UNIVERSITY OF SOUTHAMPTON

# Inference from binary gene expression data

by

Salih Tuna

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

October 2009

UNIVERSITY OF SOUTHAMPTON

## ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Salih Tuna

Microarrays provide a practical method for measuring the mRNA abundances of thousands of genes in a single experiment. Analysing such large dimensional data is a challenge which attracts researchers from many different fields and machine learning is one of them. However, the biological properties of mRNA such as its low stability, measurements being taken from a population of cells rather than from a single cell, etc. should make researchers sceptical about the high numerical precision reported and thus the reproducibility of these measurements. In this study we explore data representation at lower numerical precision, down to binary (retaining only the information whether a gene is expressed or not), thereby improving the quality of inferences drawn from microarray studies. With binary representation, we propose a solution to reduce the effect of algorithmic choice in the pre-processing stages.

First we compare the information loss if researchers made the inferences from quantized transcriptome data rather than the continuous values. Classification, clustering, periodicity detection and analysis of developmental time series data are considered here. Our results showed that there is not much information loss with binary data. Then, by focusing on the two most widely used inference tools, classification and clustering, we show that inferences drawn from transcriptome data can actually be improved with a metric suitable for binary data. This is explained with the uncertainties of the probe level data. We also show that binary transcriptome data can be used in cross-platform studies and when used with Tanimoto kernel, this increase the performance of inferences when compared to individual datasets.

In the last part of this work we show that binary transcriptome data reduces the effect of algorithm choice for pre-processing raw data. While there are many different algorithms for pre-processing stages there are few guidelines for the users as to which one to choose. In many studies it has been shown that the choice of algorithms has significant impact on the overall results of microarray studies. Here we show in classification, that if transcriptome data is binarized after pre-processed with any combination of algorithms it has the effect of reducing the variability of the results and increasing the performance of the classifier simultaneously.

# Contents

# List of Figures

# List of Tables

# Nomenclature

$\mathbf{x}$ : vector<br>
$x$ : scalar
$M$ : number of components for GMM
$Th$: threshold
$T$ : Tanimoto coefficient
$TP$: True positive
$FP$: False positive
$TN$: True negative
$FN$: False negative
$f(x)$: Probability function
$F(x)$: Cumulative Density Function (CDF)
superscript $T$ denotes the transpose of a matrix

# DECLARATION OF AUTHORSHIP

I, Salih Tuna declare that the thesis entitled `Inference from binary gene expression data` and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
    - Tuna, S. and Niranjan, M. (2009) *Cross-platform analysis with binary gene expression data*, Proceedings of the 4th international conference in Pattern Recognition in Bioinformatics (PRIB), LNBI Vol.5780 P.439-449.
    - Tuna, S. and Niranjan, M. (2009) *Classification with binary gene expression data.* Journal of Biomedical Sciences and Engineering. In Press.
    - Tuna, S. and Niranjan, M. (2009) *Inference from low precision transcriptome data representation*, Journal of Signal Processing Systems [Online, 22 April 2009], doi:10.1007/s11265-009-0363-2.
    - Delivered as a talk at the Molecular and Statistical Aspects of Molecular Biology (MASAMB), 2008, Glasgow/Scotland.
    - Presented as a poster at the 11th Microarray and Gene Expression Data (MGED) meeting, 2008, Riva del Garda/Italy.
    - Presented as a poster at the 9th MGED meeting, 2006, Seattle/USA.

Signed:.................................. Date:......................................

## Acknowledgements

I would like express my gratitude to those, whom without them this dream would not be real. I would like to thank to my PhD supervisor Prof. Mahesan Niranjan, for his support, belief and patience in me. My family, for their unlimited love, support, encouragement and believing in me. Mr. Orn Gunnarson for his friendship and for making the start up process of this PhD much more easier by sharing his knowledge, Mr. Amirthalingam Ramanan and Mr. Somjet Suppharangsan for listening the same story again and again and giving me feedback for my work. Miss Berna Keskin for her unlimited support, patience and for being a great friend, my brother Mr. Hasan Tuna, Mr. Ali Hassan for their friendship and support, University of Sheffield panel group members: Dr. Neil Lawrence, Dr. Richard Clayton, Dr. Guido Sanguinetti for their useful comments on my work. Dr. Sandor Szedmak for his very helpful discussions on SVM. Also thanks to the ML Group members of The University of Sheffield and ISIS Group members of University of Southampton.

*To My Parents: Mumine and Unal Tuna*

# Chapter 1

# Introduction

Microarrays are used to measure the abundance of mRNA level of thousands of genes in a single experiment (Schena et al., 1995). This property of microarrays made them very popular in the last decade. Within its short history, microarray technology has been applied to a wide range of problems, aiming to understand life, improve the clinical outcomes of illnesses and thus increase the standards of life. Problems addressed include gene function prediction, analysing differentially expressed genes (i.e., when exposed to certain environmental conditions), monitoring the gene expression with time, detecting cell cycle genes or classifying diseased samples from normal ones. These are achieved by examining the pattern of the gene expression values. Genes or samples in the same group have similar patterns of the gene expression values and different patterns across different groups i.e., diseased samples having similar patterns but differing when compared to the normal ones. This data come in huge dimensions which attracts researchers from different fields. To be able to analyse such noisy, huge dimensional data has been a challenge to machine learning, statisticians and bioinformaticians. Combining biological knowledge with machine learning makes this field unique and very interesting. The points missed by genetics researchers can be completed by machine learning people and vice versa. In this work we consider and combine biological knowledge with statistical knowledge to present a novel approach for making inferences from gene expressions data which improves the performance of inferences drawn from transcriptome data and addresses some issues regarding the biological properties of mRNA. Our approach use quantized gene expression values. This work of using quantized data is motivated with the biological properties of mRNA. Before giving the main motivation behind this work, we will explain how microarrays are used for extracting biological information.

## 1.1 Extracting biological information with microarrays

A typical microarray study aims to answer questions like which genes are expressed under certain conditions or which genes are responsible for a particular disease. Figure 1.1 shows a general framework of how microarrays can be used to extract biological information. Motivated by a biological question, microarray experiments are carried out. The microarray data need to be pre-processed before applying any statistical analyses. After these stages are completed, statistical analysing techniques can be applied to reach biological conclusions. These conclusions further needs biological or statistical verification and validation.

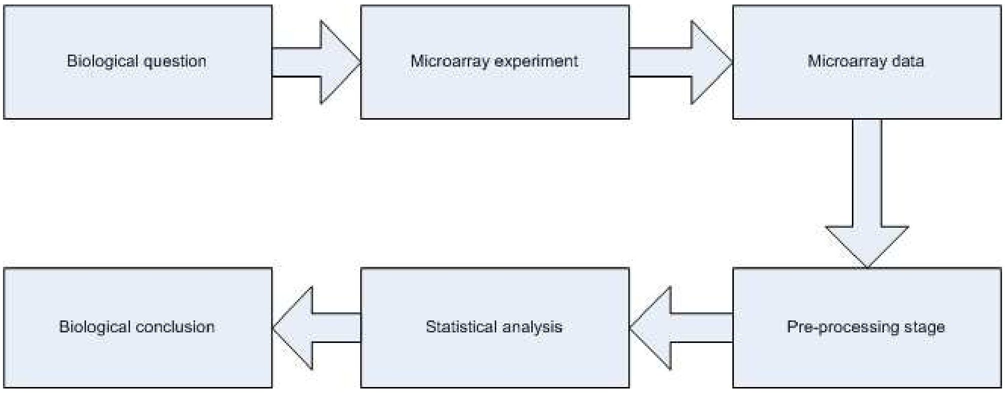


FIGURE 1.1: A general framework of how microarrays are used to extract biological information.

For a successful microarray experiment, replicates must be used. There are two types of replicate: biological and technical replicates.

### 1.1.1 Biological and technical replicates

The statistical analysis of gene expression data not only involves analysing data but also the design of the microarray experiments. At this point using replicates plays an important part in the conclusions reached by the researchers. In order to reduce the effect of variability, replicates need to be used. There are two types of replicates used in microarray studies (see Fig. 1.2):

**Biological replicates**: These replicates are taken independently from two or more biological specimens taken from the same biosource and treated in identical ways. Biological replicates are used to assess the natural variability in the system. The necessity to use biological replicates also suggests that these measurements are not reproducible. There are studies showing the importance of these replicates. Lee et al. (2000) analyse the effects of biological replicates. The experimental results of the study show that any single microarray output is subject to substantial variability and they conclude that designing experiments with replications reduces misclassification rates.

**Technical replicates**: This type of replicate uses measurements from the same sample on different arrays. Technical replicates are used to assess the experimental noise.

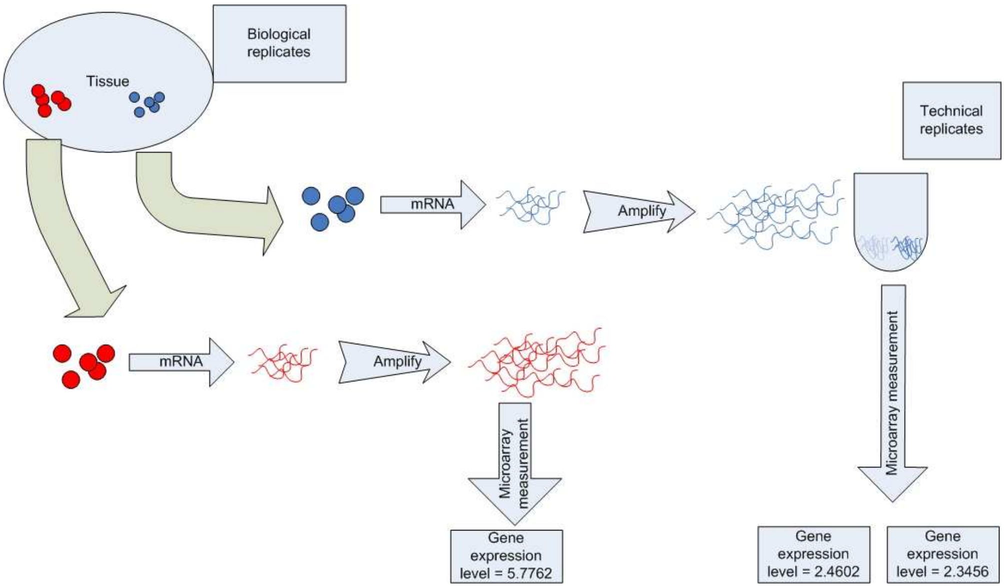


FIGURE 1.2: Microarray measurements produced with technical replicates are very similar and reproducible. However, in biological replicates, the situation is different. As mRNA is an inherently unstable molecule, mRNA samples taken from the same tissue may cause some variability in measurements. This shows that the high numerical precision for representing and processing gene expression values is not realistic.

#### 1.1.1.1 The difference between biological and technical replicates

Figure 1.2 shows a graphical representation of the difference between the two types of replicates. Most of the researchers focus on the technical replicates part of these experiments, suggesting that these measurements are reproducible i.e., if the specific sample is amplified, copied and tested afterwards at different labs, these measurements will surely be identical and reproducible (see MAQC (2006) as an example). However, when biological replicates are considered, i.e., when two different samples of mRNA are taken from the same tissue, reported gene expression measurements will have a variability. This is known as biological variability in literature (Kendziorski et al., 2005; Shi et al., 2007). Therefore it is hard to suggest the same thing for biological replicates as these measurements are less reproducible. Instead we can only say whether the gene is expressed or not. We further analysed biological and technical replicates in Chapter 4 to support this claim. This variability is explained with the mRNA properties mentioned in the following subsection. Critical appraisals of microarray technology, while recognising good reproducibility of technical replicates, often identify large variations with respect to biological replicates. One such survey by Draghici et al. (2006) conclude:

> "...the existence and direction of gene expression changes can be reliably detected for the majority of genes. However, accurate measurements of absolute expression levels and the reliable detection of low abundance genes are currently beyond the reach of microarray technology."

## 1.2 Motivation and hypothesis

This work on questioning the high numerical precision of microarray measurements is motivated by several observations about the properties of mRNA.

1. mRNA is an inherently unstable molecule which is subject to decay at different rates (Iyer and Struhl, 1996; Hume, 2000). The entire mRNA content of a cell, transcriptome, is continuously restructured. To allow this, mRNAs degrade after their synthesis. The half-lives of bacterial mRNAs are no more than few minutes while in eukaryotes, it is about a few hours after synthesis (Brown, 1999). Table 1.1 shows the descriptive statistics of mRNA half-lives for three different species. The studies mention in table measured the mRNA half life for Arabidopsis, Human and Yeast by using microarray technology.

   TABLE 1.1: mRNA half-life for three different species.

   | | Arabidopsis | Human | Yeast |
   |---|---|---|---|
   | Study | Narsai et al. (2007) | Yang et al. (2003) | Wang et al. (2002) |
   | No. of genes studied | 13012 | 5245 | 4687 |
   | Median half-life | 3.8h | 10h | 20min |
   | Max. half-life | 24h+ | 24h+ | 90min+ |
   | Min. half-life | <0.5 | <2h | 3min |

   Thus the process of of extracting cellular mRNA will be subject to significant variability prior to the subsequent amplification process.

2. There is only a finite number of mRNA molecules in any particular cell. Average number of mRNA in a cell, regardless of the organism is about 10 molecules per cell for a gene of interest (Lockhart and Winzeler, 2000; Levsky and Singer, 2003). With the underlying biology of such small numbers, pooling of cells and subsequent amplification of the extracted mRNA population can potentially give rise to a very noisy environment, against which the arrays are set to hybridize.

3. Microarray hybridization is carried out against mRNA taken from a population of cells from a biological sample of interest. Thus microarray-based measurements are far from giving estimates of mRNA counts per cell in the sample (Brazma and Vilo, 2000). Except in few studies, different cells in such a population can potentially be in different states (e.g., the expression of a gene will vary from cell to cell) (Elowitz et al., 2002; Levsky et al., 2002; Levsky and Singer, 2003). The few exceptions include forcing cells into synchrony, i.e., starving and arresting cells at a certain point, then releasing these arrested cells would be expected to lead to synchronized growth of the cell, for investigating cell cycle behaviour (*e.g.* Spellman et al. (1998)) and entraining cultured cells for observing circadian rhythm (*e.g.* Storch et al. (2002)). While there is apparent broad acceptance of synchronization in the literature, Cooper (2004) argue that these techniques do not synchronize cells.

4. The process of a gene being expressed is stochastic (Kuznetsov et al., 2002). Depending on the time and whether samples are taken from the same tissue, the amount that a gene is expressed will be a random number, as the process itself is random (Levsky et al., 2002). Raj et al. (2006) explored the cell-to-cell variations of mRNA levels in mammalian cells. For this purpose the study analyse the distribution of a specific mRNA per cell over the entire cell population stating that if mRNA were produced at a constant rate it would be expected that distribution of mRNA molecules per cell would have a Poisson distribution with mean and variance being equal. As a result of their calculations, mRNA molecules per cell for mammalian cells are reported to be approximately 40 and the variance of these as 1600. So, it can be seen there is even greater variability in mRNA number in each cell than a Poisson distribution due to gene expression process being stochastic. Thus, the mean expression level over a population of cells, as reported by microarrays, is not realistic.

5. mRNAs undergoing translation can be bound to several ribosomal complexes, the effect of which may be to restrict the availability of these molecules to amplification and hybridization (Brown, 1999).

Thus the overall picture is quite different from one in which molecules vary high abundance and free-floating in the medium to be hybridized efficiently onto the probes in a microarray. Apart from the biological properties of mRNA, numerical precision of these measurements is not usually addressed at the pre-processing stage of transcriptome measurements. There are many different pre-processing algorithms for microarray data. The choice of algorithm affects the overall results (Allison et al., 2006). As this is the case for the pre-processing stage, one may not always expect to get the same numerical precision when an experiment is repeated. Treating the differential expression (often on a log scale) as a real valued number, the numerical precision with which researchers report them is quite arbitrary. Different datasets archived in the ArrayExpress (`http://www.ebi.ac.uk/arrayexpress/`) and Gene Expression Omnibus (GEO) (`http://www.ncbi.nlm.nih.gov/geo/`) repositories use different numerical precisions, some truncating to first decimal point (e.g. 0.1) while others respecting up to six decimal places (e.g. 0.123456). Clearly much of the higher digits in the latter come from processing artefacts in the image intensity measurement and floating point calculations done in normalization, and do not have significance with respect to the underlying biology that is being measured.

All these issues: biological variability (Kendziorski et al., 2005; Shi et al., 2007), noise (Barash et al., 2004) and variability of pre-processing algorithms should make researchers sceptical about the high numerical precision reported for gene expression data. Motivated by the biological properties of mRNA we use quantized gene expression data to make statistical inferences. However, this work should not be considered as questioning the accuracy of microarray technology. Microarrays are very precise at measuring the abundance of mRNA level but these measurements come from a population of cells and not a single cell. The difference between the two situations (microarrays measuring the mRNA abundance very precisely and the high numerical precision not being very realistic) is explained with biological and technical replicates (see Fig. 1.2). As Barash et al. (2004) mention, the high numerical precisions of microarray measurements are far from being a clear one to one mapping of the mRNA level of a gene. As a result we claim that these measurements are exaggerated, not reproducible and thus the high numerical precision is not realistic. We show that binary transcriptome data representation is more appropriate and this can improve the performance of inferences drawn from microarray data with a right choice of metric which is suitable for binary data. Besides this, binary gene expression data address some other issues such as reducing the effect of algorithm choice for pre-processing gene expression from probe level data (algorithmic variability) and enabling cross-platform analysis. Even though biological and algorithmic variabilities have been known for a long time, solutions to such problems have not yet been proposed. In this work we aim to address these issues and show the benefits of using quantized, particularly binary, gene expression measurements. The term ‘continuous data’ refers to the high numerical precision measurements in this thesis.

## 1.3 Outline of the thesis

The thesis is structured as follows:

- Chapter 2 presents background information for microarray technology and machine learning techniques used in this work.
- Chapter 3 summarizes related existing work in the literature.
- In Chapter 4, we present our experimental results, showing that with quantized transcriptome data, the information loss is not much and the same conclusions can

be reached with binary data.

- In Chapter 5, we show that the performance of inference problem can be improved with the right choice of metrics suitable for binary data (i.e., Tanimoto similarity) The success of Tanimoto similarity is explained with that of the uncertainties associated at the probe level data. Also it has been shown that binary gene expression data with Tanimoto kernel improve the performance of classification in cross-platform analyses.

- In Chapter 6, we show an important aspect of using binarized data i.e., when transcriptome data is binarized and used with Tanimoto kernel in classification, the effect of the choice of algorithms for pre-processing raw data is significantly reduced.

- In Chapter 7, we present conclusions and give directions for future work.

- Appendix A shows $K$-means clustering results as a supplementary material for Chapter 4.

- Appendix B shows the solution to Support Vector Machines.

- Appendix C shows that Tanimoto kernel is a valid kernel for Support Vector Machines.

- Appendix D gives the description of the different methods in `expresso (affy)` for pre-processing probe-level data.

## 1.4 Our contribution

There are three main contributions arising from this study. Firstly, we show in a wide range of experiments, including classification, clustering, periodicity detection, analysing time series data and Singular Value Decomposition (SVD) analysis for cell cycle detection, that when binary transcriptome data (whether a gene is expressed or not) is used information loss is not much. Secondly, we employ a similarity metric, namely Tanimoto coefficient, suitable for binary data and focus on classification and clustering. Tanimoto coefficient, which is widely used in chemoinformatics, is used for the first time with microarray data. With binary data oriented similarity metric it has been shown that the performance of inferences can be improved. With this particular metric we point out an unnoticed systematic variation in oligonucleotide type arrays and show that as there are more expressed genes in an array the associated average uncertainty is lower. While we offer no biological level explanation to this we show that this property improves the performance of inferences. Another advantage is that using binarized data make cross-platform analysis possible and this also improves the performance of inferences in classification when compared to individual datasets. Finally, by focusing on classification, we show that using binarized data has the advantage of reducing the effect of the choice of algorithms for pre-processing raw data and simultaneously improving the performance of the classifier. This has been shown on a range of classification problems. We achieve this by addressing the uncertainties of the low signal measurements. This is the other advantage of Tanimoto coefficient. It ignores the low signal measurements which are associated with higher uncertainties and are represented as 0 in binary approach. So by taking these uncertainties into consideration we improve the performance of inferences drawn from transcriptome data. We hope that all these points will improve the quality of inferences drawn from microarray data and better analyses will be done in the future for making better clinical outcomes.

# Chapter 2

# Background and methods

## 2.1 Introduction

This chapter is divided into two. In the first part we give basic biological information about genes, DNA (deoxyribonucleic acid) and RNA (ribonucleic acid). Then the description of microarray technology, including two-colours array (cDNA - complementary DNA) and one-colour array (Affymetrix) will follow. Also we will describe how gene expression matrix is obtained and how it is analysed in order to get some biological information. In the second part of this chapter, the statistical methods we used for learning from gene expression data and finally the evaluation metrics used in this thesis will be described.

## 2.2 Background

In this section, biological background information will be described. First we will define the central dogma of molecular biology in order to understand what gene expression is. Then a description of microarray technology will be given, including how it is used to obtain gene expression data. The last part will be about the gene expression data itself.

### 2.2.1 DNA, RNA and protein

DNA is found in the chromosomes in the cell nucleus. Each chromosome contains two very long strands of DNA, which are bound to each other by hydrogen bonds. DNA is in a double helix structure. DNA is made up of nucleotides, which consist of three parts: a sugar, a nitrogeneous base and phosphates. There are four different nitrogenous bases: Adenine ($A$), Guanine ($G$), Cytosine ($C$) and Thymine ($T$). Two strands are linked to each other with these bases. $A$ is always paired with $T$ and $C$ is always paired with $G$.

Because of this, the two strands of DNA that form the double helix are complementary in sequence.

Genome, the entire genetic complement of a living organism, consists of discrete functional regions known as genes. Gene expression is the process where Messenger RNA (mRNA) and protein is synthesized from DNA. During the conversion of genetic information from DNA to proteins, RNA is produced before protein. RNA is also made of nucleotides but differs from DNA in having ribose instead of deoxyribose, and Uracil ($U$) instead of Thymine ($T$). Unlike DNA, RNA is single-stranded. Messenger RNA (mRNA) carries the genetic information from DNA to protein.

The biological information contained in a genome is encoded in the nucleotide sequence of its DNA or RNA molecules and is divided into units called genes. The information contained in a gene is read by proteins that attach to the genome at the appropriate position and initiate a series of biochemical reactions. This process, termed gene expression, mainly consists of two stages (transcription and translation) and is known to be the central dogma of molecular biology (Crick, 1970) (see Fig. 2.1).

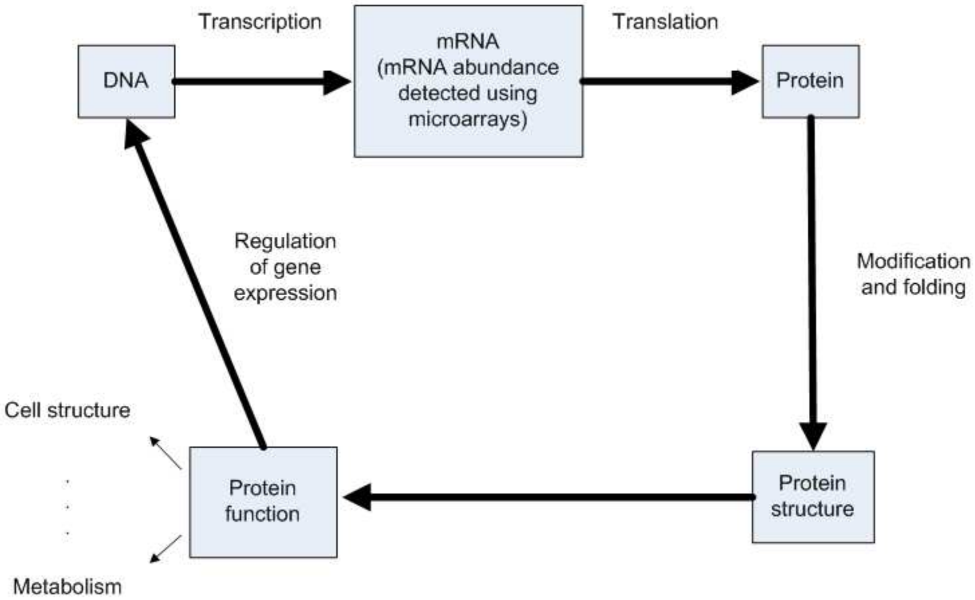


Figure 2.1: Summary of the central dogma of molecular biology. The process of gene translation into mRNA followed by mRNA transcription into protein is called gene expression.

### 2.2.2 Microarray technology

Microarrays were first introduced by Schena et al. (1995) and Lockhart et al. (1996) who measured the expression level of thousands of genes in a single experiment. This property of microarrays makes them very useful and practical. The main idea behind microarray technology is the chemical process of two complementary strands attaching to each other which is known as hybridization. The two most widely used microarrays are the cDNA arrays (Schena et al., 1995), developed at Stanford University, and the synthetic oligonucleotide microarrays, produced by Affymetrix, (Lockhart et al., 1996). Although both these technologies exploit hybridization, they differ in how DNA sequences are laid on the array and in the length of these sequences. A typical microarray experiment involves the following steps: (Liu, 2006)

1. Isolate RNA from tissue of interest and prepare fluorescently labelled samples.
2. Hybridise the labelled targets to the microarray.
3. Wash and scan the microarray to obtain a two dimensional image.
4. Process the resulting image to obtain a quantitative measurement of the intensity for each probe.

Details for the two most widely used array types are described below.

#### 2.2.2.1 cDNA microarrays

A cDNA microarray or two-color array is a glass slide where single-stranded DNA molecules, called probes, are attached at fixed spots (Schena et al., 1995). Probes are built in order to be complementary to labelled target cDNA sequences, and bind to them by hybridization (see Fig. 2.2). Hybridization is the process where RNA sequences bind to their complementary probes (i.e., $A$ with $T$ and $C$ with $G$).

Total mRNA is extracted from two samples which are hypothesized to carry genotypic differences, for example, assay of a cell line before and after drug treatment. Each sample is dyed with two different colours, usually red and green fluorescent dyes. For each sample, double stranded cDNA molecules are produced by reverse transcription of single stranded mRNA molecules. Reverse transcription is the process where single-stranded RNA is transcribed into double-stranded DNA. Following hybridization, a laser beam excites each spot and the emitted fluorescence is detected which results in a two-dimensional image. The relative amounts of red and green fluorescence measured in each spot reflects the relative abundance of target mRNA in the two original samples. Consequently, when detecting a non-fluorescent (i.e., black) spot, it is inferred that neither of the two samples contained cDNA complementary to the probe set of interest.

#### 2.2.2.2 Oligonucleotide arrays

Oligonucleotide arrays or one-color platforms (Lockhart et al., 1996) are developed by Affymetrix and are the commercial arrays. Oligonucleotide arrays estimate the mRNA

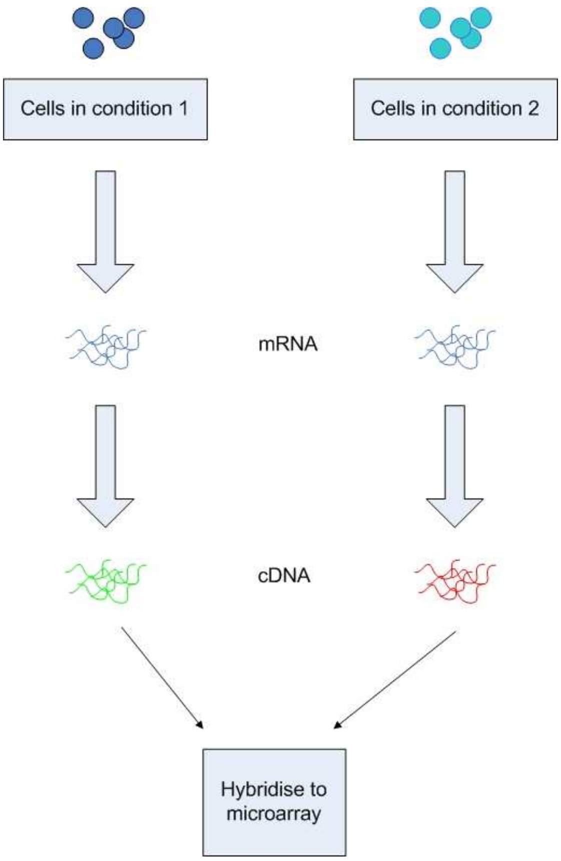


FIGURE 2.2: Typical cDNA array experiment

transcript expression levels based on the hybridization of entire mRNA population to high density arrays of synthetic oligonucleotides. The arrays contain more than 65000 different 20-mer oligonucleotides of defined sequence on the surface of an array which is 1.6 $cm^2$ (Lockhart et al., 1996). The location where each oligonucleotide is synthesized is called a feature.

Oligonucleotide arrays contain 11 - 20 pairs of probes for each of the RNAs that is being monitored. Each probe pair consists of a 20-mer that is perfectly complementary (perfect match or PM probe) to a subsequence of a particular message and a companion that is identical except for a single base difference in a central position (mismatch or MM). There are two types of hybridization in oligonucleotide arrays. The first one is the specific-hybridization where a double-stranded molecule is formed from two perfectly complementary strands. The second hybridization is the cross-hybridization where hybridization occurs between two strands which are not perfectly complementary. The MM probe of each pair serves as an internal control for hybridization intensity. The analysis of PM/MM pairs allows low-intensity hybridization patterns from rare RNAs to be sensitively and accurately recognized in the presence of cross-hybridization signals. Fig. 2.3 shows a typical oligonucleotide array experiment.

Affymetrix microarrays are often referred to as chips or arrays. For each chip a two-

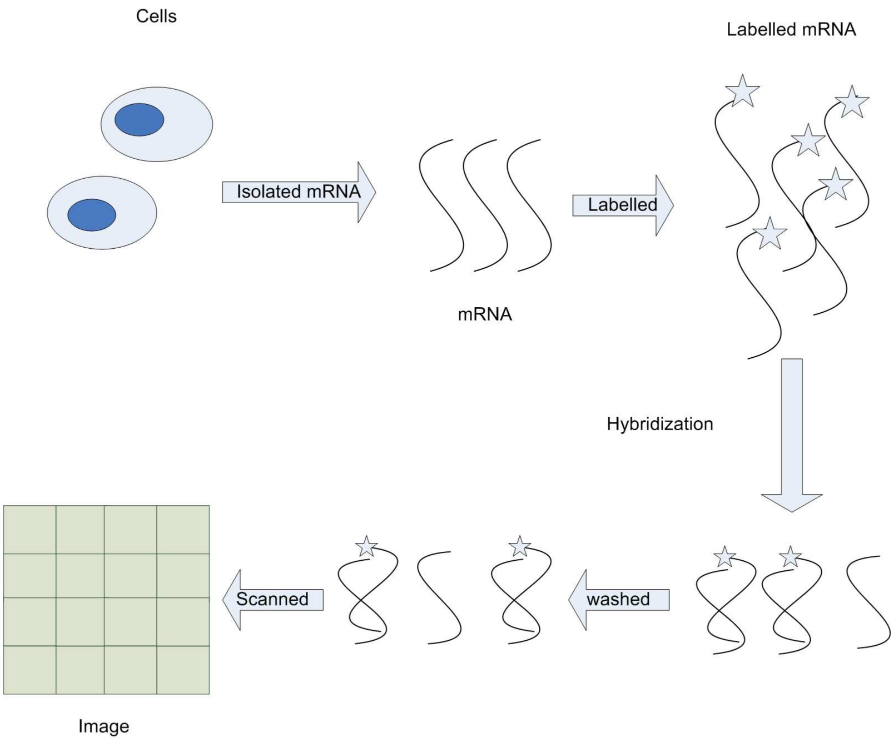


Figure 2.3: Typical oligonucleotide array experiment

dimensional image is created with each probe being identified by its coordinates on the array and measured for its fluorescent intensity. These images need to be further processed to obtain the data known as probe level data. The measured intensity values represent the expression level of the related gene and coordinates on the array and are stored in a cell intensity file (*.CEL) as the final results of the experiments. Each chip corresponds to a CEL file. Probe level analysis methods start with the intensity measurements in CEL files. After gene expression measurements are obtained this data can be used for further analyses such as detecting differentially expressed genes, making inferences from data such as classification or clustering and etc.

### 2.2.3 Gene expression data

Below we will describe how a gene expression matrix is obtained. Biologically patterns of expression are identified by comparing measured expression levels between different states on a gene-by-gene basis. To allow this comparison some transformations, pre-processing algorithms, must be applied to remove the effect of systematic variability (Quackenbush, 2002). As a result of this, an $N$ by $M$ matrix is obtained. $N$ corresponds to the number of genes and $M$ to the number of arrays. We will give the general guideline for pre-processing oligonucleotide arrays. Gene expression measurements are extracted from raw probe level data with pre-processing stage algorithms. Each of these steps can be carried out by different algorithms. Leaving image analysis algorithms out, in the `expresso` code of package `affy` for statistical software `R` there are 315 combinations. However, there is no universally accepted guideline about which method to choose for any algorithm (Cope et al., 2004; Irizarry et al., 2006; Allison et al., 2006; Pearson, 2008). The choice of these methods has an impact on the inferences made from gene expression data. The details of these methods and studies showing experimental results that these algorithms have an impact on the inferences are presented in Chapter 6. These algorithms are applied to probe level data in the order shown below:

- **Image analysis**: In a microarray experiment first quantified values are contained in a two dimensional image produced by the scanner. This image needs to be further analysed to obtain the relative fluorescence intensities, which are mainly known as probe level data. There are various techniques and softwares for doing this (e.g. Brown et al. (2001); Yang et al. (2001); Jain et al. (2002)). The probe level data (CEL files) need to be processed to get the gene expression measurements. We will not discuss the details of image analysis for microarray data in this thesis.

- **Background correction**: The image analysis used to obtain the probe level data includes nearby fluorescence in addition to the spot fluorescence (Qin et al., 2004). This is mainly known as the background noise and the background correction is used to adjust this noise level (Gautier et al., 2004). This is basically done by subtracting local background measurements from spot intensity measurements.

- **Normalization**: There is always a systematic variability between chips used for the microarray experiment (Holloway et al., 2002). Normalization balances the individual hybridization intensities which are caused by the microarray process itself (Quackenbush, 2002). In other words, the normalization step removes the variability between chips so that data from different chips can be compared (Gautier et al., 2004).

After the gene expression matrix is obtained it is ready for further analysis such as making inferences or extracting other useful information.

The gene expression matrix can be analysed in two ways:

1. **Analysing genes by comparing the rows of the matrix**. This includes gene function prediction (Brown et al., 2000), clustering genes with similar functions (Eisen et al., 1998), detecting differentially expressed genes (Causton et al., 2001), analysing periodically expressed genes (Spellman et al., 1998) and the changes in expression of genes according to time (Hooper et al., 2007).

2. **Analysing the samples by comparing the columns of the matrix**. This is usually the case when healthy samples are separated from unhealthy samples (e.g. normal vs. cancer) (Alon et al., 1999).

In this study we show experimental results from both types of problem.

## 2.3 Methods

In this section we will describe the machine learning methods and evaluation measures used throughout this thesis.

### 2.3.1 Learning methods for microarray data

Supervised and unsupervised learning methods are widely applied to microarray data. Supervised learning uses prior information about the datasets, i.e., the class labels for the learning algorithm. Unsupervised learning is used to gain some understanding of the data such as grouping objects with similar patterns (Cristianini and Shawe-Taylor, 2000; Brazma and Vilo, 2000). Different algorithms exist in literature for classification but for the gene expression data Support Vector Machines (SVM) are the state-of-the-art classifiers (Brown et al., 2000; Statnikov et al., 2008). Hierarchical and $K$-means are widely used clustering techniques for gene expression data (Brazma and Vilo, 2000). Below we will give the basic descriptions of the SVM and $K$-means clustering.

