is proposed to find a starting point for the k-th cluster centre.

Algorithm 3. An algorithm for finding the starting point.

Step 1. For each $a^i \in D_0 \bigcap A$ compute the set $S_2(a^i)$, its centre c^i and the value $\bar{f}^k_{a^i} = \bar{f}^k(c^i)$ of the function $\bar{f}^k$ at the point c^i.

Step 2. Compute

$$\bar{f}^k_{min} = \min_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

$$a^j = \arg \min_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

the corresponding centre c^j and the set $S_2(c^j)$.

Step 3. Recompute the set $S_2(c^j)$ and its centre until no more data points escape or return to this cluster.

Let $\bar{x}$ be a cluster centre generated by Algorithm 3. Then the point $\bar{x}$ is a local minimum of the function $\bar{f}^k$.

2.2.4 An incremental clustering algorithm

In this subsection we describe an incremental algorithm for solving cluster analysis problems.

Algorithm 4. An incremental algorithm for clustering problems.

Step 1. (Initialization). Select a tolerance $\epsilon > 0$. Compute the centre $x^{1*} \in \mathbb{R}^n$ of the set A. Let f^{1*} be the corresponding value of the objective function (2.7). Set $k = 1$.

Step 2. (Computation of the next cluster centre). Let $x^{1*}, \ldots, x^{k*}$ be the cluster centres for the k-partition problem. Apply Algorithm 3 to find a starting point $y^{k+1,0} \in \mathbb{R}^n$ for the $(k+1)$-st cluster centre.

Step 3. (Refinement of all cluster centres). Take $x^{k+1,0} = (x^{1*}, \ldots, x^{k*}, y^{k+1,0})$ as a new starting point, apply the k-means algorithm to solve the $(k+1)$-partition problem. Let $u^{1*}, \ldots, u^{k+1,*}$ be a solution to this problem and $f^{k+1,*}$ be the corresponding value of the objective function (2.7).

Step 4. (Stopping criterion). If

$$\frac{f^{k*} - f^{k+1,*}}{f^{1*}} < \varepsilon$$

then stop, otherwise set $x^{i*} = u^{i*},\ i = 1, \ldots, k+1$, $k = k+1$ and go to Step 2.

It is clear that $f^{k*} \geq 0$ for all $k \geq 1$ and the sequence $\{f^{k*}\}$ is decreasing, that is,

$$f^{k+1,*} \leq f^{k,*} \text{ for all } k \geq 1.$$

This implies that after $\bar{k} > 0$ iterations, the stopping criterion in Step 4 will be satisfied. Thus Algorithm 4 computes as many clusters as the data set A contains with respect to the tolerance $\varepsilon > 0$.

The choice of the tolerance $\varepsilon > 0$ is crucial for Algorithm 4. Large values of ε can result in the appearance of large clusters, whereas small values can produce small and artificial clusters. The recommended values for ε are $\varepsilon \in [10^{-2}, 10^{-1}]$.

2.3 Results of numerical experiments

To verify the effectiveness of the proposed clustering algorithm and to compare it with similar algorithms, several numerical experiments with ten gene expression data sets have been carried out on a Pentium-4, 2.0 GHz, PC. We use multi-start k-means (MSKM) and the global k-means (GKM) algorithms for comparison. 100 randomly generated starting points are used in MSKM. In the tables below, MGKM stands for the modified global k-means algorithm. In the tables we present the number of clusters, values of the clustering function (f_{val}) obtained by different algorithms and CPU time.

2.3.1 Data set 1

For description of this data set, refer to Appendix A, SectionA.1.

Results for this data set are presented in Table 2.1. We can see from these results that the MSKM algorithm produces better results than two other algorithms when the number of clusters $N \leq 15$. It outperforms the GKM algorithm in all cases and produces worse results than the MGKM algorithm only when $N = 20$. The MGKM algorithm outperforms the GKM algorithm. For this data set the MSKM is the most time consuming and the GKM is the least time-consuming algorithms.

Table 2.1: Results for Data set 1

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$4.100 \cdot 10^{11}$	33.609	$4.381 \cdot 10^{11}$	7.156	$4.377 \cdot 10^{11}$	9.609
5	$3.276 \cdot 10^{11}$	127.187	$3.383 \cdot 10^{11}$	28.688	$3.362 \cdot 10^{11}$	38.250
10	$2.804 \cdot 10^{11}$	164.859	$3.031 \cdot 10^{11}$	65.391	$2.879 \cdot 10^{11}$	91.250
15	$2.519 \cdot 10^{11}$	229.500	$2.716 \cdot 10^{11}$	103.312	$2.541 \cdot 10^{11}$	158.172
20	$2.270 \cdot 10^{11}$	263.109	$2.410 \cdot 10^{11}$	143.828	$2.207 \cdot 10^{11}$	241.094

2.3.2 Data set 2

For description of this data set, refer to Appendix A, Section A.2.

Results for this data set are presented in Table 2.2. Results presented demonstrate that the MSKM algorithm produces better results when the number of clusters $N \leq 10$. However as the number of clusters increases, MGKM outperforms the other two algorithms.

GKM requires less CPU time however its solutions are not good. MGKM requires significantly less CPU time than MSKM.

Table 2.2: Results for Data set 2

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$8.441 \cdot 10^{10}$	542.812	$8.441 \cdot 10^{10}$	59.312	$8.441 \cdot 10^{10}$	102.469
5	$6.644 \cdot 10^{10}$	1652.078	$6.769 \cdot 10^{10}$	240.391	$6.712 \cdot 10^{10}$	415.578
10	$5.703 \cdot 10^{10}$	2714.593	$6.094 \cdot 10^{10}$	545.188	$5.696 \cdot 10^{10}$	962.938
15	$5.467 \cdot 10^{10}$	4086.984	$5.556 \cdot 10^{10}$	862.453	$5.177 \cdot 10^{10}$	1543.297
20	$4.900 \cdot 10^{10}$	5016.281	$5.041 \cdot 10^{10}$	1199.984	$4.812 \cdot 10^{10}$	2150.469

2.3.3 Data set 3

For description of this data set, refer to Appendix A, Section A.3.

Results for this data set are presented in Table 2.3. One can see from these results that the MSKM algorithm outperforms two other algorithms when the number of clusters $N \leq 5$. For the number of clusters $N = 10, 15$ the MGKM algorithm outperforms others and for $N = 20$ the GKM algorithm achieved the best result. For this data set the MGKM is the most time consuming and the GKM is the least time-consuming algorithms.

Table 2.3: Results for Data set 3

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$4.429 \cdot 10^{10}$	0.031	$4.527 \cdot 10^{10}$	0.047	$4.527 \cdot 10^{10}$	0.062
5	$2.611 \cdot 10^{10}$	0.203	$2.919 \cdot 10^{10}$	0.109	$2.784 \cdot 10^{10}$	0.172
10	$1.935 \cdot 10^{10}$	0.796	$2.025 \cdot 10^{10}$	0.234	$1.883 \cdot 10^{10}$	0.625
15	$1.409 \cdot 10^{10}$	1.546	$1.373 \cdot 10^{10}$	0.375	$1.334 \cdot 10^{10}$	1.953
20	$1.085 \cdot 10^{10}$	2.000	$9.168 \cdot 10^{9}$	0.547	$9.392 \cdot 10^{9}$	4.234

2.3.4 Data set 4

For description of this data set, refer to Appendix A section A.4.

Results for this data set are presented in Table 2.4. One can see from this table that algorithms produce almost the same results when the number of clusters $N \leq 5$. However GKM requires significantly less CPU time. As the number of clusters increases, MGKM

produces better solutions than the other two algorithms. Again, MGKM requires less CPU time than MSKM.

Table 2.4: Results for Data set 4

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$9.212 \cdot 10^{10}$	0.812	$9.212 \cdot 10^{10}$	0.188	$9.212 \cdot 10^{10}$	0.297
5	$5.024 \cdot 10^{10}$	3.296	$5.032 \cdot 10^{10}$	0.609	$5.032 \cdot 10^{10}$	1.031
10	$3.424 \cdot 10^{10}$	6.703	$3.408 \cdot 10^{10}$	1.359	$3.351 \cdot 10^{10}$	2.875
15	$2.849 \cdot 10^{10}$	10.125	$2.897 \cdot 10^{10}$	2.156	$2.812 \cdot 10^{10}$	5.984
20	$2.470 \cdot 10^{10}$	11.421	$2.556 \cdot 10^{10}$	3.000	$2.422 \cdot 10^{10}$	10.234

2.3.5 Data set 5

For description of this data set, refer to Appendix A section A.5.

Results are presented in Table 2.5. We calculate maximum 10 clusters because this data set contains only 38 samples. Results from Table 2.5 show that MSKM produces better solutions than the other two algorithms, however it requires more computational time. MGKM produces better solutions than the GKM algorithm.

Table 2.5: Results for Data set 5

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$7.880 \cdot 10^{10}$	3.062	$8.137 \cdot 10^{10}$	0.578	$7.880 \cdot 10^{10}$	0.672
5	$5.537 \cdot 10^{10}$	8.171	$5.837 \cdot 10^{10}$	2.016	$5.729 \cdot 10^{10}$	2.641
10	$4.104 \cdot 10^{10}$	10.468	$4.399 \cdot 10^{10}$	4.594	$4.271 \cdot 10^{10}$	8.188
15	$2.954 \cdot 10^{10}$	13.578	$3.291 \cdot 10^{10}$	7.344	$3.002 \cdot 10^{10}$	19.188

2.3.6 Data set 6

For description of this data set, refer to Appendix A, Section A.6.

Computational results for this data set are presented in Table 2.6. We can see the MSKM algorithm outperforms other two algorithm when the number of clusters $N \leq 5$. However, the MGKM algorithm outperforms two other algorithms as the number of clusters increase. The GKM algorithm requires less computational time than two other algorithms and the MGKM is the most time-consuming among three algorithms.

Table 2.6: Results for Data set 6

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$3.622 \cdot 10^{10}$	5.140	$3.885 \cdot 10^{10}$	0.656	$3.885 \cdot 10^{10}$	0.891
5	$2.667 \cdot 10^{10}$	8.218	$2.707 \cdot 10^{10}$	2.328	$2.684 \cdot 10^{10}$	3.453
10	$1.894 \cdot 10^{10}$	14.343	$1.962 \cdot 10^{10}$	5.328	$1.893 \cdot 10^{10}$	11.109
15	$1.371 \cdot 10^{10}$	20.312	$1.370 \cdot 10^{10}$	8.594	$1.332 \cdot 10^{10}$	25.078
20	$9.986 \cdot 10^{9}$	25.578	$9.325 \cdot 10^{10}$	12.094	$9.153 \cdot 10^{9}$	45.281

2.3.7 Data set 7

For description of this data set, refer to Appendix A section A.7.

Computational results for this data set are presented in Table 2.7. For this data set, all three algorithms give the same solutions when the number of clusters $N \leq 5$. However for larger number of clusters, MGKM outperforms two other algorithms. The GKM algorithm requires the least CPU time and the MSKM algorithm requires the maximum CPU time.

Table 2.7: Results for Data set 7

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$2.777 \cdot 10^{13}$	7.468	$2.777 \cdot 10^{13}$	0.969	$2.777 \cdot 10^{13}$	1.812
5	$1.939 \cdot 10^{13}$	20.437	$1.939 \cdot 10^{13}$	3.547	$1.939 \cdot 10^{13}$	6.812
10	$1.671 \cdot 10^{13}$	36.437	$1.685 \cdot 10^{13}$	7.859	$1.626 \cdot 10^{13}$	15.203
15	$1.570 \cdot 10^{13}$	51.671	$1.555 \cdot 10^{13}$	12.344	$1.480 \cdot 10^{13}$	25.141
20	$1.534 \cdot 10^{13}$	60.359	$1.473 \cdot 10^{13}$	17.016	$1.364 \cdot 10^{13}$	36.094

2.3.8 Data set 8

For description of this data set, refer to Appendix A, Section A.8.

Results for this data set are given in Table 2.8. Results presented demonstrate that the three algorithms produce almost the same solutions when the number of clusters $N \leq 5$. As the number of clusters increases, the MGKM algorithm significantly outperforms the other algorithms. Again the GKM algorithms is the least time-consuming and the MSKM algorithm is the most time-consuming.

Table 2.8: Results for Data set 8

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$1.588 \cdot 10^{10}$	5.281	$1.589 \cdot 10^{10}$	0.703	$1.589 \cdot 10^{10}$	1.234
5	$1.068 \cdot 10^{10}$	24.296	$1.067 \cdot 10^{10}$	2.328	$1.067 \cdot 10^{10}$	4.469
10	$8.698 \cdot 10^{9}$	39.937	$8.797 \cdot 10^{9}$	5.266	$8.620 \cdot 10^{9}$	10.047
15	$8.595 \cdot 10^{9}$	50.671	$8.191 \cdot 10^{9}$	8.234	$7.813 \cdot 10^{9}$	15.609
20	$8.242 \cdot 10^{9}$	53.453	$7.656 \cdot 10^{9}$	11.234	$7.259 \cdot 10^{9}$	22.469

2.3.9 Data set 9

For description of this data set, refer to Appendix A section A.9.

Results for this data set are presented in Table 2.9. These results demonstrate that the MSKM algorithm outperforms other two algorithms when the number of clusters $N \leq 5$ and the MGKM algorithm outperforms two other algorithms for larger number clusters. Again the GKM algorithms is the least time-consuming and the MSKM algorithm is the most time-consuming.

Table 2.9: Results for Data set 9

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$2.073 \cdot 10^{11}$	0.671	$2.168 \cdot 10^{11}$	0.047	$2.168 \cdot 10^{11}$	0.062
5	$1.307 \cdot 10^{11}$	1.718	$1.417 \cdot 10^{11}$	0.125	$1.322 \cdot 10^{11}$	0.188
10	$1.040 \cdot 10^{11}$	2.890	$8.599 \cdot 10^{10}$	0.281	$8.505 \cdot 10^{10}$	0.844
15	$8.800 \cdot 10^{10}$	3.750	$5.885 \cdot 10^{10}$	0.453	$5.690 \cdot 10^{10}$	2.609
20	$8.975 \cdot 10^{10}$	5.171	$4.009 \cdot 10^{10}$	0.656	$4.015 \cdot 10^{10}$	5.609

2.3.10 Data set 10

For description of this data set, refer to Appendix A section A.10.

Computational results for this data set are presented in Table 2.10. Results from Table 2.10 demonstrate that for small number of clusters ($N \leq 5$) MSKM outperforms the other algorithms, however as the number of clusters increases GKM and MGKM produce better solutions. The MGKM algorithm is the best for larger numbers of clusters ($N \geq 10$). The MSKM algorithm is computationally more expensive and the GKM algorithm requires the least CPU time among all three algorithms.

Table 2.10: Results for Data set 10

No of clusters	MSKM		GKM		MGKM	
	f_{val}	CPU time	f_{val}	CPU time	f_{val}	CPU time
2	$1.554 \cdot 10^{11}$	2.156	$1.589 \cdot 10^{11}$	0.203	$1.582 \cdot 10^{11}$	0.359
5	$1.040 \cdot 10^{11}$	7.062	$1.064 \cdot 10^{11}$	0.688	$1.065 \cdot 10^{11}$	1.234
10	$6.553 \cdot 10^{10}$	14.281	$6.509 \cdot 10^{10}$	1.516	$6.327 \cdot 10^{10}$	2.688
15	$5.258 \cdot 10^{10}$	23.578	$4.614 \cdot 10^{10}$	2.438	$4.529 \cdot 10^{10}$	4.859
20	$4.760 \cdot 10^{10}$	29.781	$3.515 \cdot 10^{10}$	3.375	$3.489 \cdot 10^{10}$	8.266

2.3.11 Content of clusters

In order to compare clusters generated by different algorithms we use the notion of the cluster purity. The cluster purity is defined as follows:

$$P(A^i) = 100\frac{1}{n_{A^i}} \max_{j=1,\ldots,l} \; n^j_{A^i},$$

where $n_{A^i} = |A^i|$ is the cardinality of the cluster A^i, $n^j_{A^i}$ is the number of samples in the cluster A^i that belong to the true class j and l is the number of true classes. Then the total purity $P(A)$ for the data set A can be calculated as:

$$P(A) = \frac{n_{A^i} P(A^i)}{m}.$$

We used the data set 10. This data set contains 13 cancer types and therefore we calculated 30 clusters. Results are as follows.

- The MSKM algorithm produced 13 empty, 6 mixed and 11 pure clusters with total

purity $P(A) = 64.44$;

- The GKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 83.33$. In mixed clusters the results were as follows:
  - Cluster 1 - 17 tumors: breast(1), lung(2), colon(2), germinal center cells (1), bladder(1), uterus(2), kidney(3), pancreas(5);
  - Cluster 2 - 4 tumors: bladder(1), uterus(3);
  - Cluster 3 - 5 tumors: whole brain(2), cerebellum(3).
- The MGKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 85.56$. In mixed clusters the results were as follows:
  - Cluster 1 - 14 tumors: breast(1), lung(2), colon(1), bladder(2), kidney(3), pancreas(5);
  - Cluster 2 - 3 tumors: colon(1), germinal center cells (1), bladder(1).
  - Cluster 3 - 5 tumors: bladder(1), uterus(3), whole brain(1).

We can see that the MGKM algorithm generates better cluster structure than other two algorithms. Thus, we can say that the MGKM algorithm is efficient at solving cluster analysis problems in gene expression data sets and it produces better cluster structure of such data sets.

2.4 Conclusion

In this chapter we discussed the problems of cluster analysis in gene expression data sets. We considered the clustering problems in a sample space. Only a few clustering algorithms can be applied to solve such problems, including the k-means algorithm. However, this algorithm is inefficient and very sensitive to the choice of a starting point. The global k-means algorithm which has been introduced recently, is a significant improvement over the k-means algorithm. However, results of our computational experiments show that this algorithm cannot find a proper cluster structure in gene expression data sets. The main reason for this is the sparsity of the data in the sample space.

We developed a new version of the global k-means algorithm; the modified global k-means algorithm. This algorithm is very efficient for solving clustering problems in gene expression data sets.

We presented the results of numerical experiments on ten gene expression data sets to support this claim. These results clearly demonstrate that the modified global k-means algorithm outperforms the other two algorithms: the multi-start k-means and global k-means algorithms. However the modified global k-means algorithm is computationally more expensive than the global k-means algorithm.

Chapter 3

Gene selection algorithms

In this chapter we develop new gene selection algorithms for gene expression data sets. All these algorithms are based on the computation of overlaps between classes for a given gene or for a given group of genes.

3.1 Introduction

The gene expression patterns in microarray data have already provided some valuable insights into a variety of problems, and it is expected that the knowledge gleaned from microarray data will contribute significantly to advances in fundamental questions in biology as well as in clinical medicine. Microarrays provide large amounts of data about the inner life of a cell. These data sets contain thousands of genes. There are only a few genes that have features describing the cell and the rest of the genes have only very little information any at all. One can call those few genes as informative ones and the aim of the gene selection is to identify such genes. These informative genes can then be used to classify unknown tumours.

Feature (gene) selection can be applied to both supervised and unsupervised learning. Feature selection for unsupervised learning (clustering) is an interesting and complex issue (for more details, see [130]). In this thesis we do not consider this problem, but will concentrate on the problem of feature (gene) selection for supervised classification, where the class labels are known beforehand.

There are many reasons for the selection of a minimal subset of genes. We list here some of them:

1. A microarray might generate high dimensional data containing thousands or even tens of thousands of genes [122]. In general, in gene expression data sets the number of genes is two or even three orders of magnitudes more than the number of tumours and most genes are not relevant. Gene selection helps to reduce the number of non-informative genes and genes which contribute only noise [13]. This enables the development of cost-effective models.

2. The presence of the large number of genes in gene expression data sets reduces the generalization abilities of many classifiers. A large number of genes increases computational complexity. Many pattern recognition techniques were originally not designed to cope with large numbers of irrelevant features.

   The gene selection allows one to avoid overfitting and to improve model performance, that is prediction performance in the case of supervised classification and better cluster detection in the case of clustering [108].

3. The gene selection algorithms also may allow finding a subset of genes which might help to clarify how cancer is developing [69]. These genes can help us to gain a deeper insight into the underlying processes that generated the data [108].

Thus dimensionality reduction is a key to developing efficient tumour classification algorithms. Many gene selection algorithms have been proposed in the context of microarray data analysis over the last decade. These algorithms can be classified into three different groups: filter algorithms, wrapper algorithms and embedded algorithms. Currently there are a large number of gene selection algorithms. A review of some of these algorithms can be found in [108]. We now mention here some of those algorithms.

In the paper [132], the author develops a gene selection algorithm in gene expression based tumour classification. The authors suggest the use of a simple Fisher linear method for classification and heuristic stepwise and Monte Carlo methods for selecting the optimal subset of genes. Additionally, the authors of this paper compare the accuracy of four statistical procedures, in classifying tumours. These procedures are Stepwise discriminant analysis, Monte Carlo methods, t statistic and a PS statistic suggested in [55]. The results obtained show that the stepwise and Monte Carlo methods work similarly and both methods work better than t-statistic or PS statistics methods. Of the two previously mentioned methods, the stepwise method needs much less computational time, so stepwise discriminant

analysis provides a better method for tumour classification using gene expression profiles.

In [88] the authors propose a gene selection criterion for discriminant microarray data analysis based on extreme value distribution. Discriminant microarray data analysis compares and classifies expression levels of samples from two groups.

The paper [79] proposes a joint classifier and feature optimization algorithm (JCFO) for cancer diagnosis using gene expression data. This algorithm uses a sparse Bayesian approach to identify both the optimal non-linear classifier for diagnosis and the optimal subset of genes on which this diagnosis should be based. The algorithm is designed to automatically identify the small subsets of genes which have the highest discriminative information.

In the paper [80] the authors develop a gene selection algorithm using random forest approach and the scatter search methods. They apply a three-step process in order to select features and evaluate them:

1. Feature space reduction;
2. Feature subset optimization in order to select a small subset of near optimal features;
3. Evaluating the selected features in order to measure their efficiency and accuracy in classifying the samples in data set.

This approach uses a randomized process and consequently we get a forest of decision trees for feature selection. The more a feature appears in the nodes of the trees, the more informative it is in classifying the samples and it is assumed to be a more relevant gene.

The paper [13] introduces a gene selection algorithm for multi-class cancer diagnosis which is based on the notion of an overlap of a gene for all possible pairs of tumour classes. Genes that have the smallest overlaps for as many pairs as possible are chosen as informative genes.

The papers [8] and [82] propose hierarchical Bayesian models, the paper [39] considers a gene selection algorithm based on sparse logistic regression with Bayesian regularization and the paper [133] develops a Bayesian averaging method.

The papers [49] and [73] introduce algorithms based on clustering. An algorithm, which uses a hybrid of Pearson correlation coefficient and signal-to-noise ratio methods combined with an evolving classification function, is developed in the paper [54]. A semi-parametric two-sample test is proposed in [56] to find the most informative genes.

In the paper [69] different gene selection algorithms are proposed where genes of interest are selected by ranking them according to a test-statistic and then choosing the top k genes. A multivariate gene selection algorithm is developed in [84]. A recursive support vector machines approach is discussed in [85]. A combined genetic algorithm and the k-NN algorithm approach is proposed in [86]. An algorithm for gene selection using maximum likelihood is developed in [88]. The paper [116] introduces a gene selection algorithm based on logistic regression whereas the paper [124] proposes a mixed integer programming model which simultaneously selects genes and constructs a classification model.

A kernel-based framework for gene selection based on the Hilbert-Schmidt independence criterion and backward elimination, called BAHSIC, is defined in [118]. Gene selection algorithms based on support vector machines are proposed in [136] and [139]. The paper [138] proposes an algorithm based on least squares support vector machines and using least squares bounds. A comparative study of some gene expression algorithms is presented in [87].

One of the interesting areas in application of the mathematical methods in bioinformatics is the study of gene networks. Gene networks have been extensively studied in [53,128]. The parameter estimation on these networks is of Chebyshev approximation type, hence it is a semi-infinite optimization problem. See [53, 128] for more details.

In the next section we present new gene selection algorithms that are based on the use of hyperboxes containing classes. For a given gene or group of genes, we compute hyperboxes containing each class and then we compute overlaps between hyperboxes from different classes. Genes or groups of genes providing the smallest overlaps are identified as the most informative genes. The algorithms are the extension of the gene selection algorithm proposed in [13]. The new algorithms are filter algorithms which are fast and independent of the classifiers. This means they should be applied before the use of classifiers.

3.2 Definition of overlaps

In this section we define one-dimensional and multi-dimensional overlaps between different tumour types using expression levels of genes. Overlaps can be defined between two classes as well as between a given class and the rest of a data set. We start with the definition of overlaps between two classes.

3.2.1 Binary univariate overlaps

Suppose we are given a data set A which contains $m \geq 2$ classes, n_i tumours in the i-th class and p genes. We denote by d_{kj}^i the j-th gene expression value for k-th sample in the i-th class, where $i = 1, \ldots, m,\ j = 1, \ldots, p,\ k = 1, \ldots, n_i$ and introduce the following numbers:

$$a_{ij}^{min} = \min_{k=1,\ldots,n_i} d_{kj}^i,\ \ a_{ij}^{max} = \max_{k=1,\ldots,n_i} d_{kj}^i,$$

$$j = 1, \ldots, p,\ i = 1, \ldots, m.$$

Here a_{ij}^{min} and a_{ij}^{max} are the minimum and maximum expression values for the j-th gene in the i-th class, respectively. Then the j-th gene in the i-th class can be identified by a segment $[a_{ij}^{min}, a_{ij}^{max}]$. We call this segment the expression level segment of the j-th gene in the i-th class. For a given gene $j = 1, \ldots, p$ and two different classes i and l we define:

$$c_{1j}(i,l) = \max(a_{ij}^{min}, a_{lj}^{min}),\ \ c_{2j}(i,l) = \min(a_{ij}^{max}, a_{lj}^{max}),$$

$$e_{1j}(i,l) = \min(a_{ij}^{min}, a_{lj}^{min}),\ \ e_{2j}(i,l) = \max(a_{ij}^{max}, a_{lj}^{max}).$$

It is clear that the interval $[e_{1j}(i,l), e_{2j}(i,l)]$ contains expression levels of the j-th gene of all samples from classes i and l and the interval $[c_{1j}(i,l), c_{2j}(i,l)]$, if it is not empty, contains samples from both classes. Overlaps for the j-th gene between these two classes can be defined either using the length of both intervals or the number of samples whose the expression level of the j-th gene are in these intervals.

The use of the length of intervals.

Consider

$$b_1(i,l) = \max\left\{0, c_2(i,l) - c_1(i,l)\right\},\ \ b_2(i,l) = e_2(i,l) - e_1(i,l).$$

One can note that $b_1(i,l) = 0$ if the expression level segment of the j-th gene in classes i and l either has no intersection or their endpoints coincide. Always $b_2(i,l) \geq 0$ and $b_2(i,l) = 0$ if and only if $a_{ij}^{max} = a_{lj}^{max} = a_{ij}^{min} = a_{lj}^{min}$. Consider the following number

$$z = \left(a_{lj}^{min} - a_{ij}^{min})(a_{ij}^{max} - a_{lj}^{max}\right).$$

If $z \geq 0$ then either

$$[a_{lj}^{min}, a_{lj}^{max}] \subseteq [a_{ij}^{min}, a_{ij}^{max}]$$

or

$$[a_{ij}^{min}, a_{ij}^{max}] \subseteq [a_{lj}^{min}, a_{lj}^{max}].$$

In particular, if $b_2(i,l) = 0$ then $z = 0$.

The number

$$O_{il}^j = \begin{cases} 1, & z \geq 0, \\ \frac{b_1(i,l)}{b_2(i,l)}, & \text{otherwise.} \end{cases}$$

is said to be the overlap of the j-th gene between classes i and l.

The use of the number of samples.

Overlaps can also be defined using the number of samples in the interval $[c_{1j}(i,l), c_{2j}(i,l)]$. Consider the following sets:

$$Q_t = \left\{k = 1, \ldots, n_t : \;\; c_{1j}(i,l) \leq d_{kj}^t \leq c_{2j}(i,l)\right\}, \; t = i, l.$$

Let $q = |Q_i| + |Q_l|$ where $|Q_t|$ is the cardinality of the set $Q_t, \; t = i, l$. Then the number

$$O_{il}^j = \begin{cases} 1, & z \geq 0, \\ \frac{q}{n_i + n_l}, & \text{otherwise.} \end{cases}$$

is said to be the overlap of the j-th gene between classes i and l.

It is clear that $O_{il}^j = O_{li}^j$, $O_{il}^j \in [0,1]$ and $O_{il}^j = 1$ for any $j = 1, \ldots, p$ and $i, l = 1, \ldots, m$. Thus, we can define the following $m \times m$ matrix for the gene j:

$$O^j = \begin{pmatrix} 1 & O_{12}^j & O_{13}^j & \ldots & O_{1m}^j \\ O_{21}^j & 1 & O_{23}^j & \ldots & O_{2m}^j \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ O_{m1}^j & O_{m2}^j & O_{m3}^j & \ldots & 1 \end{pmatrix}.$$

O^j is a symmetric matrix.

3.2.2 One-Vs-All univariate overlaps

For a given class $i \in \{1, \ldots, m\}$ and gene $j \in \{1, \ldots, p\}$ we define

$$\bar{a}_{ij}^{min} = \min_{l=1,\ldots,m, l\neq i} a_{lj}^{min}, \;\; \bar{a}_{ij}^{max} = \max_{l=1,\ldots,m, l\neq i} a_{lj}^{max},$$

$$\bar{c}_{1j}(i) = \max\left(a_{ij}^{min}, \bar{a}_{ij}^{min}\right), \;\; \bar{c}_{2j}(i) = \min\left(a_{ij}^{max}, \bar{a}_{ij}^{max}\right),$$

$$\bar{e}_{1j}(i) = \min\left(a_{ij}^{min}, \bar{a}_{ij}^{min}\right), \;\; \bar{e}_{2j}(i) = \max\left(a_{ij}^{max}, \bar{a}_{ij}^{max}\right),$$

$$\bar{b}_1(i) = \max\left\{0, \bar{c}_2(i) - \bar{c}_1(i)\right\}, \;\; \bar{b}_2(i) = \bar{e}_2(i) - \bar{e}_1(i),$$

$$\bar{z} = \left(\bar{a}_{ij}^{min} - a_{ij}^{min}\right)\left(a_{ij}^{max} - \bar{a}_{ij}^{max}\right).$$

$$Q_{ti}^0 = \left\{k = 1, \ldots, n_t : \;\; \bar{c}_{1j}(i) \leq d_{kj}^t \leq \bar{c}_{2j}(i)\right\}, \; t = 1, \ldots, m,$$

$$\bar{Q}_i = \bigcup_{t=1}^{m} Q_{ti}^0, \;\; \bar{q} = |\bar{Q}_i|, \;\; n = \sum_{i=1}^{m} n_i.$$

Then we can define the overlap between the class i and the rest of the data set by

$$\bar{O}_i^j = \begin{cases} 1, & \bar{z} \geq 0, \\ \frac{\bar{b}_1(i)}{\bar{b}_2(i)}, & \text{otherwise.} \end{cases}$$

or by

$$\bar{O}_i^j = \begin{cases} 1, & z \geq 0, \\ \frac{\bar{q}}{n}, & \text{otherwise.} \end{cases}$$

Then we can define a vector of overlaps for a given gene j as follows:

$$\bar{O}^j = \left(\bar{O}_1^j, \ldots, \bar{O}_m^j\right).$$

3.2.3 Multi-dimensional overlaps

A hyperbox $B = [x, y], \; x, y \in R^n$ in n-dimensional space R^n is defined as follows:

$$B = \left\{x \in R^n : \; a_i \leq x_i \leq b_i, \; i = 1, \ldots, n\right\}. \tag{3.1}$$

The volume of the hyperbox B is defined as:

$$V(B) = \prod_{i=1}^{n}(x_i - y_i).$$

Assume that we are given two hyperboxes $B_1 = [x^1, y^1]$ and $B_2 = [x^2, y^2]$. Their intersection is empty if and only if there exists at least one $i \in \{1, \ldots, n\}$ such that either $x_i^1 > y_i^2$ or $x_i^2 > y_i^1$. In other words the intersection of two boxes B_1 and B_2 is empty if and only if

$$\max_{i=1,\ldots,n} \max\left\{x_i^1 - y_i^2, x_i^2 - y_i^1\right\} > 0.$$

This means that two hyperboxes B_1 and B_2 have an intersection if and only if:

$$\max_{i=1,\ldots,n} \max\left\{x_i^1 - y_i^2, x_i^2 - y_i^1\right\} \leq 0.$$

Then we get that $x_i^1 \leq y_i^2$ and $x_i^2 \leq y_i^1$ for all $i \in \{1, \ldots, n\}$ which implies that $\max\{x_i^1, x_i^2\} \leq \min\{y_i^1, y_i^2\}$ for all $i \in \{1, \ldots, n\}$. Two hyperboxes do not intersect if and only if $\max\{x_i^1, x_i^2\} > \min\{y_i^1, y_i^2\}$ for at least for one $i \in \{1, \ldots, n\}$.

The intersection of two hyperboxes B_1 and B_2 is also a hyperbox and it can be described as follows:

$$B_{12} = [\alpha, \beta],\ \alpha, \beta \in R^n$$

where $\alpha_i = \max\{x_i^1, x_i^2\}$ and $\beta_i = \min\{y_i^1, y_i^2\},\ i = 1, \ldots, n$.