### 2.3.2 Support Vector Machines

Support vector machines (SVMs) were first introduced by Vapnik (1998). The SVM was first applied to linearly separable data and then to noisy data with the introduction of slack variables. After the introduction of the kernels, SVM is applied to non-linearly separable data.

SVM has been successfully applied in different fields. The first use of SVM for microarray data was carried out by Brown et al. (2000) for gene function classification of *Saccharomyces cerevisiae*. After the successful application of SVM on transcriptome data, SVM became the most widely applied classification method to such data. Studies showed that SVM is the state-of-the-art classifier for transcriptome data.

#### 2.3.2.1 Linear SVM

Below we will give a description of how SVM works on two-class, linearly separable data. Fig. 2.4(a) shows a two-dimensional linearly separable data. There are many different ways to select a hyperplane that separates the two classes without an error but there is one optimal hyperplane which maximizes the margin between the two classes. SVM finds this optimal hyperplane that maximizes the distance between the closest points of each class. The hyperplane is defined with:

$$y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{2.1}$$

$$= \sum_{i=1}^{n} w_i x_i + b \tag{2.2}$$

where $\mathbf{w}$ is the normal from the origin to the hyperplane and $b$ is the bias. $\mathbf{w}$ and $b$ are learned from the data. A point $\mathbf{x}$ which lies on the hyperplane satisfies $\mathbf{w} \cdot \mathbf{x} + b = 0$ and the perpendicular distance from $\mathbf{x}$ to the origin is defined by $|b|/||\mathbf{w}||$. Let $d_+$ $(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) example and define the "margin" of the separating hyperplane to be $d_+ + d_-$. For linearly separable cases SVM looks for the separating hyperplane with largest margin. This is formulated as: suppose all the training data satisfy the following constraints:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \qquad \text{for} \qquad y_i = +1 \tag{2.3}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \qquad \text{for} \qquad y_i = -1 \tag{2.4}$$

These can be combined into one set of inequalities:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \qquad \forall_i \tag{2.5}$$

For a point which satisfies Eq. 2.3 lying on the hyperplane $H_1 : \mathbf{w} \cdot \mathbf{x}_i + b = 1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $|1 - b|/||\mathbf{w}||$. For a point which satisfies Eq. 2.4 lying on the hyperplane $H_2 : \mathbf{w} \cdot \mathbf{x}_i + b = -1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $|-1-b|/||\mathbf{w}||$. Since $d_+ = d_- = \frac{1}{||\mathbf{w}||}$, the margin is $\frac{|1-b|}{||\mathbf{w}||} - \frac{|-1-b|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$. We want to find the pair of hyperplanes that gives the maximum margin (maximize $\frac{1}{||\mathbf{w}||}$ with respect to Eq. 2.5) by minimizing $||\mathbf{w}||$:

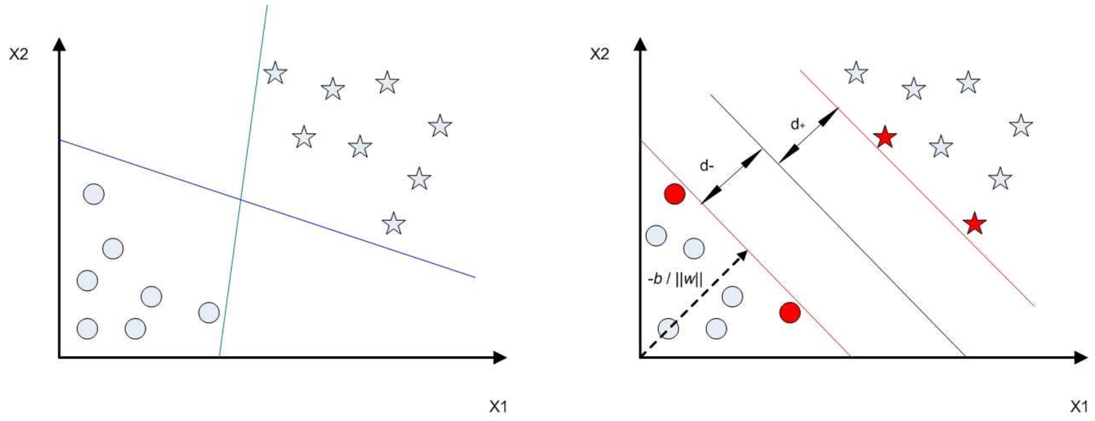
$$\text{Minimize} \quad \frac{1}{2}||\mathbf{w}||^2 \tag{2.6}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0; \quad i = 1, \cdots n; \tag{2.7}$$

This can be done with Lagrange multipliers and details are presented in Appendix B.

#### 2.3.2.2 Soft-margin SVMs

In some cases the training data cannot be separated without error. In this case, the error should be minimized. Cortes and Vapnik (1995) suggest the use of soft-margin



(a)

(b)

FIGURE 2.4: Linearly separable data. Fig. 2.4(a) shows possible hyperplanes that separate the data. Fig. 2.4(b) shows the optimal hyperplane (black) which separates the two classes. Support vectors are marked in red for both classes.

SVMs which will separate the training data with minimum number of errors. A set of variables $\epsilon_i$ are introduced to allow the possibility of examples violating constraint (Eq. 2.5), where $\epsilon_i \geq 0$, $i = 1, \cdots, n$. Eq. 2.5 is now written as:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \epsilon_i \tag{2.8}$$

which use relaxed separation constraint. Any large $\epsilon_i$ will satisfy the constraint on Eq. 2.8. In order to penalize this, it is multiplied by a constant $C$. The new objective function is written as:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \epsilon_i^p \tag{2.9}$$

where $C$ is a penalty parameter that controls the trade off between training errors and maximization of the margin. A small value for $C$ will increase the number of training errors and decrease the margin. A large of $C$ will decrease the number of training errors and maximize the margin. Setting a high value of $C$ will lead to a similar behaviour of hard-margin SVM as in the previous section. This can be seen in Appendix B.1 where it has been shown that $C$ is the upper bound of $\lambda$. Soft margin SVM problem is now:

$$\text{Minimize} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \epsilon_i \tag{2.10}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \epsilon_i; \quad \epsilon_i \geq 0; \quad i = 1, \cdots n; \tag{2.11}$$

$C$ needs to be tuned to find the best value. In our analysis, we tried to find the best

$C$ by partitioning the data into training and testing sets. $C$ which gives the highest accuracy was kept. However it should be noted that there is not only one best $C$ (Gunn, 1998). There may be other values of $C$ that may yield to similar results.

#### 2.3.2.3 Non-linear SVMs

However, real world problems are not linearly separable most of the time. But this type of data can be mapped in feature space using Kernel functions (Burges, 1998). Kernel functions simply replace the dot product in the Eq. 2.2 so that data can be linearly separable in the feature space. The kernel approach is the most important property of SVM which allow the user to classify data set which has noise or simply which is harder to classify with other classification techniques. Below, we will give a description of kernel functions as used in SVM.

#### 2.3.2.4 Kernel methods

Kernel representation offers an alternative solution by projecting the data into a high dimensional feature space (Cristianini and Shawe-Taylor, 2000). In kernel representation, the inner product in linear SVM is replaced by a kernel function:

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{2.12}$$

where all $\mathbf{x}$ and $\mathbf{z} \in X$ and $\phi$ is a mapping from $X$ to feature space $F$.

The input space is defined with $X$ and the feature space is $F = \{\Phi(\mathbf{x}) : \mathbf{x} \in X\}$. Fig. 2.5 shows an example of a feature mapping from two-dimensional input space to two-dimensional feature space. After mapping to two dimensional feature space the two classes can be separated by using a linear classifier.

$$y(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{w}_i \phi_i(\mathbf{x}) + b \tag{2.13}$$

where $\phi : X \longrightarrow F$

Kernel function $K$ computes the inner product $\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$ in feature space directly as a function of the original input space which merges the two steps needed to build a non-linear learning machine.

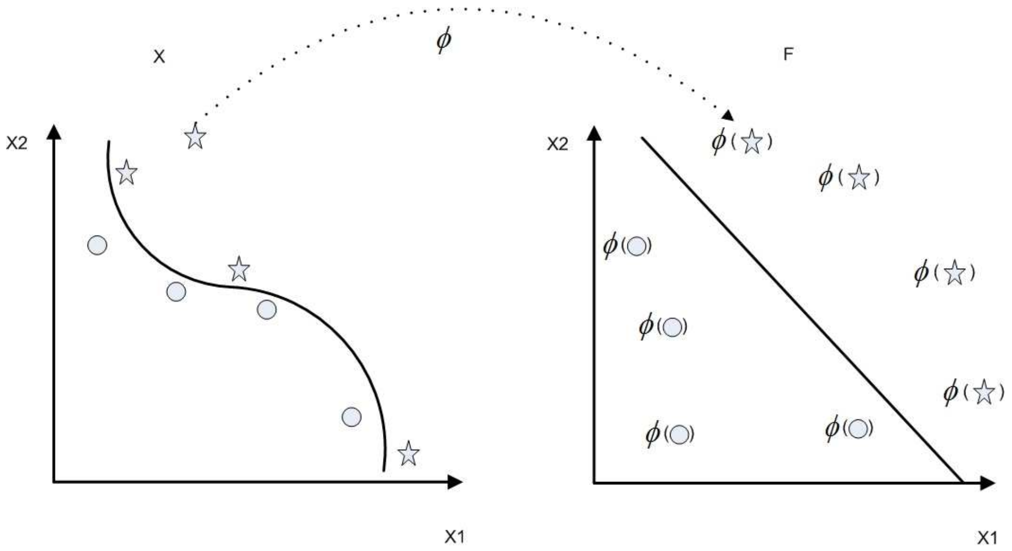


FIGURE 2.5: Example showing a mapping of two dimensional input space to two dimensional feature space where the classes can be separated much easier.

Some examples of kernels are linear kernel:

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z} \tag{2.14}$$

and Gaussian radial basis function (RBF):

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right). \tag{2.15}$$

### 2.3.3 *K*-means clustering

Clustering is an unsupervised learning method, which is used without any prior information, and widely applied to microarray data for discovering genes with similar functions. $K$-means (MacQueen, 1967) and hierarchical clustering are the most widely applied clustering techniques to gene expression data (Causton et al., 2004). Although $K$-means and hierarchical clustering are widely applied, there is no evidence about which clustering is best for gene expression data because of the underlying biology (Causton et al., 2004). Torrente et al. (2005) present some experimental results about the instability of $K$-means clustering on both real and simulated datasets. The study also mentions that it may be possible to get stable results if the data are homogeneously distributed among the groups. However, considering the noise and variability in microarray data, it is hard to expect to get stable results with $K$-means or hierarchical clustering with gene expression data.

$K$-means algorithm starts with random centroids, depending on the number of clusters specified by the user. Usually by using Euclidean distance, closest points, i,e., the sum of squares of the distance to centroids that are minimum, to these centroids are assigned. With the new formed clusters centroids are calculated again and this procedure continues until there is no change in the centroids. The main drawback of $K$-means clustering is the choice of the initial centroids. Overall clustering results are very sensitive to the initial choice of the centroids (MacQueen, 1967). The other main drawback of this method is that it assumes the data from a particular cluster come from a normal distribution.

Hierarchical clustering is the other most widely applied unsupervised learning method for microarray data. The object of hierarchical clustering is to compute a dendogram that collects all elements into a single tree. For any set of $n$ genes, a pairwise similarity matrix is computed. Genes with the highest similarity scores are identified and a node is created for joining these two genes, and a gene expression profile is computed for the node by averaging observation for the joined elements. The similarity matrix is updated with this new node replacing the two joined elements, and the process is repeated $n-1$ times until only a single element remains (Eisen et al., 1998). However, as mentioned in Causton et al. (2004) each iteration produces a fixed cluster and the algorithm does not re-evaluate the clusters that were formed early. This makes the hierarchical clustering less robust i.e., small changes in the data can produce a different clustering and this shows that the hierarchical clustering is very sensitive to noise. Also hierarchical clustering is not very suitable for noisy data. Another criticism about the hierarchical clustering for gene expression data is that there may not always be a hierarchical structure in gene expression data (Causton et al., 2004).

### 2.3.4 Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a linear combination of $M$ Gaussian densities, each having their own mean and standard deviation:

$$p\left(\mathbf{x}\right) = \sum_{k=1}^{M} \pi_k \, \mathsf{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \, \boldsymbol{\sigma}_k\right) \tag{2.16}$$

where $\pi_k$ is called the mixing coefficient and must satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{M} \pi_k = 1$. Each Gaussian density, $\mathsf{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$, is called a component of the mixture model with its own parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$. Figure 2.6 shows an example of GMM with two components.

The goal is to maximize the likelihood function with respect to the parameters (means and covariances of the components and mixing coefficients): The log of the likelihood function is given by:

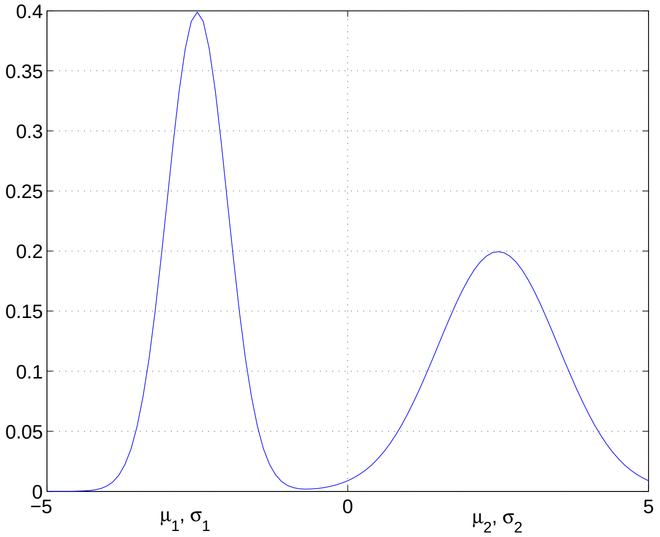


Figure 2.6: A mixture of two Gaussians.

$$\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma}) = \sum_{l=1}^{n} \ln \left\{ \sum_{k=1}^{M} \pi_k \mathsf{N}(\mathbf{x}_l|\boldsymbol{\mu}_k,\boldsymbol{\sigma}_k) \right\} \tag{2.17}$$

The log likelihood function can be maximized with Expectation-Maximization (EM) algorithm. EM is used to estimate the parameters of a distribution and while doing this, it ensures that the likelihood values increases monotonically. The basic procedure is summarized as (Bishop, 2006):

1. Initialize the distribution parameters, $\boldsymbol{\mu}_k$, $\boldsymbol{\sigma}_k$ and $\boldsymbol{\pi}_k$ and evaluate the initial value of log likelihood.

2. Expectation (E)-Step: Evaluate the posterior probabilities of P$(\mathbf{x}|z)$ where $z$ is a latent variable and indicates the probability of $\mathbf{x}$ belonging to which component. Posterior probability is calculated using the current parameter values:

$$\psi(z_{lk}) = \frac{\pi_k \mathsf{N}(\mathbf{x}_l|\boldsymbol{\mu}_k,\boldsymbol{\sigma}_k)}{\sum_{j=1}^{k} \pi_k \mathsf{N}(\mathbf{x}_l|\boldsymbol{\mu}_j,\boldsymbol{\sigma}_j)} \tag{2.18}$$

3. Maximization (M)-Step: Re-estimate the distribution parameters which will maximize the likelihood function. In order to find the new parameters, $\boldsymbol{\mu}^{\text{new}}$ and $\boldsymbol{\sigma}^{\text{new}}$, set the derivatives of $\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma})$ with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ respectively to zero. In order to find the new $\pi^{\text{new}}$, maximize $\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma})$ with respect to $\pi$, use the Lagrange multipliers since $\pi$ has the condition of $\sum \pi = 1$. The new values are

calculated by using the following equations:

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{l=1}^{n}\psi(z_{lk})\mathbf{x}_l \tag{2.19}$$

$$\boldsymbol{\sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{l=1}^{n}\psi(z_{lk})(\mathbf{x}_l - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_l - \boldsymbol{\mu}_k^{\text{new}})^T \tag{2.20}$$

$$\pi_k^{\text{new}} = \frac{N_k}{n} \tag{2.21}$$

where

$$N_k = \sum_{l=1}^{n}\psi(z_{lk}). \tag{2.22}$$

4. Evaluate the log likelihood and check the convergence. If the convergence criterion is not satisfied return to step 2.

These steps are repeated until it convergences. We used the `gmm` function in `NETLAB` software (`http://www.ncrg.aston.ac.uk`) for this purpose.

### 2.3.5 Spectral clustering

Spectral clustering is an unsupervised clustering technique mainly depended on the similarity metric used. Spectral clustering uses the eigenvalue decomposition of the similarity matrix to separate clusters. It has been introduced by Shi and Malik (2000) for image processing and it has also been applied to a wide range of problems. This includes bioinformatics applications such as the work of Higham et al. (2007), Tritchler et al. (2005) and Xing and Karp (2001). Even though $K$-means or hierarchical clustering are the main clustering methods used for microarray data, we chose spectral clustering as our method. The reason for this is that the clusters obtained by $K$-means and hierarchical clustering rely on some other conditions, such as selection of initialization centroids which are discussed in section 2.3.3. As mentioned earlier, spectral clustering uses eigenvectors of the pairwise similarity matrix to partition the data. So, the clusters obtained with spectral clustering mainly relies on the similarity metric used. The most widely used similarity matrix is derived of the negative exponential of a scaled Euclidean distance:

$$\mathbf{A}(i,j) = \exp(-\frac{\parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2}{\sigma^2}) \tag{2.23}$$

where the scale parameter $\sigma$ is a free tuning parameter.

The steps involved in spectral clustering (following the Shi and Malik (2000) algorithm) are summarized as follows:

1. Pairwise similarity matrix $\mathbf{A}_{ij}$ between the genes $i$ and $j$ is calculated by using the Euclidean distance (Eq. 2.23).

2. Compute the normalized Laplacian matrix:

$$\mathbf{L} = \mathbf{D}^{-1/2} \times \mathbf{A} \times \mathbf{D}^{-1/2} \tag{2.24}$$

   where $D(i,i) = \sum_j A(i,j)$

3. Compute the generalized eigenvalue decomposition of $\mathbf{L}$.

$$(\mathbf{D} - \mathbf{L})y_i = \lambda_i \mathbf{D}\mathbf{y}_i \tag{2.25}$$

4. Select the eigenvector corresponding to the second smallest eigenvalue.

The aim of this process is to minimize the disassociation between groups and maximize the association within the group (Shi and Malik, 2000).

Shi and Malik (2000) show that the second generalized eigenvector corresponding to the second smallest eigenvalue manages this and therefore suggests the use of second generalized vector for partitioning the clusters.

Although there are different approaches in choosing the eigenvector (e.g., Perona and Freeman (1998) which use the first eigenvector) for separating clusters, microarray studies use the second eigenvector for separating the clusters. For example Xing and Karp (2001) use Shi and Malik (2000) approach on microarray data and make experiments on leukaemia dataset of Golub et al. (1999), showing that subtypes of cancer can be correctly clustered with this approach. Furthermore, Higham et al. (2007) test different choices of eigenvectors on binary and multi-class microarray data. On two-class problem they found that the second eigenvector, similar to Shi and Malik (2000)' s approach, gives the best partition. On multi-class problem they cannot reach a clear separation of the clusters. Tritchler et al. (2005) introduce a spectral clustering approach which use the covariance matrix rather than the similarity matrix. The first eigenvector of the covariance matrix is used to separate the classes. They show that this approach separates the ALL and AML of leukaemia dataset.

### 2.3.6 Singular value decomposition

Singular Value Decomposition (SVD): for a real $m$ by $n$ matrix $\mathbf{X}$, there exist an $m \times m$ orthogonal matrix $\mathbf{U}$ with $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and a $n \times n$ orthogonal matrix $\mathbf{V}$ with $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ such that:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{2.26}$$

where the $m \times n$ matrix $\mathbf{S}$ has entries $S_{i,i} \geq 0$ for $i = 1, 2, \ldots, \min(m, n)$ and the others are zero (Golub and Van Loan, 1989). The positive constants $\mathbf{S}_i$ are called the singular values of $\mathbf{X}$. The columns of $\mathbf{U}$ are called left singular vectors and the rows of the $\mathbf{V}^T$ are called the right singular vectors.

The singular values are sorted in decreasing order of significance such that $S_1 \geq S_2 \geq \ldots S_l \geq 0$. The fraction of a singular value indicates the relative significance of component $l$ and is calculated as:

$$\text{relative significance} = \frac{\mathrm{S}_\mathrm{l}^2}{\sum_{\mathrm{k}=1}^{\mathrm{L}} \mathrm{S}_\mathrm{k}^2} \tag{2.27}$$

The magnitude of relative significance is a numerical indicator of how much each principal component capture the data.

In the context of microarray data, let $\mathbf{X}_{i,j}$ be the expression value of $i^{th}$ gene in the $j^{th}$ array, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. Then SVD is the linear transformation of the expression data from $N$ genes × $M$ array space to reduced $L$ "eigenarrays" × $L$ "eigengenes" space, where $L = \min\{M, N\}$ (Alter et al., 2000).

The columns of $\mathbf{U}$ are called *eigenarrays* and the rows of the $\mathbf{V}^T$ are called the *eigengenes* in the terminology used by Alter et al. (2000). By analogy with Principal Component Analysis (PCA) eigengenes or eigenarrays can also be referred to as *components*. Figure 2.7 shows the graphical illustration of SVD in the context of microarray data.

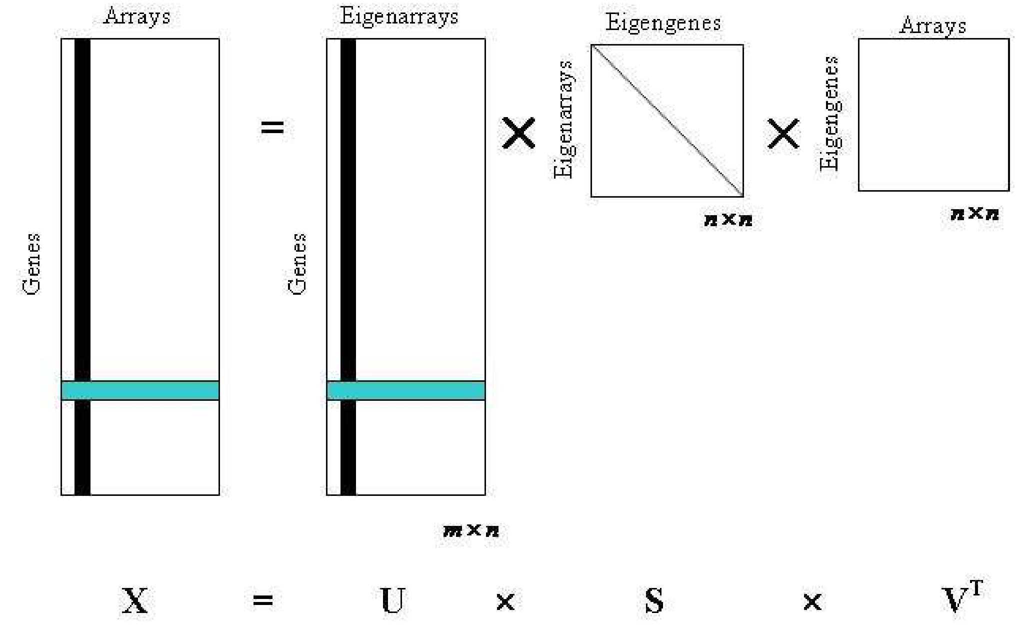


FIGURE 2.7: Graphical representation of SVD

### 2.3.7 Gamma models for oligonucleotide array signals

To analyse probe level uncertainties (Milo et al. (2003)) we used the Propagating Uncertainty in Microarray Analysis (`PUMA`) (Pearson et al., 2009) Bioconductor package with `R`, downloaded from the site (`www.bioinf.manchester.ac.uk/resources/puma/`). In particular, we used multi-mgMOS (Liu et al., 2005) where gamma models are used to extract the expression values and uncertainties from probe level, CEL files. multi-mgMOS is the modified version of gMOS (Milo et al., 2003). The definition of the basic model gMOS will be followed by the definition of multi-mgMOS below.

In Affymetrix GeneChip technology each gene is represented by a set of 11-20 pairs of probes (Irizarry et al., 2003a). Each probe pair is composed of a perfect match (PM) and mismatch (MM). PM probe is designed to measure the specific hybridization and MM probe is used to measure the cross hybridization. There are many different algorithms for achieving this. Milo et al. (2003)' s probabilistic gMOS analysis assumes that PM and MM comes from a Gamma distribution with the same inverse scale factor, $b$, and different shapes, $\alpha$ and $a$.

The Gamma distribution, $\Gamma$, can be defined as:

$$\Gamma(\tau|a,b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} \exp(-b\tau) \tag{2.28}$$

The multi-mgMOS shares the scale parameters in gamma distribution across all chips to reflect the actual characteristics of probe sequences of the same type of chip. Let $y_{gjc}$ and $m_{gjc}$ represent the $j$th PM and MM intensities respectively for the $g$th probe set under the $c$th condition (chip). The multi-mgMOS model is represented as:

$$y_{gjc} = \Gamma(a_{gc} + \alpha_{gc}, b_{gj}) \tag{2.29}$$

$$m_{gjc} = \Gamma(a_{gc} + \phi\alpha_{gc}, b_{gj}) \tag{2.30}$$

From Eq. 2.29 and 2.30, thue signal for the $j$th probe pair in the $g$th probe set of the $c$th chip, $s_{gjc}$ follows the gamma distribution:

$$s_{gjc} = \Gamma(\alpha_{gc}, b_{gj}) \tag{2.31}$$

Including uncertainties from the probe level had been shown previously by Rattray et al. (2006) and Sanguinetti et al. (2005) to improve the results of the Principal Component Analysis (PCA) of the microarray studies, clustering (Liu et al., 2007) and detecting differentially expressed genes (Liu et al., 2006).

### 2.3.8 `mas5calls`

Detection calls are used to detect whether the transcript of a gene is present or absent (Liu et al., 2002). These detection calls for $i^{\text{th}}$ probe set are calculated with the difference between PM and MM. Discrimination score is defined as:

$$R_i = \frac{\text{PM}_i - \text{MM}_i}{\text{PM}_i + \text{MM}_i} \tag{2.32}$$

The `mas5calls` function of package `affy` in Bioconductor calculates p-values with Wilcoxon signed-rank test for the hypothesis test:

$$\begin{aligned} H_0 &= \text{median}(R_i) = \tau \\ H_1 &= \text{median}(R_i) > \tau \end{aligned}$$

where $\tau$ is small positive constant. Default value of $\tau$ for `mas5calls` is 0.0015.

$$\begin{gathered} \text{if p-value} < \alpha_1 \text{ then it is present (P),} \\ \text{if } \alpha_1 \leq \text{p-value} < \alpha_2\text{, then it is marginal (M),} \\ \text{if } \alpha_2 \leq \text{p-value then it is absent (A).} \end{gathered}$$

Default values for `mas5calls` are $\alpha_1 = 0.04$ and $\alpha_2 = 0.06$. Default values are used for all the `mas5calls` experiments.

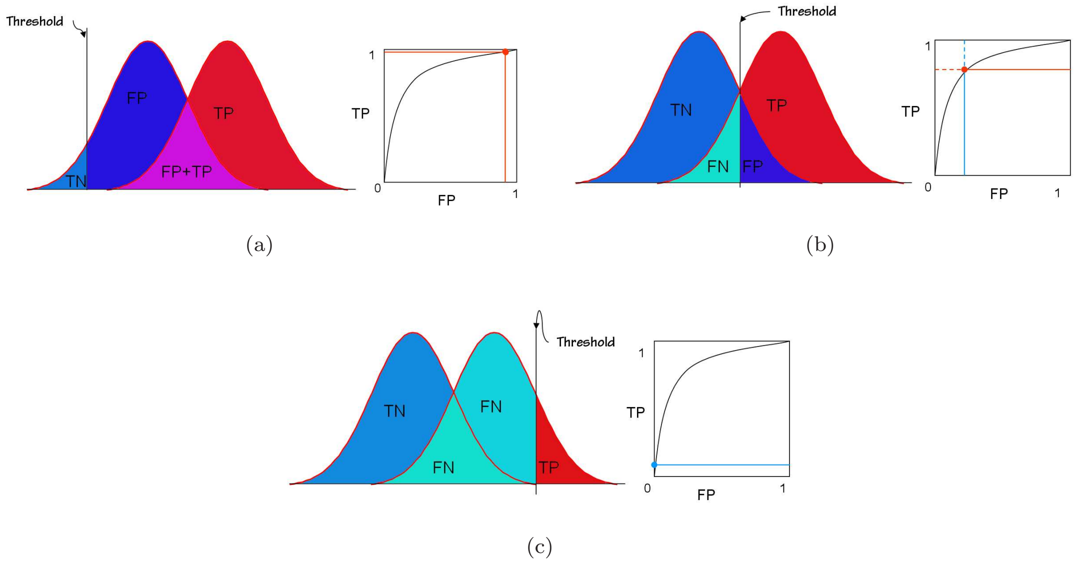
`mas5calls` is a non-parametric approach and therefore it is less sensitive to outliers. Also `mas5calls` is a widely applied method for Affymertix GeneChip technology. In order to show that the uncertainties obtained with multi-mgMOS are not affected by noise, p-values and uncertainties are compared. We used `mas5calls` to obtain p-values for the detection calls and `mas5` function to extract the associated gene expression values. We calculated the average p-values for the expressed genes only. These results were compared with the uncertainties, obtained with multi-mgMOS, for expressed genes.

### 2.3.9 Evaluation measures

In the following subsection we give the details of the evaluation measures used throughout this work. For classification we mostly used Area Under ROC Curve (AUROC) and for the rest of the inferences used correlation coefficient, and Fisher ratio and F1 score.

#### 2.3.9.1 ROC curve and AUROC

Receiver Operating Characteristic (ROC) curve is a method used to evaluate the relationship between sensitivity (True Positive) and specificity (True Negative) for all



(a)

(b)

(c)

Figure 2.8: Figure showing how the Area under ROC Curve is obtained. For every threshold, a single point on the curve is obtained. The most desirable point is where TP is high and FP is low.

possible threshold values. The vertical axis shows the true positive rates and horizontal axis shows the false positive rates (1 - true negative). See Table 2.1 for the definition of True positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Each point on the ROC curve represents the combination of sensitivity and specificity at a given threshold value (Fig. 2.8). The advantage of ROC curve over the accuracy of a classifier is that ROC removes the effect of the choice of threshold and reports more general and applicable results (van Erkel and Pattynama, 1998). The ideal case would be TP = 1 and FP = 0 which will correspond to the upper-left hand corner of the ROC curve.

Table 2.1: Confusion matrix, summary of possible classification results.

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | TP | FP |
| Predicted Negative | FN | TN |

For evaluating the performance of a classifier we used Area under ROC curve (AUROC) which is a probability measure between 0.5 (no discrimination) and 1 (perfect discrimination). AUROC is a rank statistic and it is the probability that a classifier will rank randomly chosen positive instance higher than a randomly chosen negative instance. The statistical interpretation of AUROC is given by Hand and Till (2001). Let $p(x)$ be the probability that an object with measurement $x$ belongs to class 1.

- $g(p) = g(p(x)|1)$ is the probability of an object belonging to class 1 which actually belongs to class 1. Corresponding cumulative distribution function is then represented as $G(p)$. This corresponds to TP.

- $f(p) = f(p(x)| - 1)$ is the probability of an object belonging to class 1 which actually belongs to class $-1$. Corresponding cumulative distribution function is then represented as $F(p)$. This corresponds to FP.

By plotting $F(p)$ or FP on the $x$-axis and $G(p)$ or TP on the $y$-axis we get the ROC curve. This plot lies in a unit square. So, any point above the diagonal corresponds to $G(p) > F(p)$. By using a threshold $t$, we can get obtain a value of $G(p)$ corresponding to $F(p)$. By plotting these we get the ROC curve. The area under ROC curve (AUROC), thus give us the probability of ranking TP.

From this definition the area under the ROC curve is defined as:

$$\begin{aligned} \text{AUROC} &= \int G(p)dF(p) \qquad (2.33) \\ &= \int G(p)f(p)dp \qquad (2.34) \end{aligned}$$

Now that we define the AUROC, lets think about an arbitrary point, $t$ which is randomly chosen from class 1 points. The probability that any randomly chosen point being higher or equal to $t$ is $G(t)$. If $t$ is chosen from the distribution $F$, which is class $-1$, then the probability that the randomly chosen member of class 1 will have a higher probability of belonging to class $-1$ is:

$$\int G(p)f(p)dp \qquad (2.35)$$

which is equal to the AUROC as defined in Eq. 2.34.

We took an alternative approach to calculate the AUROC. Trapezoidal rule is used to approximate the area under the curve. Trapezoidal approach uses the unit spacing, i.e., by calculating the area of rectangles and triangles under the curve, to calculate the area, which can be formulated as:

$$A = \frac{\sum_{i=2}^{n} ((x_i - x_{i-1})(y_i + y_{i-1}))}{2} \qquad (2.36)$$

AUROC is superior to accuracy measure of a classifier, as accuracy only considers one threshold but AUROC eliminate the choice of the threshold and reports a more general result. For that reason we used AUROC to evaluate the performance of classifiers except in Chapter 5 where using accuracy as the evaluation measure is more appropriate for comparison reasons.

#### 2.3.9.2 Correlation coefficient

Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the linear correlation coefficient is calculated as:

$$r(\mathbf{x},\mathbf{y}) = \frac{\mathrm{Cov}(\mathbf{x},\mathbf{y})}{\sqrt{\mathrm{Var}(\mathbf{x})\mathrm{Var}(\mathbf{y})}} \tag{2.37}$$

$$= \frac{(1/N)\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\left[(1/N)\sum_{i=1}^{N}(x_i - \overline{x})\right]^{1/2}\left[(1/N)\sum_{i=1}^{N}(y_i - \overline{y})\right]^{1/2}} \tag{2.38}$$

where $\overline{x}$ and $\overline{y}$ are the mean of $\mathbf{x}$ and $\mathbf{y}$ respectively. If $\mathbf{x}$ and $\mathbf{y}$ are uncorrelated, $r$ is expected to be around zero. For perfect correlation (anticorrelation), $r$ is expected to be close to $+1(-1)$.

#### 2.3.9.3 Fisher ratio

Fisher score/Fisher ratio is a measure of class distinction which reflects the difference between classes relative to the standard deviation within the classes. High dimensional data is projected onto one dimensional space and by considering the parameters ($\mu_i$ and $\sigma_i$ where $i = 1, 2$) obtained from the two classes. It is calculated as:

$$\text{Fisher score} = \frac{\mathrm{abs}(\mu_1 - \mu_2)}{\sigma_1 + \sigma_2} \tag{2.39}$$

where $\mu_1$, $\sigma_1$ and $\mu_2$, $\sigma_2$ are the mean and the standard deviation for the first and second classes respectively. Tighter classes have smaller variance. The difference between the means should be higher and the standard deviation of each class should be lower for linearly separable cases. Fisher ratio provides an insight of how much two classes are separable. The higher the score the more separable are the two classes.

Fisher ratio has been used for gene expression data for selecting the most discriminant genes (e.g., Golub et al. (1999)).

#### 2.3.9.4 F1 measure

To compare overlap between genes in a particular cluster when clustering is applied at different levels of precision, we used the $F1$ measure, used widely in information retrieval problems, defined as

$$\text{recall} = \frac{\text{relevant documents categorized as relevant}}{\text{total relevant documents}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.40}$$

$$\text{precision} = \frac{\text{relevant documents categorized as relevant}}{\text{total documents categorized as relevant}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.41}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.42}$$

Here, the term *recall*, equivalent to true positives in a classification problem, refers to the fraction of genes in the original cluster correctly identified in clusters obtained with continuous data. *Precision* is the true positives expressed as a fraction of the sum of true positives and false positives.

## 2.4 Summary

In this chapter we gave the basic descriptions of the gene expression data, the machine learning methods to analyse such data and the evaluation metrics used throughout this study. First we described the central dogma of molecular biology, how it is related to the data we are using, then the basics of what microarray technology is, how gene expression measurements are obtained and how these measurements can be used. We gave details of the two most commonly used array types i.e., cDNA and oligonucleotide arrays. Then we gave the definition of statistical methods used for making inferences from gene expression data and also the statistical evaluation metrics that are used. The aim of this chapter is to make it easier for the reader to follow the discussions later in this work.

# Chapter 3

# Literature review

## 3.1 Introduction

In this chapter we review the use of quantized / binary gene expression data in literature. We have focused on the studies which use statistical inferences from the transcriptome data. Although there are wide applications of boolean network (which uses binarized data naturally) for gene regulatory network, we do not give detailed description of these studies. Only for two studies which are network studies, we briefly describe the quantization method used. Since we are not using any gene regulatory networks in this study we have not included those parts. We give details of how each study quantizes the data, how they use this for statistical inferences and make comparisons, identify the drawbacks of the existing studies, and how our work fills the gaps. Quite simply we tried to explain where our work stands compared to others. This chapter will help explain our contribution.

## 3.2 Use of quantized microarray data in literature

From the start of the wide usage of microarray analysis, quantizing transcriptome data has been used in the literature. One of the earliest works using quantized gene expression data is carried out by Brazma et al. (1998) where the study uses discretized data for the purpose of clustering genes on time series data. Yeast genes are the data used. The study considers three or five intervals defined by one and two thresholds respectively. After log of the expression values is taken, these values are discretized with the predefined threshold(s) according to the following criteria: for three level with one threshold $h$ : $(-\infty, -\log h]$, $(-\log h, \log h]$ and $[\log h, +\infty)$ for $-1$, 0 and $+1$ and for five intervals with two thresholds $h_1$ and $h_2$ with the criteria $(-\infty, -\log h_2]$, $(-\log h_2, -\log h_1]$, $(-\log h_1, \log h_1)$, $[logh_1, \log h_2)$ and $[logh_2, +\infty)$ for $-2$, $-1$, 0, $+1$ and $+2$. However, the study does not give any details about how the thresholds $h$ or $h_1$ and $h_2$ are fixed. After the discretizing procedure is completed, genes that have the same discretized sequence are assigned to the same clusters. The study claims that this technique works well for clustering genes with similar functions. By showing that quantized representation works fine for gene expression data, the work of Brazma et al. (1998) is a good example to support our claim that quantized gene expression data do not lose much information.

Friedman et al. (2000) use a probabilistic model, namely Bayesian network, to analyse gene expression data. They took two approaches to learn a local probabilistic model for specifying a Bayesian network. Their first model, *multinomial* model, uses the discretization approach where gene expression data is quantized into three levels: under-expressed ($-1$), normal (0) and over-expressed (1). They define the threshold to be the average expression level of the gene across the experiments. They compare these results with the second model, *linear Gaussian* model. Their conclusion is that the multinomial model does a better job than the linear Gaussian model, but they also mention that there is clearly information loss when data is quantized. However, as Friedman et al. (2000) conclude the quantized data does a better job and as can also be seen from the results of our study information loss with quantized data is not the case. By making inferences from binary data and comparing it with continuous data, we show that binarized data preserves enough information to make correct inferences. The choice of the threshold affects the overall results obtained in this study. It is another issue to discuss whether taking the average as the threshold is a good measure or not. The level of noise in the data affects the choice of threshold. Also there is no underlying properties of the data for defining the threshold. Pe'er et al. (2001) expand the work of Friedman et al. (2000) by using a different method to quantize data. Pe'er et al. (2001) quantization uses a mixture of Gaussians where each component corresponds to a specific state. $K$-means clustering is used to estimate the mixture. There are two problems facing this approach. The first is to determine the number of states of a gene. The second one is what should be the initial value of $K$-means. To overcome these problems the study defines the control expression values for a gene. This control expression value is basically the baseline expression value (i.e., measurements of expression without disruption) of the gene. From the baseline expression values the distribution is estimated. Then by considering the state of that specific gene in perturbed samples the states of the gene (over-expressed, under-expressed, etc.) are determined with respect to its baseline measurements. The number of the different states of the gene is also the number of $K$. After $K$ is obtained, $K$-means clustering is run and expression values are discretized according to the outcome of this procedure. However, the results of $K$-means clustering analysis is very sensitive to the noise present in the data and this will have a effect on the overall result of the analysis.

Park et al. (2001) propose a non-parametric scoring function for selecting the informative genes in a microarray study. They focus on the phenotype classification problem (i.e., cancer vs. normal). The algorithm first sorts the data in such a way that samples (patients) in one group are separated from the second group. Then the gene expression values for samples in the first group are assigned the score 0 and the gene expression values for samples in the second group are assigned the score 1. Next the expression values of that particular gene are ranked across the sample. Based on this sequence they compute a score statistic that measures the disorder of 0's and 1' s. The score is calculated as the number of swaps of consecutive digits necessary to reach to the perfect split which is all the 0' s on the left and all the 1' s on the right hand side. The smaller the score the better discriminant is the gene. Their claim is that since this approach uses the rank, rather than the actual gene expression values, it is robust to the outliers. They only test this algorithm on the benchmark dataset of Golub et al. (1999) which is one of the easiest problems to discriminate. The algorithm finds only one gene, *Zyxin*, that has a perfect separation between the two classes. However, they fail to test this algorithm on a more complex problem or make any comparison with the existing algorithms for finding the discriminant genes, such as Fisher score, as described in Golub et al. (1999), Significance Analysis of Microarrays (SAM) (Tusher et al., 2001) or Principal Component Analysis (PCA). This algorithm needs to be further analysed on harder problems before drawing any conclusions.

The studies mentioned so far use quantization as a pre-processing step for their inference algorithm, rather than studying the effect of quantization on gene expression data. Shmulevich and Zhang (2002)' s work analyses the effect of quantization. Their study mainly analyses making inferences with binary gene expression data. However, the study is limited to unsupervised learning and to two datasets. Before applying binarization, data is normalized gene by gene to remove systematic variability as described in section 2.2.3. After this, a threshold is selected and gene expression measurements which are higher than the threshold are given the value 1 and defined as expressed genes. The gene expression measurements which are lower than the threshold are given the value 0 and defined as not expressed genes. Threshold is selected by examining the sorted data and finding the 'highest jump' between the simultaneous points. The choice of threshold in this study is subjective and thus results are seen to be dubious. After data is binarized, similarity matrix is calculated with Hamming distance. The similarity matrix is further analysed with Multidimensional Scaling (MDS) to separate the different classes. Their method is successfully implemented on two different datasets, i.e., tumour types are successfully discriminated using binary data, and thus the authors point out that information loss, when data is quantized, is not much. However, the authors do not compare their results with the inferences made with continuous data. This study should have considered other types of inferences before reaching this conclusion. Our study considers different types of microarray data (cDNA array and Affymetrix) in a wider range of inferences including classification, clustering, periodicity detection, analysing time series data and detecting cell cycle genes using singular value decomposition. We also present a comparison with the inferences made with continuous data.

Another study to classify the leukaemia dataset (Golub et al., 1999) was conducted by Mircean et al. (2002). Their approach evaluates different types of metrics for use with k-NN classifiers. The study considers four different types of metrics: correlation coefficient, Euclidean distance, Mahalanobis distance and Entropy correlation coefficient. First they apply these metrics to continuous data and then to quantized data. Prior to applying k-NN classifier they apply gene selection to continuous data. After this data is quantized and different metrics are tested with k-NN classifier. However, they fail to present how results would be affected if they had applied gene selection to quantized data rather than the continuous data, one of the drawbacks of this study. Two different quantization methods are considered in this study. First one assumes that data is normally distributed and it quantize the data into three levels. The thresholds are defined with $\mu-\frac{\sigma}{2}$ and $\mu+\frac{\sigma}{2}$, where $\mu$ and $\sigma$ are the global parameters The partition would be: [minValue, $\mu - \frac{\sigma}{2}$); $(\mu-\frac{\sigma}{2}$ , $\mu+\frac{\sigma}{2})$ and $(\mu+\frac{\sigma}{2}$, maxValue]. This method of selecting threshold is not the best way as microarray data hardly has Gaussian distribution due to the noise caused at the pre-processing stage of the microarray data. The second quantization method considered in this study is Lloyd algorithm (Lloyd, 1982). Lloyd algorithm as described in this paper starts with a randomly chosen centroid and by using the nearest neighbor search, assigns the data into three classes. This procedure goes on until average distance is below the pre-defined threshold. Again the same problem of data not always being Gaussian distribution arises here. Also the choice of initial centroids is one of the known drawbacks of such algorithms. e.g. K-means clustering. For K-means clustering it is widely known that it is very rare to get similar clusters using the same datasets. The threshold obtained with Lloyd algorithm thus is not reproducible and results can change, making it very hard to draw conclusions about the metric used. Despite all these drawbacks of this work, the study concludes that Entropy correlation coefficient performs best for the k-NN classifier with quantized data (using Lloyd algorithm). However, this study only comes to this conclusion by comparing the results obtained from the leukaemia dataset, and nothing else. There is no comparison with the other classifiers such as the state-of-the-art classifier, SVM. Following the discussion above, the conclusion reported in this study may be obtained solely by chance.

Zhou et al. (2003) use mixture of Gaussians to binarize gene expression data. Their approach considers expressed and not expressed genes each having a density function. GMM model is a mixture of two or more Gaussian distributions, each having their own mean $\mu_i$ and standard deviation $\sigma_i$ (Bishop, 2006). Each Gaussian density is called the component of the mixture model (Bishop, 2006). Microarray measurements having expressed and not expressed genes can be fitted to a two centre GMM model. Expressed genes can be considered as one component and the not expressed ones can be considered as the other component of the GMM distribution (Zhou et al., 2003). Zhou et al. (2003) binarization model uses the ratios of the microarray measurements. Assume a gene whose values, $U$, come from a normal distribution. This model has a multiplicative factor $K$ where $K > 1$ and the expressed genes are represented as $KU$. $KU$ also follows a normal distribution due to $U$ being normal. $B$ would be a random variable modelling the values of the reference channel. If we take the logs of these values not expressed genes can be represented as:

$$\log \frac{U}{B} = \log U - \log B \tag{3.1}$$

And the expressed genes can be represented as

$$\log \frac{KU}{B} = \log K + \log U - \log B \tag{3.2}$$

If $B$ is simply not considered as a random variable the variable $\log K$ in the expression $\log KU/B$ can be considered as a shift which makes the data to fit to a mixture of two Gaussian distributions. Figure 4.2 (a) and (b) show two and three centre GMM distributions respectively applied to microarray data by following the above idea.

The parameters of the distribution for microarray data is estimated by fitting them to a mixture model. This is done by using NETLAB Toolbox package available online at `http://www.ncrg.aston.ac.uk/netlab/index.php` (Nabney, 2002). After obtaining the parameters of each component ($\mu_1$, $\sigma_1$ and $\mu_2$, $\sigma_2$) from the distribution and assuming that $\mu_1 < \mu_2$ a threshold $Th$ is defined as below:

$$Th = \frac{\mu_1 + \sigma_1 + \mu_2 - \sigma_2}{2} \tag{3.3}$$

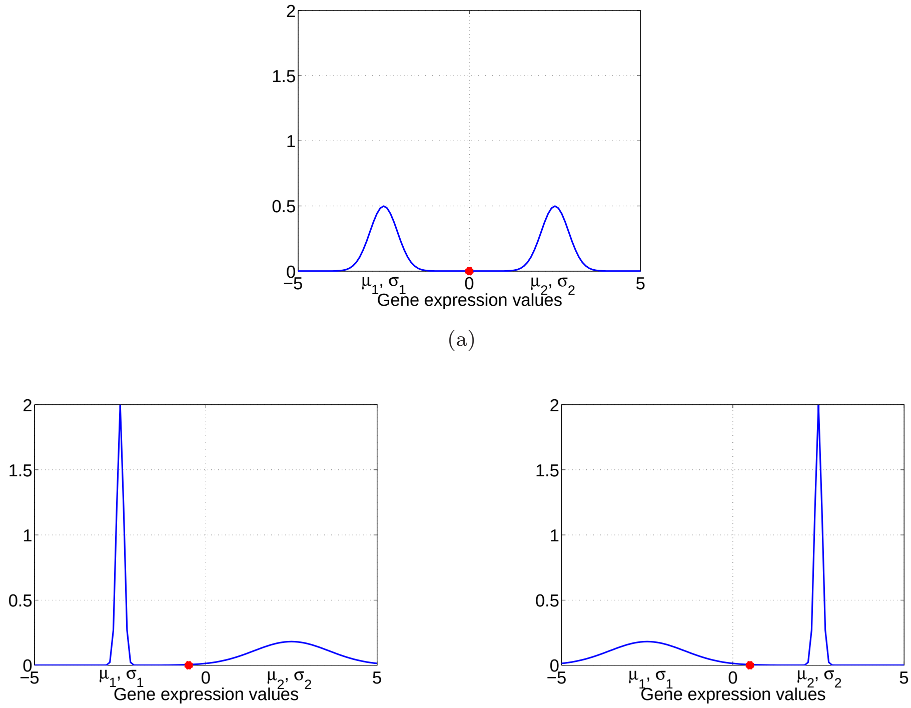
To illustrate how the formulae (Eq. 3.3) of Zhou et al. (2003) successfully defines the threshold we generated three different mixture of Gaussians:

$$p(\mathbf{x}) = \sum_{k=1}^{M} \pi_k \, \mathsf{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \tag{3.4}$$

where $\pi_k = 0.5$ and $M = 2$ for this particular example. The three models considered here are:

- $\sigma_1 = \sigma_2$ as illustrated in Figure 3.1(a)
- $\sigma_1 < \sigma_2$ as illustrated in Figure 3.1(b)
- $\sigma_1 > \sigma_2$ as illustrated in Figure 3.1(c)

In order to capture the expressed and not expressed genes there are three possible combinations of GMM. Fig. 3.1(a) shows when expressed and not expressed genes have equal variance. In this case the threshold is expected to be right in the middle, so $\sigma_1 - \sigma_2$ will balance the threshold to be right in the middle. In the second case (Fig. 3.1(b)) if $\sigma_1 < \sigma_2$ we can not expect the threshold to be just in the centre and we need to shift the threshold from right to left. This is achieved by $\sigma_1 - \sigma_2$. The negative value will shift the threshold from the center to the left. In the last case Fig. 3.1(c) when $\sigma_1 > \sigma_2$ we do not want the threshold to be right in the middle, and want it to be shifted to the right. This is achieved by $\sigma_1 - \sigma_2$ which will give us a positive value to shift the threshold from center to the right. Figure 3.1 shows an example where two center GMMs are generated in MATLAB using Eq 3.4. Red dots in the figures show the thresholds obtained with Zhou et al. (2003)' s formula.



(a)

(b)

(c)

Figure 3.1: Figure showing why the Zhou et al. (2003)' s formula of discretization work well on microarray data.

Potamias et al. (2004) propose a gene selection method and a new classifier metric based on quantized gene expression data. The genes are quantized and then the gene selection procedure is applied. The approach taken for selecting genes is very similar to Park et al. (2001)' s work. The main difference between the work of Potamias et al. (2004) and the work of Park et al. (2001) is that Potamias et al. (2004) apply gene selection to quantized data. First, the expression values ($L$) of a gene are sorted. For all consecutive pairs, the mid-points ($\mu_k$) are calculated simply by taking their means. By using these mid-points, subsets of the original expression values are formed ($H_k$ and $L_k$). For each subset authors compute the information gain by using the formulae: $IG(\mu_k) = E(L) - E(H_k, L_k)$ where $E(L)$ stands for the original set of numbers $L$, with respect to their assignment to true classes and $E(H_k, L_k)$ stands for the entropy of the system when the set of numbers

L is split into the disjoint sets $H_k$ and $L_k$. The subset which gives the maximum information gain is selected and this procedure is carried out for each gene separately. After quantizing the gene expression values they assign a score for each gene depending on the number of $h$ s and $l$ s. The higher the score the more discriminant is the gene. Such a simple approach to such complicated data can not always be expected to give good results all time. The authors also propose a method which determines the number of best discriminative genes. By using the score mentioned above they sort the genes and, by comparing the highest jump points between the scores, the authors group the genes. The number of groups depends on the number of high jumps. They apply this to positive and negative classes separately. And select the best subset of genes that give the highest accuracy on the proposed classification method. The authors further propose a classification method for classifying gene expression data. With simplest explanation, the proposed metric consider the number of $h$ s and $l$ s in the selected genes and assign the genes into positive or negative class depending on the number of $h$ s and $l$ s. The authors show this application on five different datasets.

The authors proposed a gene selection method and also a classification method in the same paper. However, it is hard to evaluate which one is more effective as they do not make any comparison with the existing algorithms. Therefore it is not obvious whether classifier or the gene selection procedure is more effective. In general the results reported here seem dubious. The authors compare some results from the published papers but results presented in the paper and the results presented in actual papers do not seem to match (e.g., leukaemia dataset' s result).

Mircean et al. (2004) use quantized gene expression data to classify three subtypes of cancer with k-NN classifier. As different from the first study (Mircean et al., 2002), authors suggest that quantizing gene expression data is not a loss of information because of the underlying biological properties. However, they do not present any details about these properties in the paper except that there is a biological variability and thus the concern about reproducibility of microarray measurements. Secondly, they suggest that quantizing remove the effect of noise. The authors compare this procedure of quantizing gene expression data to the *rate-distortion theory* where there is 'controlled' loss of information and which showed some advances in communication technology. The method of quantization is Lloyd's algorithm ($K$-means) again, which is very sensitive to the distribution of the data and also results heavily rely on the choice of the starting points. The study applies gene selection prior to the classification procedure. Genes are selected by sorting the ratio of between-class and within-class variances: $R_{BW}(i) = B_i/W_i$, genes with largest ratio are selected. The authors note that the procedure of selecting genes with quantized and continuous data returned different set of genes. With ternary quantized data, using correlation coefficient as the similarity metric and with 40 selected genes, the smallest error rate is 0.0025 while for unquantized data, smallest error rate is 0.05 with 320 genes. They also used these 40 genes, as continuous data and apply the k-NN classifier finding the error rate to be 0.08. From these, authors conclude that quantizing gene expression data has a beneficial effect. As a second step study compares the effect of similarity metrics and uses entropy correlation coefficient as the distance measure. With entropy correlation coefficient as the similarity metric smallest error rate is 0.007 with the selected 80 genes. The study also makes predictions with binary and four-level quantization but they argue the best quantization level should be the same as number of classes. When they test the error rate for binary and four-level quantized data, they do not find any improvements in the performance of classifier. Study also applied hierarchical clustering to the 40 selected set of genes with correlation coefficient as the similarity metric and managed to cluster samples correctly. The authors reach all these conclusions using only one single dataset and do not compare their results with anything else. Study fails to use state of the art classifier SVM to make a stronger claim. It may well be that this data is not discriminable with k-NN. The drawbacks of such methods like distance-to-template classifiers are further discussed below. Even though this is not an extensive study, the results are good examples of the usefulness of quantized gene expression data.

Quantized gene expression data had been applied for integrating data from cross-platform analysis. Warnat et al. (2005) used quantized gene expression data for cross-platform analysis, focusing on classification using SVM. The study uses breast and prostate cancer datasets from different platforms, namely cDNA and oligonucleotide arrays, and find the common genes in two different platforms by using UniGene database. But due to different platforms using different protocols, integrating data directly is not possible. For this purpose Warnat et al. (2005) use two different approaches and quantization is one of them. First one is to use Median Rank Score (MRS) and the second one is Quantile Discretization (QD). With QD the expression values of all arrays are discretized into predetermined number of bins. In this study the number bins used is 8. QD is based on equal frequency binning. Each array is divided into 8 subsets, each having equal number of values. The cut points are the quantiles of the array expression. After defining the bins every expression value is replaced with an integer according to the bins they are assigned to. The two central bins are merged into one. The new central bin is assigned 0 and the values above the median are assigned positive integers and the ones below the median are assigned negative integer values. By using these approaches the study concludes that cross-platform analysis yields better results than individual datasets. They suggest that the main reason for this is that there are more samples. However, they note that using quantized gene expression results in losing information. This proposition is not backed up with any experimental results. The reason to choose the predetermined number of bins as 8 is not explained in the study and also there is no experiment with different number of bins. The study do not make any comparison with these two methods. But by looking at their results it can be seen that results are very close.

Fuller et al. (2005) studied the sub-type classification of brain tumours. The study considered k-nearest neighbour (k-NN) and Fisher discriminant as the classifier. These methods were applied to quantized gene expression data. By calculating the error rates the authors evaluate the performance of their algorithm. Quantization is achieved with Lloyd's algorithm ($K$-means) and applied to each sample separately. The study considered binary, three level and four level quantized data. The reasons for authors to suggest quantized gene expression data, like in the rest of the studies in the literature, is to get rid of the effect of noise i.e., quantization will remove the unwanted sources of variability. After data is quantized, the most discriminative 50 genes out of 2303 are selected according to the ratio of within group sum of squares (WSS) and between group sum of squares (BSS) as described in Mircean et al. (2002). The study concludes that quantized gene expression data is useful in detecting the sub-types of brain cancer and they concluded that four level quantization gives the best result compared to the binary and three level data. Even though the study aims to remove the effect of noise they used Lloyd algorithm as their quantization method which is sensitive to noise and only works well if data comes from a normal distribution. Another drawback of this study is that it just tests this algorithm on one dataset. Despite these small drawbacks of the study the results can be interpreted as showing same usefulness of quantized data.

Di Camillo et al. (2005) propose a preliminary step for identifying relations among genes and to construct gene regulatory networks for short time series data. The proposed method uses quantized gene expression values. The authors' motivation is that the number of samples are very small when compared to the number of genes and this makes it hard to get model parameters right. By quantizing the data the authors aim to simplify the data and to reduce the probability of finding random association between genes. The study considers three different quantization methods and tests these quantization methods on two different methods, Dynamic Bayesian Networks (DBN), and Reveal by using a simulated data. The authors do not show how this proposed method would work on a real dataset.

The first quantization method is the proposed one by the authors and data is quantized into three level, $-1$ for the under expressed, 0 for not differentially expressed and $+1$ for over expressed values compared to the baseline. The estimated threshold is based on a model of experimental error. Quantization procedure follows as below:

$$(x(t) - x_b) > \theta \Longrightarrow x(t) = +1 \tag{3.5}$$

$$(x(t) - x_b) < -\theta \Longrightarrow x(t) = -1 \tag{3.6}$$

$$|x(t) - x_b| \leq \theta \Longrightarrow x(t) = 0) \tag{3.7}$$

where $x(t)$ is the expression value of gene x and $t$ is time and $x_b$ is the basal value of the same gene. The novelty of this quantization is the selection of the $\theta$. $\theta$ is selected according to the significance level $\alpha$ where $\alpha$ is optimized to adjust between false positives (FP) and false negatives (FN). The expected value of FP and true negatives (TN) are estimated as $FP = N_0 \times \alpha$ and $TN = N_0 - FP = N_0 \times (1 - \alpha)$ respectively, where $N_0$ is the number of not differentially expressed genes. Following these, false negative (FN) are calculated as $FN = N - TN - (TP + FP) = N - N_0 \times (1 - \alpha) - S_\alpha$. The compromise between FP and FN is achieved if $FN = FP \Leftrightarrow N - N_0 \times (1 - \alpha) - S_\alpha = N_0 \times \alpha \Leftrightarrow N - N_0 = S_\alpha$. $\alpha$ that guarantees $S_\alpha = N - N_0$ is selected. Then $\theta$ corresponding to the chosen $\alpha$ is used as the threshold. The second quantization method used is to fix the value of $\alpha$ as 0.05. The third one is achieved by simply ranking the values and taking the smallest 1/3 of the data as $-1$, the next 1/3 of the data as 0 and the last 1/3 of the data as $+1$.

The authors compare these three different quantization methods on simulated data for constructing gene regulatory networks by using the methods mentioned above. Area under precision and recall are used as the evaluation metric and the authors conclude that quantizing data with the first technique, which adjusts between FP and FN improves the performance of Reveal and DBN. The study also compares these results with continuous simulated data and testing it on a method called ARACNe (Basso et al., 2005). As described in the paper, ARACNe is used for defining subnetworks in a dataset and is used with continuous data. The authors finds that ARACNe used with continuous data does not do as well as the other methods which use quantized data. And the authors conclude that when there is a limited number of samples using quantized data is better, but as the number of samples increase continuous data will do better. As this study only carried experiments on simulated data, and the techniques used are different for testing the performance of quantized data, it is hard to reach a conclusion about the performance of this algorithm. So the conclusion reached by the authors is only tentative.

Kim et al. (2005) uses quantized gene expression values either binary or three levels to identify patterns that show consistency within cellular context. The context here corresponds to the cancer or normal state. As an example, cancer samples are represented with $S$ and the normal samples with $S^c$ or vice versa. The aim of the study is to define a cellular context and its corresponding genes. Genes within the same context are expected to have a consistency of expression and some randomness outside this context. The authors calculate consistency with entropy. For a particular gene of interest, $g$, entropy is calculated as:

$$H(g) = -\sum_{i=1}^{n} p_i \log p_i \tag{3.8}$$

where $n$ is the number of features and $p$ is the relative frequencies of the discrete expression values (e.g. $-1$, 0 or 1) so that $p_1 + \cdots + p_n = 1$.

If the expression pattern of a gene is consistent within a group, say cancer, the entropy would be low, otherwise high. Their algorithm starts with creating two empty sets, one for cancer and the other for the normal samples. By testing every gene as a starting point, samples are assigned to either of the class by using the consistency measure described above. The study does not give any details about how data is quantized, but they show on a real world dataset, by using their algorithms and comparing their results to published ones, they identify some of the top rank genes. This study is another example of how quantized representation data can be useful and the results can be interpreted as to support the idea of quantized gene expression data.

Chung et al. (2006) introduces two new quantization algorithms for gene expression data. The first model, model base quantization approach (MBQA) is a parametric model and the second one, model free quantization algorithm (MFQA) is a non parametric approach. The introduced quantization methods have the flexibility of quantizing gene expression values into arbitrary number of state changes ($E_s$) rather than into predefined number of state changes which is usually the case of binary or three level. These models are claimed to have more flexible description of the gene expression values but even though the study claim this they test their approach by quantizing the data into two $(1, 0)$ and three states $(1, 0, -1)$ and do not present any further experiments or results with more number of state change examples. The approach taken here is similar to $K$-means clustering; once the number of state change is specified, genes are discretized according to the clusters they are assigned rather than defining a threshold from the parameters of the mixture model as Zhou et al. (2003) do. The first step for both quantization algorithms is to sort the expression values for a particular gene in ascending order. MBAQ uses mixture of $E_s$ Gaussian distribution and selects the model which gives maximum posterior odds. On the other hand MFQA groups expression values into $E_s$ homogeneous groups in a way that the distance between different groups are sufficiently large. The authors find the highest jump between the successive points as the distance. But this process is very sensitive to the noise in the gene expression data and may lead to inaccurate results. These algorithms were tested on two simulated and four real world microarray datasets. Simulated data is modelled as normal distribution and it identifies the state of the change according to the mean with the same variance. Knowing which gene belongs to which state of change they compared MBAQ, MFQA with two other quantization algorithms by using the error rates. The authors conclude that MBQA is the best among the tested quantization algorithms. As tests are carried out with simulated data it is hard to reach these conclusions because microarray data is highly unlikely to have such a perfect normal distribution and when it does not, these methods can fail to show good performances. Considering this fact, the authors also did some experiments on four real data microarray data. However the authors took a different approach to analyse the performance of the real world data. For the real microarray data, before quantization is applied, genes with missing values are either removed or filled. For all data, base 2 logarithm is applied. The two proposed quantization algorithms are tested on the bases of the number of genes assigned to each state. Except for one dataset they found that results are similar, but they did not compare whether the genes match when assigned different change of stages.

By taking another different approach, the authors also compared the performance of quantized data with continuous one. The aim is to find the common most informative genes in both quantized and continuous data. They do this by calculating the correlation coefficient for both continuous and binary data. Genes with higher correlations are regarded as more important candidates and they check how many genes match which have high correlations. For two datasets they find a good ratio match while for the other two the ratio is not that high. The authors explain this by saying when small difference of expression measurement are normalized with variance, they end up with high correlation and they suggest to check the difference of expression values directly to get the most discriminant genes. And they found that the most informative genes which have large differences in expression levels between two sample classes are common in both quantized and continuous data.

The authors conclude on the basis of selecting most informative genes, quantized gene expression data preserves enough information for microarray data and using quantized data has certain advantages such as decreasing computational complexity. Even though it is a very simple and basic study in the sense of comparing the performance of quantized and continuous data they only evaluated the most common discriminative genes by using correlation coefficient. The results show that when data is quantized there is not much information loss. More analysis needed to be done before drawing conclusions. The authors only considered sample classification problems but there are also gene classification problems and also different types of arrays. In our study we show this in a more comprehensive way and also show the other important aspects of using binarized data.

Akutsu and Miyano (2006) propose a gene selection algorithm by using Leukaemia dataset of Golub et al. (1999). The study considers binary representation of the data and the reason why they use binary gene expression data is the same with the rest of the studies in literature, i.e., the noise in the data. Gene expression values are binarized as follow: First the two classes are separated. Then $e_{i,j}$ representing the observed gene expression value for gene $i$ and sample $j$. For each gene $i$, $e_{i,1}$, $e_{i,2}$, $\ldots$, $e_{i,m}$ are sorted in ascending order. Let $e_{i,p}$ denotes the largest of the first class and $e_{i,q}$ denote the smallest expression value of the second class. The threshold $\hat{e}_i$ is simply the mean value of these values $\hat{e}_i = (e_{i,p} + e_{i,q})/2$. Then the new binarized gene expression values $x_{i,j}$ is defined as:

$$x_{i,j} = \begin{cases} 1 & \text{if} \quad e_{i,j} > \hat{e}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

After the data is binarized, the selection problem is applied. The study uses *r-of-k threshold functions* for this purpose. This function is true if at least $r$ variables among $k$ variables are true. So the inference problem select $k$ genes so that correct predictions can be made. However, the study assumes that the number of informative genes is known in advance. This is a big drawback of this study. As in most studies selecting the most informative genes is a data mining procedure and depending on the aim of the study, the required number of genes is selected.

The study concludes that using binarized with the specific dataset information loss is not much. The authors noting this, interested in doing search on more datasets. These results add strength to our work. The study further compares three more algorithms for selecting genes but at this point it is not very clear whether they used binary or continuous data.

Willbrand et al. (2005) works with cell cycle genes aiming to identify patterns of these genes. The authors check for the successive points a particular gene' s expression values if there is an increase (+) or decrease (−). The aim in this study is to find correlated genes which have similar patterns. Randomness is identified by using the correlation between random data and the probability $P(\sigma)$ of up-down signature. Random data have many sign changes and thus have a higher $P(\sigma)$, whereas non-random data has less sign changes and thus less $P(\sigma)$. The authors call the string which only consists of (+) and (−), $\sigma$ and the probability of a gene having up-down signature $P(\sigma)$.

$$P(\sigma) = \frac{C(\sigma)}{(N + 1!)} \tag{3.10}$$

where $C(i) = 1$ and $C(i, j) = \binom{i+j}{i}$ for $n \leq 2$ and $n$ is the number of groups $(i, j, \ldots)$ and $i$ is the number of + and $j$ is the number of −. $N + 1$ is the length of the gene expression vector.

When $n > 2$, recursive relation applies:

$$C(i_1, \ldots i_n) = C(i_1 - 1, \ldots i_n) + \ldots + C(i_1, \ldots i_n - 1) \tag{3.11}$$

with boundary conditions

$$C(\ldots, i, 0, j, \ldots) = C(\ldots, i + j, \ldots) \tag{3.12}$$

and

$$C(0, i, \ldots) = C(i, \ldots) \tag{3.13}$$

This method has certain advantages over the continuous data as the authors claim: (1) $P(\sigma)$ is not dependent on the distribution of the data, (2) $P(\sigma)$ remains unchanged if a transformation is applied to data and (3) since it is a discrete approach calculations are easier to make.

By comparing their methods on two yeast cell cycle data, the authors show that periodically expressed genes can be identified with their method.

The study does not apply quantization to microarray data directly but by checking the successive point if they are increasing or decreasing they are actually ignoring the high numerical precision of these measurements. This point is important in the sense of showing it only matters whether the expression values increase or decrease, similar to ideas such as it only matters whether a gene is expressed or not. This study is further developed in the work of Ahnert et al. (2006) who propose a discretized method for finding patterns in microarray time series data. The method of Ahnert et al. (2006) simply ranks the expression values of a gene and gives them a sequence number instead of using the actual expression values. They applied this to two yeast data and compared the matches. The authors criticize the results of these two studies i.e., both studies are analysing the same data but there are only a limited number of matches between the two for defining the periodically expressed genes. Ahnert et al. (2006) shows that by using their techniques most of the genes can be identified as being common in both studies.

A very simple study, which only analyses simulated data, is conducted by Ruusuvuori et al. (2006). The study quantized the simulated data and compares the performance of quantized data with that of continuous data by using three different classifiers: (1) linear discriminat analysis (LDA), (2) linear SVM and (3) k-nearest neighbour (k-NN). The quantization method used here is the equidistant quantization which is $\Delta_i = (\max_i - \min_i)/(q-1)$ where $i = 1, \ldots, m$ ($m$ being the number of features) and $q$ is the number of quantization levels. By only evaluating simulated data the authors conclude that when used with LDA and SVM there is a slight information loss with binarized data but with k-NN results get worse. The authors left the investigation of the poor performance of k-NN as a future study. The authors also mention that quantized data may give higher error rates when used with hyperplane classifiers. However, as seen from our results which evaluate real world microarray data this prediction with the hyperlanes is not true at all. This study suffers from not using any real data and just reporting assumptions about the results.

Zilliox and Irizarry (2007)' s work uses binary representation of gene expression data to predict tissue types. The study constructed a database of binarized gene expression from a certain type of array, namely Affymetrix HGU133A human array. The study considered more than one hundred tissues to construct this database. By checking which genes are expressed and not expressed in tissues they formed the bar code which is a binary representation of the expression data. For quantizing data, authors used GMM as well. When a new tissue from the certain array is fed in to the code it predicts the tissue type by finding the similarity of the tissues in the database. They applied leave-one-out cross validation with Euclidean distance for finding the similarity. The authors report superior results with this technique. But distance to template classifiers such as bar code has certain drawbacks for microarray studies and they can not always be expected to perform well, especially if the assumption of data coming from a normal distribution is violated. Distance to template classifiers only performs well when the feature space of the data is isotropic and the data is spread evenly along all directions (Duda et al., 2001). It is nearly impossible to expect microarray data to show such a perfect distribution with all equally spread around the mean as microarray data contains a lot of noise. Fig. 3.2 shows a very simple two dimensional example (data generated in MATLAB) where this situation is explained. There are examples which bar code method produces very poor results. See Table 5.3 for an example where bar code method only performs %50 for predicting *lung*. This experiment is carried out by using authors' own `R` code which is made available on `http://rafalab.jhsph.edu/barcode/`. The study does not make any comparison with the state-of-the-art classifier SVM but with a method called predictive analysis of microarray (PAM) (Tibshirani et al., 2002). It has been shown by Dettling (2004) that PAM can not compete with SVM. Inspired by the bar code, we compared our techniques used in this study with distance to template classifiers but found that it does not do any better than SVM (these results are discussed in more detail and can be seen in Chapter 5).