Binary multi-dimensional overlaps

First we define the multi-dimensional overlaps between two classes i and $l,\ i, l \in \{1, \ldots, m\}$. Let $J \subset \{1, \ldots, p\}$ be a subset of genes and $J\{j_1, \ldots, j_n\},\ 0 < n \leq p$. Then the group of genes J in class t can be identified by the following n-dimensional hyperbox:

$$B_t^J = [x^t, y^t],\ \ x^t, y^t \in R^n,\ x_k^t = a_{tj_k}^{min},\ y_k^t = a_{tj_k}^{max},$$

$$k = 1, \ldots, n, t = i, l.$$

Let $B_{il}^J = B_i^J \bigcap B_l^J$. We assume that $V_{il}^0 = \max\{V(B_t^J),\ t = i, l\} > 0$. The overlap for the subset J between classes i and l is defined as

$$O_{il}^J = \begin{cases} 1, & B_i^J \subseteq B_l^J \text{ or } B_l^J \subseteq B_i^J, \\ \frac{V(B_{il}^J)}{V_{il}^0}, & \text{otherwise.} \end{cases}$$

We can also define the multi-dimensional overlaps using the number of samples in hyperbox B_{il}^J. Consider the set

$$Q_t^J = \{k = 1, \ldots, n_i :\ u^k = (d_{kj_1}^t, \ldots, d_{kj_n}^t) \in B_{il}^J\}.$$

Let $q = |Q_i^J| + |Q_l^J|$. Then

$$O_{il}^J = \begin{cases} 1, & B_i^J \subseteq B_l^J \text{ or } B_l^J \subseteq B_i^J, \\ \frac{q}{n_i + n_l}, & \text{otherwise.} \end{cases}$$

It is again clear that $O_{il}^J = O_{li}^J,\ O_{il}^J \in [0, 1]$ and $O_{il}^J = 1$ for any $J \subset \{1, \ldots, p\},\ J \neq \emptyset$ and $i, l = 1, \ldots, m$. Thus, we can define the following $m \times m$ matrix for the subset of genes J:

$$O^J = \begin{pmatrix} 1 & O_{12}^J & O_{13}^J & \ldots & O_{1m}^J \\ O_{21}^J & 1 & O_{23}^j & \ldots & O_{2m}^J \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ O_{m1}^J & O_{m2}^J & O_{m3}^J & \ldots & 1 \end{pmatrix}.$$

O^J is a symmetric matrix.

One-Vs-All multi-dimensional overlaps

We can define overlaps between a given class $i \in \{1, \ldots, m\}$ and the rest of the data set for a subset of genes J in a similar way as in the case of univariate overlaps. First we define the following hyperboxes:

$$B_i^J = [x^i, y^i],\ x^i, y^i \in R^n,\ x_k^i = a_{ij_k}^{min},\ y_k^i = a_{ij_k}^{max},$$

$$\bar{B}_i^J = [\bar{x}^i, \bar{y}^i],\ \bar{x}^i, \bar{y}^i \in R^n,\ \bar{x}_k^i = \bar{a}_{ij_k}^{min},\ \bar{y}_k^i = \bar{a}_{ij_k}^{max},$$

$$\bar{B}^0 = B_i^J \bigcap \bar{B}_i^J.$$

$$Q_t^{J0} = \left\{k = 1, \ldots, n_t : \ u^k = (d^t_{kj_1}, \ldots, d^t_{kj_n}) \in \bar{B}_0\right\},$$

$$\bar{Q}_i^J = \bigcup_{t=1}^{m} Q_t^{J0}, \ \ \bar{q} = |\bar{Q}_i^J|.$$

Let $V^0 = \max\{V(B_t^J), \ t = 1, \ldots, m\} > 0$. Then we can define the overlap for the subset J between the class i and the rest of the data set by

$$\bar{O}_i^J = \begin{cases} 1, & B_i^J \subseteq \bar{B}_i^J \text{ or } \bar{B}_i^J \subseteq B_i^J, \\ \frac{V(\bar{B}^0)}{V^0}, & \text{otherwise.} \end{cases}$$

or by

$$\bar{O}_i^J = \begin{cases} 1, & B_i^J \subseteq \bar{B}_l^J \text{ or } \bar{B}_l^J \subseteq B_i^J, \\ \frac{\bar{q}}{n}, & \text{otherwise.} \end{cases}$$

Then we can define a vector of overlaps for a given subset of genes J as follows:

$$\bar{O}^J = \left(\bar{O}_1^J, \ldots, \bar{O}_m^J\right).$$

3.3 Computation of informative genes

In this section we consider three different algorithms to compute the informative genes.

It is clear that the smaller the overlap the better a gene or a group of genes is for separation of different tumour types. Let

$$I(n) = \{J \in \{1, \ldots, p\} : \ \ |J| = n\}, \ \ 0 < n \leq p$$

be the set of all possible subsets of genes that contain n different genes.

The following algorithms can be used to determine the most informative genes. In all algorithms we will consider binary and one-vs-all overlaps.

Algorithm 5. The use of minimum overlaps

Binary overlaps. For any $J \in I(n)$ we define the following numbers

$$r_J = \max_{i=1,\ldots,m} \max_{l=i+1,\ldots,m} O_{il}^J$$

and

$$R = \min_{J \in I(n)} r_J.$$

We assume that $R \in [0, 1)$. The subset of genes $J \in I(n)$ is said to be the most informative subset if $r_J = R$.

One can take any tolerance $\varepsilon > 0$ such that $\varepsilon \leq 1 - R$ and define a subset of informative genes with respect to this tolerance. A subset of genes $J \in I(n)$ is a subset of informative genes with respect to the tolerance $\varepsilon > 0$ if

$$r_J \leq R + \varepsilon.$$

If $\varepsilon = 0$ we get the most informative genes and we get all genes as informative ones if $\varepsilon = 1 - R$. Increasing ε from 0 to $1 - R$ we can get a sequence of subsets with the increasing number of genes.

One-vs-all overlaps. Here we define

$$\bar{r}_J = \max_{i=1,\ldots,m} \bar{O}_i^J, \quad \bar{R} = \min_{J \in I(n)} \bar{r}_J.$$

and assume that $\bar{R} \in [0, 1)$. The subset of genes $J \in I(n)$ is said to be the most informative subset if

$$\bar{r}_J = \bar{R}.$$

Again we can take any tolerance $\varepsilon > 0$ such that $\varepsilon < 1 - \bar{R}$ and define a subset of informative genes with respect to this tolerance. A subset of genes $J \in I(n)$ is a subset of informative genes with respect to the tolerance $\varepsilon > 0$ if

$$\bar{r}_J \leq \bar{R} + \varepsilon.$$

Increasing ε from 0 to $1 - \bar{R}$ we get a sequence of subsets with the increasing number of genes, where $\varepsilon = 0$ corresponds to the subset of the most informative genes and $\varepsilon = 1 - \bar{R}$ corresponds to the whole set of genes.

Algorithm 6. The use of the sum of overlaps.

Binary overlaps. For each subset $J \in I(n)$ of genes we compute

$$f_J = \sum_{i=1}^{m} \sum_{k=i+1}^{m} O_{il}^J,$$

and

$$F = \min_{J \in I(n)} f_J.$$

The subset of genes $J \in I(n)$ is called the most informative subset if $f_J = F$.

Let $\varepsilon > 0$ be a given tolerance. Then a subset of genes $J \in I(n)$ is called a subset of informative genes with respect to $\varepsilon > 0$ if

$$f_J \leq F + \varepsilon.$$

One-vs-all overlaps. Here for given $J \in I(n)$ we compute

$$\bar{f}_J = \sum_{i=1}^{m} \bar{O}_i^J.$$

Let

$$\bar{F} = \min_{J \in I(n)} \bar{f}_J.$$

The subset $J \in I(n)$ is called the subset of most informative genes if

$$\bar{f}_J = \bar{F}.$$

Let $\varepsilon > 0$ be a any positive number. A subset $J \in I(n)$ is called a subset of informative genes with respect to $\varepsilon > 0$ if

$$\bar{f}_J \leq \bar{F} + \varepsilon.$$

In both cases $0 \leq \varepsilon < \infty$ and increasing ε from 0 to ∞ we get a sequence of subsets with the increasing number of genes, where we get the subset of the most informative genes if $\varepsilon = 0$ and the set of all of genes if ε is sufficiently large.

Algorithm 7. The use of the number of well-separated classes.

Binary overlaps. Let

$$\theta = \min_{i=1,\ldots,m} \min_{l=i+1,\ldots,m} O_{il}^J.$$

and $\alpha \in [\theta, 1]$. For the subset $J \in I(n)$ of genes we define the following set:

$$N_J(\alpha) = \left\{(i,l) :\ i = 1,\ldots,m,\ l = i+1,\ldots,m,\ O_{il}^J \leq \alpha\right\}.$$

Let

$$N_0 = \max_{J \in I(n)} |N_J(\alpha)|,$$

where $|Q|$ is the cardinality of the set Q. The subset J is called the subset of most informative genes if

$$|N_J(\alpha)| = N_0.$$

It is clear that $N_0 \leq \frac{m(m-1)}{2}$. Let $q > 0$ be any integer such that $0 \leq q \leq N_0$. A subset of genes $J \in I(n)$ is called a subset of informative genes with respect to the number q if

$$|N_J(\alpha)| \geq q.$$

One-vs-all overlaps. Let

$$\bar{\theta} = \min_{i=1,\ldots,m} \bar{O}_i^J,$$

and $\bar{\alpha} \in [\bar{\theta}, 1]$. For the subset $J \in I(n)$ of genes we define the following sets:

$$\bar{N}_J(\bar{\alpha}) = \left\{i \in \{1,\ldots,m\} :\ \bar{O}_i^J \leq \bar{\alpha}\right\}.$$

Let

$$\bar{N}_0 = \max_{J \in I(n)} |\bar{N}_J(\bar{\alpha})|.$$

The subset of genes J is called the most informative subset if

$$|\bar{N}_J(\alpha)| = \bar{N}_0.$$

It is clear that $\bar{N}_0 \leq m$. Let $\bar{q} > 0$ be any integer such that $\bar{q} \leq \bar{N}_0$. A subset $J \in I(n)$ of genes is called a subset of informative genes with respect to the number $\bar{q}$ if

$$|\bar{N}_J(\alpha)| \geq q.$$

We can get a sequence of subsets with increasing numbers of genes by increasing α ($\bar{\alpha}$)

from 0 to 1 and by decreasing $q(\bar{q})$ from $N_0(\bar{N}_0)$ to 0.

It should be noted that for $m = 2$ all algorithms coincide, that is they produce the same results. Furthermore, Algorithms 5 and 6 are the same in this case. However, for larger numbers of classes they may differ, sometimes significantly. If Algorithm 5 determines genes which are good for separation of a few classes only, Algorithms 6 and 7 are efficient in finding genes for the separation of all classes. Since most gene expression data sets contain more than two classes Algorithms 6 and 7 are more effective for computing informative genes in such data sets. Therefore in our computational experiments in the next section we will use only these two algorithms. Algorithm 6 tries to find genes with least overall overlaps and Algorithm 7 finds genes that are good for separating as many classes as possible.

3.4 Results of numerical experiments

To verify the effectiveness of the proposed algorithms we carried out numerical experiments using ten gene expression data sets. Numerical experiments have been carried out on a Pentium-4, 2.0 GHz, PC.

In numerical experiments we use only Algorithms 6 and 7 with both binary and one-vs-all overlaps. We consider subsets of genes with $n = 1, 2$, assuming that only pairs of genes may interact with each other. To compute overlaps we use the number of samples. Algorithm 5 does not give satisfactory results if the number of classes is large. Also the use of volumes of hyperboxes for the computation of overlaps in gene expression data sets is not always good since the volume of overlaps can quickly go to 0, while still containing relatively large number of samples.

We apply Algorithms 6 and 7 to find a sequence of subsets with increasing numbers of genes using different values of $\varepsilon > 0$ in Algorithm 6 and $\alpha, \bar{\alpha}, q$ and $\bar{q}$ in Algorithms 7. Then we apply the k-NN algorithm to data sets with the reduced number of genes to perform classification. Since the number of samples is not large the leave-one-out estimates are used in all data sets and we take $k = 1$.

The following versions of Algorithms 6 and 7 are applied in numerical experiments:

1. A2B - Algorithm 6 with binary overlaps and with $|J| = 1, 2$. (n=1,2)
2. A2O - Algorithm 6 with one-vs-all overlaps and with $|J| = 1, 2$. (n=1,2)
3. A3B - Algorithm 7 with binary overlaps and with $|J| = 1, 2$. (n=1,2)
4. A3O - Algorithm 7 with one-vs-all overlaps and with $|J| = 1, 2$. (n=1,2)

In all tables below we present the number of genes (N_g) and the classification accuracy on test sets. In these tables the best results for a number of genes are given in bold font and overall best results are given in italic bold font.

3.4.1 Data set 1

For description of this data set refer to Appendix A section A.1. Results for this data set are presented in Table 3.1.

Table 3.1: Results for Data set 1

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	**97.9**	-	**97.9**	-	**97.9**	-	**97.9**	-
2	**99.0**	96.9	**99.0**	96.9	**99.0**	91.7	**99.0**	91.7
3	**99.0**	-	**99.0**	-	**99.0**	-	**99.0**	-
4	97.9	**99.0**	97.9	**99.0**	97.9	94.8	97.9	94.8
5	***100.0***	-	***100.0***	-	***100.0***	-	***100.0***	-
6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	93.8	**100.0**	93.8
10	100.0	99.0	100.0	99.0	100.0	94.8	100.0	94.8
3000	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
7129	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0

One can see from Table 3.1 that algorithms with one-dimensional overlaps could find two genes that provide the same accuracy as the use of all 7129 genes. Moreover, they found 5 genes that give 100 $\%$ accuracy for supervised classification. They are genes MYRL2, SMARCA3, XDH, FHL3 and RHAG. However, algorithms with two-dimensional overlaps (except algorithms A3B with $n = 2$ and binary overlaps) could find 6 genes with the same results. We can conclude that algorithms with $n = 1$ and binary overlaps are more efficient in finding informative genes in this data set.

3.4.2 Data set 2

For description of this data set refer to Appendix A, Section A.2. Results for this data set are presented in Table 3.2.

Results presented in Table 3.2 show that Algorithm A2B with both $n = 1$ and $n = 2$ is quite effective in finding the most informative genes. We can see that they can find 50 genes which have greater accuracy than the use all 12625 genes. The most informative genes are among those 50 genes, because further increase of the number of genes does not lead to significant improvement in accuracy. The greatest accuracy was achieved using 150 genes that were found by Algorithm A2B with $n = 2$. This means the use of multi-dimensional overlaps may produce better results.

Table 3.2: Results for Data set 2

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	**70.8**	-	61.9	-	59.9	-	65.3	-
2	71.3	**78.2**	67.3	**78.2**	70.3	68.8	67.3	74.8
4	76.7	80.2	73.8	**83.2**	79.7	70.8	73.3	76.2
10	80.2	84.7	79.2	85.1	83.2	**87.1**	73.3	86.1
50	92.1	**93.6**	91.6	89.6	92.1	90.6	82.2	91.1
80	**95.5**	94.6	90.6	93.1	94.1	88.6	87.1	89.6
100	**96.0**	94.6	90.6	94.1	94.1	86.1	83.2	89.6
150	93.1	***96.5***	89.6	92.1	89.6	89.6	82.2	93.1
250	92.1	93.6	92.6	94.6	92.1	87.6	81.7	91.1
500	93.1	94.1	92.1	94.1	92.1	84.7	84.7	92.6
1000	93.6	93.6	94.6	93.6	94.1	87.6	85.6	92.6
12625	88.1	88.1	88.1	88.1	88.1	88.1	88.1	88.1

3.4.3 Data set 3

For description of this data set refer to Appendix A section A.3. Results for this data set are presented in Table 3.3.

Table 3.3: Results for Data set 3

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	73.7	-	73.7	-	**76.3**	-	**76.3**	-
2	55.3	**94.7**	55.3	84.2	65.8	78.9	65.8	65.8
3	**76.3**	-	**76.3**	-	68.4	-	68.4	-
4	76.3	92.1	71.1	89.5	**94.7**	60.5	**94.7**	81.6
6	86.8	**97.4**	89.5	**97.4**	92.1	78.9	86.8	86.8
10	86.8	94.7	***100.0***	97.4	92.1	84.2	94.7	78.9
14	97.4	100.0	100.0	94.7	97.4	86.8	97.4	81.6
18	100.0	94.7	100.0	94.7	100.0	92.1	94.7	89.5
20	100.0	94.7	94.7	94.7	100.0	89.5	94.7	89.5
100	97.4	100.0	100.0	100.0	92.1	94.7	89.5	89.5
999	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

We can see from Table 3.3 that for this data set algorithms with univariate overlaps are also successful in finding the most informative genes. A3B and A3O with $n = 1$ could find 4 genes that provide very high (94.7 $\%$) classification accuracy. However, Algorithm A2B with $n = 2$ found two genes with the same accuracy. Algorithm A2O could achieve

100 % classification accuracy with only 10 genes. These genes are No. 164, 309, 391, 548, 600, 603, 606, 686, 769 and 925 in the list of genes of this data set. We can conclude that algorithms with both binary and one-vs-all overlaps are efficient in finding informative genes in this data set.

3.4.4 Data set 4

For description of this data set refer to Appendix A section A.4. Results for this data set are presented in Table 3.4.

Table 3.4: Results for Data set 4

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	47.6	-	**51.5**	-	43.7	-	43.7	-
2	74.8	**77.7**	57.3	71.8	52.4	67.0	52.4	62.1
3	**80.6**	-	58.3	-	59.2	-	59.2	-
4	86.4	**91.3**	68.0	86.4	57.3	72.8	57.3	86.4
6	91.3	**92.2**	73.8	83.5	59.2	77.7	59.2	87.4
10	93.2	**95.1**	68.9	88.3	77.7	78.4	64.1	89.3
20	94.2	**98.1**	86.4	89.3	92.2	92.2	85.4	96.1
50	97.1	***99.0***	91.3	96.1	91.3	96.1	***99.0***	***99.0***
100	99.0	98.1	97.1	98.1	97.1	98.1	96.1	98.1
200	99.0	98.1	98.1	98.1	98.1	99.0	98.1	98.1
1000	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0

Results presented in Table 3.4 show that Algorithm A2B with $n = 2$ could find the smallest number of most informative genes. 10 genes found by this algorithm can be considered as the most informative genes since they can produce quite high classification accuracy. These genes are No. 189, 242, 407, 451, 560, 631, 684, 800, 984 and 989 in the list of genes of this data set. However, the comprehensive list of most informative genes is among the first 50 genes found by Algorithm A2B as well as by Algorithm A3O with both $n = 1$ and $n = 2$.