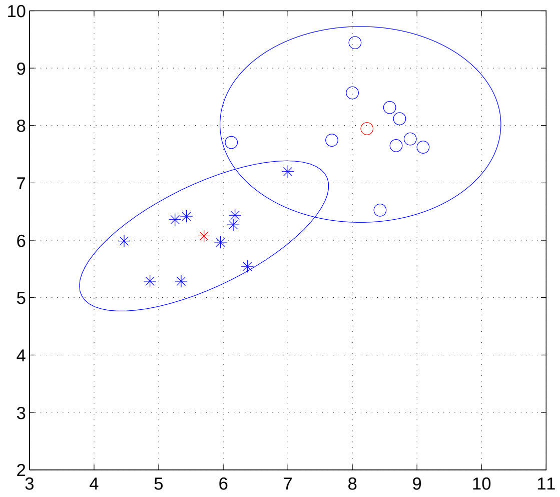


Figure 3.2: A small example of why distance to template classifiers perform poorly. When data points do not have isotropic variances, even though they come from a normal distribution, they may be misclassified. Red marks show the centres of each cluster.

Despite all the drawbacks of this work it is worth mentioning that by this method authors remove the lab effect of measurements but still fail to compete with SVM. The study also does not make any comparison for continuous data. They just tested bar code with very simple datasets and ignored all the other things.

Bar code method of Zilliox and Irizarry (2007) was successfully applied by Yegnasubramanian et al. (2008) for discriminating cancer from the normal cell lines at a prostate cancer study. The study expects that genes in normal prostate cell are not expressed and genes in prostate cancer cell to be expressed and they quantify this with bar code which uses the binary representation of the transcriptome data. The authors suggest that absolute expression pattern (expressed or not) has advantages over relative expression (high numerical precision) and for that reason use bar code approach. Secondly, the study favours bar code over the present-absent call of Affymetrix claiming that perfect match and mismatch may result in wrong calls. The study shows that binary representation can be useful in discriminating cancer and normal cell lines. However, it is only limited to one dataset and one type of inference. As bar code method gives good results there are also situations in which it performs very poor.

Sahoo et al. (2007) use a step function, called StepMiner, for identifying genes which have sudden changes for time course data. The study does not directly apply quantization to data but instead analyses when there is a sudden increase or decrease in the profile of a certain gene. By analysing the sudden changes in the profile, the study is ignoring the high numerical precision of these measurements which is an example to add further weight to our work. They have shown that genes which have similar patterns have relevant Gene Ontology annotations. The study showed this on one simulated and one real world dataset.

Another study aiming to remove the lab effect of microarray studies is conducted by Kim et al. (2008) using discretized gene expression data. The study first ranks the gene expression values and gives them the sequence number. Then those datasets are combined. The authors compare this method with individual and combined datasets by using Out of Bag (OOB) error rate. Rather then focusing on the performance of the discretized data, this study is more interested in finding the effect of combining dataset by checking error rates. The authors conclude that combined datasets are better than individual datasets. The study does not make any conclusion about the performance of quantized data.

However, none of these studies mentions the underlying biology as their reason to support the usage of quantized data. Apart from these none of them makes a comparison of the inferences drawn from binary data with the continuous data. While we show biological reasons to support our idea we also make comparisons with continuous data and show two other advantages of using binary data.

Quantized gene expression data have not only been used for making statistical inferences.

There are many studies using boolean networks for gene regulatory networks. Some examples of boolean network studies include the work of Pal et al. (2005), Shmulevich et al. (2002a), Shmulevich et al. (2002b), Smith et al. (2002), Huang (1999), Kim et al. (2000a), Kim et al. (2000b). Since gene regulatory network problems are not considered in this study, no detailed explanation is given about these studies. Another study which does not make any statistical inference but check the reliability of microarray data by using binary representation of transcriptome data is carried out by Bilke et al. (2003) where the study took a Bayesian approach to assess the reliability of microarray data. The authors used the Affymetrix approach of Presence (P) and Absence (A) call for discretizing the data. The study show that using binarized gene expression data has compatible results when compared to t-test statistics used with continuous data for comparing the reliability of microarray data.

## 3.3 Summary

Quantized gene expression data has been used and been shown to have beneficial effects for making inferences. The common concern of all of these works is to reduce the effect of noise at continuous data which is due to various pre-processing stages of microarray data or image processing. Most of the studies which uses quantized gene expression, except Mircean et al. (2004), do not count the biological variability as a cause of the noise. Biological variability is due to the underlying biological properties of mRNA and it is highly expected to have such noise in the data. By examining the results of the existing works, we can tell that there is no extensive study that analyses the performance of quantized gene expression data. Quantization has only been used as a simple step within the algorithm. Apart from few studies, no comparison is made with continuous and quantized data. The studies which makes a comparison between continuous and quantized data, fail to apply state-of-the-art classifier SVM. Those studies used methods like k-NN classifier which is extremely dependent on noise or results vary depending on the selection of the initial centroid. It is also quite obvious that quantization is not carefully studied as most of the papers do not even cite each other.

Our study differs from all of them in the sense of evaluating a lot of data and using different inference problems such as classification, clustering, detecting periodically expressed genes and developmental time series data. We based our reasons to quantize data on biological reasons and the nature of the microarrays. We also address the problem at the pre-processing stage and even come up with a solution rather than just saying quantization is good to get rid of noise. None of the works mentioned above provide biological reasons why quantized data make sense due to the underlying biology. Quantization seems to be done as a ‘random’ step to analyse the data. The authors who used quantized data does not seem to do a through literature search on the quantization. And even though they use quantize data most of them also mention that when microarray data is quantized there is a loss of information. However, when quantization is handled with care, and with a suitable choice of metric for quantized data, it can be seen that there is no information loss. In fact there is a gain in the performance of inferences drawn from binary transcriptome data. Throughout the rest of this study we show this with experimental results and also other advantages of using quantized data i.e., reducing the algorithmic variability.

To the best of our knowledge there is no thorough study in literature to analysis binary data domain by considering several data types or several inference problems. All the existing papers address the subject quantization as a step to remove the effect of noise. But quantization has other beneficial aspects which are to reduce the effect of algorithmic choice.

# Chapter 4

# Questioning the high numerical precision in transcriptome based microarray studies

## 4.1 Introduction

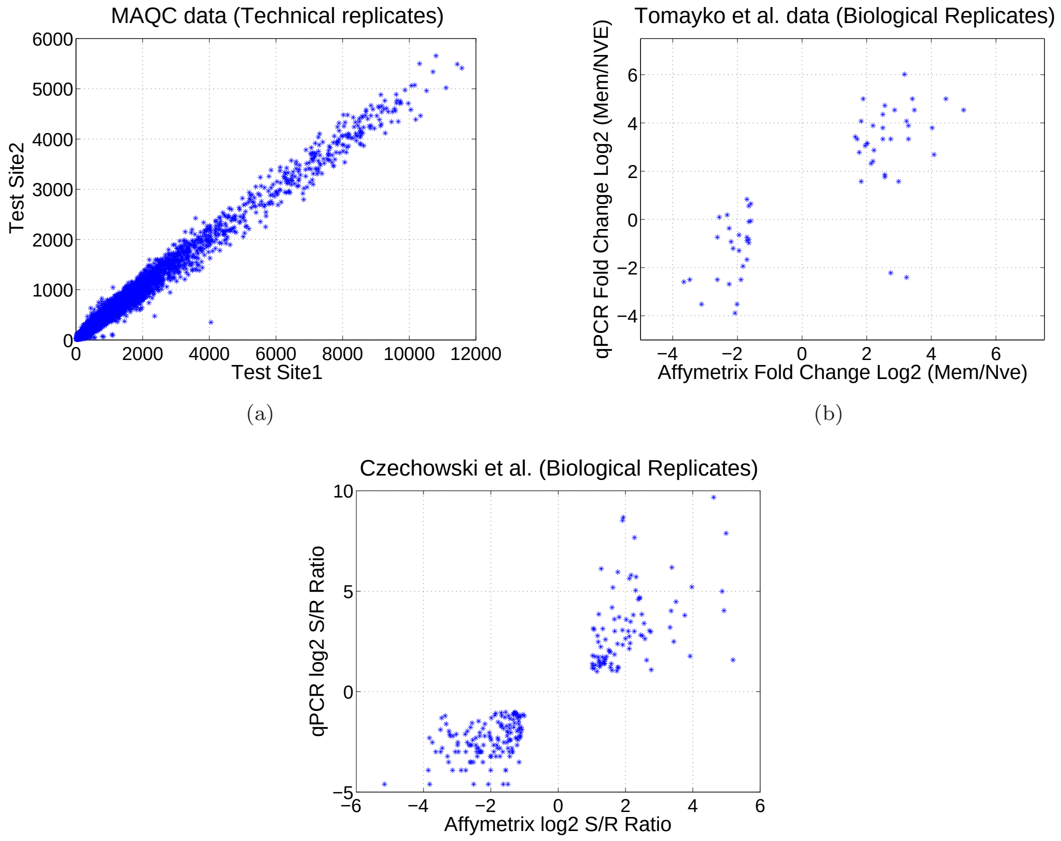
Since the first use of microarray technology by Schena et al. (1995) and Lockhart et al. (1996), analysing gene expression data became quite popular in the machine learning community. After the early successful applications of machine learning techniques to such data by DeRisi et al. (1997), Eisen et al. (1998) and Brown et al. (2000), the technology of microarray itself is developing (e.g. next generation DNA sequencing) and more data is becoming available. As the size of the data increases either new algorithms or the modifications of the existing algorithms are needed. However, biological properties of mRNA as described in Chapter 1 and thus the biological variability in measurements should make us skeptical about the high numerical precision of these measurements. Microarray measurements contain a lot of noise, caused by biological variation and pre-processing stage analysis such as normalization, background correction, etc. or noise just due to measurement errors. The number of different algorithms available for pre-processing raw microarray data and therefore the choice of different combinations have effects on the numerical precision reported and this affects the results of inferences drawn from microarray data as well. When all these are put together it makes gene expression measurements not reproducible. Here we propose a new representation of microarray data, namely binary data, where we are only interested in whether a gene is expressed or not (1 for expressed genes and 0 for not expressed genes) aiming to improve the quality of inferences drawn from microarray data. First, we ask the question whether there would be any information loss if researchers were to use quantized data for making inferences. In this chapter we present experimental results from a wide range of inferences including classification, clustering, analysing periodically expressed genes, time series data, cell cycle genes and differentially expressed genes by using standard algorithms to answer the question above. First we will present the experimental results from biological and technical replicate samples to illustrate the point of questioning high numerical precision in transcriptome based measurements.

### 4.1.1 Reproducibility and numerical precision of microarray measurements

We start with evaluating the reproducibility of microarray measurements by considering biological and technical replicates. Our claim is that microarray measurements are not reproducible if biological replicates are used. We compare these two cases and suggest that it only matters whether a gene is expressed or not and the high numerical precision reported for making inferences is not realistic. To compare these two cases, we used three different datasets. One for technical replicates (MAQC, 2006) and two biological replicates data (Tomayko et al., 2008; Czechowski et al., 2004). Technical replicates are Affymetrix data, mRNA samples from human, and are amplified and tested at different test sites. There are 54675 genes in MAQC (2006) data and the correlation coefficient for technical replicates are 0.99 (see Fig. 4.1(a)). Biological replicates are the comparison of microarray measurements with quantitative PCR (qPCR). Tomayko et al. (2008) has 69 genes from mouse and Czechowski et al. (2004) has 237 genes from Arabidopsis (we only took genes which have at least two fold change and the rest is treated as noise).

Correlations between qPCR and microarray measurements in Fig. 4.1(b) and Fig. 4.1(c) are high (0.82 and 0.6 respectively) when the whole data is taken into account. However, a better model of the data is that there are two modes (high expression and low expression), and when the modes are analysed separately, correlation between the two measurements drops to negligible levels (around 0.40). For Tomayko et al. (2008) correlation for the expressed genes and not expressed genes are 0.36 (after outliers being removed) and 0.47 respectively. For Czechowski et al. (2004) correlation for the expressed genes and not expressed genes are 0.45 and 0.40 respectively. Motivated by these, we suggest that the information in the data relates to high and low expression levels (binary) with the remainder being noise. These can be represented as 1s for expressed genes and 0s for the not expressed genes.

Following the discussion above, we take a computational approach to explore what meaningful level of precisions is for transcriptome measurements. We used Zhou et al. (2003)'s binarization method to obtain different levels of quantization, and ask if researchers would have reached different conclusions, had they worked with data represented at lower precisions. We consider six inference problems, which are: (a1) inferring gene function from gene expressions by posing a classification problem, using both two colour spotted array data and Affymetrix synthetic oligonucleotide data; (a2) classification of



(a)

(b)

(c)

Figure 4.1: Comparison of reproducibility of mRNA measurements. Fig. 4.1(a) shows the comparison of mRNA levels from human when two technical replicates are used with different sites (MAQC, 2006). Fig. 4.1(b) shows the comparison of mRNA levels when two biological replicates are used. First replicate is used with Affymetrix and the second replicate is used with qPCR to detect the fold change (Tomayko et al., 2008). 'Mem/Nve' in the axes of (b) stands for Murine Memory / Naive B cells. Data is from mouse. Fig. 4.1(c) also shows the comparison of qPCR with Affymetrix data for Arabidopsis (Czechowski et al., 2004). 'S/R ratio' on the axes of (c) stands for shoot / root ratio. Genes which have at least two fold change are considered and the rest is treated as noise.

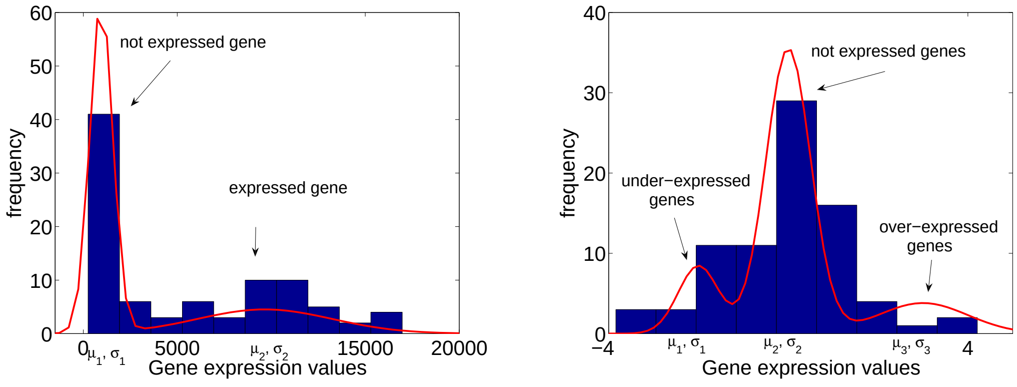
phenotypes (medical conditions) from gene expressions; (b) function inference by cluster analysis; (c) detecting periodically expressed genes in the cell cycle; (d) analyzing developmental time series data, (e) analyzing cell cycle genes with singular value decomposition and (f) analyzing differentially expressed genes. In this chapter we report the observations on a sample of problems to illustrate the critical question we pose.

## 4.2 Quantization of microarray data

Quantization of microarray has been studied in literature. Among possible methods which were also reviewed in Chapter 3, we choose the quantization method of Zhou et al. (2003) where mixture of Gaussians are used for the different states of gene expression values. Our justification for choosing Zhou et al. (2003)'s method is that it is relatively more principled than other approaches to quantization reviewed above. Arbitrary thresholds set by other researchers are not necessarily transferable across different platforms or experiments due to variabilities induced by image processing and normalization, while the method in Zhou et al. (2003) depends on the underlying probability density of the expression levels and hence the idea is portable to any situation. We focused on binary representation of these measurements. If the measurements are the ratio of intensities, logarithm of the values are taken, if the measurements are direct intensities there is no need to take the logarithms as suggested by Zhou et al. (2003). Gene expression values are quantized by fitting a mixture Gaussian model to the expression values:

$$p\left(\mathbf{x}\right) = \sum_{k=1}^{M} \pi_k \, \mathsf{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \, \boldsymbol{\sigma}_k\right) \tag{4.1}$$

where $p\left(\mathbf{x}\right)$ is the probability density of gene expression measurement, $M$, the number of mixture components, and $\mathsf{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is a Gaussian density of mean $\mu$ and standard deviation $\sigma$. We used two and three component mixtures mostly, corresponding to $M = 2$ and $M = 3$ in the above equation. Two and three-center GMM is fitted to both cDNA and Affymetrix in the beginning of this work. Then we adopted two-center GMM for both platforms. Fig. 4.2(a) shows an example of gene expression values fitted to two-center GMM from an Affymetrix array and Fig. 4.2 (b) shows three-center GMM fitted to data from cDNA array.



(a)

(b)

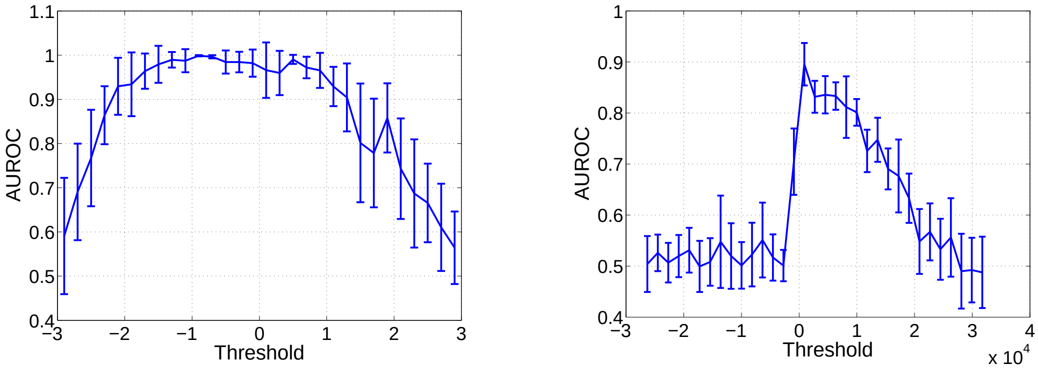
Figure 4.2: Mixture Gaussian distributions and corresponding histograms of gene expression levels for a subset of data taken from a sample of (a) *Affymetrix* gene expression measurements (Causton et al. (2001)) and (b) cDNA gene expression measurements (Brown et al. (2000)).

After learning parameters of the model, threshold $Th$ is chosen as:

$$Th = \frac{\mu_1 + \sigma_1 + \mu_2 - \sigma_2}{2} \tag{4.2}$$

to achieve binary quantization. For three level quantization, we fit a model of three Gaussian components, ordered them by their means and selected two thresholds between adjacent Gaussians using the above formula.

Another issue with quantization is that to decide whether the threshold should be local or global. At the beginning we considered both of these situations: (1) quantizing the data with a global threshold and (2) quantizing the data with local thresholds i.e., gene by gene or array by array. In the early stage of this work quantization is applied with a global threshold chosen by hand over a different range. Figure 4.3 shows the two classification problems, using linear kernel in SVM, for the performance of quantized data with different global thresholds. It is seen from Figure 4.3 that, a right choice of threshold over the whole data will give the same results when threshold is determined with GMM. Even though three-level quantization might be more appropriate for a cDNA array, the aim of this experiment was to show that a right choice of threshold over global data will result in the same performance of inference when compared to a GMM. However, tuning threshold by hand is a time consuming procedure and may be subjective to the analyst doing the experiments. So a more consistent and objective method is required which means the performance of the classifier should be repeatable by whoever or whenever the experiments are carried out. When we compare the performance of binary data obtained either by a global or local threshold, the performances are very similar suggesting that the choice will not affect the overall result. As mentioned earlier in this chapter, using GMM for defining the threshold will make sure that the obtained thresholds are reproducible. It is worth noting that more experiments need to be carried out to examine the effect of difference between two and three-level quantization on cDNA array.



(a)

(b)

FIGURE 4.3: AUROC results when gene expression data is binarized by using a global threshold. Thresholds are selected over a range by hand. Two examples are shown from (a) Brown et al. (2000) and (b) Ramaswamy et al. (2001).

Regardless of the platform, quantization is applied according to the point of interest i.e., if we are interested in classifying genes, quantization is applied gene by gene. If we are interested in classifying samples (arrays), sample by sample quantization is applied throughout this work.

## 4.3 Experiments

To compare the performance of binarized data we conducted some experiments on some published, benchmark datasets, comparing the performance of continuous data with the binary data. To be consistent we also compared the results of continuous data with the published results and saw that our implementations are good enough to reproduce the published results with continuous data. We considered six types of inference problems with 17 different datasets. These inference problems considered include classification (four different datasets), clustering (nine different datasets), detection of periodically expressed genes (one dataset), analysing cell cycle genes with singular value decomposition (one dataset), analysing developmental time series data (one dataset). and analysing differentially expressed genes with correlation coefficient (one dataset)

### 4.3.1 Datasets

We give a short description of the datasets used in this chapter[1]

- Three **Yeast datasets** one from Brown et al. (2000), one from Causton et al. (2001) and one from Eisen et al. (1998). Brown et al. (2000) use 7129 genes with 79 features. 121 ribosomal genes vs. rest are classified. Array type used for this study cDNA array. Causton et al. (2001) used Affymetrix array and studied how yeast genes react to environmental changes such as heat. Eisen et al. (1998) showed in yeast genes that genes with similar functions cluster together. These are the examples of gene function prediction problems.

- **Leukemia data** from Golub et al. (1999). There are 72 samples (47 ALL vs. 25 AML) with 7129 genes. Array type used for this study is Affymetrix. This is an example of phenotype prediction problem.

- **Cancer vs. normal patients** by Ramaswamy et al. (2001). This is a multi-class classification problem where the study uses subtype of cancers vs. normal samples. Here we classified tumor samples (all subtypes of cancer as one class) vs. normal. There are 16064 genes and 256 samples. 190 cancer and 66 normal samples. Array type used for this study is Affymetrix. Another example of phenotype prediction problem.

[1] Detailed information about for the datasets used for $K$-means clustering can be found in the Appendix A.

- **Response of human fibroblast to serum** are studied by Iyer et al. (1999). 517 genes clustered into 10 clusters according to the response to human fibroblast to serum. Array type used for this study is cDNA microarrays.

- **Time series data for Drosophila melanogaster** by Hooper et al. (2007). Drosophila embryo with 14064 genes were examined during the first 24 hours of development.

- **Periodic gene expression data** from de Lichtenberg et al. (2005) for detecting periodically expressed genes. There are 5000 genes and four phases of periodicity. The second periodic gene expression data is from Spellman et al. (1998) for analysis with singular value decomposition. There are 784 genes with five phases and 14 arrays (elutriation-synchronized cell cycle)

- **Colon cancer dataset**. Alon et al. (1999) classifies the patients with colon cancer vs. normal. There are 42 cancer and 20 normal samples with 2000 genes. Array type used for this study is cDNA.

- **Differentially expressed genes** from Tirosh et al. (2008). Mating expression values from three different species are studied. There are 2198 genes with 10 arrays (3, 3 and 4 replicates for each species).

### 4.3.2 Classification

Classification is one of the most widely used supervised inference technique applied to microarray data. Generally there are two types of classification problems applied to transcriptome data, (1) function prediction of genes where labels correspond to genes in a particular function group and (2) phenotype prediction where class labels correspond to different outcomes in a clinical settings (e.g. cancer vs. normal) (Causton et al., 2004). Phenotype prediction has been widely applied in studies such as Golub et al. (1999) and Alon et al. (1999) and gene function prediction has also been widely applied, such as the work from Brown et al. (2000).

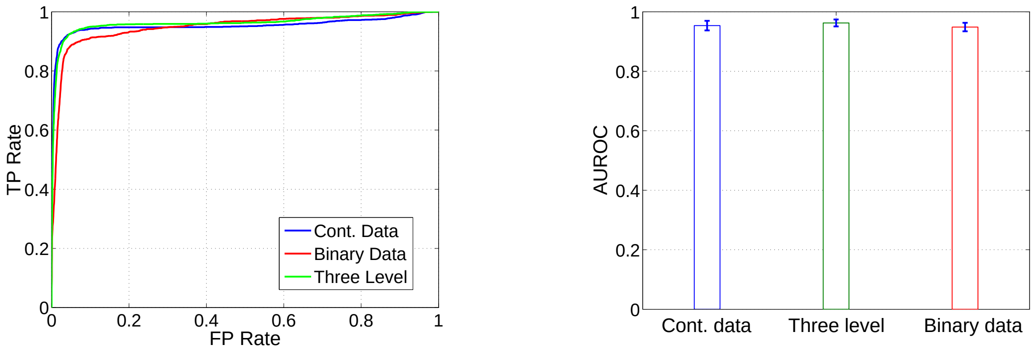
The state-of-the-art classification method for gene expression data is SVM (Brown et al., 2000; Guyon et al., 2002; Cristianini and Shawe-Taylor, 2000; Statnikov et al., 2008). The first use of SVM to microarray data is carried out by Brown et al. (2000). The study classifies genes according to their function. They compare the performance of SVM to four other classifiers. Their results show that classification with SVM gives the minimum error rate.

We used SVM as our classifier and implemented it by using `SVMLight` package (Joachims, 1999). We applied four classification problems two of which are gene function prediction (Brown et al., 2000; Causton et al., 2001) and two of which are cancer vs. normal classification (Golub et al., 1999; Ramaswamy et al., 2001). For all datasets, parameters

$C$ and $\sigma$ are calculated by cross-validation. We trained the parameters on the first half and test on the second half. Data are randomly divided into training and testing for 25 times. The performance of SVM is measured by using Area Under Receiver Operating Characteristic curves (AUROC). We compared the performance of classification by using binary data and then compared these results with the performance of classifying continuous data, confirming that these results are identical or very similar to what was claimed in each of the original studies. Missing values in the datasets were simply replaced by zeros. Further analysis of classification, such as breast cancer, is also studied and the results can be seen in Chapter 5.

#### 4.3.2.1 Results

In Table 4.1, we present the AUROC results of the classification problems and show that even when gene expression data is represented as binary (whether a gene is expressed or not), discriminability between these classes is retained and the loss of information is not much. Except in one dataset (Golub et al. (1999)), inference made with binary data is the same as the ones made with the continuous data. A similar conclusion can be reached by looking at the Figure 4.4 where we show AUROC performance of Causton et al. (2001)' s dataset. The caption 'Three level' in Fig. 4.4(b) corresponds to when data is partitioned into three levels (i.e., $-1, 0, 1$). These results show that with binary data information loss is not much when used with regular algorithms. In the next chapter we will overcome this problem and show that with a proper selection of kernel for SVM, the inferences with binary data can be improved.



(a)

(b)

FIGURE 4.4: Classification with continuous and quantized expression levels for the problem of discriminating ribosomal yeast genes considered by Causton et al. (2001). Receiver operating characteristic curves and area under the curves, averaged over 25 bootstrap partitions of the data, are shown in (a) and (b) respectively. Error bars over these partitions are also shown in (b).

Table 4.1: Loss of discriminability in a sample of classification problems when expression data is quantized to three and two levels. Averages and standard deviations across 25 random bootstrap partitions of area under the receiver operating characteristics curve are shown for a sample of problems.

| Dataset | Cont. Data | 3 level of Quantization | Binary |
|---|---|---|---|
| Golub et al. (1999) | $0.92 \pm 0.05$ | $0.89 \pm 0.06$ | $0.89 \pm 0.07$ |
| Ramaswamy et al. (2001) | $0.90 \pm 0.03$ | $0.89 \pm 0.03$ | $0.90 \pm 0.04$ |
| Brown et al. (2000) | $0.99 \pm 0.004$ | $0.99 \pm 0.001$ | $0.99 \pm 0.001$ |
| Causton et al. (2001) | $0.95 \pm 0.02$ | $0.96 \pm 0.01$ | $0.95 \pm 0.01$ |

### 4.3.3 Clustering

Clustering is a widely used unsupervised inference tool in transcriptome analysis. One of the early works that cluster genes according to their functions was carried out by Eisen et al. (1998). To study how the results of cluster analysis are affected if we worked with quantized data, we used three different approaches and analysed nine published datasets. The different approaches used in this chapter can be summarised as follow:

1. Calculate the pairwise correlation coefficient of genes within a cluster and compare it with randomly selected genes.
2. Calculate the pairwise correlation coefficient of genes for within the clusters and between the clusters, and evaluate the results with Fisher ratio.
3. Apply $K$-means clustering (results presented in Appendix A).

#### 4.3.3.1 Results

Our aim is to show how the ability to detect clusters in the expression of these genes degrades with quantization of the data.

**First evaluation**: To compare the quality of the clusters we computed the average pairwise correlation between gene expression profiles for genes (a) taken from within an identified cluster, (b) random pairs of genes, and (c) pairs of genes taken from across different clusters. Figure 4.5 for the data taken from Iyer et al. (1999) shows an example where we show that even at binary level genes forming a cluster can be separated from the randomly selected genes. Here the important point not to be missed is even though correlation coefficient degrades with quantization the difference between the clusters and randomly selected genes can still be kept. Therefore binarizing data would not cause much information loss.

**Second evaluation**: Calculate correlation coefficient for within clusters and between clusters. We used data from Eisen et al. (1998) (Figure 4.6). As can be seen from

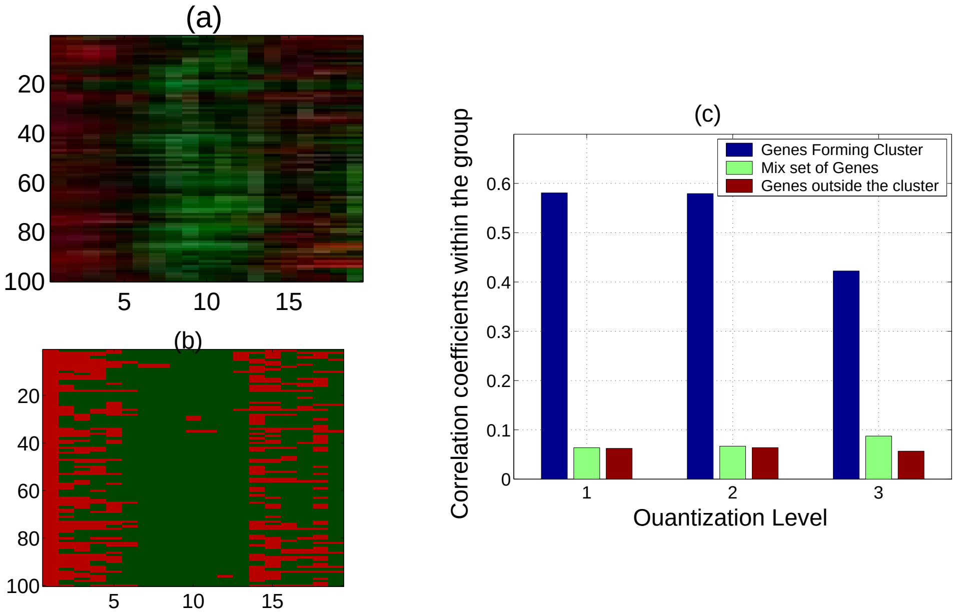
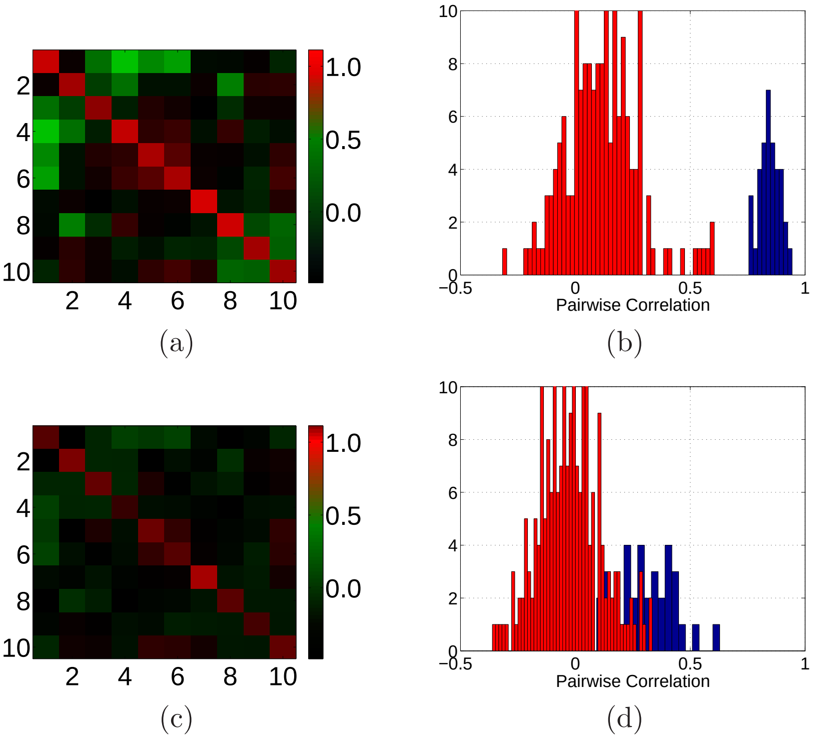


Figure 4.5: Average within and cross group correlations for a cluster of genes taken from Iyer et al. (1999)'s study of human fibroblast response to serum. (a) and (b) are the expression levels of an identified cluster of 100 genes, with continuous and binary-quantized data. (c) shows correlations, illustrating that the average within group correlations stay much higher than cross group correlations even under extreme quantizations.

Figure 4.6 within cluster correlation is higher in all ten clusters. To quantify this judgement we took the distribution of pairwise correlation coefficient within cluster B and the distribution of pairwise correlation coefficient between cluster B and cluster C and calculated the Fisher Ratio as described in Golub et al. (1999) to evaluate the level of discrimination between the two. In the continuous data Fisher Ratio is 3.81 and with the quantized data the Fisher Ratio is 1.25. This loss is not much if we consider the relation between the Fisher Ratio and the area under ROC curve. AUROC decrease from 1 to 0.96 when binary data is used instead of continuous values. Figure 4.7 compares Fisher scores and the corresponding AUROC. It is generated by randomly generating several Gaussian densities and measuring AUROC and Fisher scores. This analysis also shows that we are still keeping the required information with binary data.

**Third evaluation**: Apply $K$-means clustering to the continuous and binary data. Overlap between genes in a particular cluster when clustering is applied at different levels of numerical precision are compared with $F1$ measure. In order to compare the discriminability within and between-clusters we used Fisher ratio. These results are presented in detail in Appendix A. With $K$-means clustering results, there are not always perfect matches in the whole datasets. However, this may well be because of the drawbacks of the $K$-means clustering itself which are mentioned in the beginning of the thesis in section 2.3.3. As Quackenbush (2001); Torrente et al. (2005) mention the results of $K$-means clustering can change with the distance metric used. And which of these results



(a) (b) (c) (d)

FIGURE 4.6: Average pairwise correlations, within and cross-group, of ten clusters taken from Eisen et al. (1998), shown as intensity plots. (a) and (c) are 10 × 10 average correlation matrices computed using the continuous expression levels and binary-quantized expression levels respectively. (b) shows within group and cross group correlations of genes in clusters identified by labels $B$ and $C$ in Eisen et al. (1998) as histograms. (d) shows the same histograms when the data is quantized to binary precision.

are biologically meaningful needs verification by a biology expert. But still, in our results there are matches that are worth mentioning here. Our use of binary data is still reserve the information required for making inferences. In the next chapter we will use spectral clustering to remove those unwanted effects of $K$-means clustering. Spectral clustering uses the eigenvalue decomposition of the similarity matrix. So the results obtained with this method will mainly depend on the similarity metric used and will not be affected by other conditions.

### 4.3.4 Detecting periodically expressed genes

In this subsection we present results from analysing periodically expressed genes from de Lichtenberg et al. (2005). Spellman et al. (1998) was the first work to detect periodically expressed genes using microarray data. Spellman et al. (1998) used Fourier transform in order to detect those genes. Here we analyse periodically expressed genes by using correlation coefficient. Correlation coefficient can be used for analysing periodic genes as mentioned by Cooper and Shedden (2003). First we took a subset of genes

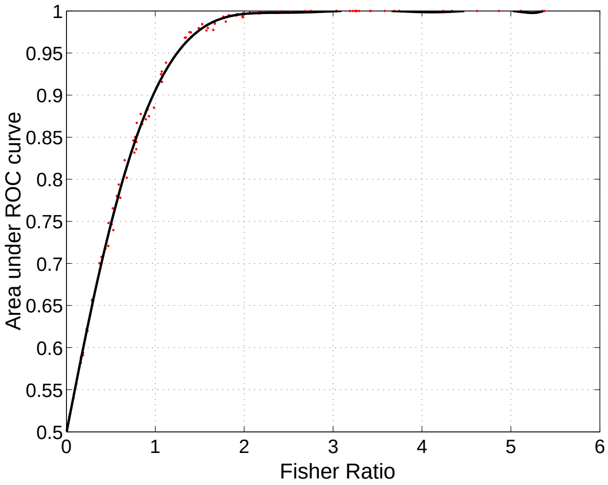


FIGURE 4.7: Comparing Fisher ratios with area under receiver operating characteristics curves (AUROC). Our interest is how much discrimination is lost when the Fisher ratio between clusters reduces (from 3.81 to 1.25, in the example considered) as a result of quantization. We randomly generated several pairs of one dimensional Gaussian densities and measured the two figures of merit for their separation. The points on the scatter diagram correspond to pairs of Gaussians and the continuous line is an interpolation through them, obtained by curve fitting. Note that at a Fisher ratio of 1.25, AUROC has only reduced to 0.95, demonstrating that significant discriminability is retained between the clusters.

identified as cell cycle regulated, with peak expression in the $S$ phase of the cycle. Our objective is to show how the ability to detect periodicity in the expression of these genes degrades with quantization of the data. We adopted a computational strategy similar to that used in clustering above, and measured the average pairwise correlation amongst three groups of genes: (a) the 99 genes which are known to be periodically expressed, yielding an average correlation measure, averaged across $(99 \times 98)/2 = 4851$ pairwise correlations; (b) similar average correlation across an arbitrary group of 100 genes taken from the dataset, but not overlapping with those in group (a); and (c) average correlation between the above 99 genes and $99 \times 100$ genes whereby we picked 100 genes at random to correlate against the 99 above. For correct detection of periodically expressed genes we would expect group (a) to show higher average correlation than those of groups (b) and (c). Our interest is if the average correlation amongst group (a) genes continues to be higher than the other two groups under increasing levels of quantization.

#### 4.3.4.1 Results

Figure 4.8 shows the comparison of discriminating periodically expressed genes from the others when continuous or binary data is used.

We find (Fig. 4.8) that the correlation difference we measured, *i.e.* differences within class correlation of the periodically expressed genes from those for the other two groups

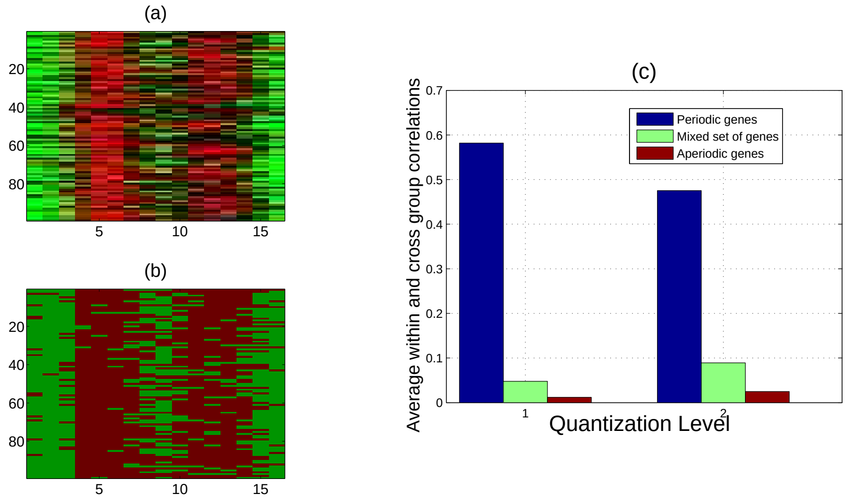


FIGURE 4.8: Expression profiles of a subset of periodically expressed genes, (a), and binary expression profiles after coarse quantization, (b). (c) shows the within class average pairwise correlation for three groups of genes considered (see text), showing that the discriminability of the set of periodic genes from the remainder is robust enough to be maintained at low precisions of the expression levels. Quantization levels one and two refer to the use of continuous data and binary levels $+1$, and $-1$.

(mixture of periodic and aperiodic genes and the random set of genes) do not change. Thus even under such coarse quantization, we would have picked out these genes as expressed in regulation with the cell cycle. Here our demonstration of the effect of quantization on periodicity determination is based on correlation, within and across, groups of genes identified as periodic.

In the following subsection we analysed periodically expressed genes from Spellman et al. (1998) with SVD, following a similar strategy to Alter et al. (2000).

### 4.3.5 Analysing periodical genes with SVD

An alternative approach for analysing gene expression data is the use of singular value decomposition (SVD) of the expression matrix, described in Section 2.3.6. Alter et al. (2000) used SVD to study cell cycle regulation and show that a principal component projection of the data matrix on two dimensions helps visualize periodically expressed genes grouping together in a two dimensional plot according to their phase of expression where phase corresponds to the time of peak expression. We asked the question if quantized expression data, of cell cycle regulated genes, will also exhibit this property. Where they differ, we worked to quantify how much the difference was. To do this we introduce the derivation of an ROC curve with two sliding thresholds on the two dimensional projected space in the next section. Five phases of cell cycle as defined in Spellman et al. (1998) are used here. These phases are S, G1, M/G1, G2/M and S/G2 and the corresponding stages are:

- S phase: The stage of the cell cycle when DNA synthesis occurs.
- G1 phase: The first gap period of cell cycle.
- G2 phase: The second gap period of cell cycle.
- M phase: The stage of the cell cycle when mitosis or meiosis occurs.

First we will describe how we used SVD analysis to project cell cycle genes on reduced two dimensions, namely first principal and second principal components on $x$ and $y$-axis respectively. The cell cycle gene expression matrix is decomposed as described in section 2.3.6.

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{4.3}$$

In order to project the gene expression data onto first and second eigengenes (or first and second principal components) the original matrix is multiplied with the corresponding eigengenes:

$$\mathbf{X}' = \mathbf{X} \times \mathbf{v} \tag{4.4}$$

where $\mathbf{v}$ is the first two eigengenes of the $\mathbf{V}^T$.

#### 4.3.5.1 Two-thresholds ROC

Fig 4.9 shows genes expressed in the G1 phase of the cell cycle (green stars) against the rest of the genes (red stars) on a two-dimensional principal subspace. We see that there is some grouping of the G1 phase genes and some overlap with the remainder. To quantify this overlap we need to slide a threshold such that those on the slide of a threshold are treated as positives (i.e., G1 phase) and those on the others are treated as negatives (i.e., non G1-phase). With the subspace of interactions being circular, due to cyclic nature of the continuous data, we need two thresholds to do this classification.

Two-thresholds ROC can be described as when data is projected onto the subspace by using first and second eigenvectors, two thresholds, $T_1$ and $T_2$ which are moving around the circle is defined. $T_1$ is kept stable and $T_2$ moves around the circle with a 1 degree at a time. As $T_2$ moves, true positives (TP) and false positives (FP) falling between the two thresholds are calculated (see Fig. 4.9 as an example). When $T_2$ completes the circle an ROC curve is obtained. However, there is no certainty about where to start $T_1$. For that reason after the full circle of $T_2$, $T_1$ starts to move around and the same procedure for $T_2$ applies. By fixing $T_1$ we seek to get the maximum AUROC for a particular phase. For each phase, one vs. all classification is applied. This new approach with two-thresholds ROC will be very useful in evaluating the cell cycle gene classification problems in later

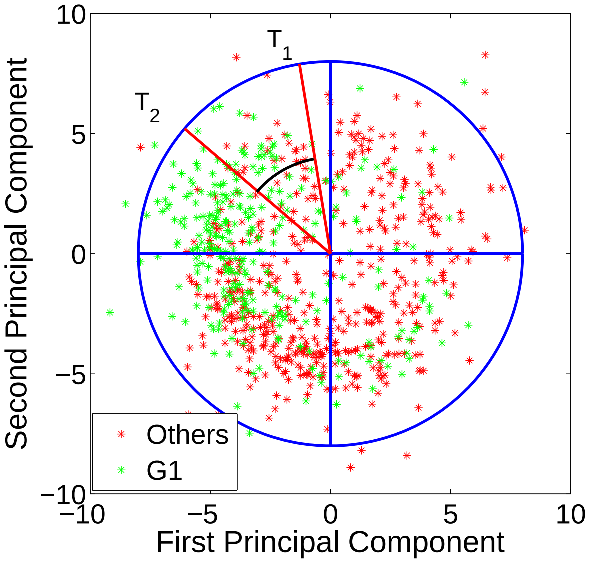


Figure 4.9: Two-threshold ROC. The two thresholds, T1 and T2 are shown with red lines. Genes falling between the two thresholds are used to calculate the TP and FP.

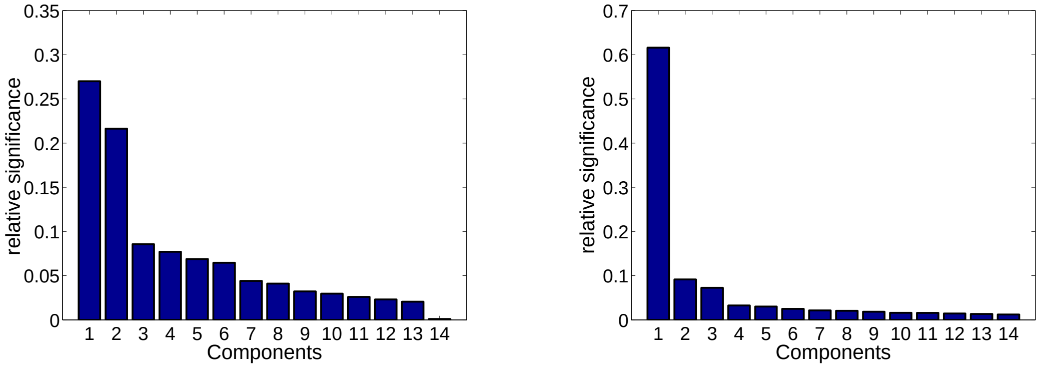
applications. By using two-thresholds ROC, we compare the discrimination of genes with different phases by using continuous and binary data by means of AUROC. The result of this analysis is presented in the following subsection.

#### 4.3.5.2 Results

We calculated the relative variance or the significance of the components for continuous and binary data (Fig. 4.10). By using the first and second eigengenes to represent the data, we are using 49% of the data with continuous (Fig. 4.10(a)) and 71% with the binary data (Fig. 4.10(b)). However, with binary data most of the information comes from the first eigenvector (around 62%). The remaining components are considered as noise.

AUROC results obtained with two-threshold ROC for different phases are presented in Table 4.2. Fig. 4.11 and 4.12 show how genes with different phases are distributed when projected onto first and second principal components and the ROC curve with maximum AUROC for continuous and binary data respectively. The red lines on the projected data shows the fixed $T_1$ to obtain the maximum AUROC.

When the results are compared between continuous and binary data, the performance of binary data is a little worse than the performance of continuous data, but still binary data is doing a good job. Here it should be kept in mind that SVD is for continuous data. We instinctively believe that this problem of the performance obtained by binary data can be improved if we used discrete PCA analysis by following the work of Buntine and Jakulin (2004) as does Tanimoto coefficient for binary classification (see Chapter



(a)

(b)

Figure 4.10: Relative importance of eigengenes (components). (a) is obtained with continuous and (b) with binary data.

Table 4.2: Comparing the performance of SVD analysis for continuous and binary data by means of two threshold-AUROC.

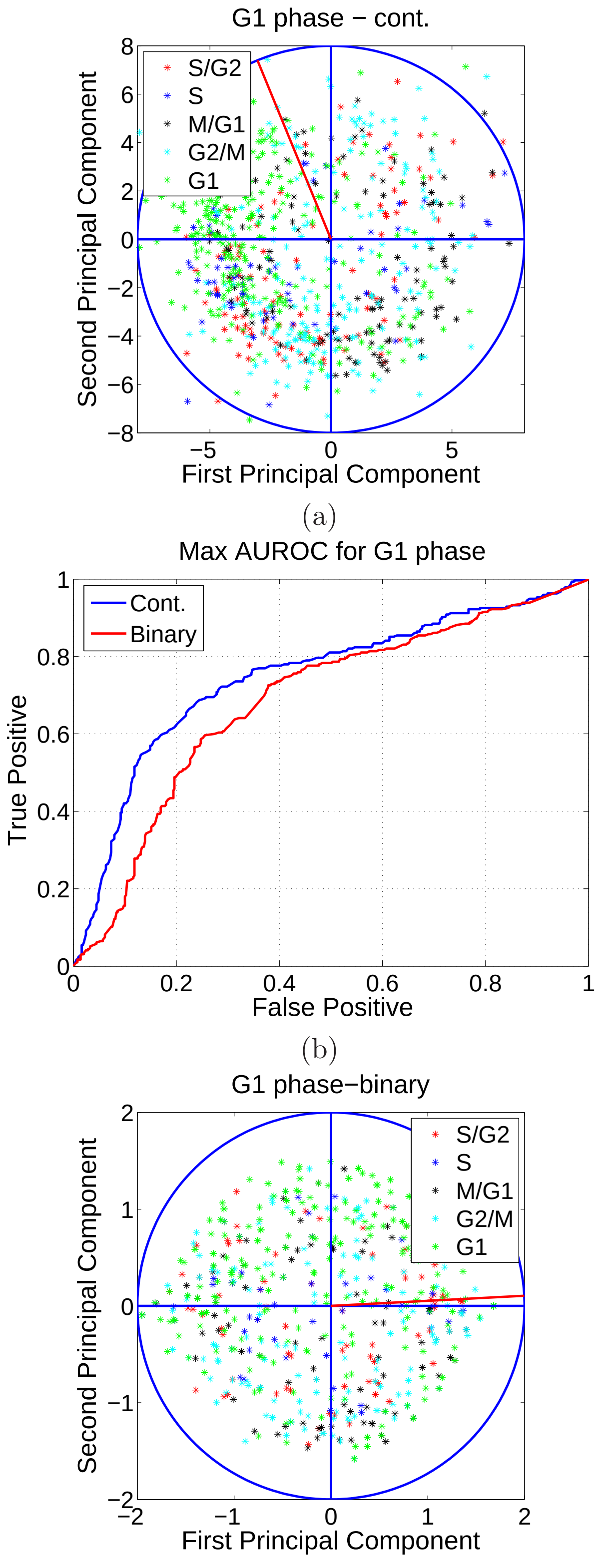
| Dataset | Phase | AUROC | |
|---|---|---|---|
| | | Cont. data | Binary data |
| Spellman et al. (1998) | S/G2 | 0.61 | 0.58 |
| | S | 0.68 | 0.61 |
| | M/G1 | 0.70 | 0.64 |
| | G2/M | 0.70 | 0.64 |
| | G1 | 0.75 | 0.69 |

5). Due to the time restriction of this work applying discrete PCA to this work remains as a future plan.

### 4.3.6 Developmental time series

Here we analysed time series data to detect significant changes in gene expression. Hooper et al. (2007) analysed the significance changes in the gene expression during the embryonic development of the fruit fly *Drosophila melanogaster*. Measurements are taken at every 1 hour (for the first 6.5 hours measurements are taken at overlapping 1 hour) for 24 hours which is the period for fertilized eggs to develop into a larva. In order to detect significant genes, local convolution with two steps function is used in the original study:

- [+1 +1 +1 +1 -1 -1 -1 -1], to detect down-regulation; and
- [-1 -1 -1 -1 +1 +1 +1 +1], to detect up-regulation.

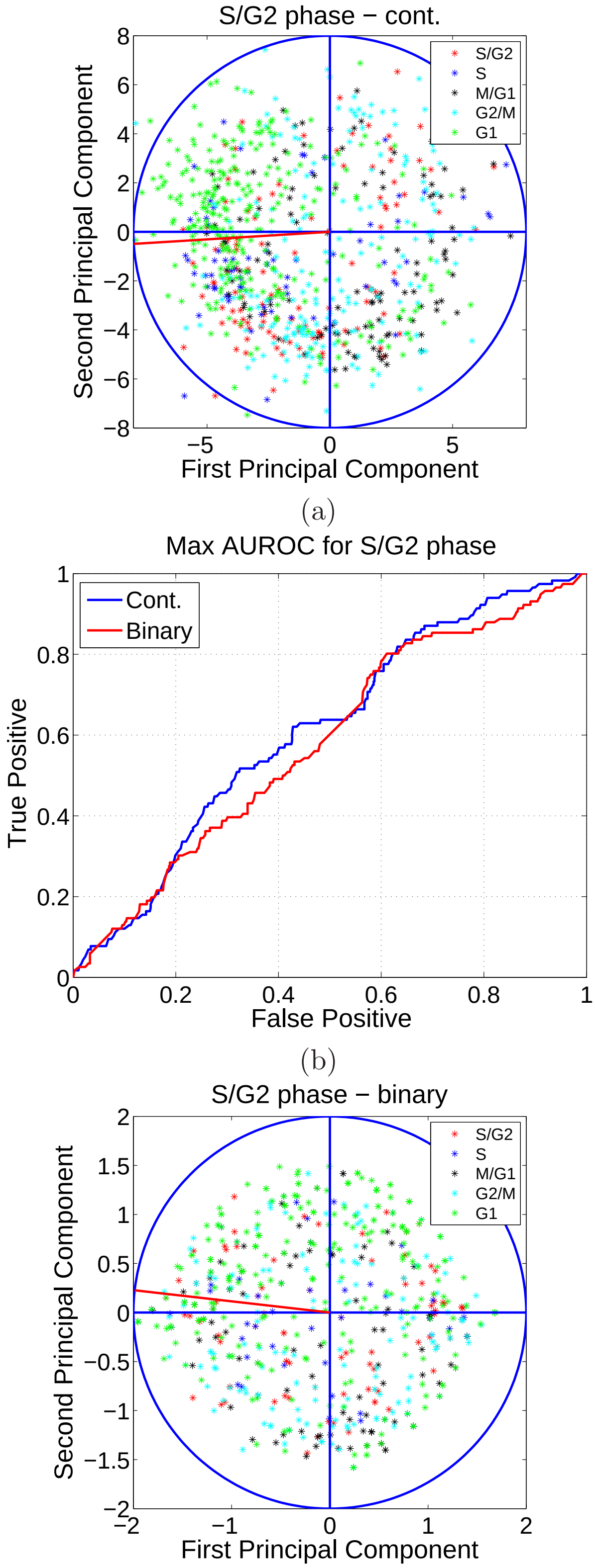


(a)

(b)

(c)

FIGURE 4.11: SVD results for G1 phase genes for continuous and binary data ((a) and (c)). Red line shows the point where $T_1$ is fixed to obtain the max. AUROC. (b) shows the max. AUROCs obtained with with continuous and binary data.



(a)

(b)

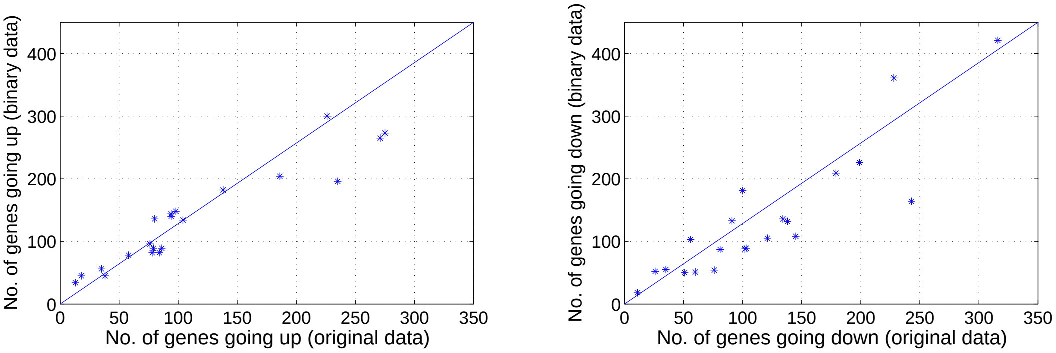
(c)

Figure 4.12: SVD results for S/G2 phase genes for continuous and binary data ((a) and (c)). Red line shows the point where $T_1$ is fixed to obtain the max. AUROC. (b) shows the max. AUROCs obtained with with continuous and binary data.

To be more specific, four points of successive low expression and four points of high successive expression or vice versa is required. This approach requires a sharp increase or decrease in expression level and at the same time requires the the change in transcript level to be consistent over a period of time. Therefore this approach reduces the effect of outliers. We count the number of genes which follow the patterns above for continuous and binary data. Results are presented in Figure 4.13.

#### 4.3.6.1 Results

The number of genes that undergo significant changes in expression give a picture of major regulatory changes during the stages of development. We re-analysed this data at the original precision and after discretizing it to binary precision. Fig. 4.13 shows this comparison, demonstrating that the number of genes detected as significantly up-regulated (or down-regulated) along the developmental time-course of interest is very much the same at the lowest possible precision.



(a)

(b)

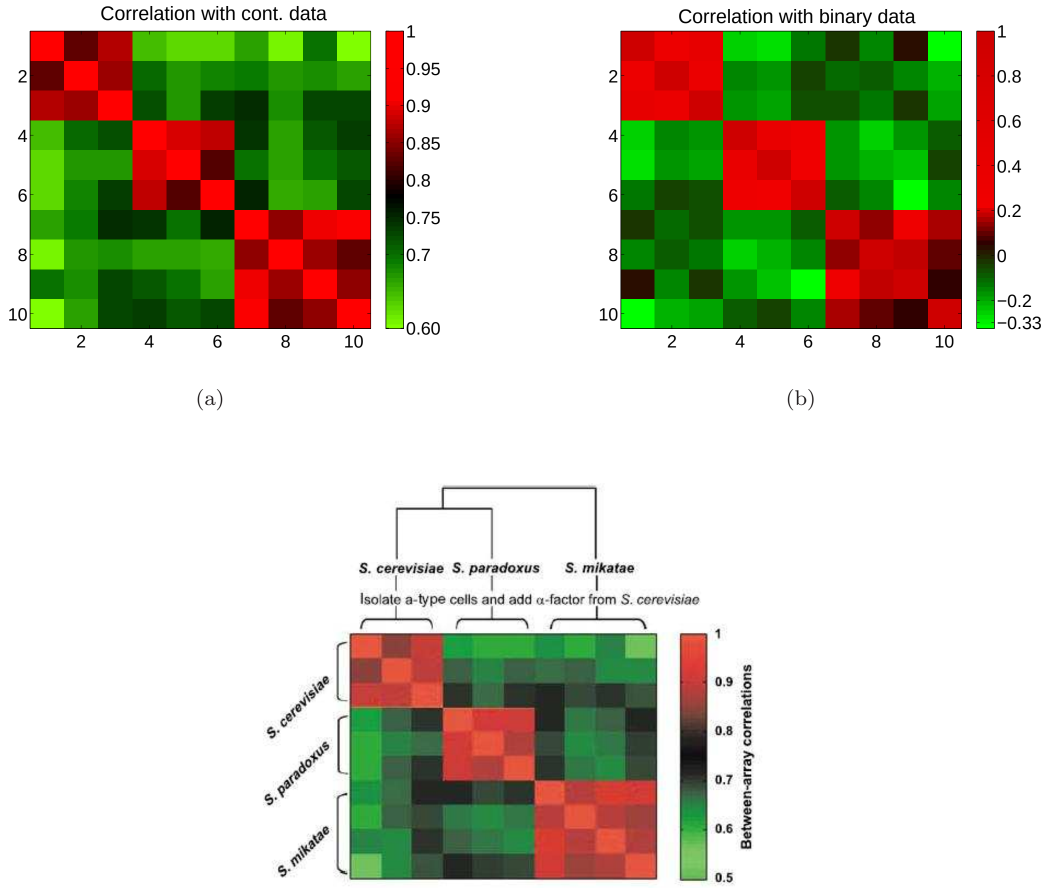
FIGURE 4.13: Comparison of the numbers of significantly up-regulated, (a), and significantly down-regulated, (b), genes at different stages of development, using continuous gene expression measurements and binary quantized expression levels.

### 4.3.7 Differentially expressed genes

In this subsection we used correlation coefficient to discriminate the differentially expressed genes under certain conditions. In a recent study by Tirosh et al. (2008), the effects of mating on three different species (*s. cerevisiae*, *s. paraxous* and *s. mikatae*) are compared. By taking 3, 3 and 4 replicates for each species respectively, they show by using correlation coefficient that these three species can be discriminated with their response to mating. The mean of the correlation coefficients within and across the species are reported to be 0.90 and $0.60 - 0.70$ respectively. This difference indicates that these species can be discriminated with their response to mating.

Here we downloaded the full data from Gene Expression Omnibus with the accession no. `GSE7525` and removed the genes which has missing values (2198 genes left out of 6143) and re-calculated the correlation coefficient for within and across the species with continuous and quantized data. Fig. 4.14 shows the intensity plot of the correlation coefficients for both continuous and binary data.

With our own calculations and by using the same dataset, the mean of the correlation coefficient for within and across species for continuous data are 0.90 and 0.692 respectively (Fig 4.14(a)).



(a)

(b)

(c)

FIGURE 4.14: The intensity plot of the correlation coefficients for the species *s. cerevisiae*, *s. paraxous* and *s. mikatae*. First three columns and rows are *s. cerevisiae*, the following three columns and rows are *s. paraxous* and the last four columns and rows are *s. mikatae*. While (a) and (b) show the results obtained with continuous and binary data respectively, (c) is showing the original figure from Tirosh et al. (2008)

The exact same procedure above is applied to quantized data. Data is binarized by array by array using Gaussian Mixture Model (GMM) and the correlation coefficients are calculated. The mean of the correlation for the within and across species with binary data is 0.46 and −0.15 respectively (Fig 4.14(b)).

By using quantized data we can still show that the different species can be discriminated according to their response to mating. Even there is a drop in the mean of the correlation coefficients, the difference for between and across species can still be clearly observed. Our conclusion is that if the original authors were worked with the binarized data, the conclusion they would reached would be the same.

## 4.4 Summary

In this chapter we have shown that if we were to use quantized data to make inferences from microarray data, information loss is not much. We showed experimental results from several inference problems including classification, clustering, periodicity detection, analysing time series data, cell cycle genes with SVD and differentially expressed genes. Comparisons are made with continuous and quantized data. We also compared our performance of inferences with the published results (data not shown). We used several different datasets to illustrate the point. As our results showed, binary data (whether a gene is expressed or not) can still reserve the needed information if correctly quantized. We conclude that the seemingly high numerical precision measurements reported should be regarded as the effects of biological variability and artifacts of measurement systems, such as image processing or normalizations applied to microarray data. Quantizing microarray data removes those unwanted effects. Biological variability still remains as one of the main problems in microarray data with no solution. Using binary data reduces this unwanted effect and when used wisely, as shown in the next chapter even improve the performance of inferences drawn from transcriptome data. The inference problems described here can be expanded and measurements from different array types needs to be considered. This is a part of the future plan of this work.

A very early work by Dougherty et al. (1995) is worth mentioning here. The study compares the performance of two classifiers with continuous and discrete data by randomly selecting 16 datasets from UC Irvine machine learning repository (Asuncion and Newman, 2007). The result of the study claims that quantized data can improve the performance of specific classifiers compared to continuous data. The study is restricted to classification and tests only two algorithms. However, this can not be universally true to always expect binary or quantized data to perform better than continuous data. In our study we support our idea of using binary transcriptome data with the underlying biology which is mentioned in section 1.2. We are suggesting the use of binary transcriptome data due to the biological properties of mRNA and suggest that the reason for the success of binary transcriptome data is the biological reasons lying underneath. For this, it is worth mentioning that this work should not be considered as questioning the accuracy of the microarray technology or measurements, but our claim is that the measurements themselves may be precise and the inferences drawn from them do not change at lower precisions, even at binary level.

Considering that binary data with regular algorithms do not lose much information, adopting algorithms designed for binary data should improve the overall performances of the inference problem. In the next chapter we show examples of how the performance of inference problems can be improved by using a similarity metric designed for binary data and show examples in classification and clustering. We adopted Tanimoto coefficient, widely used in chemoinformatics field for detecting similar chemicals, where all data are represented as binary strings.

# Chapter 5

# Improving the performance of inferences with a signal sensitive metric

## 5.1 Introduction

In the previous chapter we showed, with a range of inference problems that if gene expression values were represented as binary data, i.e., whether a gene is expressed or not, and make inferences from that, information loss would not be much. It should be kept in mind that those algorithms were not especially designed for binary data. In this Chapter we make inferences from binary gene expression values by using algorithms and metrics especially designed for binary data, i.e., Tanimoto coefficient (Tanimoto, 1958). Tanimoto coefficient is the most widely used similarity metric in the field of chemoinformatics where all data is represented as binary strings (fingerprints) and similarity metrics are used for retrieving similar chemicals with certain functional properties. Following the experimental results of Willett et al. (1998) which states that Tanimoto performs the best for long vectors in binary data for detecting similar chemicals, we then applied Tanimoto coefficients to microarray data.

In this chapter we present experimental results obtained from binary gene expression values by using algorithms designed for binary data. Our results show that using Tanimoto similarity metric for binary data improves the performances of inferences made with binary gene expressions. We show this in kernel framework (Swamidass et al., 2005; Trotter, 2006) in SVM and spectral clustering. We further show that the success of Tanimoto similarity can be explained with the unnoticed systematic variability in probe level measurements. Evaluation metric used is accuracy in this chapter. The reason for doing this is to make comparison easier with the published results in literature.

## 5.2 Tanimoto Coefficient

Tanimoto coefficient is a similarity metric used for binary vectors to detect the similarity between them. It ranges from 0 (completely different vectors) to 1 (exactly same vectors) (Willett, 2006) and is the rate of the number of common bits on to the total number of bits on (1s) two vectors. It focuses on the number of common bits that are on.

Tanimoto coefficient, $T$, is defined as follow:

$$T = \frac{c}{a+b-c} \tag{5.1}$$

where
$a$: the number of expressed points for gene x,
$b$: the number of expressed points in gene y and
$c$: the number of common expressed points in two genes, x and y.

Similarity matrix obtained with Tanimoto coefficient is symmetric and positive and semi-definite (the eigenvalues of the matrix are $\geq 0$). The denominator of Tanimoto coefficient can be considered as a normalization factor which helps to reduce the bias of the vector size (i.e., with larger vectors Tanimoto coefficients work better (Willett et al., 1998; Holliday et al., 2003)). In chemoinformatics data is represented as binary fingerprints, and the research in this field focuses on finding similar chemicals for drug discovery or similar other purposes. As all data are long binary vectors Tanimoto coefficient is the preferred similarity measure in chemoinformatics. Willett et al. (1998) showed experimentally by comparing twenty different metrics Tanimoto gives the best results in the sense of detecting similar chemicals.

Tanimoto coefficient has also been used as kernel for SVM for classifying similar chemicals by Trotter (2006). The successful applications of Tanimoto in the chemoinformatics literature lead us to use this coefficient for making inferences from binary microarray data. By using Tanimoto coefficient, classification and clustering experiments are carried out in this chapter. Results and the details of the experiments are presented in the following section.

## 5.3 Experiments

Experiments with binary gene expression values using Tanimoto kernel is carried out for classification using SVM and Tanimoto similarity for spectral clustering. We compare our results with the inferences made with continuous data and with the published results.

### 5.3.1 Datasets

We give a short description of the datasets used in this chapter.

- Two **Yeast** datasets. One from Brown et al. (2000), cDNA, 7129 genes, 121 of them are ribosome and the rest non-ribosome with 79 features. The second one is from Causton et al. (2001), Affymetrix, same as above but only the array type used to extract the expression values is different, which is Affymetrix.

- **Leukaemia data set** Golub et al. (1999), there are 5000 genes with 38 samples (27 ALL, 11 AML). We used the 50 genes which has the highest correlation as mentioned in the original work.

- **Colon data set** Alon et al. (1999), 2000 genes with 62 samples (20 normal and 42 tumour samples).

- Three **Breast cancer data sets**, West et al. (2001) 7129 genes and 49 samples, (25 $ER^+$ and 24 $ER^-$), Huang et al. (2003) 12625 genes with 89 samples (depending on LN status) and Gruvberger et al. (2001) 2166 genes and 58 samples, (28 $ER^+$ and 30 $ER^-$) (this dataset is used for cross-platform analysis with West et al. (2001)' s data; so we only used the common genes in these two studies[1]).

- Two **Prostate cancer data sets**, Welsh et al. (2001) 4344 genes and 33 samples, (9 normal and 24 tumor) and Dhanasekaran et al. (2001) 4344 genes with 53 samples (19 normal and 34 tumor) (these datasets are used for cross-platform analysis. So we only used the common genes in these two studies[1].)

- Two **Lung cancer**, one from Gordon et al. (2002), 12533 genes and 181 samples (31 malignant pleural mesothelioma (MPM) and 150 adenocarcinoma (ADCA)) and one from Landi et al. (2008), 22283 genes with 107 samples (58 tumor and 49 normal samples).

- 53 randomly selected datasets from ArrayExpress (`http://www.ebi.ac.uk/arrayexpress/`), Gene Expression Omnibus (GEO) (`http://www.ncbi.nlm.nih.gov/geo/`) and author's web page for probe level uncertainty analysis. Accession numbers and the web links of these datasets are:

  - GEO: `GSE5666`, `GSE7041`, `GSE8000`,`GSE8505`,`GSE6487`,`GSE6850`, `GSE8238`, `GSE2665`
  - Array Express: `E-GEOD-6783`, `E-GEOD-6784`, `E-MEXP-1403`, `E-ATMX-30`, `E-GEOD-6647`, `E-GEOD-6620`, `E-ATMX-13`, `E-MEXP-1443`, `E-GEOD-2450`, `E-GEOD-2535`, `E-MEXP-914`, `E-MEXP-268`, `E-GEOD-2848`, `E-GEOD-2847`, `E-MEXP-430`, `E-GEOD-6321`, `E-MEXP-70`, `E-GEOD-1588`, `E-MEXP-727`, `E-TABM-291`, `E-GEOD-3076`, `E-GEOD-1938`, `E-GEOD-7763`,

[1] Data is provided to us by Arthur Gretton and Karsten Borgward.

E-GEOD-3854, E-GEOD-1639, E-TABM-169, E-MAXD-6, E-MEXP-526, E-GEOD-2343, E-GEOD-3846, E-MEXP-26, E-GEOD-1723, E-GEOD-1934, E-MAXD-6, E-MEXP-879, E-GEOD-10262, E-GEOD-10422, E-MEXP-998, E-MEXP-580, E-GEOD-10072, E-GEOD-10627

– Web pages: `http://yeast.swmed.edu/cgi-bin/dload.cgi`, `http://data.genome.duke.edu/west.php`, `http://data.genome.duke.edu/lancet.php`, `http://www.chestsurg.org/publications/2002-microarray.aspx`

- **Simulated data** is produced by using Dettling (2004)'s code in R. Data is produced according to the mean and correlation structure of leukaemia data (Golub et al., 1999). The size of the data produced is 200 by 250. 200 samples to classify (100 positive classes and 100 negative classes) with 250 features.