3.4.5 Data set 5

For description of this data set refer to Appendix A section A.5. Results for this data set are presented in Table 3.5.

Table 3.5: Results for Data set 5

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	**65.8**	-	**65.8**	-	**65.8**	-	**65.8**	-
2	76.3	**84.2**	76.3	**84.2**	71.1	52.6	71.1	52.6
4	76.3	**78.9**	76.3	**78.9**	76.3	73.7	76.3	73.7
10	**81.6**	**81.6**	**81.6**	**81.6**	63.2	55.3	63.2	55.3
30	***84.2***	76.3	***84.2***	76.3	60.5	57.9	60.5	57.9
50	78.9	71.1	78.9	71.1	60.5	57.9	60.5	57.9
200	76.3	78.9	76.3	78.9	68.4	71.1	68.4	71.1
500	71.1	84.2	71.1	84.2	63.2	63.2	63.2	63.2
5000	68.4	68.4	68.4	68.4	68.4	68.4	68.4	68.4

We can see from Table 3.5 that Algorithms A2B and A2O with both $n = 1$ and $n = 2$ were effective in finding the most informative genes for Data set 5. Just two genes found by these algorithms have greater classification accuracy than the use of all 5000 genes. These genes are No. 1332 and 1851. Recall that in these algorithms we use the sum overlaps between classes whereas in Algorithms A3B and A3O we use the number of well separated classes. Although 10 genes found by Algorithms A2B and A2O can provide very good classification accuracy the most genes are among the first 30 genes.

3.4.6 Data set 6

For description of this data set refer to Appendix A section A.6. Computational results for this data set are presented in Table 3.6.

One can see from Table 3.6 that results for this data set are mixed. However, we can see that algorithms with the sum of overlaps (A2B and A2O) work better than algorithms with hihgest number of well-separated classes (A3B and A3O). Algorithm A2O could find 200 most informative genes with highest classification accuracy reducing the number of genes almost 30 times. The classification accuracy is significantly better than that for 5893 genes. Algorithms could find 2 genes (Genes No. 3539 and 4412) which give the same classification accuracy as the use of all 5893 genes.

Table 3.6: Results for Data set 6

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	**61.8**	-	**61.8**	-	26.5	-	26.5	-
2	**52.9**	**52.9**	41.2	**52.9**	**52.9**	32.4	**52.9**	50.0
10	44.1	44.1	47.1	52.9	23.5	32.4	32.4	**61.8**
50	52.9	50.0	61.8	61.8	50.0	52.9	44.1	**67.6**
100	55.9	58.8	58.8	**64.7**	50.0	55.9	41.2	47.1
200	61.8	55.9	***76.5***	58.8	50.0	44.1	41.2	47.1
300	70.6	55.9	73.5	55.9	50.0	38.2	41.2	47.1
400	64.7	58.8	76.5	58.8	47.1	44.1	35.3	41.2
500	67.6	58.8	73.5	55.9	41.2	41.2	32.4	41.2
1000	70.6	52.9	70.6	52.9	32.4	41.2	35.3	38.2
2000	67.6	58.8	58.8	55.9	47.1	41.2	41.2	32.4
5893	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9

3.4.7 Data set 7

For description of this data set refer to Appendix A section A.7. Results for this data set are presented in Table 3.7.

Table 3.7: Results for Data set 7

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	**50.8**	-	48.4	-	41.5	-	41.5	-
2	70.2	**71.0**	66.5	**71.0**	53.6	62.1	53.6	61.3
4	80.2	82.3	81.0	**85.1**	62.5	74.6	72.6	74.6
10	91.1	92.7	88.7	**96.0**	79.8	76.2	85.1	85.5
20	94.0	**97.2**	90.7	96.0	91.5	85.5	85.9	88.3
50	98.0	**99.2**	96.0	98.4	98.4	96.8	93.1	96.0
100	99.2	**99.2**	98.8	98.8	98.4	98.4	96.4	96.8
200	99.2	***100.0***	99.2	***100.0***	98.4	98.4	98.0	96.4
250	99.2	100.0	99.6	99.6	99.2	98.4	98.4	96.4
300	99.2	100.0	100.0	100.0	98.8	98.0	98.4	96.4
985	98.8	98.8	98.8	98.8	98.8	98.8	98.8	98.8

Results presented in Table 3.7 show that algorithms with the sum of overlaps (A2B and A2O with $n = 2$) work better than algorithm with the highest number of well-separated classes. It is likely that the most informative genes are among the first 20 genes found by Algorithm A2B with $n = 2$. However, 100 $\%$ classification accuracy was achieved by 200

genes found by both Algorithm A2B with $n = 2$ and Algorithm A2O with $n = 2$.

3.4.8 Data set 8

For description of this data set refer to Appendix A section A.8. Results for this data set are presented in Table 3.8.

Table 3.8: Results for Data set 8

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
2	80.2	**87.8**	77.7	87.3	73.6	80.2	73.6	82.7
4	86.8	**90.7**	78.7	88.8	80.2	80.2	80.2	82.2
10	86.8	**94.4**	85.8	92.4	86.3	84.3	86.3	86.8
20	91.9	91.4	91.4	**93.4**	86.8	91.4	87.3	88.8
50	92.4	94.4	90.4	**95.4**	90.9	92.9	91.4	90.4
100	93.4	92.4	95.4	94.4	**95.9**	**95.9**	94.9	94.4
150	94.9	**96.4**	**96.4**	94.9	95.9	95.4	94.9	93.9
200	96.4	***98.0***	97.0	94.9	94.9	96.4	94.9	94.4
250	95.9	96.4	96.4	95.4	94.9	94.9	94.9	95.9
500	95.9	95.4	94.4	96.4	97.0	94.4	93.9	95.4
1000	93.9	93.9	93.9	93.9	93.9	93.9	93.9	93.9

We can see from Table 3.8 that Algorithms A2B and A2O with $n = 2$ are the most successful in finding the most informative genes. Although the best classification accuracy was achieved using 200 genes found by Algorithm A2B with $n = 2$ it is likely that the most informative genes are among the first 10 genes found by this algorithm. These genes provide better classification accuracy than the use of all 1000 genes and this result is close to the best accuracy for 200 genes. The ten most informative genes are genes No. 166, 291, 437, 563, 608, 616, 624, 867, 986 and 991. Again for this data set algorithms that use the sum overlaps performed better than algorithms that used the number of well-separated classes.

3.4.9 Data set 9

For description of this data set please refer to Appendix A section A.9. Results for this data set are given in Table 3.9.

Results presented in Table 3.9 show that Algorithm A2B and A2O with $n = 2$ could find informative genes. Algorithm A2B found 20 most informative genes and these genes

Table 3.9: Results for Data set 9

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	38.1	-	38.1	-	35.7	-	**45.2**	-
2	61.9	52.4	57.1	52.4	54.8	**69.0**	66.7	47.6
4	66.7	**76.2**	61.9	**76.2**	66.7	73.8	57.1	64.3
10	**88.1**	**88.1**	76.2	**88.1**	59.5	69.0	66.7	73.8
20	90.5	***97.6***	83.3	88.1	73.8	90.5	66.7	90.5
50	90.5	97.6	88.1	95.2	71.4	90.5	66.7	92.9
100	92.9	92.9	85.7	95.2	73.8	90.5	81.0	92.9
150	92.9	92.9	88.1	92.9	73.8	88.1	76.2	92.9
200	90.5	90.5	88.1	95.2	78.6	92.9	85.7	88.1
500	88.1	92.9	90.5	92.9	81.0	85.7	83.3	85.7
989	81.0	81.0	81.0	81.0	81.0	81.0	81.0	81.0

produce significantly better classification accuracy than the use of all 989 genes. Again we can see that these algorithms can substantially reduce the number of genes and significantly improve the performance of the k-NN algorithm. Again algorithms using the sum of overlaps performed better than algorithms using the number of well-separated classes.

3.4.10 Data set 10

For description of this data set refer to Appendix A section A.10. Results for this data set are presented in Table 3.10.

One can see from Table 3.10 that for Data set 10 Algorithms A2B with $n = 2$ and A3B with both $n = 1$ and $n = 2$ outperform other algorithms. We also note that the results for Algorithm A2B with $n = 2$ are also consistent and good. Fifthy genes found by Algorithm A2B with $n = 2$ and Algorithm A3B with $n = 1$ produce better classification accuracy than the use of all 1277 genes. However, the most informative genes are among 200 genes found by Algorithm A2B with $n = 2$.

Table 3.10: Results for Data set 10

	A2B		A2O		A3B		A3O	
N_g	1	2	1	2	1	2	1	2
1	12.2	-	14.4	-	12.2	-	**22.2**	-
2	**42.2**	27.8	25.6	27.8	**42.2**	35.6	25.6	20.0
4	57.8	46.7	36.7	55.6	52.2	**61.1**	23.3	25.6
10	72.2	75.6	50.0	73.3	71.1	**84.4**	38.9	54.4
20	77.8	**85.6**	58.9	80.0	74.4	**85.6**	51.1	52.2
50	85.6	**88.9**	66.7	86.7	**88.9**	86.7	75.6	65.6
100	88.9	**91.1**	81.1	89.8	**91.1**	90.0	73.3	74.4
150	91.1	**92.2**	84.4	91.1	**92.2**	90.0	85.6	83.3
200	94.4	***95.6***	88.9	94.4	92.2	93.3	84.4	82.2
250	93.3	92.2	85.6	92.2	91.1	93.3	86.7	87.8
500	92.2	93.3	87.8	93.3	90.0	92.2	85.6	86.7
1277	87.8	87.8	87.8	87.8	87.8	87.8	87.8	87.8

3.5 Conclusion

In this chapter we developed new gene selection algorithms. These algorithms are based on the use of the overlaps between different tumour types (classes). We introduced different types of the overlaps: the univariate and multi-dimensional overlaps as well as the binary and one-vs-all overlaps. In the case of the univariate overlaps we consider genes separately and compute segments of the expression levels of genes for each class. For each gene the overlap between classes is computed as the overlap between these segments.

Multi-dimensional overlaps are defined for the subsets of genes. In this case for each subset of genes we compute the hyperboxes of expression levels and then overlaps are computed as the overlaps between these hyperboxes. One can compute the overlaps between two classes (binary overlaps) as well as between a given class and the rest of a data set (one-vs-all).

We ranked genes or group of genes using either the total sum of overlaps or the number of "well-separated" classes. The definition of "well-separated" classes depends on a data set. Genes or groups of genes with small overlaps can be considered as the informative genes. The use of different overlaps leads to the different gene selection algorithms.

We tested our algorithms on ten publicly available gene expression data sets. The k-NN algorithm was applied to validate the results obtained by gene selection algorithms. It is well know that this algorithm is one of the most efficient classification algorithms. We considered the genes separately (in this case $n = 1$) or pairs of genes (in this case $n = 2$), assuming that the genes may interact pairwise or not interact at all. Results of the numerical experiments clearly demonstrate that the proposed algorithms are able to find a subset with a small number of genes and with a high classification accuracy. Results also show that the use of the sum of overlaps leads to the design of better gene selection algorithms. Results obtained by these algorithms are more consistent. They can significantly reduce the number of genes and substantially improve performance of the k-NN algorithm. However, for some gene expression data sets, algorithms with use of the number of "well-separated" classes can be also useful.

It should be noted that univariate versions of the overlapping algorithms are very fast. It requires only a few seconds on PC Pentium 4 with CPU of 1.83 GHZ and RAM of 1 GB to compare a subset of the most informative genes. The overlapping algorithm with two-dimensional bases, requires a reasonable CPU time, however, as the dimension of the

hyperboxes increase, CPU time increases drastically too, and the algorithm is not applicable with high dimensional hyperboxes in large scale gene expression data sets.

Chapter 4

Classification algorithm for gene expression data sets

In this chapter we introduce an algorithm for solving supervised data classification problems in gene expression data sets. This algorithm is especially effective when the number of genes is very small. Therefore we will apply the new algorithm to gene expression data sets together with gene selection algorithms from the previous chapter.

4.1 Introduction

The aim of classification or supervised learning is to determine whether an object belongs to a certain class. Classification of patients into existing disease classes using gene expression information is a typical application. In microarray analysis, classification is used to predict sample phenotypes based on gene expression patterns. Classifiers based on gene expression normally predict that a certain percentage of individuals that have a given expression profile will also have the phenotype of interest [100]. When working with complex data variables (features), such as what might be seen in large, noisy and incomplete microarray data sets, supervised methods are more efficient than the unsupervised ones.

The term classification in its broadest sense covers any context in which some decision or forecast is made according to currently available information [92]. Classification procedures include some formal methods in order to make judgment in new situations. More strictly, classification is constructing a procedure that will be applied to continuing cases, and the aim is to assign each new case to one of pre-defined classes on the basis of observed

attributes or features. The construction of a classification procedure using a set of data for which the true classes are known is called pattern recognition, or supervised learning. An example of this is assigning a credit status to an individual on the basis of financial and personal information.

Three main approaches that historically have been applied in this area include: statistical approaches, machine learning and neural networks.

Statistical approaches are generally characterised by inclusion of a probability model. This model provides the classification as well as the probability of belonging to a particular class. Since techniques are used by humans, some intervention in variable selection or structuring the problem is expected.

Classification within the statistical community has occurred in two main phases. The first phase is the classical phase which focuses on derivatives of Fisher's early work on linear discrimination. The second phase known as the modern phase, uses more classes of models which try to provide an estimation of the joint distribution of the features within each class, which in turn can be used for developing a classification rule.

Machine learning includes computing procedures that are based on logical or binary operations, and which learn a task from a series of examples. Machine learning tries to make classifying expressions simple enough to be understood by the human. They try to mimic human reasoning in order to provide insight into the decision process. Machine learning uses background knowledge, as statistical approaches use, however operation is conducted without human intervention. Machine learning focuses on decision-tree approaches, in which classification is a result of a sequence of logical steps.

Neural networks have different applications ranging from understanding and imitating the human brain, to practical scientific, commercial and engineering disciplines of pattern recognition, modelling, and prediction.

Neural networks might include different techniques however they all include layers of interconnected nodes, where each node produces a non-linear function of its input. The input to a node might come from the input data or from other nodes. A complete network represents a complex set of interdependencies that may incorporate any degree of nonlinearity, which allows general functions to be modelled. It has been argued that to a certain extent neural networks mirror the behaviour of networks of neurons in the brain.

Optimization based classification algorithms are based on the separation of known

classes by means of certain, not necessarily linear, functions. Classification algorithms based on linear separability have been developed in work by Bennet and her colleagues work [24, 25].

Over the last decade different approaches have been proposed to find piecewise linear functions separating two sets. Bennet et al. [27] develop the bilinear separability concept where two hyperplanes are used to separate sets. Astorino et al. [7] introduce the concept of polyhedral separability. In the latter case, one of the sets is approximated by a polyhedral set and the rest of the space is used to approximate the second set. The number of hyperplanes is not restricted, however the piecewise linear function is polyhedral, that is it is convex. However in many real situations, sets cannot be separated using only a few hyperplanes nor by using convex piecewise linear functions.

Support Vector Machines algorithms have been developed by Burges [35], Vapnik [129] and Thorsten [126].

An algorithm based on polyhedral separability has been introduced by Astorino and colleagues [7] and another algorithm based on max-min separability has been developed by Bagirov [18].

It should be mentioned that among these algorithms, only Support Vector Machines algorithms have been applied to gene expression data sets.

Supervised microarray data analysis (like any supervised data analysis process) includes four stages:

1. Construction of a classifier or model: We need a set of genes (training set), functional classes to which these genes belong (dependent variables), and independent variables that describe characteristics of the genes.

2. A learning phase: Training data are analysed by a classification algorithm.

3. A testing phase: The test data are used to assess the accuracy of the classifier.

4. An application phase: Classifier predicts the class label of the unknown gene expression values. There are other methods to analyse microarray data including linear discriminant analysis, decision trees, nearest neighbours, support vector machine.

Validation will require the use of data other than those used to develop the classifier. Validation issues arise including questions regarding the applicability of the new algorithms

to new individuals. [122]. The use of training and test sets and cross validation are the methods that are used to overcome this problem.

Training and test sets are not very good for validating classifiers with small data sets. Cross validation or leave-one-out estimators can be used in such cases.

The use of one training set and one test sets is the most commonly used method for validating the results of a classification algorithm. In this method, for example, two thirds of the data are used to train the algorithm. An algorithm is optimized to classify the training data. After training, the remaining part of the data might be used for verification and quantification of the success of the algorithm.

For middle-sized data sets, the most commonly used validation method is cross- validation. This method has a *fold* associated with it that determines how the algorithm is implemented. k-fold cross validation divides the data randomly into k equal parts. By running the algorithm k times, $k-1$ of the parts are used as a training set, and the other part as a test set. Every time the algorithm is run, a different test set is used so over k runs of the algorithm, all data are used as a test set.

For small data sets (with several hundred samples or fewer) leave-one-out validation method is used. In this method each time one data point is used as a test set and the rest of the data set is used as a training set. This procedure is repeated for all possible samples.

4.2 Supervised data analysis

Because the expression data sets contain a large number of genes, not all of the above mentioned classification algorithms can be directly applied to them. However, most of the algorithms can be applied along with gene selection algorithms. Over the last decade a large number of papers were devoted to the design of algorithms for solving problems of supervised data classification in gene expression data sets. It is not possible to cover all these papers here. We will concentrate only on those which are widely applied. Due to some similarities between support vector machines algorithms (SVM) and the algorithm proposed in this chapter, we will mention papers which apply SVM to gene expression data sets. Review of some of these algorithms can be found in [106].

One of the methods for tumour classification is molecular diagnostics, which offers the promise of precise, objective, and systematic cancer classification. However, these tests are

not widely applied because characteristic molecular markers for most solid tumours have yet to be identified [40]. The second method is the use of DNA microarray-based tumour gene expression profiles. Recently, they have been used for cancer diagnosis. In Alizadeh et al. [3], Bittner et al. [29], Dhanasekaran et al. [46], Golub et al. [55], Hedenfalk [65], Perou et al. [103] studies have been limited to a few cancer types and have spanned multiple technology platforms thereby complicating comparison among different data sets.

The paper [36] proposes Pattern Classification Program (PCP), which is an open-source machine learning program for supervised classification of patterns. The implementation of this program integrates gene selection and tumour prediction stages.

The paper [42] shows that the modified t-statistics and shrunken centroids employed by The Prediction Analysis of Microarrays tend to increase misclassification error when compared with their simpler counterparts. Based on these observations, the author proposes a classification method called Classification to Nearest Centroids, which ranks genes by standard t-statistics, does not shrink centroids and uses a class-specific gene-selection procedure.

In the paper [99] the authors present the classification algorithm based on partial least squares. This algorithm was applied to four gene expression data sets with multiple classes: a hereditary breast cancer data set with BRCA1-mutation, BRCA2-mutation and sporadic breast cancer samples; an acute leukaemia data set with: acute myeloid leukaemia (AML), T-cell acute lymphoblastic leukemia (T-ALL) and B-cell acute lymphoblastic leukaemia (B-ALL) samples; a lymphoma data set with: diffuse large B-cell lymphoma, B-cell chronic lymphocytic leukemia and follicular lymphoma (FL) samples; and the NCI60 data set with cell lines derived from cancers of various sites of origin.

The paper [45] proposes a modification of the generic boosting algorithm to improve its classification accuracy in the context of gene expression data. In particular, the authors present a feature pre-selection method, a more robust boosting procedure and a new approach for multi-categorical problems. This leads to significant improvement in the performance of the boosting algorithm.

In the paper [94] an approach called VizRank is applied to score and rank point-based visualizations according to the degree of separation of data instances of different classes. The paper [90] proposes an algorithm where a simultaneous reduction of genes and classification of tumours is applied. This algorithm tries to identify genes that are able to

distinguish between two different classes of tissue samples.

In [22] six machine learning techniques have been investigated for their classification accuracy focusing on two metabolic disorders, phenylketo nuria and medium-chain acyl-CoA dehydrogenase deficiency. The paper [105] considers spectral pattern comparison methods for tumour classification. In [97] the kernel-based Naive Bayesian algorithm is developed for breast cancer prediction.

Support vector machines algorithms are among the most commonly applied algorithms based on optimization techniques. In [107] a support vector machines algorithm has been applied to solving the classification of tumorus based on gene expression data gathered from microarray analysis. The paper [121] proposes a new multi-category support vector machines algorithm. Results presented demonstrate that this algorithm outperforms a number of popular machine learning algorithms, including the k-NN algorithm. In the paper [96] the SVM algorithm with automatic kernel selection was applied to tumour classification. The paper [51] discusses the computational complexity of the SVM algorithms and presents an algorithm to reduce the number of support vectors. In the paper [43] presents a comparison of different kernels in the SVM algorithms.

4.3 Max-min separability concept

The concept of max-min separability was introduced in [12]. In this approach two classes are separated using a piecewise linear function. Since a continuous piecewise linear function can be represented as a max-min of linear functions, such separability is called max-min separability. This function need not to be convex. It is proved that any two finite point sets can be separated by a piecewise linear function. Results presented in [18] demonstrate that an algorithm based on max-min separability is an efficient algorithm for solving supervised data classification problems in many large scale data sets. We summarise papers [12] and [18] to describe the concept of max-min separability. We start with the definition of linear and polyhedral separability. The max-min separability is the generalisation of both linear and polyhedral separability.

4.3.1 Linear separability

Let A and B be given sets containing m and p n-dimensional vectors, respectively:

$$A = \{a^1, \ldots, a^m\},\ a^i \in \mathbb{R}^n,\ i = 1, \ldots, m,$$

$$B = \{b^1, \ldots, b^p\},\ b^j \in \mathbb{R}^n,\ j = 1, \ldots, p.$$

The sets A and B are linearly separable if there exists a hyperplane $\{x, y\}$, with $x \in \mathbb{R}^n,\ y \in \mathbb{R}^1$ such that

1) for any $j = 1, \ldots, m$

$$\langle x, a^j \rangle - y < 0,$$

2) for any $k = 1, \ldots, p$

$$\langle x, b^k \rangle - y > 0.$$

The sets A and B are linearly separable if and only if $\operatorname{co} A \bigcap \operatorname{co} B = \emptyset$.

In practice, it is unlikely for the two sets to be linearly separable. Therefore it is important to find a hyperplane that minimizes some misclassification cost. In [26] the problem of finding this hyperplane is formulated as the following optimization problem:

$$\text{minimize } f(x, y) \text{ subject to } (x, y) \in \mathbb{R}^{n+1} \tag{4.1}$$

where

$$f(x, y) = \frac{1}{m} \sum_{i=1}^{m} \max\left(0, \langle x, a^i \rangle - y + 1\right) + \frac{1}{p} \sum_{j=1}^{p} \max\left(0, -\langle x, b^j \rangle + y + 1\right)$$

is an error function. Here $\langle \cdot, \cdot \rangle$ stands for the scalar product in $\mathbb{R}^n$. An algorithm for solving this problem (4.1) was described in [26] . It was shown that the problem (4.1) is equivalent to the following linear program:

$$\text{minimize } \frac{1}{m} \sum_{i=1}^{m} t_i + \frac{1}{p} \sum_{j=1}^{p} z_j$$

subject to

$$t_i \geq \langle x, a^i \rangle - y + 1,\ \ i = 1, \ldots, m,$$

$$z_j \geq -\langle x, b^j \rangle + y + 1, \;\; j = 1, \ldots, p,$$

$$t \geq 0, \;\; z \geq 0,$$

where t_i is nonnegative and represents the error for the point $a^i \in A$ and z_j is nonnegative and represents the error for the point $b^j \in B$.

The sets A and B are linearly separable if and only if $f^* = f(x^*, y_*) = 0$ where (x^*, y_*) is the solution to the problem (4.1). It is proved that the trivial solution $x = 0$ cannot occur.

4.3.2 Polyhedral separability

The concept of h-polyhedral separability was developed in [7]. The sets A and B are h-polyhedrally separable if there exists a set of h hyperplanes $\{x^i, y_i\}$, with

$$x^i \in \mathbb{R}^n, \; y_i \in \mathbb{R}^1, \; i = 1, \ldots, h$$

such that

1) for any $j = 1, \ldots, m$ and $i = 1, \ldots, h$

$$\langle x^i, a^j \rangle - y_i < 0,$$

2) for any $k = 1, \ldots, p$ there exists at least one $i \in \{1, \ldots, h\}$ such that

$$\langle x^i, b^k \rangle - y_i > 0.$$

It is proved in [7] that the sets A and B are h-polyhedrally separable, for some $h \leq p$ if and only if

$$\text{co } A \bigcap B = \emptyset.$$

The problem of polyhedral separability of the sets A and B is reduced to the following problem:

$$\text{minimize } \; f(x, y) \; \text{ subject to } \; (x, y) \in \mathbb{R}^{(n+1) \times h} \tag{4.2}$$

where

$$f(x, y) = \frac{1}{m} \sum_{j=1}^{m} \max \left[0, \max_{1 \leq i \leq h} \left\{ \langle x^i, a^j \rangle - y_i + 1 \right\} \right] +$$

$$\frac{1}{p}\sum_{k=1}^{p}\max\left[0,\min_{1\leq i\leq h}\left\{-\langle x^i,b^k\rangle+y_i+1\right\}\right]$$

is an error function. Note that this function is a nonconvex piecewise linear function. It is proved that $x^i=0,\ i=1,\ldots,h$ cannot be the optimal solution. Let $\{\bar{x}^i,\bar{y}_i\}, i=1,\ldots,h$ be a global solution to the problem (4.2). The sets A and B are h-polyhedrally separable if and only if $f(\bar{x},\bar{y})=0$. If there exists a nonempty set $\bar{I}\subset\{1,\ldots,h\}$ such that $x^i=0,\ i\in\bar{I}$, then the sets A and B are $(h-|\bar{I}|)$-polyhedrally separable. In [7] an algorithm for solving problem (4.2) is developed. The calculation of the descent direction at each iteration of this algorithm is reduced to a certain linear programming problem.

4.4 Max-min separability

In this section we describe the concept of max-min separability and introduce an error function [12].

4.4.1 Definition and properties

Let $H=\{h_1,\ldots,h_l\}$, where $h_j=\{x^j,y_j\},\ j=1,\ldots,l$ with $x^j\in\mathbb{R}^n,\ y_j\in\mathbb{R}^1$, be a finite set of hyperplanes. Let $J=\{1\ldots.,l\}$. Consider any partition of this set $J^r=\{J_1,\ldots,J_r\}$ such that

$$J_k\neq\emptyset,\ k=1,\ldots,r,\ \ J_k\bigcap J_j=\emptyset,\ \ \bigcup_{k=1}^{r}J_k=J.$$

Let $I=\{1,\ldots,r\}$. A particular partition $J^r=\{J_1,\ldots,J_r\}$ of the set J defines the following max-min-type function:

$$\varphi(z)=\max_{i\in I}\ \min_{j\in J_i}\ \left\{\langle x^j,z\rangle-y_j\right\},\ z\in\mathbb{R}^n. \tag{4.3}$$

Let $A,B\subset\mathbb{R}^n$ be given disjoint sets, that is $A\bigcap B=\emptyset$.

Definition 1. *The sets A and B are max-min separable if there exist a finite number of hyperplanes $\{x^j,y_j\}$ with $x^j\in\mathbb{R}^n,\ y_j\in\mathbb{R}^1,\ j\in J=\{1,\ldots,l\}$ and a partition $J^r=\{J_1,\ldots,J_r\}$ of the set J such that*

1) for all $i \in I$ *and* $a \in A$

$$\min_{j \in J_i} \left\{ \langle x^j, a \rangle - y_j \right\} < 0;$$

2) for any $b \in B$ *there exists at least one* $i \in I$ *such that*

$$\min_{j \in J_i} \left\{ \langle x^j, b \rangle - y_j \right\} > 0.$$

Remark 1. It follows from Definition 1 that if the sets A and B are max-min separable then $\varphi(a) < 0$ for any $a \in A$ and $\varphi(b) > 0$ for any $b \in B$, where the function φ is defined by (4.3). Thus the sets A and B can be separated by a function represented as a max-min of linear functions. Therefore this kind of separability is called a max-min separability.

Remark 2. Linear and polyhedral separability can be considered as particular cases of the max-min separability. If $I = \{1\}$ and $J_1 = \{1\}$ then we have the linear separability and if $I = \{1, \ldots, h\}$ and $J_i = \{i\},\ i \in I$ we obtain the h-polyhedral separability.

Proposition 1. *[12]. The sets* A *and* B *are max-min separable if and only if there exists a set of hyperplanes* $\{x^j, y_j\}$ *with* $x^j \in \mathbb{R}^n,\ y_j \in \mathbb{R}^1,\ j \in J$ *and a partition* $J^r = \{J_1, \ldots, J_r\}$ *of the set* J *such that*

1) for any $i \in I$ *and* $a \in A$

$$\min_{j \in J_i} \left\{ \langle x^j, a \rangle - y_j \right\} \leq -1;$$

2) for any $b \in B$ *there exists at least one* $i \in I$ *such that*

$$\min_{j \in J_i} \left\{ \langle x^j, b \rangle - y_j \right\} \geq 1.$$

Proposition 2. *[12]. The sets* A *and* B *are max-min separable if and only if there exists a piecewise linear function separating them.*

Remark 3. It follows from Proposition (2) that the notions of max-min and piecewise linear separability are equivalent.

Proposition 3. *[12]. The sets* A *and* B *are max-min separable if and only if they are disjoint:* $A \bigcap B = \emptyset$.

In the next proposition we show that in most cases the number of hyperplanes necessary for the max-min separation of the sets A and B is limited.

Proposition 4. *[12]. Assume that the set A can be represented as a union of sets $A_i,\ i = 1,\dots,q$ and the set B as a union of sets $B_j,\ j = 1,\dots,d$ such that*

$$A = \bigcup_{i=1}^{q} A_i, \quad B = \bigcup_{j=1}^{d} B_j$$

and

$$co\ A_i \bigcap co\ B_j = \emptyset \ \text{ for all } i = 1,\dots,q,\ j = 1,\dots,d. \tag{4.4}$$

Then the number of hyperplanes necessary for the separation of the sets A and B is at most $q \cdot d$.

Remark 4. Proposition 4 demonstrates that in most cases the cardinality of all sets of indices $J_i,\ i \in I$ are the same. If the assumptions of Proposition 4 are satisfied then the cardinality of all these sets is either p or q. We will use this fact for the design of an incremental algorithm.

4.4.2 Error function

Given any set of hyperplanes $\{x^j, y_j\},\ j \in J = \{1,\dots,l\}$ with $x^j \in \mathbb{R}^n,\ y_j \in \mathbb{R}^1$ and a partition $J^r = \{J_1,\dots,J_r\}$ of the set J, we say that a point $a \in A$ is well separated from the set B if the following condition is satisfied:

$$\max_{i \in I} \min_{j \in J_i} \left\{ \langle x^j, a \rangle - y_j \right\} + 1 \leq 0.$$

Then we can define the separation error for a point $a \in A$ as follows:

$$\max \left[0, \max_{i \in I} \min_{j \in J_i} \left\{ \langle x^j, a \rangle - y_j + 1 \right\} \right]. \tag{4.5}$$

Analogously, a point $b \in B$ is said to be well separated from the set A if the following condition is satisfied:

$$\min_{i \in I} \max_{j \in J_i} \left\{ -\langle x^j, b \rangle + y_j \right\} + 1 \leq 0.$$

Then the separation error for a point $b \in B$ can be written as

$$\max\left[0, \min_{i\in I}\max_{j\in J_i}\left\{-\langle x^j, b\rangle + y_j + 1\right\}\right]. \tag{4.6}$$

Thus, an averaged error function can be defined as

$$f(x,y) = (1/m)\sum_{k=1}^{m}\max\left[0, \max_{i\in I}\min_{j\in J_i}\left\{\langle x^j, a^k\rangle - y_j + 1\right\}\right]$$

$$+(1/p)\sum_{t=1}^{p}\max\left[0, \min_{i\in I}\max_{j\in J_i}\left\{-\langle x^j, b^t\rangle + y_j + 1\right\}\right] \tag{4.7}$$

where $x = (x^1, \ldots, x^l) \in \mathbb{R}^{l\times n},\ y = (y_1, \ldots, y_l) \in \mathbb{R}^l$. It is clear that $f(x,y) \geq 0$ for all $x \in \mathbb{R}^{l\times n}$ and $y \in \mathbb{R}^l$.

Proposition 5. *[12]. The sets A and B are max-min separable if and only if there exists a set of hyperplanes $\{x^j, y_j\}, j \in J = \{1, \ldots, l\}$ and a partition $J^r = \{J_1, \ldots, J_r\}$ of the set J such that $f(x,y) = 0$.*

Proposition 6. *[12]. Assume that the sets A and B are max-min separable with a set of hyperplanes $\{x^j, y_j\}, j \in J = \{1, \ldots, l\}$ and a partition $J^r = \{J_1, \ldots, J_r\}$ of the set J. Then*

1) $x^j = 0,\ j \in J$ cannot be an optimal solution;

2) if

(a) for any $t \in I$ there exists at least one $b \in B$ such that

$$\max_{j\in J_t}\left\{-\langle x^j, b\rangle + y_j + 1\right\} = \min_{i\in I}\max_{j\in J_i}\left\{-\langle x^j, b\rangle + y_j + 1\right\}, \tag{4.8}$$

(b) there exists $\tilde{J} = \{\tilde{J}_1, \ldots, \tilde{J}_r\}$ such that $\tilde{J}_t \subset J_t,\ \forall t \in I,\ \tilde{J}_t$ is nonempty at least for one $t \in I$ and $x^j = 0$ for any $j \in \tilde{J}_t,\ t \in I$.

Then the sets A and B are max-min separable with a set of hyperplanes $\{x^j, y_j\}, j \in J^0$ and a partition $\bar{J} = \{\bar{J}_1, \ldots, \bar{J}_r\}$ of the set J^0 where

$$\bar{J}_t = J_t \setminus \tilde{J}_t,\ t \in I \ \text{ and } \ J^0 = \bigcup_{i=1}^{r}\bar{J}_i.$$

Remark 5. The error function (4.7) is nonconvex and if the sets A and B are max-min separable with a certain number of hyperplanes, then the global minimum of this function $f(x^*, y_*) = 0$ and the global minimizer is not unique.