### 5.3.2 Classification

As mentioned in the previous chapter, classification is one of the most widely used inference methods for microarray data. We showed experimental results using binary microarray data with standard kernels, particularly linear and radial basis function (RBF) kernels and concluded that information loss is not much when binary data is used instead of continuous data. Here we used a kernel, called Tanimoto kernel (Swamidass et al., 2005; Trotter, 2006), designed for binary data and did experiments for classification on six different datasets. Trotter (2006) has introduced Tanimoto kernel by following the basic definition of Tanimoto coefficient (Eq. 5.1) for SVM and successfully applied it for chemoinformatics data for classifying similar chemicals. Following the success of Tanimoto kernel in SVM we also implemented Tanimoto Kernel using MATLAB SVM toolbox (Gunn, 1998) to microarray data. In Appendix C we show that Tanimoto kernel is a valid kernel.

Since all data only consist of zeros and ones, the number of bits switched on in a string, containing $m$ bits in total is:

$$a = \sum_{i=1}^{m} x_i \quad \text{or} \quad a = \mathbf{x}^T\mathbf{x} \tag{5.2}$$

and the same applies to the calculation of $b$ and $c$ in Eq. 5.1:

$$b = \sum_{i=1}^{m} z_i \quad \text{or} \quad b = \mathbf{z}^T\mathbf{z} \tag{5.3}$$

$$c = \sum_{i=1}^{m} x_i \cdot zi \quad \text{or} \quad c = \mathbf{x}^T\mathbf{z} \tag{5.4}$$

Following the definition of Tanimoto coefficient (Eq. 5.1), Tanimoto kernel is defined as:

$$K_{Tan}(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}^T \mathbf{z}}{\mathbf{x}^T \mathbf{x} + \mathbf{z}^T \mathbf{z} - \mathbf{x}^T \mathbf{z}} \tag{5.5}$$

By following a similar approach in section 4.3.2 we used cross-validation to determine the value of $C$ for linear kernel. The data is partitioned into random training and testing sets for 25 times. Mean and standard deviation are presented in Table 5.1.

Tanimoto kernel has no free parameters to tune and therefore it is computationally fast. Six different datasets, two of which are from the previous chapter and four new cancer datasets are used in this section. From the four new datasets two of them are breast cancer, one of them is a lung cancer and one is a colon cancer study. These datasets are used for comparing results from linear kernel SVM and distance-to-template classifiers which is inspired by a recent work of Zilliox and Irizarry (2007). The details of Zilliox and Irizarry (2007)' s work with further experiments are presented in Section 5.4.

#### 5.3.2.1 Results

We show the comparison of these results in Table 5.1. In the easy problems where the accuracy of the classification is higher and classes are easily separable (e.g., datasets from Gordon et al. (2002) and Brown et al. (2000)), Tanimoto and linear kernels perform the same. In some harder problems (e.g., datasets from West et al. (2001) and Alon et al. (1999)) Tanimoto kernel outperforms linear kernel using both continuous and binary data. Where there is an improvement with the use of Tanimoto kernel we calculated the statistical level of significance, i.e., p-values, which are presented in brackets in Table 5.1. p-values test whether the accuracy obtained with binary data using Tanimoto-SVM is higher than the accuracy obtained with continuous data using linear-SVM. Using Tanimoto kernel improves the performance of the classification problems in microarray studies. Results obtained with linear kernel are very similar to the published ones in literature. For that reason we did not implement any further experiments with any other kernel.

In all the cases Tanimoto kernel outperforms the distance-to-template optimizer (so called Bar code (Zilliox and Irizarry, 2007)). Our results show that Tanimoto can be successfully implemented to binary microarray data even improving the performance of classification if we were to use continuous data. The success of Tanimoto kernel over distance-to-template classifier is highly expected as distance-to-template classifiers only use the distance metrics whereas Tanimoto kernel uses the kernel trick to improve the performance.

We also tried to find templates which may be better than class means for a distance-to-template classifier by implementing a stochastic search by means of a genetic algorithm.

Table 5.1: Comparison of classification with different types of kernels for SVM. "D.O.T" stands for "Distance to optimized template"

| Dataset | Data type | Method | Accuracy |
|---|---|---|---|
| West et al. (2001) | Cont. | Linear-SVM | $0.83 \pm 0.10$ ($p = 0.068$) |
| | Binary | Linear-SVM | $0.86 \pm 0.08$ |
| | Binary | Tanimoto-SVM | $0.87 \pm 0.08$ |
| | Binary | Distance-to-class mean | $0.79 \pm 0.08$ |
| | Binary | D.O.T | $0.77 \pm 0.11$ |
| Huang et al. (2003) | Cont. | Linear-SVM | $0.63 \pm 0.12$ ($p = 0.098$) |
| | Binary | Linear-SVM | $0.67 \pm 0.08$ |
| | Binary | Tanimoto-SVM | $0.67 \pm 0.10$ |
| | Binary | Distance-to-class mean | $0.60 \pm 0.11$ |
| | Binary | D.O.T | $0.66 \pm 0.11$ |
| Gordon et al. (2002) | Cont. | Linear-SVM | $0.99 \pm 0.01$ |
| | Binary | Linear-SVM | $0.96 \pm 0.03$ |
| | Binary | Tanimoto-SVM | $0.99 \pm 0.01$ |
| | Binary | Distance-to-class mean | $0.88 \pm 0.07$ |
| | Binary | D.O.T | $0.90 \pm 0.07$ |
| Brown et al. (2000) | Cont. | Linear-SVM | $0.99 \pm 0.01$ |
| | Binary | Linear-SVM | $0.98 \pm 0.01$ |
| | Binary | Tanimoto-SVM | $0.98 \pm 0.01$ |
| | Binary | Distance-to-class mean | $0.67 \pm 0.02$ |
| | Binary | D.O.T | $0.75 \pm 0.03$ |
| Alon et al. (1999) | Cont. | Linear-SVM | $0.78 \pm 0.11$ ($p = 0.02$) |
| | Binary | Linear-SVM | $0.82 \pm 0.07$ |
| | Binary | Tanimoto-SVM | $0.84 \pm 0.03$ |
| | Binary | Distance-to-class mean | $0.80 \pm 0.07$ |
| | Binary | D.O.T | $0.72 \pm 0.10$ |
| Golub et al. (1999) | Cont. | Linear-SVM | $0.96 \pm 0.05$ |
| | Binary | Linear-SVM | $0.95 \pm 0.03$ |
| | Binary | Tanimoto-SVM | $0.96 \pm 0.04$ |
| | Binary | Distance-to-class mean | $0.94 \pm 0.02$ |
| | Binary | D.O.T | $0.92 \pm 0.09$ |

Templates were initialized to class means. At every step in an iterative search, we randomly changed 20% of the elements in the two templates, to derive mutated bar codes in their vicinity. Throughout the search, we retained ten best template pairs at any iteration. Large search steps were implemented by crossover operation between pairs of templates whereby half the bits in the patterns were swapped between pairs.We evaluated the accuracy of the resulting classifier and if there was an improvement we retained the mutated templates, and discarded them if there was no improvement. Even with the optimized distance-to-templates Tanimoto-SVM outperforms all.

In order to test the efficiency of Tanimoto coefficient over other metrics in this content, a method needs to be used, just to compare the distance metrics. To compare this we applied spectral clustering using Tanimoto coefficient and Euclidean distance. The details and the results are presented in the following subsection.

### 5.3.3 Spectral clustering

Showing that Tanimoto kernel improves the performance of classifier for quantized gene expression data, our next aim is to test how Tanimoto coefficient performs with clustering. But instead of using standard $K$-means or hierarchical clustering, spectral clustering is used. In Appendix A we took expression profiles of published clusters of genes and re-did the clustering algorithm with continuous and quantized data. However, clustering is generally not a stable procedure and results i.e., which gene gets associated with which cluster depend on factors such as initialisation of iterative algorithms (Torrente et al., 2005). The drawbacks of $K$-means or hierarchical clustering are also mentioned in section 2.3.3. Whereas spectral clustering uses the eigenvalue decomposition of the similarity matrix. For that reason, using spectral clustering, which mainly relies on the similarity matrix will give a better insight of the effect of the distance or similarity measure used. In the following experiments we compare Tanimoto coefficient with Euclidean distance by using five different datasets. We report the experimental results from five different datasets (four real datasets and one simulated data). Simulated data is produced by using Dettling (2004)'s code in `R`. Data is produced according to the mean and correlation structure of leukaemia data (Golub et al., 1999). The size of the data produced is $200 \times 250$; 200 samples to classify (100 positive classes and 100 negative classes) with 250 features.

As mentioned earlier, spectral clustering uses eigenvectors of the pairwise similarity matrix to partition the data. The most widely used distance metric to calculate the similarity matrix is the negative exponential of a scaled Euclidean distance.The steps involved in spectral clustering (following Shi and Malik (2000)' s algorithm), in which we replace the Euclidean distance by Tanimoto similarity, are summarized as follows:

1. Pairwise similarity matrix $\mathbf{A}_{ij}$ between the genes $i$ and $j$ is calculated by using Tanimoto coefficient (Eq. 5.1).

2. Following Brewer (2007) (see Eq. 5.6) an exponential is applied:

$$\mathbf{A}^F_{ij} = \exp^{-\alpha(\mathbf{A}_{ij}-1)^2} \tag{5.6}$$

where $\alpha$ is a free parameter and it plays an important role in adjusting the clusters

3. Compute the normalized Laplacian matrix (Eq. 5.7).

$$\mathbf{L} = \mathbf{D}^{-1/2} \times \mathbf{A}^F \times \mathbf{D}^{-1/2} \tag{5.7}$$

where $D(i,i) = \sum_j A(i,j)$.

4. Compute the generalized eigenvalue decomposition of $L$.

$$(\mathbf{D} - \mathbf{L})y_i = \lambda_i \mathbf{D}\mathbf{y}_i \tag{5.8}$$

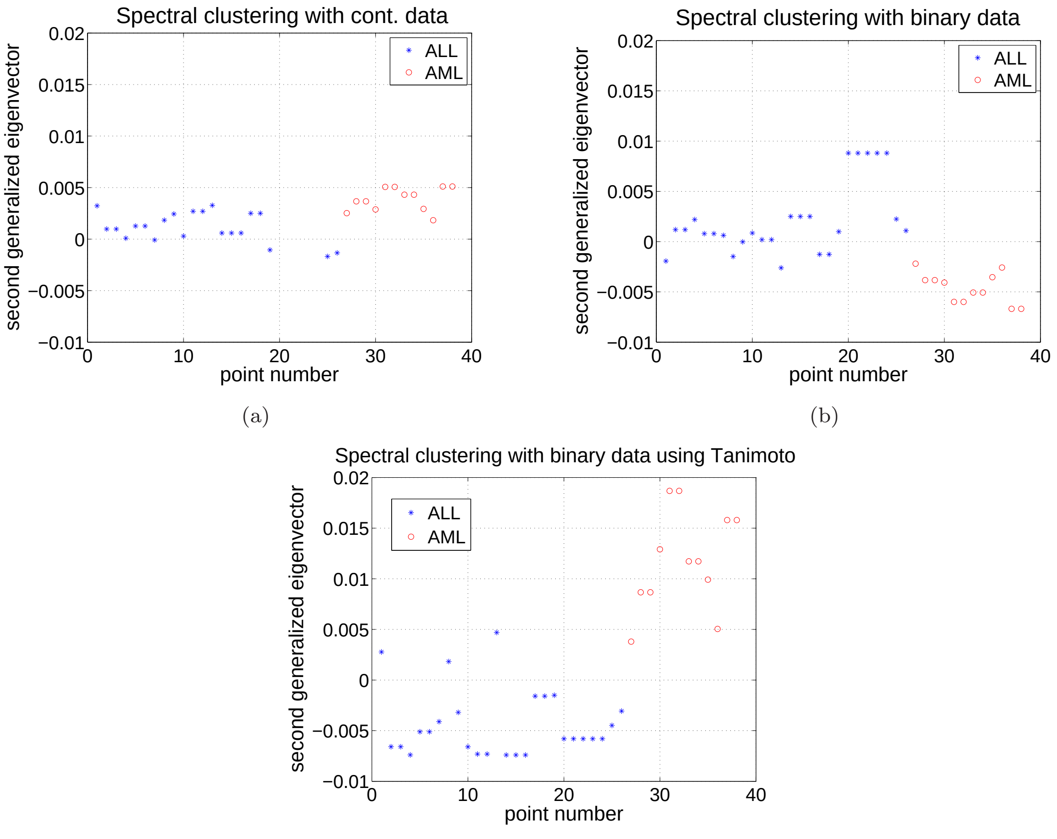
5. Select the eigenvector corresponding to the second smallest eigenvalue.

#### 5.3.3.1 Results

Table 5.2 shows spectral clustering results. A comparison between continuous data-Euclidean distance, binary data-Euclidean distance and binary data-Tanimoto similarity is made. We report the results by using Fisher score, to see how much they are separable, and the error rates, to see how much error is made. Originally classification problems were chosen on purpose here so that without using the labels in clustering process, we can still keep track of the labels and evaluate them for the results. Fig 5.1 shows an example of this analysis applied to Golub et al. (1999) ALL vs. AML problem. When a horizontal line as threshold is selected, it can easily be seen how much these two clusters are separable.

As a second step for spectral clustering we applied feature filtering to four different datasets. By using Fisher Ratio, genes which are easily separable are sorted. In Figure 5.2 we show the results.

In the spectral clustering results, except in one dataset (Huang et al. (2003)), Tanimoto similarity outperforms Euclidean distance with continuous and binary data. The clusters obtained with Tanimoto similarity are more discriminant (see Fisher Ratio) with less errors (see error rate). This is also true when a subset of genes is selected for the same analysis (see Fig. 5.2 and the last column in Table 5.2). For the simulated data, Tanimoto coefficient still outperforms the Euclidean distance with binary data in both Fisher Ratio and error rates. For simulated data we did not calculate the best subset of genes as simulated data is not expected to have any noise.



(a)

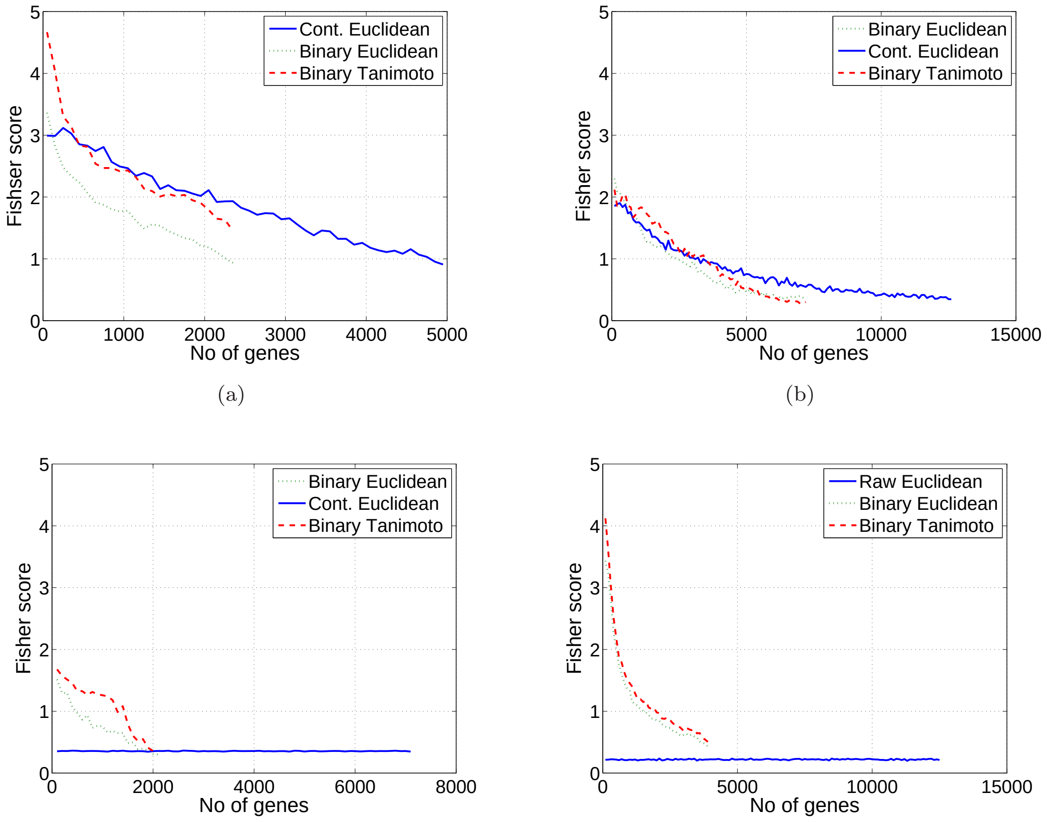
(b)

(c)

FIGURE 5.1: Figures showing spectral clustering results for different type of metrics. In (a) spectral clustering is applied to continuous data by using Euclidean distance, in (b) binary data is used with Euclidean distance and in (c) binary data is used with Tanimoto coefficient for spectral clustering. Data from Golub et al. (1999)

## 5.4 Bar code vs. Tanimoto-SVM

A recent work of Zilliox and Irizarry (2007) contains ideas similar to ours. By considering all the tissue samples from a certain type of Affymetrix array, namely HGU133A human array, they quantize the whole data and for each tissue by taking the means they define a bar code. By using Euclidean distance, they predict the tissue types. They claim bar code gets good results, but they fail to compare their results with the-state-of-art classification method, SVM. Bar code or distance-to-template classifiers can not always be expected to perform well, especially if the assumption of data coming from a normal distribution is violated (Duda et al., 2001). The results presented in their work seem to be fairly easy problems. By using two datasets from Zilliox and Irizarry (2007) and one random dataset from GEO database (accession no. E-GEOD-10072) which is a cancer lung vs. normal lung classification problem, we compared our method of Tanimoto-SVM with bar code (Table 5.3) We used the `R` code which is made available by the authors' at their web page: `http://rafalab.jhsph.edu/barcode/`.



(a) (b) (c) (d)

Figure 5.2: Comparison of spectral clustering results for four different datasets at various number of genes selected with Fisher Ratio. (a) is for Golub et al. (1999),(b) is for Huang et al. (2003), (c) is for West et al. (2001) and (d) is for Gordon et al. (2002).

### 5.4.1 Results

In the lung cancer problem, we found out that bar code can only detect 50% accuracy if the tissue is lung or not, and not even making a discrimination if it is cancerous or not. We evaluated this dataset in two ways, first classifying normal lung from the cancer lung with an accuracy of 99% and the second evaluation was separating lung from the other tissues such as breast and lymph node/tonsil. Here we got an accuracy of 89% which are quite higher than the accuracy of bar code. For the other two datasets even though results from barcode are quite high, these results could not out perform the performance of Tanimoto-SVM. We even tested Tanimoto-SVM with the optimized distance-to-template classifiers (see Table 5.1). Our results show that bar code or optimized distance-to-template classifiers can not compete with Tanimoto-SVM.

However, the main aim of bar code to remove the lab effect of microarray measurements should not be missed. As bar code considers all the arrays to construct the bar code, this may help to remove the lab effect of the microarray measurements. But there should be better method to do this as bar code fails to compete with Tanimoto-SVM. This can

Table 5.2: Comparison of the spectral clustering results by using Tanimoto and Euclidean distance with Fisher Ratio and error rate.

| Dataset | Data type | Distance metric | Fisher Ratio | Error Rate | Error Rate (best subset of genes) |
|---|---|---|---|---|---|
| Simulated Data | Cont. | Euclidean | $2.47 \pm 0.50$ | $0.14 \pm 0.08$ | |
| | Binary | Euclidean | $0.47 \pm 0.49$ | $0.33 \pm 0.02$ | |
| | Binary | Tanimoto | $0.66 \pm 0.21$ | $0.21 \pm 0.10$ | |
| Golub et al. (1999) | Cont. | Euclidean | $0.98 \pm 0.41$ | $0.32 \pm 0.23$ | $0.05 \pm 0.11$ |
| | Binary | Euclidean | $1.01 \pm 0.43$ | $0.10 \pm 0.08$ | $0.02 \pm 0.04$ |
| | Binary | Tanimoto | $1.49 \pm 0.42$ | $0.05 \pm 0.05$ | $0.004 \pm 0.02$ |
| Huang et al. (2003) | Cont. | Euclidean | $0.35 \pm 0.22$ | $0.21 \pm 0.05$ | $0.04 \pm 0.05$ |
| | Binary | Euclidean | $0.37 \pm 0.18$ | $0.22 \pm 0.05$ | $0.03 \pm 0.05$ |
| | Binary | Tanimoto | $0.33 \pm 0.17$ | $0.21 \pm 0.05$ | $0.02 \pm 0.04$ |
| West et al. (2001) | Cont. | Euclidean | $0.35 \pm 0.04$ | $0.45 \pm 0.06$ | $0.45 \pm 0.06$ |
| | Binary | Euclidean | $0.30 \pm 0.18$ | $0.33 \pm 0.08$ | $0.21 \pm 0.15$ |
| | Binary | Tanimoto | $0.35 \pm 0.24$ | $0.28 \pm 0.09$ | $0.11 \pm 0.07$ |
| Gordon et al. (2002) | Cont. | Euclidean | $0.21 \pm 0.07$ | $0.17 \pm 0.03$ | $0.16 \pm 0.03$ |
| | Binary | Euclidean | $0.41 \pm 0.19$ | $0.13 \pm 0.02$ | $0.09 \pm 0.03$ |
| | Binary | Tanimoto | $0.52 \pm 0.19$ | $0.12 \pm 0.02$ | $0.08 \pm 0.02$ |

Table 5.3: Comparison of Tanimoto-SVM with Zilliox and Irizarry (2007)' s barcode.

| Dataset | Data type | Method | Accuracy |
|---|---|---|---|
| E-GEOD-10072 | Binary | Bar code | 0.50 |
| Lung | Binary | Tanimoto-SVM | $0.89 \pm 0.03$ |
| Lung tumor vs. normal | Binary | Tanimoto-SVM | $0.99 \pm 0.03$ |
| GSE2665 | Binary | Bar code | 0.95 |
| lymph node/tonsil | Binary | Tanimoto-SVM | $0.99 \pm 0.02$ |
| lymph node vs. tonsil | Binary | Tanimoto-SVM | $1.0 \pm 0.0$ |
| GSE2603 | Binary | Bar code | 0.90 |
| Breast Tumor | Binary | Tanimoto-SVM | $0.99 \pm 0.01$ |
| Breast Tumor vs. normal | Binary | Tanimoto-SVM | $0.99 \pm 0.01$ |

be done by using a similar approach to the Warnat et al. (2005) where the study uses two different arrays but same type of data (i.e., breast cancer or prostate cancer data). By using this we can remove the platform effects, as analysed further in the following subsection.

## 5.5 Cross-platform analysis

The same types of problems have been studied in literature using different technologies. Two examples to this are the breast cancer classification (cancer vs. normal) by West et al. (2001) and Gruvberger et al. (2001) and prostate cancer classification (cancer vs. normal) by Welsh et al. (2001) and Dhanasekaran et al. (2001) (see Table 5.4 and 5.5 for details). However, different platforms use different methods to report the gene expression measurements. Due to these differences in protocols used, it is not possible to directly combine these measurements or make comparisons.

Warnat et al. (2005) and Gretton et al. (2009) offer novel algorithmic approaches to dealing with cross platform variations. In their formulation training data for a cancer vs non-cancer SVM classifier is assumed to come from a particular microarray platform and the unseen test data is assumed to come from a different platform. As one would expect, with no adjustment to the data, test set performance is very poor. Warnat et al. (2005) offer two solutions to improving on this: the use of median rank scores (MRS) and quantile discretizations (QD). The former approach uses ranks of genes as features in computing similarity metrics while the latter quantizes data into eight bins, the ranges of which are set to equalize bin occupancy. This second method is similar in spirit to the method we advocate in that ours is to quantize down to binary levels. Gretton et al. (2009) develop an approach aimed at the more generic problem of test set distributions being different from training set distributions. A weighting scheme known as kernel mean matching (KMM) is developed and microarray cross-platform inference is used as a test problem to evaluate their algorithm.

To demonstrate how binary representations help in cross platform inference, we carried out experiments on breast and prostate cancer datasets. These datasets are the same as those used in Warnat et al. (2005) and Gretton et al. (2009) and were given to us by the authors in processed format (i.e., we worked with the expression levels rather than with the raw data at the CEL file or image levels). These data come from spotted cDNA and Affymetrix platforms, and details of the four datasets are summarized in Tables 5.4 and 5.5. Warnat et al. (2005) preprocessed all the data and found the subset of common genes by means of the Unigene database (http://www.ncbi.nlm.nih.gov/unigene).

Table 5.4: Details of breast cancer datasets

| Study | Breast cancer | | | |
|---|---|---|---|---|
| | Platform | #common genes | Samples | Target variable |
| West et al. (2001) | Affymetrix | 2166 | 49 | ER-status: 25(+), 24(-) |
| Gruvberger et al. (2001) | cDNA | 2166 | 58 | ER-status: 28(+), 30(-) |

### 5.5.1 SVM classification

In implementing SVM classifiers, we first ensured that our implementation achieves the same results as reported in Warnat et al. (2005). Table 5.8, "cont-not normalized"

Table 5.5: Details of prostate cancer datasets

| Study | Prostate cancer | | | |
|---|---|---|---|---|
| | Platform | #common genes | Samples | Target variable |
| Welsh et al. (2001) | Affymetrix | 4344 | 33 | 9 normal, 24 tumor |
| Dhanasekaran et al. (2001) | cDNA | 4344 | 53 | 19 normal, 34 tumor |

column confirms that our implementation achieves the same results reported previously. Then, following the suggestion in Gretton et al. (2009), we normalized each array to have a mean of zero and standard deviation one, and trained and tested our SVM implementations. This normalization has a significant impact on the results ("cont-normalized", in Table 5.8). We then quantized the data and applied Tanimoto kernel SVM. Note this kernel has no tuning parameters. We implemented quantization on an array by array basis. In chapter 4 we have experimented with different ways of quantization (array by array, gene by gene and a global method), and noted only small differences between these over a range of quantization thresholds.

### 5.5.2 Results

Tables 5.6 and 5.7 show the difference in classification between continuous and binary representations on the two cancer classification problems. Accuracies are shown for 25 random partitions of the data into training and test sets, along with standard deviations quantifying the uncertainty in this process. We see that in three out of the four cases, binarization, and the use of Tanimoto kernel, offers significant improvements, and performs no worse than continuous data in the fourth. Warnat et al. (2005)' s, results are averaged over 10 cross validation runs, but the paper does not report the variation across results.

Table 5.6: Breast cancer results on individual datasets. Data is randomly partitioned into training and testing for 25 times.

| Dataset | Data type | Method | Accuracy |
|---|---|---|---|
| Gruvberger et al. (2001) | Cont. | Linear-SVM | 0.80±0.07 |
| Gruvberger et al. (2001) | Binary | Tanimoto-SVM | 0.82±0.08 |
| West et al. (2001) | Cont. | Linear-SVM | 0.76±0.15 |
| West et al. (2001) | Binary | Tanimoto-SVM | 0.79±0.11 |

Table 5.8 presents results of training SVMs with one type of data and testing the performance on data from a different platform. In this cross platform comparison, normalization as a first step has a big impact. Further improvement is obtained by our binarized Tanimoto approach. While in one of the four experiments this approach gives

Table 5.7: Prostate cancer results on individual datasets. Data is randomly partitioned into training and testing for 25 times.

| Dataset | Data type | Method | Accuracy |
|---|---|---|---|
| Dhanasekaran et al. (2001) | Cont. | Linear-SVM | 0.89 ± 0.06 |
| Dhanasekaran et al. (2001) | Binary | Tanimoto-SVM | 0.89 ± 0.05 |
| Welsh et al. (2001) | Cont. | Linear-SVM | 0.92 ± 0.06 |
| Welsh et al. (2001) | Binary | Tanimoto-SVM | 0.96 ± 0.06 |

poor performance, it proves useful in the other three.

Table 5.8: Cross-platform classification results. Array by Array quantization. The notation "Gruvberger → West" indicates that we train on Gruvberger' s data and test on West' s data.

| Dataset | Data type | Accuracy |
|---|---|---|
| Gruvberger → West | Cont.(not normalized) | 0.49 |
| Gruvberger → West | Cont.(normalized) | 0.94 |
| Gruvberger → West | Binary | 0.96 |
| West → Gruvberger | Cont.(not normalized) | 0.52 |
| West → Gruvberger | Cont.(normalized) | 0.93 |
| West → Gruvberger | Binary | 0.90 |
| Dhanasekaran → Welsh | Cont.(not normalized) | 0.27 |
| Dhanasekaran → Welsh | Cont.(normalized) | 1 |
| Dhanasekaran → Welsh | Binary | 1 |
| Welsh → Dhanasekaran | Cont.(not normalized) | 0.64 |
| Welsh → Dhanasekaran | Cont.(normalized) | 0.93 |
| Welsh → Dhanasekaran | Binary | 1 |

In Table 5.9 we give a comparison with other previously published results on the same datasets, namely the median rank and quantile discretization of Warnat et al. (2005) and the kernel mean matching approach of Gretton et al. (2009). While the number of experiments is small, we note that the binarized Tanimoto method we propose has merit in terms of its performance in a cross platform setting.

Table 5.9: Comparison of our approach to the published results in literature. Accuracies obtained by SVM are compared.

| Study | Train → Test | Method | | | |
|---|---|---|---|---|---|
| | | MRS | QD | KMM | Binary |
| Breast cancer | Gruvberger → West | 0.63 | 0.86 | 0.94 | **0.96** |
| | West → Gruvberger | **0.95** | 0.92 | **0.95** | 0.90 |
| Prostate cancer | Dhana → Welsh | 0.88 | 0.97 | 0.91 | **1** |
| | Welsh → Dhana | 0.89 | 0.91 | 0.83 | **1** |

## 5.6 Probe level uncertainty in microarray measurements

Tanimoto coefficient, as a kernel in SVM and as a similarity metric in clustering, had been shown to improve the performance of the inferences drawn from microarray data. We seek the answer for this in the uncertainties of microarray measurements. The property of Tanimoto coefficient that it focuses on the number of bits on (1s) in an vector hide the answer to the question why Tanimoto coefficient actually does better than the Euclidean distance. i.e., Tanimoto similarity metric attaches higher scores to profiles with large numbers of expressed genes. For example if we consider two pairs of vectors $\mathbf{x}$ and $\mathbf{y}$

$$\begin{array}{ll}
\mathbf{x} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0] & \mathbf{x} = [1\ 1\ 0\ 0\ 0\ 0\ 0\ 0] \\
\mathbf{y} = [1\ 1\ 0\ 0\ 0\ 0\ 0\ 0] & \mathbf{y} = [1\ 1\ 1\ 0\ 0\ 0\ 0\ 0] \\
\text{Hamming distance} = 1 & \text{Hamming distance} = 1 \\
\text{Tanimoto similarity} = 0.5 & \text{Tanimoto similarity} = 0.66
\end{array}$$

In both cases Hamming distance, thus Euclidean distance, is 1. The Tanimoto similarities between these pairs, however, are different: 0.5 for the first pair and 0.66 for the second. We suggest that a weighting on the similarity scores translates to improved clustering and class prediction performance which comes from the uncertainties associated with microarray measurements. In this subsection we focus on Affymetrix GeneChip array and analyse the uncertainties of the microarray measurements from the probe level. Including uncertainties from the probe level had been shown previously by Rattray et al. (2006) and Sanguinetti et al. (2005) to improve the results of the Principal Component Analysis (PCA) of the microarray studies. We mainly focus on multi-mgMOS (Liu et al., 2005) and show some preliminary results from the `mas5calls` as well.

### 5.6.1 Experiments on uncertainty

By randomly selecting 53 datasets from ArrayExpress and GEO databases, we calculated the expression values and uncertainties with multi-mgMOS. It should be noted that these uncertainties are technical variances. For three of these datasets we applied the classification. And for the same three datasets we also applied the mas5calls to get the p-values. A comparison between the uncertainties obtained with multi-mgMOS and the p-values obtained with `mas5calls` is made.

After obtaining the uncertainties, each array is quantized with GMM and for those genes which are represented as expressed after binarized, uncertainties or p-values are recorded. Then we calculated the average uncertainty or (p-values for `mas5calls`) for the expressed genes in an array. This procedure is summarized below:

$$\begin{array}{ccccc} & \text{Exp1} & \text{Std1} & \text{Exp2} & \text{Std2} \\ \text{Genes} & \begin{bmatrix}1\\0\\\vdots\\1\\0\end{bmatrix}_{(n,1)} & \begin{bmatrix}0.7\\0.9\\\vdots\\0.3\\0.8\end{bmatrix}_{(n,1)} & \begin{bmatrix}1\\1\\\vdots\\1\\0\end{bmatrix}_{(n,1)} & \begin{bmatrix}0.4\\0.5\\\vdots\\0.2\\0.8\end{bmatrix}_{(n,1)} \\ & \sum n & & \sum n & \end{array}$$

#### 5.6.1.1 Results

First we present the results from multi-mgMOS analysis. After binarizing the expression values for each experiment by using GMM, the number of expressed genes are counted and for those genes which are expressed the mean of uncertainties are calculated. The number of expressed genes vs. the average uncertainty of expressed genes is plotted for three of the datasets used in the previous section of classification (see Fig. 5.3). The result of this experiment showed some unnoticed systematic variations in microarray measurements i.e., average uncertainty over expressed genes is lower when there are more expressed genes. This should not be confused with lower signal measurements having higher uncertainties. The above result is reached solely by analysing expressed genes and the corresponding uncertainties.

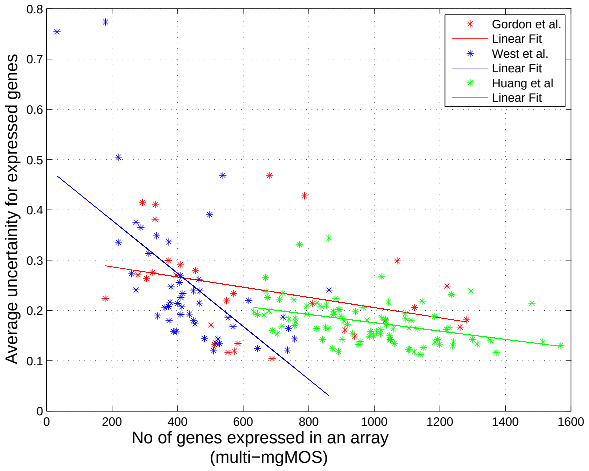


FIGURE 5.3: A systematic variation in probe level uncertainty of *Affymetrix* microarray data. Scatter plots of uncertainties against number of expressed genes, and the linear regression lines, for the three datasets whom classification had been applied in the Section 5.3.2.

Looking at these three datasets, as there are more expressed genes in an array, the uncertainty is lower; we wanted to test this in a more systematic way. For this we downloaded 53 randomly selected datasets from public repositories. Except in one of these experiments (data not shown) we found that as there are more expressed genes, microarray measurements are more certain (see Fig. 5.4). For this particular reason we suggest that the Tanimoto coefficient performs better than the other metrics and improves the performances of inference problems.