The problem of the max-min separability is reduced to the following mathematical programming problem:

$$\text{minimize } f(x,y) \text{ subject to } (x,y) \in \mathbb{R}^{(n+1)\times l} \tag{4.9}$$

where the objective function f has the following form:

$$f(x,y) = f_1(x,y) + f_2(x,y)$$

and

$$f_1(x,y) = \frac{1}{m}\sum_{k=1}^{m} \max\left[0, \max_{i\in I} \min_{j\in J_i} \left\{\langle x^j, a^k\rangle - y_j + 1\right\}\right], \tag{4.10}$$

$$f_2(x,y) = \frac{1}{p}\sum_{t=1}^{p} \max\left[0, \min_{i\in I} \max_{j\in J_i} \left\{-\langle x^j, b^t\rangle + y_j + 1\right\}\right]. \tag{4.11}$$

In order to solve this problem, we will apply the discrete gradient method. Figure 4.1 shows max-min separability for two classes and figure 4.2 shows multi-class max-min separability.

4.5 An incremental algorithm

The number of hyperplanes l necessary to separate two sets is not known a priori. In this section we suggest an algorithm for the computation of a piecewise linear function separating two sets and this algorithm computes hyperplanes incrementally. It computes as many hyperplanes as necessary for separating the sets with respect to a given tolerance.

There are some difficulties when one applies max-min separability to solve supervised data classification problems. The first is that the number of hyperplanes necessary to separate two sets is not known a priori. The second is that the number of variables in an error function increases as the number of hyperplanes increases and as a result the problem of minimization of an error function becomes a large scale optimization problem. The third is that the number of local minimizers of the error function drastically increases as the number

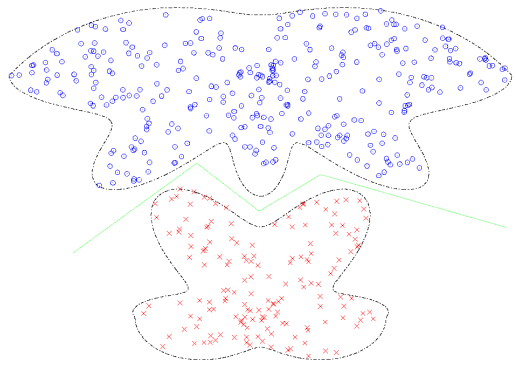

Figure 4.1: Max-min separability

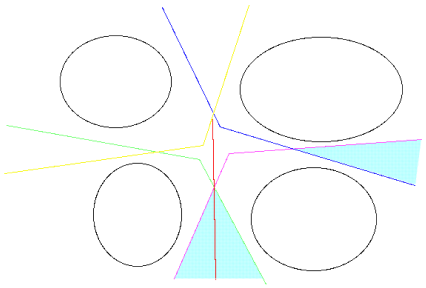

Figure 4.2: Classification

of hyperplanes and the number of data points increase. The problem of minimization of an error becomes a complicated global optimization problem.

In the incremental algorithm we start with one hyperplane, that is, with the one affine

function. In this case the problem is quite a simple convex optimization problem which can be easily transformed to a certain linear programming problem. If the separation is satisfactory with respect to some tolerance, we stop. Otherwise, we compute the error function for both sets. Depending on the value of this function, we increase the number of minimum functions (under maximum) or the number of affine functions (under minimum). We define a starting point for a new problem using the final results from the previous problem. Such an approach allows us to find either a global or near global solution and to significantly reduce computational efforts. It also allows us to compute as many hyperplanes as necessary for the separation of the two sets.

Following Proposition 4 we assume that the sets $J_i,\ i \in I$ have the same cardinality. Let $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ be tolerances.

Algorithm 8. An incremental algorithm

Step 0. (Initialization) Select any starting point $(x^1, y_1),\ x^1 \in \mathbb{R}^n,\ y_1 \in \mathbb{R}^1$. Set $X^1 = (x^1, y_1),\ I_1 = \{1\},\ J_1^1 = \{1\}$, $f_1 = f(x^1, y_1), r_1 = |I| = 1,\ d_1 = |J_1| = 1$, the number of hyperplanes $l = 1$ and $k = 1$.

Step 1. (Computation of a piecewise linear function) Solve Problem (4.9) starting from the point $X^k \in \mathbb{R}^{(n+1)\times l}$. Let $X^{k,*}$ be a solution to this problem, F_k^* is the corresponding objective function value, $f_{1,k}^*$ and $f_{2,k}^*$ are the values of functions f_1 and f_2, respectively.

Step 2. (The first stopping criterion) If $f_{1,k}^* \leq \varepsilon_1$ and $f_{2,k}^* \leq \varepsilon_1$ then stop. $X^{k,*}$ is a final solution.

Step 3. (The second stopping criterion) If $k \geq 2$,

$$f_{1,k-1}^* - f_{1,k}^* \leq \varepsilon_2$$

and

$$f_{2,k-1}^* - f_{2,k}^* \leq \varepsilon_2$$

then stop. $X^{k,*}$ is a final solution.

Step 4. (Adding new hyperplanes)

1. If $f_{1,k}^* > \varepsilon_1$, then set $d_{k+1} = d_k + 1,\ J_i^{k+1} = J_i^k \bigcup \{d_{k+1}\}$ for all $i \in I_k$. Set $x^{ij} = x^{i,j-1,*},\ i \in I_k,\ j = d_{k+1}$.

2. If $f^*_{2,k} > \varepsilon_1$, then set $r_{k+1} = r_k + 1,\ I_{k+1} = I_k \bigcup \{r_{k+1}\},\ J^k_{r_{k+1}} = J^k_{r_k}$. Set $x^{ij} = x^{i-1,j,*},\ i = r_{k+1},\ j \in J^k_{r_k}$.

Step 5. (New starting point) Set $X^{k+1} = \left(x^{ij},\ j \in J^{k+1}_i,\ i \in I_{k+1}\right)$, $k = k + 1$ and go to Step 1.

Explanations of Algorithm 8. The algorithm starts by computing one hyperplane to separate sets (Steps 0 and 1). There are two different stopping criteria in this algorithm. The stopping criterion in Step 2 means that the computed piecewise linear function separates two sets with the tolerance $\varepsilon_1 > 0$. The stopping criterion in Step 3 implies that adding new hyperplanes cannot significantly decrease the value of the error function. This may happen when a large number of hyperplanes are needed to separate sets. Such a criterion allows one to avoid problems with overfitting in supervised data classification problems. However, this stopping criterion does not mean that a piecewise linear function separating two sets has been computed. Step 4 provides rules for adding new hyperplanes and defining their normal vectors. These vectors are defined to guarantee a decrease of the error function in the next iteration compared with the current iteration. Step 5 defines a starting point for the minimization of the error function for the next iteration. Since the problem of minimization of the error function is a global optimization problem, such a strategy allows us to find the "near" global solution.

We use the discrete gradient method to minimize the error functions. The discrete gradient method was introduced in [9, 10], see also [11]. This method is modified to take advantage of the special structure of the problem and thus to reduce computational effort. This modification of the discrete gradient method can be found in [12] and in more detail in [18].

4.6 Results of numerical experiments

In order to verify the effectiveness of the proposed incremental algorithm we conducted numerical experiments using 5 gene expression data sets. The incremental algorithm was applied along with the gene selection algorithm from Chapter 3. The incremental algorithm is applicable to the entire gene expression data sets because the number of genes is too large. Moreover, the incremental algorithm based on max-min separability is not efficient

when data are sparse. We compare results obtained by using the incremental algorithm with those obtained by the k-NN algorithm. We used the best results obtained by the k-NN algorithm in this comparison.

The code of the incremental algorithm was written in Lahey Fortran 95 and the numerical experiments were carried out on a PC Pentium IV with CPU 1.83 and 1GB RAM. In numerical experiments we used only five gene expression data sets. They were chosen to demonstrate the strengths and weaknesses of the proposed algorithm. Similar results can be obtained for other data sets. In tables we present the accuracy on test sets for both the k-NN algorithm and the max-min separability based algorithm, as well as CPU time for both algorithms.

4.6.1 Data set 4

For description of this data set please refer to Appendix A, Section A.4. Results for this data set are presented in Table 4.1.

Table 4.1: Results for Data set 4

	Test set accuracy		CPU time	
N_g	k-NN	Max-min	k-NN	Max-min
2	77.7	88.3	0.02	46.19
4	91.3	92.2	0.05	11.81
6	92.2	94.2	0.06	6.16
8	94.2	96.1	0.07	8.83
10	95.1	95.1	0.08	12.83
20	98.1	100.0	0.13	6.55
50	99.0	98.1	0.25	2.08
100	99.0	99.0	0.53	27.45

One can see from Table 4.1 that for a small number of genes ($N_g \leq 20$) the incremental algorithm is more accurate than the k-NN algorithm. However, as the number of genes increases, the accuracy of the incremental algorithm is no better than that of the k-NN algorithm. In this case the data becomes more and more sparse and the incremental algorithm fails to achieve good accuracy on the test set due to the problem of overfitting. We can see that the incremental algorithm requires much more CPU time than the k-NN algorithm. We can also see that the CPU time with a small number of genes is greater than that for a large number of genes. This is not unexpected, because when the data become sparse the

number of hyperplanes necessary for their separation decreases which leads to a decrease in computational effort.

4.6.2 Data set 5

For description of this data set refer to Appendix A, Section A.5. Results for this data set are presented in Table 4.2.

Table 4.2: Results for Data set 5

	Test set accuracy		CPU time	
N_g	k-NN	Max-min	k-NN	Max-min
2	84.2	89.5	0.00	0.39
4	78.9	78.9	0.00	1.02
6	78.9	78.9	0.00	0.13
8	86.8	89.5	0.02	0.09
10	81.6	94.7	0.00	0.09
20	73.7	92.1	0.02	0.09
50	78.9	84.2	0.02	0.16

Results from Table 4.2 show that the incremental algorithm performs better than the k-NN algorithm for a small number of genes ($N_g \leq 50$). We can also see that in some cases the difference between the k-NN and the incremental algorithm is significant (for $N_g = 10, 20, 50$). For this data set the CPU time used by the incremental algorithm is reasonable.

4.6.3 Data set 6

For description of this data set refer to Appendix A, Section A.6. Computational results for this data set are presented in Table 4.3.

One can see from Table 4.3 that for a small number of genes ($N_g \leq 30$), the incremental algorithm is more accurate than the k-NN algorithm. However, as the number of genes increases, the accuracy of the incremental algorithm is becoming worse than that of the k-NN algorithm. Again for this data set the CPU time used by the incremental algorithm is reasonable.

Table 4.3: Results for Data set 6

	Test set accuracy		CPU time	
N_g	k-NN	Max-min	k-NN	Max-min
2	52.9	55.9	0.00	0.77
4	50.0	64.7	0.00	0.99
6	70.6	73.5	0.00	0.25
8	67.6	76.5	0.00	0.50
10	61.8	79.4	0.00	0.16
20	67.6	79.4	0.00	0.17
30	58.8	67.6	0.00	0.16
50	67.6	55.9	0.02	0.73

4.6.4 Data set 7

For description of this data set refer to Appendix A, Section A.7. Results for this data set are presented in Table 4.4.

Table 4.4: Results for Data set 7

	Test set accuracy		CPU time	
N_g	k-NN	Max-min	k-NN	Max-min
2	71.0	80.6	0.31	1060.19
4	85.1	88.7	0.58	1737.97
6	90.7	92.3	0.70	231.00
8	92.3	91.9	0.81	579.21
10	96.0	95.6	0.98	1406.08
20	97.2	96.8	1.69	62.95
20	99.2	99.2	3.84	121.92

One can see from Table 4.4 that for a small number of genes ($N_g \leq 6$) the incremental algorithm is more accurate than the k-NN algorithm. However, as the number of genes increases the k-NN algorithm outperforms the incremental algorithm. Again in this case the data become more and more sparse and the incremental algorithm fails to achieve good accuracy on the test set due to the problem of overfitting. The incremental algorithm requires significantly more CPU time than the k-NN algorithm. Again, the CPU time with a small number of genes is greater than that for a large number of genes. This means the algorithm computes significantly more hyperplanes for a small number of genes, which increases computational effort.

4.6.5 Data set 8

For description of this data set refer to Appendix A, Section A.8. Results for this data set are presented in Table 4.5.

Table 4.5: Results for Data set 8

	Test set accuracy		CPU time	
N_g	k-NN	Max-min	k-NN	Max-min
2	87.8	88.8	0.19	349.82
4	90.7	94.4	0.28	442.84
6	90.9	93.4	0.34	473.45
8	92.4	93.4	0.38	177.50
10	94.4	91.4	0.47	726.83
20	93.4	94.9	0.81	189.11
50	95.4	95.4	1.89	27.41

Results from Table 4.5 demonstrate that for a small number of genes ($N_g \leq 8$) the incremental algorithm outperforms the k-NN algorithm. However, as the number of genes increases, the k-NN algorithm outperforms the incremental algorithm. The incremental algorithm requires significantly more CPU time than the k-NN algorithm. The CPU time for the incremental algorithm is variable for different numbers of genes, because the algorithm computes different numbers of hyperplanes.

Thus, based on results presented in this section we can conclude that the max-min separability incremental algorithm is very effective when the number of genes is not large. This algorithm is not effective at solving classification problems in sparse data sets. The new algorithm requires much more CPU time than the k-NN algorithm.

4.7 Conclusion

In this chapter we developed a new algorithm for solving supervised data classification problems in gene expression data sets. This algorithm computes a piecewise linear function separating a given tumour type from all of the others. Since the number of hyperplanes in this case is not known a priori, we proposed an incremental approach to compute hyperplanes separating two sets. The problem of computation of these hyperplanes is formulated as an optimization problem where the objective function is a nonconvex piecewise linear function. The discrete gradient method is applied to solve this optimization problem.

We tested the new algorithm on five gene expression data sets and compared the results with those obtained using the k-NN algorithm, which is known to be an efficient algorithm for supervised data classification. Results show that the proposed algorithm outperforms the k-NN algorithm when the number of genes is not large, however it suffers from overfitting when the number of genes increases and the data become sparse.

Gene expression data sets have many genes and the use of all genes allows one to separate different tumour types with one hyperplane only. However, the use of all genes leads to overfitting problems, meaning that the classification of the new tumours will be very difficult. The combination of gene selection and separation algorithms allows us to design more efficient classification algorithms, however, in this case our hyperplane is not any more sufficient for separation of tumour types. Therefore we develop a classification algorithm based on the combination of overlapping gene selection and max-min separability algorithms.

Conclusion

The purpose of this research was to develop new algorithms for solving clustering, gene selection and supervised data classification problems on gene expression data sets. To design these algorithms we used optimization techniques, more specifically, nonsmooth optimization techniques.

In Chapter 1 of this research we presented an overview of molecular biology including cells, chromosomes, DNA, RNA, amino acids, proteins and genes followed by genetic engineering. In regards to microarray, its history, types, technology and applications were discussed, followed by microarray data analysis and microarray gene expression.

In Chapter 2 we presented a new algorithm for solving clustering problems in gene expression data sets. It should be noted that one can consider two types of clustering problems in gene expression data sets: clustering with respect to samples and clustering with respect to genes. The first type of clustering can help to find similar samples whereas the second type of clustering can help to find similar genes and to reduce the number of genes in a data set. In this thesis we considered the clustering problem with respect to samples. A few clustering algorithms can be applied to solve such clustering problems in gene expression data sets. The k-means algorithm is among those algorithms. However, it is well known that this algorithm is very sensitive to the choice of the starting points and fails to find good cluster structure if the number of features is large and the number of clusters is relatively large (in many cases more than five).

The global k-means algorithm is a significant improvement over the k-means algorithm. In this thesis we developed a new version of the global k-means algorithm: the modified global k-means algorithm. The problem of computation of the starting point in this algorithm is reduced to a certain nonsmooth optimization problem. The latter problem is solved using the k-means algorithm. Our numerical results on ten publicly available gene expression data sets show that the modified global k-means algorithm outperforms both the

multi-start k-means and the global k-means algorithms when the number of clusters $N > 5$. However , the modified global k-means algorithm is computationally more expensive than the global k-means algorithm.

In Chapter 3 we presented new gene selection algorithms. Gene selection is a very important step in the analysis of gene expression data sets. Because these data sets contain thousands or even tens of thousands of genes and most of them contribute noise, these genes should be detected and removed from a data set to improve the performance of the supervised data classification algorithms. We developed the new gene selection algorithms which are based on the use of the overlaps between different tumour types (classes). We introduced the univariate and multi-dimensional overlaps as well as the binary and one-vs-all overlaps. To find the most informative genes we considered the sum of overlaps over all classes and also the number of "well-separated" classes. We applied these algorithms to ten gene expression data sets. The k-NN algorithm was applied to validate the results obtained by gene selection algorithms. Our results demonstrate that the developed algorithms find a subset with a small number of genes and with high classification accuracy. These results also show that the use of binary overlaps and the sum of overlaps over classes lead to the design of better gene selection algorithms.

In Chapter 4 we developed a new algorithm for solving the supervised data classification problems in gene expression data sets. This algorithm is based on max-min separability. It allows one to find a continuous piecewise linear function separating two sets. Since any continuous piecewise linear function can be represented as a max-min of linear functions, we call such a separation, max-min separation. We formulated the classification problem as an optimization problem and we applied the discrete gradient method to solve this problem. Since the number of linear functions separating two sets is not known a priori we developed a new incremental algorithm to compute piecewise linear functions. This cannot be applied to whole gene expression data sets due to the large number of genes. We applied this algorithm along with the gene selection algorithms introduced in Chapter 3. We presented the results of numerical experiments on five gene expression data sets and compared them with those obtained by the k-NN algorithm. These results show that the new algorithm is very efficient when the number of genes is not large and it outperforms, sometimes significantly, the k-NN algorithm. However, as the number of genes increases and the data set becomes more sparse the k-NN algorithm outperforms the max-min separability based on the incre-

mental algorithm. The incremental algorithm is computationally more expensive than the k-NN algorithm.

In brief we can outline the results of this thesis as follows:

1. The new clustering algorithm for gene expression data sets was developed. This algorithm outperforms other k-means algorithms, however it is computationally more expensive than the other algorithms.

2. The new gene selection algorithm was developed and this algorithm is very efficient in finding the subset of most informative genes. The algorithm is based on the computation of the overlaps between different tumour types and it is very easy to implement.

3. The new algorithm was developed to solve the supervised data classification problems in gene expression data sets. This algorithm can only be applied along with the gene selection algorithms. The algorithm produces very good results when the number of genes is not large.

In this thesis our aim was to develop new tools based on nonsmooth optimization techniques for solving clustering, gene selection and supervised classification problems in gene expression data sets. Currently, Bioinformatics is a rapidly developing area and many algorithms have been proposed to solve similar problems. The comparative study of all these algorithms is very interesting, however, it is not within the topic of this thesis and it will be the subject of our future research. The biological interpretation of results obtained by these algorithms is also very interesting, however this is not within the topic of this thesis either and will be considered in our future research.

Appendix A

Data sets

A.1 Data set 1

This data set is the Ann Arbor Lung cancer data set and was generated at the University of Michigan by Beer *et al.* [23]. It consists of 7129 genes, 67 stage *1* and 19 stage *3* tumours, as well as 10 non-neoplastic lung samples which were hybridized to the Affymetrix GeneChip Hu6800. A full description of the data set can be accessed at:
http://dot.ped.med.umich.edu:2000/ourimage/pub/Lung/index.html

A.2 Data set 2

This data set is the Boston Lung Cancer data set and was generated at the Dana Farber Cancer Institute. The data set consists of 12625 genes, 17 normal lung samples and 185 lung tumour samples. Of these, there are 138 lung adenocarcinoma, 6 small-cell lung cancer, 20 carcinoid lung cancer and 21 squamous cell. Expression profiles were generated using the Affymetrix GeneChip HG_U95Av2 [41] .This data set can be accessed at:
http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.3 Data set 3

This data set is a leukaemia data set. This data set includes 999 genes and 38 samples. Bone marrow samples were obtained from acute leukaemia patients at the time of diagnosis: 11 acute myeloid leukaemia (AML) samples; 8 T-lineage acute lymphoblastic leukaemia

(ALL) samples; and 19 B-lineage (ALL) samples. The leukaemia data set is from the previous-generation Human Genome HU6800 Affymetrix microarray [93]. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.4 Data set 4

This is the Novartis multi-tissue data set. There are 103 samples all together and 1000 genes [93]. The data set includes tissue samples from four cancer types with 26 breast, 26 prostate, 28 lung, and 23 colon samples. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.5 Data set 5

This is a leukaemia data set with 5000 genes and 38 samples including 11 acute myeloid leukaemia (AML) and 27 acute lymphoblastic leukaemia (ALL) samples [34]. The original data set is retrievable from: http://www.broad.mit.edu/cgi-bin/cancer/ datasets.cgi.

A.6 Data set 6

This data set is a medulloblastomas gene expression data which includes a set of 34 samples with 25 classic and 9 Desmoplastic (Brain-MD) samples and 5893 genes [34]. The original data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.7 Data set 7

This data set is the St.Jude leukaemia data set which includes 985 genes and 248 samples. Diagnostic bone narrow samples were taken from paediatric acute leukaemia patients corresponding to 6 prognostically important leukaemia subtypes: 43 T-lineage ALL, 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, 20 MLL rearrangements and 64 “hyperdiploid>50” chromosomes [93]. The data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.8 Data set 8

This is a lung cancer data set which includes 1000 genes and 197 samples including 139 adenocarcinomas (AD), 21 squamous cell carcinomas (SQ), 20 carcinoids (COID) and 17 normal (NL) lung samples [93]. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.9 Data set 9

This data set is a CNS tumours data set including 42 samples and 989 genes. The samples are embryonal tumours of the central nervous system (CNS) including 10 medulloblastomas (MD), 8 primitive neuroectodermal tumours (PNET), 10 atypical teratoid/rhabdoid tumours (Rhab), 10 malignant gliomas (Glio), and 4 normal cerebellums (Ncer) [93]. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

A.10 Data set 10

This data set has 1277 genes and 90 samples. It contains 13 distinct tissue types: 5 breast, 9 prostate, 7 lung, 11 colon, 6 germinal centre cells, 7 bladder, 6 uterus, 5 peripheral blood monocytes, 12 kidney, 10 pancreas, 4 ovary, 5 whole brain and 3 cerebellum [93]. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

Appendix B

Glossary

Activation An increase in the rate of transcription is called activation or up-regulation.

Affymetrix Affymetrix is the world's leading manufacturer of DNA microarrays.

Affymetrix GeneChips Affymetrix GeneChip is one of the most popular microarray platforms. They are manufactured in a unique way and can be read by the special Affymetrix machine.

Allele Alternative forms of a gene. They occur at the same position on the paired chromosomes.

Allelic The state of being alleles.

Amino Acid A biochemical building block that makes up cellular protein.

Amplification Increasing the number of copies of a specific DNA molecule.

Annealing The hybridization of a single-stranded DNA molecule to another single stranded DNA molecule of complementary sequence.

Array profile A gene expression profile that explains the expression values for many genes under a single condition or sample.

Basic alignment search tool Basic Local Alignment Search Tool is an algorithm for comparing primary biological sequence information, such as the amino acid sequences of different proteins or the nucleotides of DNA sequences.

Biochemistry The field of study that endeavours to understand the chemical basis of life by focusing on the study of DNA, RNA, proteins, and other bio-molecules.

Bioinformatics Specialised field of computer science focused on the analysis of biological data.

Bio-molecule A bio-molecule is a chemical compound that naturally occurs in living or-

ganisms. Bio-molecules consist primarily of carbon and hydrogen, along with nitrogen, oxygen, phosphorus and sulphur.

Biophysics Biophysics is an interdisciplinary science that applies the theories and methods of physics, to questions of biology.

Carcinogen A substance that causes cancer in a body.

Cell A cell is a single unit or compartment, enclosed by a border, wall or membrane.

Chromosome Large segment of genomic DNA that replicates autonomously in the cell and segregates during cell division.

Codon Any one of 64 three-nucleotide sequences or triplets in messenger RNA that specify one of the 20 amino acids used for protein synthesis.

Complementary DNA Is produced from messenger RNA using the enzyme reverse transcriptase.

Cytogenetics Cytogenetics is the study of the structure of chromosomal material.

Deletion Mutation that results in the removal of one or more nucleotides from a DNA sequence.

DNA The biopolymeric molecule that constitutes the genetic blueprint of virtually every organism in the biosphere.

DNA cloning Isolation and manipulation of a piece of DNA by incorporating it into a specially modified phage or plasmid and introducing it into a bacterial cell.

DNA sequence A DNA sequence or genetic sequence is a succession of letters representing the primary structure of a real or hypothetical DNA molecule or strand, with the capacity to carry information.

DNA sequencing The experimental process of determining the primary nuclear sequence of a DNA molecule.

E.Coli This is one of the main species of bacteria living in the lower intestines of mammals, known as gut flora.

Enhancer An enhancer is an element that alters a promoter's efficiency by increasing or decreasing the rate of transcription.

Enzyme A protein that carries out a biochemical reaction in the cell.

Eukaryote cell Has a nucleus which is separated from the rest of the cell by a membrane and contains the gene's genetic material.

Exon Segment of a gene retained in the messenger RNA after procession, and often con-

taining the codons represented as amino acids in proteins.

Expression profile If different probes matching all mRNAs in a cell are used, a snapshot of the total mRNA pool of a living cell or tissue can be obtained and this is known as the expression profile.

GenBank The GenBank sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations. This database is produced at the National Center for Biotechnology Information (NCBI) as part of the International Nucleotide Sequence Database Collaboration, or INSDC.

Gene Segment of genomic DNA that encodes a specific cellular mRNA and protein.

Genetic code The cellular alphabet that specifies one of the 20 common cellular amino acids or stop codons from the 64 triplets in messenger RNA.

Genetic engineering Genetic engineering is a term for the process of manipulating genes, usually outside the organism's natural reproductive process.

Gene expression The cellular process by which genetic information flows from gene to messenger RNA to protein.

Gene expression data matrix In this matrix rows represent genes and columns represent experimental conditions, samples or features.

Gene profile A gene profile is a gene expression profile that describes expression values for a single gene across many samples or conditions.

Genome The complete set of different genes carried by an organism or virus.

Genomics The development and application of mapping, sequencing, computational and other procedures for the analysis of entire genomes, in turn providing an understanding of the structure, function and evolution of genes and genomes.

Genotype The genetic constitution of an individual, usually referring to specific characters under consideration.

Human genome All the DNA in one set of chromosomes.

Human genome project The Human Genome Project is a project to map and sequence the 3 billion nucleotides contained in the human genome and to identify all the genes present in it.

Hybridization The chemical process by which two complementary DNA or RNA strands zipper up to form a double-stranded molecule.

Intron Segment of a gene removed from the messenger RNA during processing and not

represented in proteins.

Insertion Mutation that results in the addition of one or more nucleotides to a DNA sequence.

Leukaemia Is a cancer of the blood or bone marrow and is characterised by an abnormal proliferation of blood cells, usually white blood cells (leukocytes).

Leukocytes White blood cells that form a component of the blood.

Lymphomas Lymphoma is a variety of cancer that originates in lymphocytes or, more rarely, of histiocytes.

Macromolecule A very large molecule, composed of many atoms and having a very high molecular weight.

Microarray An ordered array of microscopic elements on a planar substrate that allows the specific binding of genes or gene products.

Microarray data Contains two basic aspects: biological and statistical. The biological aspect refers to the expression of a gene influenced by conditions, and the statistical aspect says how trustworthy the biological significance is.

Microarray gene expression data matrix Measures the expression of many genes with a number of conditions in a table. Rows in the table corrspond to genes and columns correspond to different tissues or treatments.

Mitochondria Is a cell's membrane and it provides the energy the cell needs.

Molecular biology Molecular biology is the study of biology at a molecular level. The field overlaps with other areas of biology and chemistry, particularly genetics and biochemistry.

Messenger RNA or mRNA The class of cellular RNA that undergoes extensive editing, contains the protein coding sequences of genes, and functions as an informational intermediate between DNA and protein.

Mutagen Chemical agent that alters the primary nucleotide sequence of DNA.

Mutation Any change in a DNA sequence, but typically acquired during the life span of an organism.

Nucleus An organelle of eukaryotic cells that is bounded by a nuclear membrane and contains the chromosomes whose genes control the structure of proteins within the cell.

Nucleotide A complex organic molecule forming the basic unit of nucleic acids, with a structure made up of three components: a pentose sugar, an organic base, and a phosphate group.

Outlier In a gene expression matrix outliers are inconsistent values.

Polymerase chain reaction Polymerase chain reaction allows production of millions of copies of DNA from a small amount of genetic material and is used in microarray manufacture.

Protein Data Bank The Protein Data Bank is a repository for 3-D structural data of proteins and nucleic acids.

Phenotype The total set of characteristics expressed by an organism is called its phenotype.

Polymerase Polymerase transcribes the genes for precursors to ribosomal RNA.

Primer Oligonucleotide that hybridizes to a complementary nucleic acid template and expedites enzymatic synthesis by providing a starting point for polymerase.

Probe Labelled molecule in solution that reacts with a complementary target molecule on the substrate.

Prokaryote cell This cell has a single chromosome including circular double-stranded DNA.

Promoter Genomic location upstream of a cellular gene that determines the start site for RNA polymerase.

Protein Any member of the major family of cellular biomolecules encoded by a unique cellular gene and consisting of a repeating series of amino acids linked together by peptide bonds.

Protein sequence The order of amino acid in a protein chain.

Proteome The variety of proteins generated by a genome of an organism is called its proteome.

Proteomics Study of protein structure and behaviour is called proteomics.

Recombinant DNA Revolutionary technology developed in the 1970s that allows genes from different organisms to be spliced together.

Replication Cellular process by which DNA is copied from a DNA template to produce an exact copy of the genome.

Repression A decrease in the rate of transcription is called repression or down-regulation.

Ribosomal RNA Specialized class of cellular RNA, located in ribosomes, that plays structural and catalytic roles during protein synthesis.

Ribosome Large cytoplasmic structure that facilitates protein synthesis.

RNA Nucleic acid comprising the nucleotides adenosine, cytidine, guanosine and uri-

dine; it is closely related to DNA. Principal differences are the use of ribose instead of 2′-deoxyribose and rudine in place of theymidine. Some RNA molecules encode proteins, others are components of the protein synthesis machinery.

Serial analysis of gene expression This method uses a traditional DNA sequencing to identify and count the number of mRNAs in a cell.

Sequencing by hybridization This is building a miniature DNA array or DNA chips including thousands of DNA fragments attached to a surface.

Sequence variant A change in the primary nucleotide sequence of DNA is known as a sequence variant.

Single nucleotide polymorphism Common sequence variant containing a one-base-pair change relative to the normal gene.

Target Molecule tethered to a microarray substrate that reacts with a complementary probe molecule in solution.

TATA box A promoter is an element that determines the starting site for RNA polymerase, which is an enzyme that makes mRNA from the DNA template. Many promoters have an AT-rich promoter sequence which is called a TATA box.

Transfer RNA Specialised class of cellular RNA that binds specific amino acids and facilitates protein synthesis by mediating codon recognition.

Bibliography

[1] K.S. Al-Sultan. A tabu search approach to the clustering problem. *Pattern Recognition*, 28(9):1443 – 1451, 1995.

[2] K.S. Al-Sultan and M.M. Khan. Computational experience on four algorithms for the hard clustering problem. *Pattern Recognition Letters*, 17:295 – 308, 1996.

[3] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L.Yang, G.E. Marti, T. Moore, J.J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, and W.C. Chan. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Natura*, 403:503 – 511, 2000.

[4] L. Alphey. *DNA sequencing. From experimental methods to bioinformatics*. BIOS scientific publishers Ltd., 1997.

[5] D. Amaratunga and J. Cabrera. *Exploration and analysis of DNA microarray and protein array data*. John Wiley and Sons, Inc., 2004.

[6] J. An and Y.P.P. Chen. Finding Rule Groups to Classify High Dimensional Gene Expression Datasets. In *The 18th International Conference on Pattern Recognition 2006 (ICPR2006)*, pages 1196–1199. Hong Kong, IEEE CS Press, 20–24 August 2006.

[7] A. Astorino and M. Gaudioso. Polyhedral separability through successive LP. *Journal of Optimization Theory and Applications*, 112(2):265 – 293, 2002.

[8] K. Bae and B.K. Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20:3423–3430, 2004.

[9] A.M. Bagirov. Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis. *Investigacao operacional*, 19:75 – 93, 1999.

[10] A.M. Bagirov. Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices. *Applied optimization: Progress in optimization: Contribution from Australasia*, 30:147 – 175, 1999.

[11] A.M. Bagirov. A method for minimization of quasidifferentiable functions. *Optimization methods and software*, 17(1):31 – 60, 2002.

[12] A.M. Bagirov. Max-min seperability. *Optimization methods and software*, 20(2-3):271 – 290, 2005.

[13] A.M. Bagirov, B. Ferguson, S. Ivkovic, G. Saunders, and J. YearWood. New algorithms for multi-class cancer diagnosis using tumour gene expression signature. *Bioinformatics*, 19(14):1800 – 1807, 2003.

[14] A.M. Bagirov and K. Mardaneh. Modified global k-means algorithm for clustering in gene expression data sets. In M. Boden and T. Bailey, editors, *Proceedings of the AI 2006 Workshop on Intelligent Systems of Bioinformatics WISB–2006*, volume 73, pages 23–28. Australian Computer Society Inc, Nov 2006.

[15] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova, and J. YearWood. Unsupervised and supervised data classification via nonsmooth and global optimization. *TOP: Spanish operations research journal*, 11(1):1 – 93, 2003.

[16] A.M. Bagirov, A.M. Rubinov, and J. YearWood. Using global optimization to improve classification for diagnosis and prognosis. *Topics in health information management*, 22:65 – 74, 2001.

[17] A.M. Bagirov, A.M. Rubinov, and J. YearWood. A global optimization approach to classification. *Optimization and engineering*, 3(2):129 – 155, 2002.

[18] A.M. Bagirov and J. Ugon. Supervised data classification via max-min separability. In A.M. Rubinov and V. Jeyakumar, editors, *Trends in continuous optimization, Applied optimization*, volume 99. Springer, Dordrecht, 2005.

[19] A.M. Bagirov and J. YearWood. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *European journal of operational research*, 170(2):1 – 6, 2006.

[20] A.M. Bagirov and J. Yearwood. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *European journal of operational research*, 170(2):578–596, 2006.

[21] P. Baldi and H.G. Wesley. *DNA microarrays and gene expression*. Cambridge university press, 2002.

[22] C. Baumgartner, C. Bohm, D. Baumgartner, G. Marini, K. Weinberger, B. Olgemoller, B. Liebl, and A.A. Roscher. Supervised machine learning techniques for the classification of metabolic disorders in newborns. *Bioinformatics*, 20:2985–2996, Nov 2004.