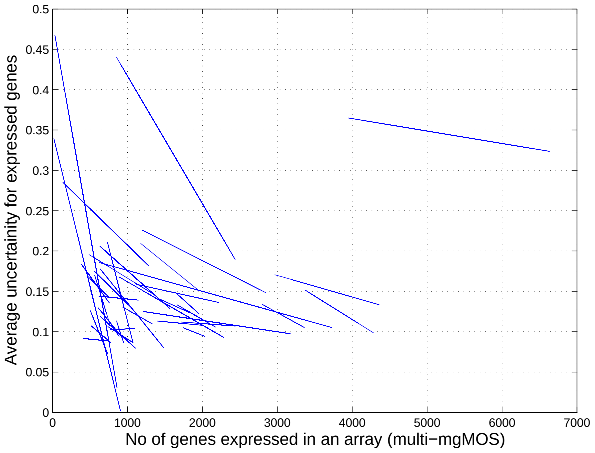


FIGURE 5.4: A systematic variation in probe level uncertainty of *Affymetrix* microarray data. 53 randomly chosen arrays we plot the average uncertainty of determining expression levels against the number of genes detected as present. Only liner regression lines are shown for clarity.

In order to show that these results are not the consequence of noise, we test the same idea above by using Affymetrix P and A calls and the corresponding p-values. Since detection calls is a non-parametric approach it is a robust method as explained in section 2.3.8. Expression values obtained with `mas5` function in `R` are binarized by using the GMM and the corresponding p-values are stored. Then by applying the same methods above we counted the number of genes expressed in an array and calculated the mean of the p-values for the expresses genes. Fig. 5.5 shows these results. It can be seen that the p-values are also lower when there are more expressed genes which strengthen our results obtained with multi-mgMOS.

The biological interpretation of this result needs to be further investigated. Whether these results are only because of the particular array property or is it the same for most of the arrays needs to be verified with experiments.

## 5.7 Summary

In this chapter we show how to improve the inferences drawn from transcriptome data when binary data is used. We use a similarity metric suitable for binary data. This metric, called Tanimoto similarity, is widely applied in chemoinformatics and has been shown to give best performance for binary data. We focused on classification and clustering and presented experimental results. Our results showed that using binary gene

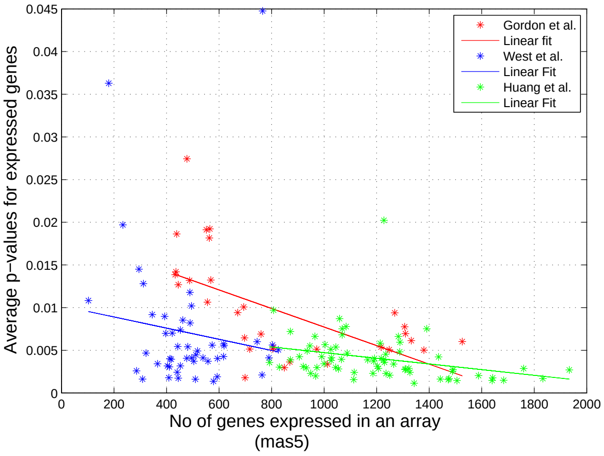


Figure 5.5: Uncertainty graph with p-values. As there are more expressed genes in an array, the average p-values are lower which supports the finding of multi-mgMOS results.

expression data with metrics suitable for binary data improves the performance of inference from gene expression data.

We further show an advantage of using binary data with Tanimoto kernel in classification. By binarizing data we can combine datasets from cross-platform studies and this improve the classification performance of the individual datasets. Our approach has been shown to outperform the existing works in literature. We compare cross-platform classification to the Zilliox and Irizarry (2007)' s bar code approach and raise a critical appraisal of this method. One aim of the bar code is to remove the lab effects of microarray measurements. However, bar code approach is only limited to one type of array and is not tested on cross-platform studies. And this is not the only drawback of bar code approach. Our experimental results show that Tanimoto kernel in SVM outperforms distance-to-template classifiers. This is highly expected for a kernel in SVM to perform much better than the distance-to-template classifiers as the kernels use higher dimensions in feature space to separate the data. Our search to find better templates than the ones obtained with mean as Zilliox and Irizarry (2007)' s bar code, fails to do better than Tanimoto kernel. But we go further and make a direct comparison between Tanimoto similarity and Euclidean distance with spectral clustering. As the results of spectral clustering mainly rely on the similarity matrix and thus the similarity metric used, we choose spectral clustering as our method. The results showed that for transcriptome data Tanimoto similarity is doing better than the Euclidean distance in this context.

We explain the success of Tanimoto coefficient with the uncertainties of the probe level data. It is worth mentioning here that the uncertainties obtained by multi-mgMOS is due to technical variance and not biological variance. Using biological variances here would be more appropriate but this is left as a future work of this work. By using 53 randomly selected datasets and two different methods, namely multi-mgMOS and p-values of detection calls where the latter is a robust method, to extract expression values from probe level data, it has been shown that as there are more expressed genes in an array the uncertainty of these genes are lower in Affymetrix GeneChip technology. This property of microarray measurements when combined with the fact that Tanimoto coefficient focuses on the number of bits on (1s) in an array, helps to explain the success of Tanimoto coefficients for the inferences drawn from microarray data. With this weighting of expressed genes, we explain the success of Tanimoto coefficient. While no molecular level explanation is offered for this, we show that when this systematic property of Affymetrix is taken into consideration it improves the quality of inferences drawn from microarray data and hope that this will be taken into account before making any further processes with such data.

By using Tanimoto coefficient, we show that the quality of inferences drawn from microarray data can be improved. Further more, certain advantages of using binarized transcriptome data e.g., in cross-platform analysis, have been shown. In the next chapter we show experimental results from classification that binary transcriptome data significantly reduces the effect of algorithm choice for pre-processing raw data especially when used with Tanimoto kernel.

# Chapter 6

# Reduction in algorithmic variability

## 6.1 Introduction

A plethora of computational methods for the statistical analysis of high throughput gene expression measurements is available to users interested in making inferences from transcriptomes. These steps are commonly known as pre-processing stages. Pre-processing stages includes background correction, within and between-array normalizations, probe-specific correction and summarization. After the raw data is processed it is ready for more sophisticated machine learning approaches such as classification, cluster analysis and the modelling of time-course data by means of dynamical systems. The pre-processing stages lead to quantifications of relative mRNA abundances, taking scanned images as input. The pre-processing stage has been shown to have an important effect on the results of statistical inference approaches (Barash et al., 2004; Cope et al., 2004; Choe et al., 2005; Ploner et al., 2005; Shedden et al., 2005; Millenaar et al., 2006; Allison et al., 2006; Qin et al., 2006). In this chapter we show that binary representation of transcriptome data has the desirable property of reducing the variability introduced at the pre-processing stages due to algorithmic choice. We review the effect of the choice of algorithms on different problems and suggest that using binary representation of microarray data with Tanimoto kernel for SVM reduces the effect of the choice of algorithm and simultaneously improves the performance of classification on transcriptome data.

### 6.1.1 Algorithmic variability during the pre-processing of raw data

Many studies have explored the effect of algorithm choice for pre-processing microarray data. The general conclusion reached by these studies is that the overall results are highly dependent on the algorithms used for pre-processing data. In this section we reviewed some of these studies to illustrate the point.

When Irizarry et al. (2003a) introduced RMA for analysing Affymetrix GeneChip data they compared their proposed method with dCHIP and MAS5.0 (for the combination of algorithms used for dCHIP and MAS5.0 see Table 6.2). While Irizarry et al. (2003a) mention there is no standard way to compare the effect of the algorithm choice, the authors evaluate their results with three criteria by using one dataset: (1) the precision of measures of expression; (2) the consistency of fold change; and (3) the specificity and sensitivity of the measures ability to detect differential expression. While they make comparisons for three different algorithms, they report that RMA performs best among the three algorithms, although there are contradictory findings in the literature (e.g., Choe et al. (2005)). Since these algorithms perform better or worse than the other this also shows that there is a variability between the algorithms chosen even though this is not stated by the Irizarry et al. (2003a) in their study.

For detecting differentially expressed genes by using spike-in data, Choe et al. (2005) analysed different combination of algorithms by using the package `affy` for pre-processing raw microarray data. The main aim of the study is to find the best combination of algorithms. They compare different combinations of algorithms with dCHIP and MAS5.0. The authors considered 152 combinations and made tests on one particular, spike-in, dataset. Their study reports the best combination of algorithms while admitting that results of defining differentially expressed genes vary depending on the combination used. However the authors do not offer a solution to this problem. As the authors also points out themselves, the findings of Choe et al. (2005) contradict the findings of Irizarry et al. (2003a). Choe et al. (2005) explain this with the different datasets used in the two studies. Furthermore, Pearson (2008) reports more contradictory results on spike-in dataset when used with different pre-processing algorithms. It is apparent that testing these combinations with just one dataset will not give the optimum solution. The authors calculate the correlation coefficient between the actual and the observed fold change for all expression datasets obtained with different pre-processing algorithms. In one comparison, this correlation is reported to be 0.508. They found out that when probe sets with low signal intensity i.e., lowest quartile of signal intensity are removed, the correlation coefficient between the actual and the observed fold changes improves significantly (0.87). The authors suggest that using a signal dependent metric will improve the success of microarray analysis. At this point considering Tanimoto coefficient is shown to improve the performance of inference problems. Tanimoto coefficient focuses on 1's which are the expressed genes. Expressed genes are measured with high signal intensities and low uncertainties. This particular metric ignores the not expressed genes which are the ones with low signal intensities and high uncertainties.

Shedden et al. (2005) compare seven algorithms on two different datasets for detecting differentially expressed genes. Algorithms considered in this study are: (1) dCHIP,

(2) GCRMA-EB, (3) GCRMA-MLE, (4)MAS5.0, (5) PDNN, (6) RMA and (7) TM. The authors use the false discovery rate to quantify the sensitivity. They found that dCHIP performs the best but also agree that to make a definitive conclusion about different algorithms, i.e., which one is the best or worst, many more datasets should be examined. The authors accept that the choice of processing algorithms have a major impact on the results.

Ploner et al. (2005) compare pre-processing stages, focusing on normalization. The study compares MAS5.0, RMA and MBEI on breast cancer, dilution and spike-in data. MAS5.0 is the best according to their results which are carried out on real world datasets.

Unlike most of the studies which compared different methods to summarize gene expression measurements, Qin et al. (2006) uses real world dataset and not spike-in data for their experiments. The study estimates the relative gene expression measurements by evaluating six different methods with qRT-PCR and using the Pearson' s correlation coefficient as the evaluation metric. The pre-processing algorithms considered in this study are: (1) MAS5.0, (2) gcRMA, (3) RMA, (4) VCN, (5) dCHIP and (6) dCHIP.mm. The reason in choosing Pearson' s correlation coefficient as their evaluation metric is that this particular metric takes into account both variance and bias of the measurements produced by arrays and thus can find the balance between the two. Their result shows that MAS5.0, gcRMA, and dCHIP have higher correlations than the rest of the three algorithms tested.

Millenaar et al. (2006) test six different algorithms: namely (1) MAS5, (2) dCHIP PMMM, (3) dCHIP PM, (4) RMA, (5) GC-RMA and (6) PDNN, for calculating the gene expression levels of Arabidopsis. The study considers five criteria to evaluate the results: (1) comparison with spike-in genes, (2) reproducibility, (3) biological relevance, (4) the use of MM probes and (5) comparison with Real Time PCR. The test results showed that all six algorithms result in different levels of gene expression and the authors' suggestion for the users is to test many possible algorithms to decide which one gives a better solution for their datasets. This solution is computationally hard to apply especially if we consider the growth of microarray technology and the size of the datasets produced. A more systematic and easier method in the sense of computational complexity is required for this problem.

Despite many studies being carried out for the pre-processing of microarray data, each study favours their own method as the best. However, as also mentioned by Allison et al. (2006), there is no clear winner in the choice of algorithms. The results reported by each author contradict the other and the real problem that the results and therefore the inferences made with them vary depending on the algorithm choice used is still out there. The most extensive study so far that shows significant algorithmic variability is due to P.C. Boutros [1] who analysed 19,446 different combinations of algorithms for

[1] Boutros P.C., Microarray Gene Expression Society Meeting (MGED), Riva del Garda, Italy (2008)

pre-processing microarray data, and evaluated the results with sensitivity and stability of the algorithms used. While this and similar studies (Choe et al. (2005)) seek the best combination of algorithms on one or two datasets, they do not offer a generic solution for practitioners to select a combination that leads to reliable results in downstream inference. Our approach has the property of reducing the algorithmic variability in class prediction problems.

## 6.2 Pre-processing algorithms

The code `expresso` of the package `affy` (Irizarry et al., 2003b; Gautier et al., 2004) in `R` is a very powerful tool as it gives the user a chance to combine and use many different combinations of algorithms. For this reason we used `expresso` to analyse the algorithmic variability through classification. Table 6.1 shows the list of the methods available in `expresso`. The `expresso` code has four methods and each has 7, 3, 3 and 5 alternative ways respectively. Some combination of algorithms are known with certain names in literature and Table 6.2 gives these common names of the certain combination of algorithms. In Appendix D the details of each method used in `expresso` are given.

Table 6.1: List of possible methods in `expresso`

| Normalization | Background correction | Probe specific correction (PM) | Summary method |
|---|---|---|---|
| constant | mas | mas | avgdiff |
| contrasts | none | pmonly | liwong |
| invariantset | rma | subtractmm | mas |
| loess | | | medianpolish |
| qspline | | | playerout |
| quantiles | | | |
| quantiles.robust | | | |

Table 6.2: The common names of most widely used combinations of algorithms for pre-processing raw microarray data.

| **Common name** | Normalization | Background correction | Probe specific correction (PM) | Summary |
|---|---|---|---|---|
| **dCHIP** | invariantset | none | pmonly | liwong |
| **MAS5.0** | constant | mas | mas | mas |
| **RMA** | quantiles | rma | pmonly | medianpolish |

## 6.3 Experiments

To evaluate the effect of the choice of algorithm, we conducted classification experiments on eight different datasets. SVM is used for this purpose and the results are evaluated by means of AUROC. Each dataset is randomly partitioned into training and test sets 50 times. Thus, each AUROC result reported are the average of 50 runs. On five datasets more than 200 and for the remaining three datasets randomly selected 38 combinations of pre-processing algorithms were used. We found that the randomly selected 38 combinations is a significant sub sample of the whole combinations available in `expresso`(see Table 6.3 for the list of the randomly selected 38 combinations). Classification with continuous data - linear kernel, binary data - linear kernel and binary data - Tanimoto kernel are compared. We worked with CEL files which are downloaded from public array repositories, ArrayExpress and GEO, whose details are given in the following section.

Table 6.3: The list of the 38 combinations of algorithms used in `expresso`.

| Combination no | Background correction | Normalization | PM correction | Summary |
|---|---|---|---|---|
| 1 | mas | constant | mas | liwong |
| 2 | mas | constant | pmonly | liwong |
| 3 | mas | constant | subtractmm | liwong |
| 4 | none | contrasts | pmonly | liwong |
| 5 | rma | contrasts | pmonly | liwong |
| 6 | rma | invariantset | mas | liwong |
| 7 | none | loess | mas | liwong |
| 8 | rma | qspline | subtractmm | liwong |
| 9 | rma | quantiles.robust | subtractmm | liwong |
| 10 | mas | constant | mas | avgdiff |
| 11 | mas | constant | pmonly | avgdiff |
| 12 | mas | constant | subtractmm | avgdiff |
| 13 | none | constant | mas | avgdiff |
| 14 | none | constant | pmonly | avgdiff |
| 15 | none | constant | subtractmm | avgdiff |
| 16 | rma | constant | mas | avgdiff |
| 17 | rma | constant | pmonly | avgdiff |
| 18 | rma | constant | subtractmm | avgdiff |
| 19 | mas | loess | subtractmm | avgdiff |
| 20 | none | loess | mas | avgdiff |
| 21 | none | loess | pmonly | avgdiff |

Continued on next page

**Table 6.3 – continued from previous page**

| Combination no no | Background correction | Normalization | PM correction | Summary |
|---|---|---|---|---|
| 22 | none | loess | subtractmm | avgdiff |
| 23 | none | qspline | pmonly | avgdiff |
| 24 | mas | quantiles | pmonly | avgdiff |
| 25 | mas | quantiles.robust | mas | avgdiff |
| 26 | none | quantiles.robust | subtractmm | avgdiff |
| 27 | rma | quantiles.robust | subtractmm | avgdiff |
| 28 | rma | constant | mas | mas |
| 29 | mas | loess | mas | mas |
| 30 | mas | qspline | pmonly | mas |
| 31 | rma | quantiles.robust | pmonly | mas |
| 32 | rma | constant | pmonly | mas |
| 33 | none | contrasts | pmonly | mas |
| 34 | rma | invariantset | mas | mas |
| 35 | rma | invariantset | pmonly | mas |
| 36 | rma | loess | mas | mas |
| 37 | mas | quantiles | pmonly | mas |
| 38 | rma | quantiles | mas | mas |

### 6.3.1 Datasets

We give a short description of the datasets together with the GEO accession number or the authors' web page, used in this chapter.

- Two **prostate cancer datasets**. First one is GSE6956 with 22277 genes with 89 samples. 69 prostate and 20 normal samples (Wallace et al., 2008). The second one is from Singh et al. (2002) with 12625 genes with 102 samples. 52 prostate and 50 normal samples (`http://www.broad.mit.edu/`)
- **lung cancer dataset** (GSE7670) 22283 genes with 66 samples. 30 normal and 36 cancer samples (Su et al., 2007).
- **breast cancer dataset** (GSE5847) 22283 genes with 95 samples. 47 normal and 48 cancer samples (Boersma et al., 2007).
- **lymph node vs. tonsil** problem (GSE2665) 22283 genes with 20 samples. 10 lymph node and 10 tonsils (Martens et al., 2006).

- **Childhood acute lymphoblastic leukemia** (ALL) (GSE3910) 22283 genes with 70 samples. 35 for each diagnosis and relapse (Bhojwani et al., 2006).
- **Breast cancer data sets**, (`http://data.genome.duke.edu/west.php`) 7129 genes and 49 samples, (25 $ER^+$ and 24 $ER^-$) (West et al., 2001).
- **Lung cancer** (GSE10072), 22283 genes with 107 samples. 58 cancer and 49 normal samples (Landi et al., 2008)

### 6.3.2 Results

Fig. 6.1 and Table 6.4 summarize the classification results obtained with different preprocessing algorithms.

Table 6.4: Summary of the algorithmic variability results. Standard deviation of the AUROCs obtained with different combinations are presented in the table. The numbers in brackets show the mean of the AUROCs.

| Study | # Comb. | Continuous | Binary-Linear | Binary-Tanimoto |
|---|---|---|---|---|
| GSE2665 | 273 | 0.07 (0.97)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.002 (0.99)<br>$p_{\text{var}} = 1$<br>$p_{\text{mean}} = 0.95$ | 0.004 (0.99) |
| GSE7670 | 215 | 0.006 (0.99)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.002 (0.99)<br>$p_{\text{var}} = 0.23$<br>$p_{\text{mean}} = 0.001$ | 0.002 (0.99) |
| GSE6956 | 211 | 0.03 (0.84)<br>$p_{\text{var}} = 0.94$<br>$p_{\text{mean}} < 0.001$ | 0.03 (0.90)<br>$p_{\text{var}} = 0.70$<br>$p_{\text{mean}} = 0.76$ | 0.03 (0.90) |
| West et al. (2001) | 210 | 0.04 (0.91)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.02 (0.91)<br>$p_{\text{var}} = 0.94$<br>$p_{\text{mean}} < 0.001$ | 0.02 (0.93) |
| Singh et al. (2002) | 203 | 0.09 (0.88)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.023 (0.91)<br>$p_{\text{var}} = 0.03$<br>$p_{\text{mean}} = 1$ | 0.02 (0.90) |
| GSE3910 | 38 | 0.04 (0.60)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.03 (0.64)<br>$p_{\text{var}} = 0.11$<br>$p_{\text{mean}}0.89$ | 0.02 (0.064) |
| GSE10072 | 38 | 0.005 (0.998)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}}0.04$ | 0.0004 (0.999)<br>$p_{\text{var}} = 0.01$<br>$p_{\text{mean}} < 0.24$ | 0.0004 (0.999) |
| GSE5847 | 38 | 0.02 (0.67)<br>$p_{\text{var}} < 0.001$<br>$p_{\text{mean}} < 0.001$ | 0.01 (0.71)<br>$p_{\text{var}} = 0.05$<br>$p_{\text{mean}} = 0.61$ | 0.009 (0.71) |

As mentioned in the description of the algorithms in Appendix D there are instances where certain combinations return NA values e.g. *subtractmm* returning negative values.

This issue is also mentioned the manual of the affy package (Bolstad, 2004). On each of the eight problems considered here, variability caused by algorithm choice was significantly reduced by the binarization of the data (box plots in the second columns of each graph in Fig. 6.1). This reduction is further improved by the use of the Tanimoto similarity metric (third column of box plots). The particular metric places emphasis on the expressed genes which have less variation of measurement; and ignores the not expressed genes which are the low signal intensity with higher variation of measurement. Admittedly, on two of these problems, the classification task is so easy that improvements are very small. But with the remaining seven of them there is a significant improvement with binary data and further improvement when used with Tanimoto kernel. Statistical significances using F-test for variance differences and t-test for mean differences, comparing against our recommended binary Tanimoto method are also given in Table 6.4. It is seen that there is a significant variance reduction in seven of the eight datasets. Even though there is no variability reduction for GSE6956, there is a significant improvement in the performance of classification when binary gene expression data is used.

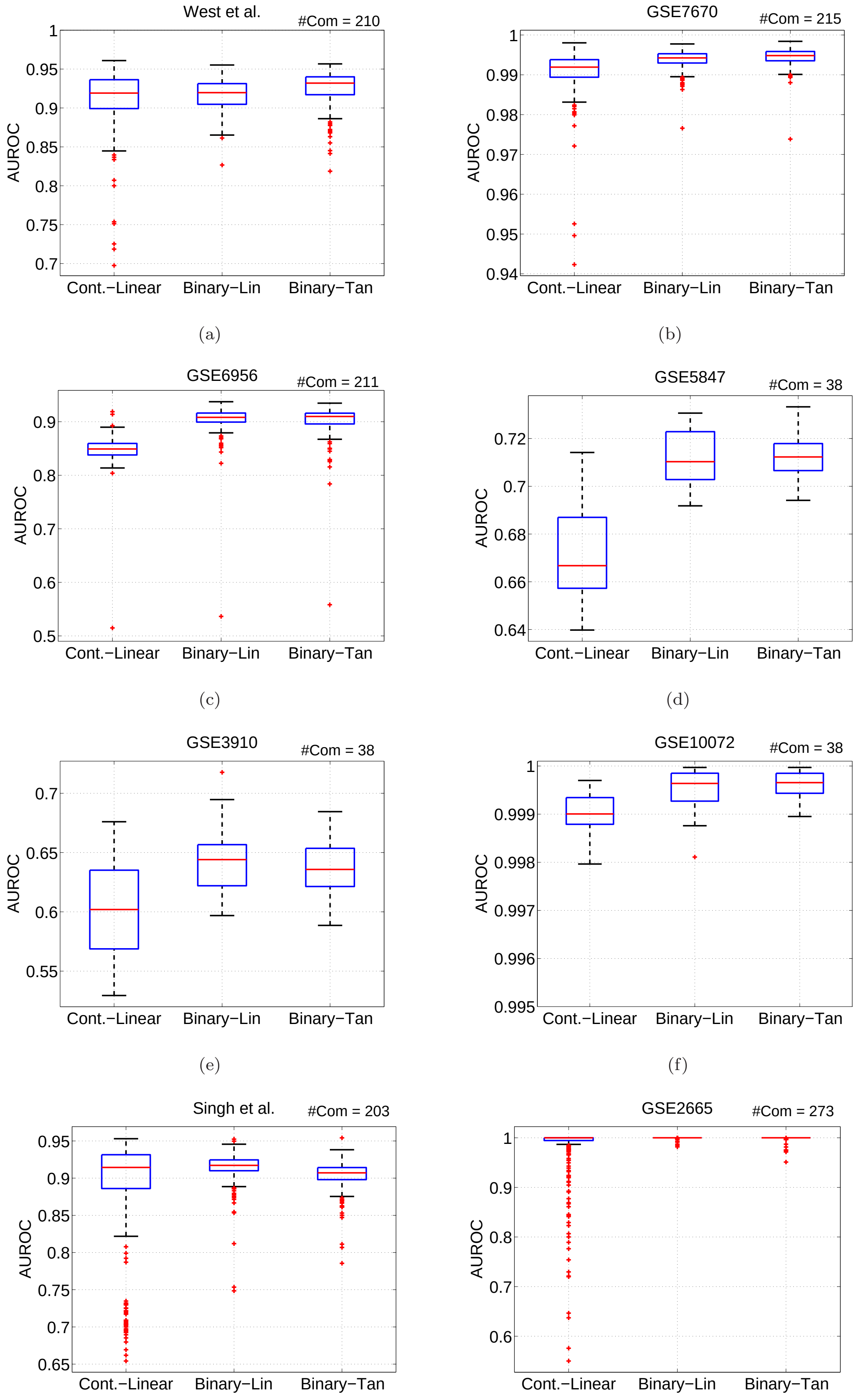
While binarizing data removes the noise and thus reduces the algorithmic variability, we also note that critical observation made by Choe et al. (2005) in stating that genes with low levels of expression being more susceptible to noise in hybridization, smooth on observed correlations. They extend this to claim that a similarity metric that suppress the effect of the genes is a desirable one. Our work on Tanimoto kernel achieves precisely this. Thus the variability of inferences drawn from microarray data can be drastically reduced when used as binary data, just the same way the so called lab effects can be reduced in a binary representation, as established previously. Use of a signal dependent metric further improves this.

## 6.4 Summary

In this chapter we reviewed the pre-processing algorithms for raw microarray data. There are many pre-processing algorithms for users to choose with no universally accepted guideline. Noting that the overall results of inferences drawn from transcriptome data are highly dependent on the choice of algorithms we propose a solution for this problem. First we reviewed some of the existing works for pre-processing algorithms where each study favours their own algorithm as the best. But a careful review suggests that there is no winner. Most of the studies focus on detecting differentially expressed genes using a single dataset (usually spike-in data) and by testing it with several algorithms. Different studies report different algorithm as the ‘best’ depending on the dataset being used. For instance, Irizarry et al. (2003a) conclude that RMA performs better than MAS5.0 (Affymetrix, 2002) and dCHIP (Li and Wong, 2001a) while Choe et al. (2005)’ s directly contradict this claim. These examples can be increased, Ploner et al. (2005) claims MAS5.0 outperforms RMA and dCHIP while Shedden et al. (2005)

suggest dCHIP is the best algorithm among seven others. These contradictory observations are in part due to the authors comparing techniques on different datasets. Our approach is distinctly different from these because we are interested in the variability seen in the downstream inference rather than the precise measurement of mRNA abundance. The basic descriptions of the pre-processing algorithms are given in Appendix D. Then experiments on 8 different real world datasets are carried out. We focused on classification and showed that when transcriptome data is binarized, after pre-processed with any choice of algorithms, it has the effect of reducing the variability of the classification results together with increasing the performance of the classifier. We used AUROC as the evaluation metric and plotted the results as box plots. This success of binary data with Tanimoto kernel addresses the issue of the low signal measurements which have higher uncertainty. By focusing on the expressed genes and ignoring the low expressed genes we show that we can make better inferences from transcriptome data. These findings of ours suggest that high numerical precision is also affected by the pre-processing algorithms and therefore is not realistic. By showing this advantage of binary transcriptome data we hope that this approach will increase the quality of the inferences drawn from microarray studies. It will be our point of interest to investigate how results are affected with other type of arrays.

Secondly, we are interested in making experiments by using the same idea on new DNA sequence data. The new era of parallel DNA sequencing is the new method to quantify transcriptome measurements (Shendure, 2008). Considering this technology is still at its early stage, obtaining such results would be more valuable. With the huge size of this new data variability is expected to be more. However experiments must be carried out to confirm this.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.1: Reduction in variability of results due to pre-processing choice of algorithms. Different combinations of pre-processing the CEL files produce large variations in classification results (leftmost columns). Working with discretized data reduces this variation in the inference.

# Chapter 7

# Conclusion and future work

Microarrays measure the mRNA abundance of thousands of genes in a single experiment. Within a short time microarrays became quite popular and a large number of statistical algorithms have been applied to gene expression data. The aim is to extract the most useful information from such data. However, one thing that the researchers are missing is that the biological properties of mRNA and the nature of the microarray technology suggest that the high numerical precision of these measurements are not realistic. The consequence of this is that microarray measurements are not reproducible. These disadvantages of microarrays have been ignored for a long time and different algorithms have regularly been introduced to the community, each time claiming that the new algorithms are better than the old ones. In this thesis we took a critical approach and questioned the high numerical precision of such measurements. Our results show that binary gene expression measurements are more convenient to the structure of the data.

## 7.1 Questioning the high numerical precision of microarray measurements

We started by questioning the high numerical precision of microarray measurements. By considering several inference problems including classification, clustering, periodicity detection, analysing cell cycle genes with SVD, analysing time series data and differentially expressed genes, we show that not much information is lost when binary data is used. We do this by implementing a simple progressive quantization procedure that dropped one digit at a time (data not shown), until we reached binary numerical precision (gene is expressed (1) or not expressed (0)). Our results show that binary representation is more realistic than the high numerical precision. However, this should not be considered as questioning the accuracy of microarray measurements. Having tested these inferences with standard algorithms, we used Tanimoto metric from chemoinformatics which is a more suitable metric for binary data. Our aim is to see if we can further improve the performance of inferences with binary data.

## 7.2 Improving the performance of inferences with a signal sensitive metric

We show that the inferences drawn from binary microarray data can be improved with the right choice of metric (i.e., Tanimoto coefficient) for binary data. We further explored the reason for the success of Tanimoto coefficient by considering the uncertainties of the microarray measurements from probe level data. With a systematic study by using multi-mgMOS, we found out that in Affymetrix measurements as there are more expressed genes in an array the mean of the uncertainties are lower. This concept should not be confused with low signal measurements having higher uncertainties. These results were also confirmed with a robust method, p-values of detection calls. Using binarized transcriptome data with Tanimoto metric also addresses the issue of the low signal measurements which are associated with higher uncertainty compared to the high signal measurements. Considering a metric which is signal sensitive improves the performance of inferences drawn from transcriptome data. We further show that a binary representation of gene expression profiles, combined with a kernel similarity metric that is appropriate for such data, has the potential to address the important problem in microarray based phenotype classifications of cross platform inference. While the experimental work is on a very small number of datasets, which were the only ones available to us at this time from previous studies, we believe this advantage comes from using a data representation that respects properties of the measurement environment.

## 7.3 Reduction in algorithmic variability

There are many algorithms for pre-processing raw microarray data but among these algorithms there is no clear winner. In many studies it has been shown that the result of inferences heavily relies on these algorithms. Here we proposed a solution to reduce the effect of algorithm choice for pre-processing stage. We focused on classification and showed that while we reduce the algorithmic variability our approach also increase the performance of the classifier simultaneously. This topic needs to be further explored with more datasets and also by including more combination of the algorithmic choice. This part will remain as a future plan of this work.

To summarize all the findings, we suggest that a low numerical precision representation is more compatible with the environment, from which microarray data are gathered, than the arbitrary length of decimal places, to which they are usually reported and archived. Except in a small number of cases like cell cycle regulation studies, where the cellular states are artificially synchronised, mRNA extraction is from a heterogeneous population of cells, each cell usually having a small number of copies of the mRNA species. This causes large variation in measurements across different sub-populations from the same biological sample - the so called biological variability. As mentioned at the beginning of the thesis:

> “...the existence and direction of gene expression changes can be reliably detected for the majority of genes. However, accurate measurements of absolute expression levels and the reliable detection of low abundance genes are currently beyond the reach of microarray technology.” (Draghici et al., 2006)

## 7.4 Future plan

The major concern of this thesis is to show that quantizing gene expression data do not lose much information and with metrics suitable for binary data and the quality of inferences drawn from microarray data can even be improved. We showed these with various experiments. By focusing on classification at the last chapter, we further show two advantages of using binary gene expression data. First one is binary gene expression data make combining data from cross-platform studies possible. The classification accuracies obtained with cross-platform studies is higher than the ones obtained with individual datasets. The second advantage is that we can reduce the effect of the choice of pre-processing algorithms and simultaneously improve the performance of classification task.

The investigation with algorithmic variability is the most promising future part of this thesis. We would be interested in using more combinations of these algorithms with more datasets to understand the effect of the choice of pre-processing algorithms. It is in our interest to see how the choice of pre-processing algorithms affects the results of other inference problems. These inference problems include the ones that were considered in this thesis such as clustering, periodicity detection or time series data. A sensible way should be sought for analysing other types of inference problem. Apart from those, we are interested in doing some experiments with the next generation DNA sequence data. As mentioned in Shendure (2008)’ s review, DNA sequence data is replacing microarrays for quantifying transcriptomes. One of the reason for this as noted in the above paper is that microarray results not being reproducible between laboratories and across platforms. However this still worth experimenting if DNA sequence data is reproducible or does these pre-processing algorithms would have the same effect on DNA sequence data. Considering the huge size of these data, tens of millions tags, by intuition there would be more noise and variability. However, these still need to be verified with experiments. If there is any variability and if this can be reduced, it would be a very valuable step towards the new emerging sequential data and thus the inference made from them. It would be also important as DNA sequencing technology is still in its beginning stage. Whether the same pattern is true for sequence data needs further investigation.

We have shown inferences from different type of arrays that binarizing these data do not lose much information. It is worth experimenting with quantized data from different type of arrays to see if there is any loss of information. Having shown that using a metric suitable for binary data improves the performance of inferences, it is worth doing some experiments using binary PCA for detecting cell cycle genes. We have shown in Chapter 4 that using SVD analysis with binary data there is some loss of information. We think that this loss can be recovered by using binary SVD. It is worth doing some experiments to see how Tanimoto coefficient affects the results for other type of inferences. Also how it can be used for periodicity detection and analysing time series data.

While we explain the success of the Tanimoto coefficient with uncertainties, as there are more expressed genes in an array the average uncertainty is lower, it is worth exploring this in more depth. For example, how other type of arrays are affected. Or how we can explain this at biological level needs further investigation. And furthermore, uncertainties obtained with multi-mgMOS are technical variabilities. It would be more appropriate to use uncertainties which are obtained with biological variabilities.

More cross-platform experiments need to be carried out. Our current work is on extending the study to a larger collection of datasets, the difficulty in doing this being the matching of the gene identities.

Apart form these experimental future plan, we are preparing an open source `R` code which will be the implementation of this thesis. Researchers who are interested in using binarized transcriptome data will be able use this code. First application will be classification using Tanimoto kernel.

# Appendix A

# *K*-means clustering results on questioning the high numerical precision of microarray measurements

Chapter 4 contains analyses of a number of illustrative problems in which we demonstrate that the quality of inferences drawn from microarray studies do not significantly degrade when the precision of the measurements is quantized to low precision. Here we provide additional computational results for cluster analysis of microarray data, which is the most widely used technique for transcriptome-based inferences. To this end we took expression profiles of published clusters of genes and re-did the clustering algorithm with continuous and quantized data. Clustering is generally not a stable procedure and results (which gene gets associated with which cluster) depend on factors such as initialisation of iterative algorithms (Torrente et al., 2005). In our computational strategy, we applied $K$-means clustering with $K$ set to the members of the published clusters and tried to match the clusters obtained by our implementation to those in the original papers. In order to achieve the most similar clustering results we forced the centres of the clusters to be the mean (or mode in the binary case) of each published cluster.

In the following tables we show the mean pairwise correlation of genes taken from within the clusters and mean pairwise cross correlation of pairs of genes taken from across clusters. Comparison is made at different level of quantization for nine different data sets, downloaded from *ArrayExpress* (`http://www.ebi.ac.uk/arrayexpress`). Fisher ratios between within cluster and across cluster pairwise correlations are given in the last row of each table.