[23] D.G Beer, S.L.R. Kardia, C.C. Huang, T.J. Giordano, A.J. Levin, D.E. Misek, L. Lin, G. Chen, T.G. Gharib, D.G. Thomas, M.L. Lizyness, R. Kuick, S. Hayasaka, J.M.G. Taylor, M.D. Iannettoni, M.B. Orringer, and S. Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *NATURE MEDICINE*, 8(8):816–824, August 2002.

[24] K.P. Bennet and J. Blue. A support vector machine approach to decision trees. *Mathematics report*, pages 97 – 100, 1997.

[25] K.P. Bennet and E.J. Brederstainer. A parametric optimization method for machine learning. *INFORMS journal on computing*, 9:311 – 318, 1997.

[26] K.P. Bennet and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1:23 – 34, 1992.

[27] K.P. Bennet and O.L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2(3):207 – 227, 1993.

[28] D.P. Berrar, W. Dubitzky, and M. Grnzow. *A practical approach to microarray data analysis*. Kluwer academic publishers, 2003.

[29] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward,

and J. Trent. Molecular classification of cutaneous malignant melonoma by gene expression profiling. *Nature*, 406:536 – 540, 2000.

[30] E. Blalock. *A beginner's guide to microarrays*. Kluwer academic publishers, 2003.

[31] H.H. Bock. Clustering and neural networks. In A. Rizzi, M. Vichi, and H.H. Bock, editors, *Advances in data science and classification*, pages 256 – 277. Springer verlag, Berlin, 1998.

[32] P.S. Bradley and O.L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization methods and software*, 13:1 – 10, 2000.

[33] D.E. Brown and C.L. Entail. A practical application of simulated annealing to the clustering problem. *Pattern recognition*, 25(4):401 – 412, 2001.

[34] J.P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov. Metagenes and Molecular Pattern Discovery using Matrix Factorization: Supplement Information. http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi., 2003.

[35] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121 – 167, 1998.

[36] L.J. Buturovic. PCP: a program for supervised classification of gene expression profiles. *Bioinformatics*, 22:245–247, 2006.

[37] R.C. Calladine, D.R. Horace, F.B. Luisi, and A.A. Travers. *Understanding DNA. The molecule and how it works*. Elsevier academic press, 2004.

[38] H.C. Causton, J. Quackenbush, and A. Brazma. *A beginner's guide. Microarray gene expression date analysis*. Blackwell publishing company, 2003.

[39] G.C. Cawley and L.C. Talbot. Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics*, 22:2348–2355, 2006.

[40] J.L. Connolly, S.J. Schnitt, H. H. Wang, J. A. Longtine, A. Dvorak, and H. F. Dvorak. Principles of cancer pathology. In J.F. Holland et al., editor, *Cancer medicine*, pages 533 – 555. Williams and Wilkins, Baltimore, 1997.

[41] A.S. Cordero, S. Mukherjee, A. Subramanian, H. You, J. Roix, C. Ladd, T.R. Golub, and T. Jacks. Supplementary Methods, Figures and Tables: An oncogenic Kras expression signature identified by cross-species gene expression analysis. http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi., 2005.

[42] A.R. Dabney. Classification of microarrays to nearest centroids. *Bioinformatics*, 21:4148–4154, Nov 2005.

[43] L. Davis, J. Hawkins, S. Maetschke, and M. Boden. Comparing SVM sequence kernels: A protein subcellular localization theme. In proceedings of the workshop on Intelligent Systems for Bioinformatics, CRPIT, 2007.

[44] O. de Merle, P. Hansen, B. Jaumard, and N. Mladenovic. An interior point method for minimum sum-of-squares clustering. *SIAM J. on scientific computing*, 21:1485 – 1505, 2001.

[45] M. Dettling and P. Buhlmann. Boosting for tumor classification with gene expression data. *Bioinformatics*, 19:1061–1069, Jun 2003.

[46] S.M. Dhanasekaran, T.R. Barrette, D. Ghosh, R. Shah, S. Varambally, K. Kurachi, K.J. Pienta, M.A. Rubin, and A.M. Chinnaiyan. Delineation of prognostic biomarkers in prostate cancer. *Nature*, 412:822 – 826, 2001.

[47] I.S. Dhillon, E.M. Marcotte, and U. Roshan. Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19(13):8, 2003.

[48] G. Diehr. Evaluation of a branch and bound algorithm for clustering. *SIAM J. Scientific and statistical computing*, 6:268 – 284, 1985.

[49] C.H.Q. Ding. Unsupervised feature selection via two-way ordering in gene expression analysis. *Bioinformatics*, 19:1259–1266, 2003.

[50] R. Dubes and A.K. Jain. Clustering techniques: The user's dilemma. *Pattern Recognition*, 8:247 – 260, 1976.

[51] L. Dufton and M. Boden. Reducing the number of support vectors to allay inefficiency of large-scale models in computational biology. In proceedings of 2007 International Symposium on Computational Models for Life Sciences - CMLS'07.

[52] B. Everitt. *Cluster analysis*. Halsted press, 1980.

[53] J. Gebert, M. Latsch, E.M.P. Quek, and G.W. Weber. Analysis and optimizing genetic network structure via path-finding. *Journal of Computational Technologies*, 9:3–12, 2004.

[54] L. Goh, Q. Song, and N. Kasabov. A novel feature selection method to improve classification of gene expression data. In *Proceedings of the second conference on Asia-Pacific bioinformatics*, volume 29, pages 161 – 166. Dunedin, New Zealand, 2004.

[55] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Dowing, M.A. Caligiuri, C.D. Bloom field, and E.S. Lander. Molecular classification of cancer: class discovery and class predictions by gene expression monitoring. *Science*, 286:531 – 537, 1999.

[56] Z. Guan and H. Zhao. A semiparametric approach for marker gene selection based on gene expression data. *Bioinformatics*, 21:529–536, 2005.

[57] J.F. Hair, R.L. Tatham, R.E. Anderson, and W. Black. *Multivariate data analysis*. Prentice-Hall englewood cliffs, New Jersey, 1998.

[58] P. Hanjoul and D. Peeters. A comparison of two dual-based procedures for solving the p-median problem. *European journal of operational research*, 20:387 – 396, 1985.

[59] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191 – 215, 1997.

[60] P. Hansen and N. Mladenovic. J-means: a new heuristic for minimum sum-of-squares clustering. *Pattern recognition*, 4:405 – 413, 2001a.

[61] P. Hansen and N. Mladenovic. Variable neighborhood decomposition search. *Journal of heuristic*, 7:335 – 350, 2001b.

[62] P. Hansen, E. Ngai, B.K. Cheung, and N. Mladenovic. Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering. *Les cahiers du gerad*, 2002.

[63] D.M. Hawkins, M.W. Muller, and J.A. Krooden. Cluster analysis. In D.M. Hawkins, editor, *Topics in applied multivariate analysis*. Cambridge University press, Cambridge, 1982.

[64] J.D. Hawkins. *Gene structure and expression*. Cambridge university Press, 1996.

[65] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O.P. Kallioniemi, B. Wilfond, A. Borg, J. Trent, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, and G. Sauter. Gene expression profiles in hereditary breast cancer. *The new england journal of medicine*, 344:539 – 548, 2001.

[66] D. Heffernan and R. Miller. *The Australian biology dictionary*. Pearson education Australia Pty limited, 2nd edition, 1997.

[67] J. Wilson T. Hunt. *Molecular biology of the cell: A problems approach*. Garland science, 2002.

[68] T. Hunt, S. Prentis, and J. Tooze. *DNA makes RNA makes Protein*. Elsevier biomedical press, 1983.

[69] J. Jaeger, R. Sengupta, and W.L. Ruzzo. Improved gene selection for classification of microarrays. In *Pacific Symposium on Biocomputing*, volume 8, pages 53 – 64, 2003.

[70] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264 – 323, 1999.

[71] R.E. Jensen. A dynamic programming algorithm for cluster analysis. *Operation Research*, 17:1034 – 1057, 1969.

[72] B.R. Jordan. *DNA microarrays: Gene expression applications*. Springer verlag, 2001.

[73] R. Jornsten and B. Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19:1100–1109, 2003.

[74] M. Kantardzic. *Data mining: Concepts, models, methods and algorithms*. John wiley & sons Inc., 2003.

[75] G.H. Keller and M.M. Manak. *DNA probes*. Macmillan publishers Ltd., 1993.

[76] J. Kieleczawa. *DNA sequencing. Optimizing the process and analysis*. Jones and Bartlett Publishers, 2005.

[77] S. Knudsen. *A biologist's guide to analysis of DNA microarray data*. John wiley and sons Inc., 2002.

[78] W.L.G. Koontz, P.M. Narendra, and K. Fukunaga. A branch and bound clustering algorithm. *IEEE Transactions on Computers*, 24:908 – 915, 1975.

[79] B. Krishnapuram, L. Carin, and A.J. Hartemink. Joint classifier and feature optimization for cancer diagnosis using gene expression data. In *RECOMB, Berlin, Germany*, pages 167 – 175, April 2003.

[80] M. Lau and M. Schultz. A feature selection method for gene expression data with thousands of features. http://zoo.cs.yale.edu/classes/cs490/02-03b/, 2003.

[81] E. Lawrence. *Hederson's dictionary of biological terms*. Pearson education Ltd., 2000.

[82] K.E. Lee, N. Sha, E.R. Dougherty, M. Vannucci, and B.K. Mallick. Gene selection: a Bayesian variable selection approach. *Bioinformatics*, 19:90–97, 2003.

[83] M.L. Lee. *Analysis of microarray gene expression data*. Kluwer academic publishers, 2004.

[84] J. Lepre, J.J. Rice, Y. Tu, and G. Stolovitsky. An efficient algorithm for pattern discovery and multivariate feature selection in gene expression data. *Bioinformatics*, 20:1033–1044, 2004.

[85] F. Li and Y. Yang. Analysis of recursive gene selection from microarray data. *Bioinformatics*, 21:3741–3747, 2005.

[86] L. Li, C.R. Weinberg, T.A. Darden, and L.G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17:1131–1142, 2001.

[87] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429–2437, 2004.

[88] W. Li and I. Grosse. Gene selection criterion for discriminant microarray data analysis based on extreme value distributions. In *RECOMB, Berlin, Germany*, pages 217 – 223. ACM, April 10-13 2003.

[89] A. Likas, M. Vlassis, and J. Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36:451 – 461, 2003.

[90] F. Martella. Classification of microarray data with factor mixture models. *Bioinformatics*, 22:202–208, Jan 2006.

[91] M. Medvedovic and S. Sivaganesan. Bayesian infinite mixture model based clustering gene expression profiles. *Bioinformatics*, 18:1194 – 1206, 2002.

[92] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine learning. Neural and statistical classification*. Prentice Hall, 1994.

[93] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus Clustering. A resampling-based method for class discovery and visualization of gene expression microarray data. http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi., December 2003.

[94] M. Mramor, G. Leban, J. Demsar, and B. Zupan. Visualization-based cancer microarray data classification analysis. *Bioinformatics*, 23:2147–2154, August 2007.

[95] U.R. Muller and D.V. Nicolau. *Microarray technology and its applications: Biological and medical physics, biomadical engineering*. Springer, 2005.

[96] J. Nahar, S. Ali, and Y.P.P. Chen. Microarray Data Classification using Automatic SVM Kernel selection. *DNA and Cell Biology*, Accepted on June 2007.

[97] J. Nahar, Y.P.P. Chen, and S. Ali. Kernel Based Naive Bayes Classifier for Breast Cancer Prediction. *Journal of Biological Systems*, 15(1):17–25, 2007.

[98] S. Neidle. *DNA structure and recognotion*. Oxford university press Inc., 1994.

[99] D.V. Nguyen and D.M. Rocke. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18:1216–1226, Sep 2002.

[100] U.A. Nuber. *DNA microarrays: BIOS advanced methods*. Taylor and francis group, LLC., 2005.

[101] B.K. Nunnally. *Analytical techniques in DNA sequencing*. Taylor and francis group, LLC., 2005.

[102] P.J. Park, M. Pagano, and M. Bonetti. A nonparametric scoring algorithm for identifying informative genes from microarray data. In *Pacific symposium on Biocomputing*, volume 6, pages 52 – 63, 2001.

[103] C.M. Perou, T. Sorlie, M.B. Eisen, R.M. van de, S.S. Jeffrey, C.A. Rees, J.R. Pollack, D.T. Ross, H. Johnsen, L.A. Akslen, O. Fluge, A. Pergamenschikov, C. Williams, S.X. Zhu, P.E. Lonning, A.L. Borresen-Dale, P.O. Brown, and D. Botstein. Molecular portraits of human breast tumours. *Nature*, 406:747 – 752, 2000.

[104] P.A. Pevzner. *Computational molecular biology. An algorithmic approach*. The MIT press, 2001.

[105] T.D. Pham, D. Beck, and H. Yan. Spectral pattern comparison methods for cancer classification based on microarray gene expression data. *IEEE Trans, Circuits and Systems I: Fundamental Theory and Applications, special issue in the Life Science Systems*, 53(11):2425–2430, 2006.

[106] T.D. Pham, C. wells, and D.I. Crane. Analysis of microarray gene expression data. *Current Bioinformatics*, 1(1):37–53, 2006.

[107] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, and T.R. Golub. Multiclass cancer diagnosis using tumour gene expression signatures. *Medical sciences, PNAS*, 98(26):15149 – 15154, December 18.

[108] Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[109] R. Sarker, H.A. Abbas, and C. Newton. *Heuristics and optimization for knowledge discovery*. Idea group publishing, 2002.

[110] M. Schena. *DNA microarrays: A practical approach*. Oxford university press, 1999.

[111] M. Schena. *Microarray biochip technology*. Natick, MA: Eaton Publishers, 2000.

[112] M. Schena. *Microarray analysis*. John wiley and sons, Inc., 2003.

[113] M. Schena. *Microarray Analysis*. Wiley-Liss, 2004.

[114] M. Schena. *Protein Microarrays*. Jones and Bartlett Publishers, Inc., 2004.

[115] S.Z. Selim and K.S. Al-Sultan. A simulated annealing algorithm for the clustering. *Pattern recognition*, 24(10):1003–1008, 1991.

[116] S.K. Shevade and S.S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19:2246–2253, 2003.

[117] R.A. Shimkets. *Gene expression profiling: Methods and protocols*. Humana press Inc., 2004.

[118] L. Song, J. Bedo, K.M. Borgwardt, A. Gretton, and A. Smola. Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, 23:1490–1498, 2007.

[119] H. Spath. *Cluster analysis algorithms for data reduction and classification of objects*. Ellis horwood Limited, Chichester, England, 1980.

[120] H. Spath. *Cluster analysis algorithms*. Ellis horwood Limited, Chichester, England, 1991.

[121] A Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(631–643), Mar 2005.

[122] D. Stekel. *Microarray bioinformatics*. Cambridge university press, 2003.

[123] L.X. Sun, Y.L. Xie, X.H. Song, J.H. Wang, and R.Q. Yu. Cluster analysis by simulated annealing. *Computers and chemistry*, 18:103–108, 1994.

[124] M. Sun and M. Xiong. A mathematical programming approach for gene selection and tissue classification. *Bioinformatics*, 19:1243–1251, 2003.

[125] M. Thain and M. Hickman. *The penguin dictionary of biology*. Penguin group, 11th edition, 2004.

[126] J. Thorsten. *Learning to classify text using support vector machines*. Kluwer academic publishers, Dordrecht, 2002.

[127] D.T. Tran and T.D. Pham. Modeling Methods for Cell Phase Classification. In T.D. Pham, H. Yan, and D.I. Crane, editors, *Advanced Computational Methods for Biocomputing and Bioimaging*. Nova Science Publishers, 2007.

[128] O. Ugur and G.W. Weber. Optimization and dynamics of gene-environment networks with intervals. *Journal of Industrial and Management Optimization*, 3(2):357–379, 2007.

[129] V.N. Vapnik. *The nature of statistical learning theory*. Springer verlag, New York, 1995.

[130] R. varshavsky, A. Gottlieb, M. Linial, and D. Horn. Novel unsupervised Feature Filtering of Biological Data. *Bioinformatics*, 22:e507–e513, 2006.

[131] R. Walker. *Genes and DNA*. King fisher publications PLC., 2003.

[132] M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle. Feature (gene) selection in gene expression based tumour classification. *Molecular genetics and metabolism*, 73:239 – 247, 2001.

[133] K.Y. Yeung, R.E. Bumgarner, and E. Raftery. Bayesian model averaging: development of an improved multi-class gene selection and classification tool for microarray data. *Bioinformatics*, 21:2394–2402, 2005.

[134] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo. Model based clustering and data transformations for gene expression data. *Bioinformatics*, 17:977 – 987, 2001.

[135] K.Y. Yeung, M. Medvedovic, and R.E. Bumgarner. Clustering gene expression data with repeated measurements. *Genome biology*, 4:R34, 2003.

[136] H.H. Zhang, J. Ahn, X. Lin, and C. Park. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22:88–95, 2006.

[137] L. Zhang, C. Tang, Y. Song, A. Zhang, and M. Ramanathan. Vizcluster and its application on clustering gene expression data. *Journal of parallel and distributed database*, 13(1):73 – 79, 2003.

[138] X. Zhou and K.Z. Mao. LS bound based gene selection for DNA microarray data. *Bioinformatics*, 21:1559–1564, 2005.

[139] X. Zhou and D.P. Tuck. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics*, 23:1106–1114, 2007.