To compare overlap between genes in a particular cluster when clustering is applied at different levels of precision, we used the $F1$ measure, used widely in information retrieval problems, defined as

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A.1}$$

Here, the term *recall*, equivalent to true positives in a classification problem, refers to the fraction of genes in the original cluster correctly identified in clusters obtained with continuous data. *Precision*, not to be confused with the numerical precision, is the true positives expressed as a fraction of the sum of true positives and false positives.

In the tables, the numbers in brackets indicates the number of genes associated with a cluster.

Genes which have more than two missing values were ignored. The missing values for genes which have one or two missing values are simply replaced by the mean of the column vector which represents the experiments (conditions).

Table A.1: Mean pairwise correlations and standard deviation for the clusters in Eisen et al. (1998). The organism on which experiments carried out on is *Saccharomyces cerevisiae*, using spotted DNA microarrays during diauxic shift, mitotic cell division cycle and sporulation.

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster B<br>(9) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.85 ± 0.04<br>-0.31 ± 0.26<br>3.87 | 0.39 ± 0.11<br>-0.09 ± 0.16<br>1.78 | 0.41 ± 0.10<br>-0.07 ± 0.14<br>2.0 |
| Cluster C<br>(23) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.70 ± 0.09<br>-0.20 ± 0.28<br>2.43 | 0.42 ± 0.14<br>-0.01 ± 0.15<br>1.48 | 0.47 ± 0.14<br>-0.03 ± 0.15<br>1.72 |
| Cluster D<br>(11) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.59 ± 0.07<br>0.17 ± 0.24<br>1.35 | 0.49 ± 0.11<br>0.09 ± 0.17<br>1.43 | 0.52 ± 0.10<br>0.05 ± 0.18<br>1.68 |
| Cluster E<br>(16) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.83 ± 0.08<br>0.26 ± 0.42<br>1.14 | 0.40 ± 0.18<br>0.12 ± 0.21<br>0.72 | 0.28 ± 0.15<br>0.06 ± 0.18<br>0.67 |
| Cluster F<br>(17) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.72 ± 0.06<br>0.24 ± 0.25<br>1.55 | 0.45 ± 0.10<br>0.14 ± 0.18<br>1.11 | 0.50 ± 0.10<br>0.13 ± 0.19<br>1.28 |
| Cluster G<br>(13) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.71 ± 0.09<br>0.21 ± 0.27<br>1.43 | 0.38 ± 0.15<br>0.14 ± 0.19<br>0.71 | 0.41 ± 0.16<br>0.12 ± 0.20<br>0.81 |
| Cluster H<br>(8) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.91 ± 0.03<br>0.10 ± 0.12<br>5.40 | 0.76 ± 0.07<br>0.15 ± 0.14<br>2.91 | 0.73 ± 0.08<br>0.12 ± 0.14<br>2.77 |
| Cluster I<br>(81) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.88 ± 0.05<br>0.03 ± 0.39<br>1.93 | 0.52 ± 0.13<br>0.07 ± 0.20<br>1.36 | 0.43 ± 0.15<br>0.04 ± 0.18<br>1.18 |
| Cluster J<br>(5) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.64 ± 0.10<br>-0.09 ± 0.18<br>2.61 | 0.41 ± 0.11<br>0 ± 0.13<br>1.71 | 0.34 ± 0.15<br>0 ± 0.13<br>1.21 |
| Cluster K<br>(13) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.66 ± 0.11<br>-0.06 ± 0.30<br>1.76 | 0.43 ± 0.14<br>0.02 ± 0.19<br>1.24 | 0.46 ± 0.13<br>0.07 ± 0.19<br>1.22 |

Table A.2: $F1$ measure, overlap results, for Eisen et al. (1998), comparing quantized clusters with continuous data. Numbers in brackets indicate the number of genes associated with the cluster.

| **Clusters** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| B (9) | 1.0 (9) | 0.95 (10) |
| C (23) | 1.0 (23) | 0.98 (22) |
| D (11) | 1.0 (11) | 0.92 (13) |
| E (16) | 1.0 (16) | 0.84 (15) |
| F (17) | 1.0 (17) | 0.94 (17) |
| G (13) | 1.0 (13) | 0.86 (15) |
| H (8) | 1.0 (8) | 1.0 (8) |
| I (81) | 1.0 (81) | 0.97 (80) |
| J (5) | 1.0 (5) | 1.0 (5) |
| K (13) | 1.0 (13) | 0.92 (11) |

Table A.3: Mean pairwise correlations and standard deviation for the clusters in Iyer et al. (1999). Human fibroblast response to serum (cDNA microarray) during growth control and cell cycle progression.

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster A (100) | Within cluster correlation | 0.57 ± 0.20 | 0.58 ± 0.21 | 0.51 ± 0.22 |
| | Cross Correlation | 0.01 ± 0.45 | 0.0 ± 0.45 | 0.0 ± 0.42 |
| | Fisher Ratio | 0.86 | 0.88 | 0.80 |
| Cluster B (142) | Within cluster correlation | 0.49 ± 0.22 | 0.42 ± 0.23 | 0.33 ± 0.25 |
| | Cross Correlation | 0.02 ± 0.42 | 0.02 ± 0.40 | 0.0 ± 0.36 |
| | Fisher Ratio | 0.73 | 0.64 | 0.54 |
| Cluster C (32) | Within cluster correlation | 0.63 ± 0.21 | 0.58 ± 0.21 | 0.55 ± 0.22 |
| | Cross Correlation | 0.04 ± 0.46 | 0.05 ± 0.45 | 0.03 ± 0.44 |
| | Fisher Ratio | 0.88 | 0.80 | 0.79 |
| Cluster D (40) | Within cluster correlation | 0.77 ± 0.22 | 0.72 ± 0.21 | 0.64 ± 0.26 |
| | Cross Correlation | 0.0 ± 0.41 | -0.01 ± 0.40 | -0.02 ± 0.39 |
| | Fisher Ratio | 1.22 | 1.20 | 1.02 |
| Cluster E (7) | Within cluster correlation | 0.69 ± 0.24 | 0.71 ± 0.22 | 0.68 ± 0.20 |
| | Cross Correlation | 0.11 ± 0.41 | 0.11 ± 0.42 | 0.09 ± 0.44 |
| | Fisher Ratio | 0.89 | 0.94 | 0.92 |
| Cluster F (31) | Within cluster correlation | 0.53 ± 0.30 | 0.50 ± 0.28 | 0.46 ± 0.30 |
| | Cross Correlation | -0.16 ± 0.38 | -0.16 ± 0.36 | -0.12 ± 0.33 |
| | Fisher Ratio | 1.01 | 1.03 | 0.92 |
| Cluster G (15) | Within cluster correlation | 0.57 ± 0.26 | 0.57 ± 0.26 | 0.56 ± 0.25 |
| | Cross Correlation | -0.12 ± 0.48 | -0.13 ± 0.48 | -0.11 ± 0.45 |
| | Fisher Ratio | 0.93 | 0.95 | 0.96 |
| Cluster H (60) | Within cluster correlation | 0.62 ± 0.24 | 0.56 ± 0.26 | 0.53 ± 0.28 |
| | Cross Correlation | -0.26 ± 0.41 | -0.24 ± 0.40 | -0.21 ± 0.39 |
| | Fisher Ratio | 1.35 | 1.21 | 1.10 |
| Cluster I (17) | Within cluster correlation | 0.40 ± 0.31 | 0.33 ± 0.31 | 0.29 ± 0.30 |
| | Cross Correlation | -0.06 ± 0.37 | -0.05 ± 0.33 | -0.05 ± 0.32 |
| | Fisher Ratio | 0.68 | 0.59 | 0.55 |
| Cluster J (18) | Within cluster correlation | 0.75 ± 0.26 | 0.65 ± 0.24 | 0.62 ± 0.23 |
| | Cross Correlation | 0.04 ± 0.35 | 0.03 ± 0.36 | 0.05 ± 0.33 |
| | Fisher Ratio | 1.16 | 1.03 | 1.02 |

Table A.4: $F1$ measure, overlap results, for Iyer et al. (1999), comparing quantized clusters with continuous data. Numbers in brackets indicate the number of genes associated with the cluster. Matlab $K$-means clustering, distance used is correlation.

| **Clusters**[1] | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| C1 (67) | 0.89 (54) | 0.94 (59) |
| C2 (41) | 0.79 (48) | 0.86 (45) |
| C3 (43) | 0.86 (43) | 0.62 (95) |
| C4 (51) | 0.59 (30) | 0.66 (95) |
| C5 (24) | 0.83 (30) | 0.83 (95) |
| C6 (30) | 0.74 (51) | 0.54 (11) |
| C7 (57) | 0.80 (38) | 0.61 (38) |
| C8 (54) | 0.87 (61) | 0.54 (20) |
| C9 (48) | 0.73 (34) | 0.63 (105) |
| C10 (47) | 0.84 (65) | 0.48 (15) |

[1] We couldn't identify the same clusters as in Iyer et al. (1999). However our results show overlap between continuous clusters and quantized clusters. We believe this is mainly because of the different clustering algorithm used and the small number of genes which are associated in some of the original clusters.

Table A.5: Mean pairwise correlations and standard deviation for the clusters in Alizadeh et al. (2000). The organism experiments carried out on is Homo sapiens by using cDNA microarray.

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster B (33) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.41 ± 0.16<br>0.09 ± 0.27<br>0.74 | 0.32 ± 0.16<br>0.02 ± 0.17<br>0.91 | 0.28 ± 0.15<br>0.07 ± 0.18<br>0.64 |
| Cluster gcm (18) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.33 ± 0.18<br>0 ± 0.20<br>0.87 | 0.14 ± 0.13<br>-0.02 ± 0.14<br>0.59 | 0.24 ± 0.16<br>0 ± 0.17<br>0.73 |
| Cluster ly (30) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.60 ± 0.11<br>0.21 ± 0.23<br>1.15 | 0.58 ± 0.11<br>0.07 ± 0.17<br>1.82 | 0.53 ± 0.11<br>0.22 ± 0.19<br>1.03 |
| Cluster panB (31) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.62 ± 0.13<br>-0.02 ± 0.23<br>1.78 | 0.48 ± 0.12<br>-0.04 ± 0.16<br>1.86 | 0.49 ± 0.13<br>0 ± 0.16<br>1.69 |
| Cluster pro (108) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.57 ± 0.13<br>0.10 ± 0.24<br>1.27 | 0.17 ± 0.15<br>0.02 ± 0.14<br>0.52 | 0.47 ± 0.13<br>0.12 ± 0.20<br>1.06 |
| Cluster t (10) | Within cluster correlation<br>Cross Correlation<br>Fisher Ratio | 0.35 ± 0.15<br>-0.06 ± 0.25<br>1.03 | 0.21 ± 0.16<br>-0.02 ± 0.17<br>0.70 | 0.19 ± 0.16<br>-0.03 ± 0.18<br>0.65 |

Table A.6: $F1$ measure, overlap results, for Alizadeh et al. (2000), comparing quantized clusters with continuous data. Numbers in brackets indicate the number of genes associated with the cluster.

| **Clusters** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| B (21) | 0.81 (31) | 0.78 (33) |
| Gcm (19) | 0.92 (20) | 0.97 (18) |
| Ly (25) | 0.88 (32) | 0.89 (31) |
| PanB (30) | 0.98 (31) | 0.98 (31) |
| Pro (111) | 0.94 (103) | 0.97 (107) |
| T (24) | 0.59 (13) | 0.59 (10) |

Table A.7: Mean pairwise correlations and standard deviation for the cluster in Causton et al. (2001). Cross correlations are calculated by selecting 22 not expressed random genes for 1000 times. Here the maximum F1 score is reported since the 22 genes used in the paper may be a subset of the whole expressed genes cluster. The organism experiment carried out on is Saccharomyces cerevisiae by using Affymetrix, during heat, acid, alkali, salt, sorbitol, diauxic shift, $H_2 O_2$.

| **Cluster** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Over expressed Yeast genes as a response to environmental changes (22) | Within cluster correlation | 0.66 ± 0.27 | 0.57 ± 0.24 | 0.54 ± 0.26 |
| | Cross Correlation | 0.06 ± 0.32 | 0.03 ± 0.32 | 0.07 ± 0.22 |
| | Fisher Ratio | 1.03 | 0.96 | 0.98 |

Table A.8: $F1$ measure, overlap results, for Causton et al. (2001). 44 genes in total were used for clustering, 22 over expressed genes and 22 randomly selected not over expressed genes. Numbers in brackets indicate the number of genes associated with the cluster. MATLAB $K$-means clustering with standard Euclidean distance is used.

| **Cluster** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| Expressed genes (37) | 0.75 (22) | 0.91 (31) |

Table A.9: Mean pairwise correlations and standard deviation for the clusters in Jones et al. (2004). Expression profiling of human breast cancer (cDNA microarray).

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster LY (45) | Within cluster correlation | 0.47 ± 0.19 | 0.41 ± 0.19 | 0.41 ± 0.20 |
| | Cross Correlation | -0.35 ± 0.18 | -0.16 ± 0.20 | -0.11 ± 0.18 |
| | Fisher Ratio | 2.22 | 1.46 | 1.37 |
| Cluster MYL (13) | Within cluster correlation | 0.77 ± 0.07 | 0.43 ± 0.14 | 0.39 ± 0.12 |
| | Cross Correlation | -0.35 ± 0.18 | -0.16 ± 0.20 | -0.11 ± 0.18 |
| | Fisher Ratio | 4.48 | 1.74 | 1.67 |

Table A.10: $F1$ measure, overlap results, for Jones et al. (2004), comparing quantized clusters with continuous data. Numbers in brackets indicate the number of genes associated with the cluster. After removing the genes which has more than two missing values, the number of genes in cluster LY reduced to 32 and the number of genes in cluster MYL reduced to 12.

| **Clusters** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| Ly (32) | 0.90(26) | 0.93 (28) |
| Myl (12) | 0.80 (18) | 0.86 (16) |

Table A.11: Mean pairwise correlations and standard deviation for the clusters in Rustici et al. (2004). Organism on which experiments carried out on is Schizosaccharomyces pombe, clustering periodic gene expressions according to their peak point.

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster 1 (66) | Within cluster correlation | 0.71 ± 0.18 | 0.61 ± 0.23 | 0.63 ± 0.20 |
| | Cross Correlation | -0.05 ± 0.56 | -0.08 ± 0.51 | -0.08 ± 0.42 |
| | Fisher Ratio | 1.03 | 0.93 | 1.15 |
| Cluster 2 (58) | Within cluster correlation | 0.77 ± 0.22 | 0.73 ± 0.29 | 0.43 ± 0.28 |
| | Cross Correlation | 0.06 ± 0.57 | 0.04 ± 0.49 | 0.02 ± 0.38 |
| | Fisher Ratio | 0.90 | 0.88 | 0.62 |
| Cluster 3 (42) | Within cluster correlation | 0.73 ± 0.20 | 0.70 ± 0.25 | 0.65 ± 0.25 |
| | Cross Correlation | 0.10 ± 0.44 | 0.08 ± 0.39 | 0.05 ± 0.35 |
| | Fisher Ratio | 0.98 | 0.97 | 1.0 |
| Cluster 4 (121) | Within cluster correlation | 0.33 ± 0.41 | 0.32 ± 0.41 | 0.27 ± 0.38 |
| | Cross Correlation | -0.34 ± 0.39 | -0.30 ± 0.37 | -0.25 ± 0.33 |
| | Fisher Ratio | 0.84 | 0.80 | 0.73 |

Table A.12: $F1$ measure, overlap results, for Rustici et al. (2004). Numbers in brackets indicate the number of genes associated with the cluster.

| **Clusters**[2] | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| C1 (36) | 0.78 (56) | 0.81 (53) |
| C2 (102) | 0.88 (85) | 0.83 (103) |
| C3 (80) | 0.97 (75) | 0.87 (63) |
| C4 (69) | 0.99(71) | 0.99 (68) |

[2] We couldn't identify the same clusters as in Rustici et al. (2004). However our results show overlap between continuous clusters and quantized clusters. We believe this is mainly because of the different clustering algorithm used and the small number of genes which are associated in some of the original clusters.

Table A.13: Mean pairwise correlations and standard deviation for the clusters in Vukkadapu et al. (2005). Genes were clusteres according to the disease progress. The organism on which experiments carried out on is Mus musculus by using Affymetrix.

| **Clusters** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Cluster 1 (70) | Within cluster correlation | 0.80 ± 0.13 | 0.74 ± 0.17 | 0.71 ± 0.22 |
| | Cross Correlation | 0.28 ± 0.45 | 0.21 ± 0.48 | 0.19 ± 0.42 |
| | Fisher Ratio | 0.90 | 0.81 | 0.81 |
| Cluster 2 (43) | Within cluster correlation | 0.77 ± 0.16 | 0.71 ±0.17 | 0.72 ± 0.26 |
| | Cross Correlation | -0.08 ± 0.40 | -0.09 ± 0.41 | -0.04 ± 0.44 |
| | Fisher Ratio | 1.52 | 1.38 | 1.09 |
| Cluster 3 (90) | Within cluster correlation | 0.86 ± 0.10 | 0.75 ± 0.17 | 0.66 ± 0.24 |
| | Cross Correlation | 0.02 ± 0.47 | -0.01 ± 0.47 | -0.02 ± 0.44 |
| | Fisher Ratio | 1.47 | 1.19 | 1.00 |
| Cluster 4 (87) | Within cluster correlation | 0.86 ± 0.14 | 0.77 ± 0.17 | 0.74 ± 0.35 |
| | Cross Correlation | -0.15 ± 0.38 | -0.17 ± 0.38 | -0.11 ± 0.42 |
| | Fisher Ratio | 1.94 | 1.71 | 1.10 |
| Cluster 5 (24) | Within cluster correlation | 0.88 ± 0.09 | 0.86 ± 0.12 | 0.88 ± 0.17 |
| | Cross Correlation | 0.36 ± 0.41 | 0.32 ± 0.45 | 0.32 ± 0.40 |
| | Fisher Ratio | 1.04 | 0.95 | 0.98 |

Table A.14: $F1$ measure, overlap results, for Vukkadapu et al. (2005), comparing quantized clusters with continuous data. Numbers in brackets indicate the number of genes associated with the cluster.

| **Clusters** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| 1 (70) | 0.95(77) | 0.93(80) |
| 2 (43) | 0.90(35) | 0.76(44) |
| 3 (79) | 0.90(66) | 0.84(83) |
| 4 (87) | 0.84(87) | 0.90(73) |
| 5 (35) | 0.83(49) | 0.99(34) |

Table A.15: Mean pairwise correlations and standard deviation for the cluster in Schonrock et al. (2006). 25 over expressed genes whose names are available in the paper are used. Cross correlations are calculated by selecting 25 different random genes for 10000 times. Here the maximum F1 score is reported since the 22 genes used in the paper may be a subset of the whole expressed genes cluster. The organism on which experiments carried out on is Arabidopsis thaliana by using Affymetrix.

| **Cluster** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Genes that are differentially expressed in CAF-1 (25) | Within group correlation | 0.57 ± 0.30 | 0.52 ± 0.30 | 0.55 ± 0.37 |
| | Cross Correlation | -0.06±0.34 | -0.04±0.31 | -0.08±0.31 |
| | Fisher Ratio | 0.98 | 0.93 | 0.92 |

Table A.16: $F1$ measure, overlap results, for Schonrock et al. (2006). 50 genes in total were used for clustering, 25 over expressed genes and 25 randomly selected not over expressed genes. Numbers in brackets indicate the number of genes associated with the cluster. Matlab $K$-means clustering with standard Euclidean as the distance is used.

| **Cluster** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| Expressed genes (21) | 0.84 (29) | 0.86 (28) |

Table A.17: Mean pairwise correlations and standard deviation for the cluster in Somers et al. (2006). 15 over expressed genes whose names are available in the paper are used. These 15 genes were cross correlated with the rest of the genes. The organism on which experiments carried out on is Bos taurus by using cDNA microarray.

| **Cluster** | **Mean pairwise correlations** | **Cont. Data** | **Three level** | **Binary Data** |
|---|---|---|---|---|
| Bos taurus, significantly differentially expressed genes between NT and IVP blastocysts (15) | Within group correlation | 0.71 ± 0.22 | 0.63 ± 0.19 | 0.62 ± 0.26 |
| | Cross Correlation | -0.10±0.50 | -0.15±0.43 | -0.03±0.44 |
| | Fisher Ratio | 1.13 | 1.26 | 0.93 |

Table A.18: $F1$ measure, overlap results, for Somers et al. (2006). 15 over expressed genes whose names are available on the paper is used against the rest og the dataset. However we should know that in the rest of the dataset there are still some genes which are over expressed. Numbers in brackets indicate the number of genes associated with the cluster. Matlab $K$-means clustering with standard Euclidean as the distance is used.

| **Cluster** | **F1 measure** | |
|---|---|---|
| | Three level | Binary |
| Expressed genes (17) | 0.87 (22) | 0.94 (19) |

## A.1 A note on the F1 Measure

In all these, by using $F1$ as measure of performance we find significant overlap between genes that are found as members of each identified cluster, even under extreme levels of quantization of the data. Where the memberships differ, we carried out a manual inspection of the source article to see if the authors make any claim about genes that were wrongly clustered, and failed to find any. We would conclude from this that the differences in clusters formed between data taken at continuous precision and data quantized to binary/tertiary precision is negligible as far as the inference drawn from them is concerned.

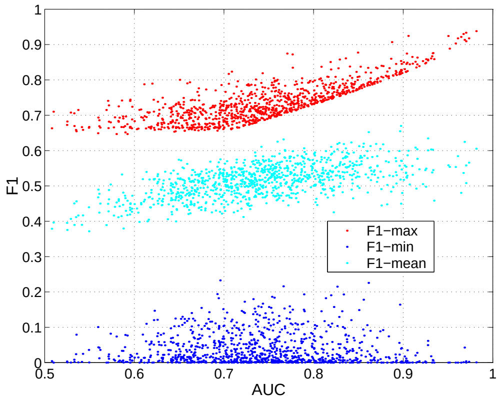


Figure A.1: Maximum, mean and minimum F1 scores against corresponding areas under the corresponding ROC curves for randomly generated one dimensional Gaussian densities.

To gain an intuitive understanding of how the $F1$ measure relates to the area under ROC curve, we simulated one dimensional random Gaussian densities of different means and standard deviations. Fig. A.1 shows a scatter plot of the maximum, minimum and average $F1$ measures obtained at given values of AUC. Unlike in the classifier designs undertaken, in the clustering setting, we have no control over changing a decision threshold to alter the balance between precision and recall.

# Appendix B

# Support Vector Machines

Given some labelled patterns

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\} \quad y_i \in \{-1, +1\} \tag{B.1}$$

Suppose that all of them satisfy the the following constraints

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \qquad \text{for} \qquad y_i = +1 \tag{B.2}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \qquad \text{for} \qquad y_i = -1 \tag{B.3}$$

They can be combined into one set of inequalities

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \qquad \forall_i \tag{B.4}$$

For a point which satisfies Eq. B.2 stands on the hyperplane $H_1 : \mathbf{w} \cdot \mathbf{x}_i + b = 1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $|1 - b|/||\mathbf{w}||$ For a point which satisfies Eq. B.3 stands on the hyperplane $H_2 : \mathbf{w} \cdot \mathbf{x}_i + b = -1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $| - 1 - b|/||\mathbf{w}||$. Since $d_+ = d_- = \frac{1}{||\mathbf{w}||}$, the margin is $\frac{|1-b|}{||\mathbf{w}||} - \frac{|-1-b|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$. We want to find the pair of hyperplanes which gives the maximum margin (maximize $\frac{1}{||\mathbf{w}||}$ with respect to Eq. B.4) by minimizing $||\mathbf{w}||$. This can be done with Lagrange multipliers and for computational ease $||\mathbf{w}||$ is replaced by $\frac{1}{2}||\mathbf{w}||^2$ for quadratic programming optimization. To solve this minimization we construct the Langrangian $L$ and the Langrange multipliers $\lambda_i$:

$$L(\mathbf{w}, b, \lambda) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \lambda_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{B.5}$$

The solution to this optimization problem can be found by taking the derivative of $L$ with respect to $\mathbf{w}$ and $b$:

$$\frac{\partial L(\mathbf{w}, b, \lambda)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0 \tag{B.6}$$

$$\frac{\partial L(\mathbf{w}, b, \lambda)}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{B.7}$$

By re substituting the relations obtained

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \tag{B.8}$$

$$0 = \sum_{i=1}^{n} \lambda_i y_i \tag{B.9}$$

into the primal form, we obtain

$$L_d(\mathbf{w}, b, \lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{B.10}$$

So the dual optimization problem is :

$$\text{Maximize} \qquad \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{B.11}$$

$$\text{Subject to} \qquad \sum_{i=1}^{n} \lambda_i y_i = 0 \quad \text{and} \quad \lambda_i \geq 0 \tag{B.12}$$

The classification function can be written as:

$$f(x) = \operatorname{sign}\left( \sum_{i=1}^{n} \lambda_i (\mathbf{x}_i \cdot \mathbf{x}_i) + b \right) \tag{B.13}$$

The optimization problem can be solved Once the multipliers $\lambda$ have been found, the optimal hyperplane is easy to get, $\mathbf{w}_{\text{opt}}$ and $b_{\text{opt}}$ can be derived as:

$$\mathbf{w}_{\text{opt}} = \sum_{SV} \lambda_i y_i \mathbf{x}_i \tag{B.14}$$

$$b_{\text{opt}} = 1 - y_i \mathbf{w}_{\text{opt}} \cdot \mathbf{x}_i^{SV} \tag{B.15}$$

where $\mathbf{x}_i^{SV}$ denotes support vectors. Support vectors are those points of sample which satisfy $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0$.

Further discussions on SVM can be found in Burges (1998); Cristianini and Shawe-Taylor (2000); Scholkopf and Smola (2002); Shawe-Taylor and Cristianini (2004).

## B.1 Soft-margin SVM

Soft-margin SVM can be expressed as:

$$\text{Minimize} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \epsilon_i \tag{B.16}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \epsilon_i; \quad \epsilon_i \geq 0; \quad i = 1, \cdots n; \tag{B.17}$$

$$L(\mathbf{w}, b, \epsilon, \lambda, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + C\sum \epsilon_i - \sum_{i=1}^{n} \lambda[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \epsilon_i] - \sum \alpha\epsilon_i \tag{B.18}$$

The solution to this optimization problem can be found by taking the derivative of $L$ with respect to $\mathbf{w}$, $b$ and $\epsilon$:

$$\frac{\partial L(\mathbf{w}, b, \epsilon, \lambda, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0 \tag{B.19}$$

$$\frac{\partial L(\mathbf{w}, b, \epsilon, \lambda, \alpha)}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{B.20}$$

$$\frac{\partial L(\mathbf{w}, b, \epsilon, \lambda, \alpha)}{\partial \epsilon} = C - \lambda_i - \alpha_i = 0 \tag{B.21}$$

$$C \geq \lambda_i \geq 0 \tag{B.22}$$

Incorporating a kernel and rewriting it in terms of Lagrange multipliers, this again leads to the problem of maximizing Eq. B.11, subject to the constraints:

$$0 \leq \lambda_i \leq C \quad \text{and} \quad \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{B.23}$$

By setting $C$ as very big number we get the same constraint as in Eq. B.12 which implies a very large of $C$ is equivalent to applying hard-margin SVM.

# Appendix C

# Tanimoto Kernel

In this section we will show that Tanimoto kernel is a valid kernel. A valid kernel must be a positive semi-definite matrix (Shawe-Taylor and Cristianini, 2004). Positive semi-definite (PSD) matrix have eigenvalues which are $\geq 0$ and $\mathbf{x}^T A\mathbf{x} \geq 0$ holds for any non-zero $\mathbf{x}$. We will start with general theorems and proofs. At the end we will relate these to Tanimoto kernel.

## C.1 Rayleigh quotient

The Rayleigh quotient corresponding to a symmetric matrix $\mathbf{A}$ is the expression:

$$\rho = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{C.1}$$

**Theorem C.1.** *If $\boldsymbol{A}$ is symmetric with eigenvalues $\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n = \lambda_{\max}$, then*

$$\lambda_{\min} \leq \rho \leq \lambda_{\max}$$

*where $\rho$ is the Rayleigh quotient for any $\boldsymbol{x} \neq 0$, and*

$$\lambda_{\min} = \min_{x\neq 0} \frac{\boldsymbol{x}^T A\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} \qquad \lambda_{\max} = \max_{x\neq 0} \frac{\boldsymbol{x}^T A\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} \tag{C.2}$$

*Proof.* If $\mathbf{A}$ is symmetric, an orthonormal set of eigenvectors exist, $\mathbf{x}_1, \cdots, \mathbf{x}_n$ where $\mathbf{x}_i$ corresponds to $\lambda_i$. Suppose that the expansion of an arbitrary vector in terms of the $\mathbf{x}_i$ is

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \tag{C.3}$$

then

$$\mathbf{Ax} = \sum_{i=1}^{n} \alpha_i \lambda_i x_i \tag{C.4}$$

$$\rho = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\lambda_1 \alpha_1^2 x_1^2 + \lambda_2 \alpha_2^2 x_2^2 + \cdots + \lambda_n \alpha_n^2 x_n^2}{\alpha_1^2 x_1^2 + \alpha_2^2 x_2^2 + \cdots + \alpha_n^2 x_n^2} \tag{C.5}$$

$$\rho - \lambda_{\min} = \frac{(\lambda_2 - \lambda_{\min}) \alpha_2^2 x_2^2 + \cdots + (\lambda_n - \lambda_{\min}) \alpha_n^2 x_n^2}{\alpha_1^2 x_1^2 + \alpha_2^2 x_2^2 + \cdots + \alpha_n^2 x_n^2} \tag{C.6}$$

Since $\lambda_i \geq \lambda_{\min}$ for all $i$ and all $\alpha_i^2 x_i^2 \geq 0$, we have $\rho \geq \lambda_{\min}$. □

$\lambda_{\max}$ is similarly proved by considering $\rho - \lambda_n$. Proof is taken from Noble (1969).

If $\mathbf{x}^T A \mathbf{x} \geq 0$ than the eigenvalues of $A$ will also be positive or zero. This can be shown with the inner product space which will be described next.

## C.2 Inner product space

A vector space $\mathbf{X}$ over the real $\mathbb{R}$ is an inner product space if it satisfies:

$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0 \tag{C.7}$$

Inner product $\langle \mathbf{x}, \mathbf{z} \rangle$ is defined as:

$$\langle e_i, e_j \rangle = \delta_{ij} \tag{C.8}$$

where

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{C.9}$$

$$x = \sum x_i e_i \tag{C.10}$$

$$z = \sum z_i e_i \tag{C.11}$$

$e_i$ are orthonormal basis

$$\langle \mathbf{x}, \mathbf{z} \rangle = \mathbf{x}^T \mathbf{z} = \sum_{i=1}^{n} x_i z_i \tag{C.12}$$

***Proposition*** 1. Kernel matrices are positive semi-definite. A matrix $\mathbf{K}$ is positive semi-definite if $\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0$ for all non-zero $\mathbf{v} \in \mathbb{R}$.

*Proof.* Consider the general case of a kernel matrix let

$$K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle, \quad \text{for} \quad i, j = 1, \cdots, n. \tag{C.13}$$

For any vector $\mathbf{v}$ we have:

$$\mathbf{v}^T \mathbf{K} \mathbf{v} = \sum_{i,j=1}^{l} v_i v_j K_{ij} = \sum_{i,j=1}^{l} v_i v_j \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle \tag{C.14}$$

$$= \left\langle \sum_{i,j}^{l} v_i \phi(\mathbf{x}_i), \sum_{i,j}^{l} v_j \phi(\mathbf{x}_j) \right\rangle \tag{C.15}$$

$$= \left\| \sum_{i=1}^{l} v_i \phi(\mathbf{x}_i) \right\|^2 \geq 0 \tag{C.16}$$

as required. □

## C.3 Tanimoto kernel

Since Tanimoto kernel is used for binary vectors, each coefficient $a$, $b$ and $c$ can be written as inner products:

$$a = \sum_{i=1}^{m} x_i \quad \text{or} \quad a = \mathbf{x}^T \mathbf{x} \tag{C.17}$$

and the same applies to the calculation of $b$ and $c$ in Eq. 5.1:

$$b = \sum_{i=1}^{m} z_i \quad \text{or} \quad b = \mathbf{z}^T \mathbf{z} \tag{C.18}$$

$$c = \sum_{i=1}^{m} x_i \cdot z_i \quad \text{or} \qquad c = \mathbf{x}^T \mathbf{z} \tag{C.19}$$

$$K_{Tan}(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}^T \mathbf{z}}{\mathbf{x}^T \mathbf{x} + \mathbf{z}^T \mathbf{z} - \mathbf{x}^T \mathbf{z}} \tag{C.20}$$

Tanimoto coefficient between two objects are represented as a combination of inner product spaces i.e., each are valid kernel on its own.

***Proposition*** 2. Let $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ be valid kernels. Then

$$k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$$

and

$$k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) \cdot k_2(\mathbf{x}, \mathbf{z})$$

and

$$k(\mathbf{x}, \mathbf{z}) = a \cdot k_1(\mathbf{x}, \mathbf{z})$$

and

if $k_1$, $k_2$, $\cdots$ are kernels, and $k(x, z) := \lim_{m \to \infty} k_m(x, z)$

are also valid kernels (Shawe-Taylor and Cristianini, 2004; Scholkopf and Smola, 2002).

By following Gower (1971), We show that Tanimoto kernel is a PSD function[1].

*Proof.* Based on Eq. C.20, Tanimoto kernel is written as:

$$K_{Tan} = \frac{c}{a + b - c} \tag{C.21}$$

where $c = \langle \mathbf{x}, \mathbf{z} \rangle$, $a = \langle \mathbf{x}, \mathbf{x} \rangle$ and $b = \langle \mathbf{z}, \mathbf{z} \rangle$. The denominator of Eq. C.21 can be written:

$$m - d = a + b - c \tag{C.22}$$

where $m$ is a constant and is the length of the feature string. $d$ is the number of common bits that are zeros and is written as $d = \langle 1 - \mathbf{x}, 1 - \mathbf{z} \rangle$ (1 is the vector of ones).

Now, Tanimoto kernel matrix can be written as:

$$K_{Tan} = c \cdot \frac{1}{m - d} \tag{C.23}$$

$$\tag{C.24}$$

[1] proof is provided by Sandor Szedmak.

$c$ is already PSD. We must show that $\frac{1}{m-d}$ is also PSD.

Following the definition of geometric series:

$$\frac{1}{m-d} = \frac{m}{1-\frac{d}{m}} = m + \frac{md}{m} + \frac{md^2}{m^2} + \cdots \tag{C.25}$$

$$= m\left(1 + \frac{d}{m} + \frac{d^2}{m^2} + \frac{d^3}{m^3} + \cdots\right) \tag{C.26}$$

where $\frac{d^2}{m^2} = \frac{d}{m} \cdot \frac{d}{m}$. Now we must show that $\frac{d}{m}$ is PSD:

$$\frac{d}{m} = \frac{\langle 1-\mathbf{x}, 1-\mathbf{z}\rangle}{m} \geq 0 \tag{C.27}$$

□

# Appendix D

# Pre-processing algorithms in `expresso`

In this appendix we give the basic descriptions of the pre-processing algorithms, mainly the ones used in `expresso`. These methods are also described in the technical report of Bolstad (2004). We start with the background correction algorithms.

## D.1 Background correction

After the image processing, the observed intensities contain noise. The observed intensities need to be adjusted to give accurate measurements. This step is known as the background correction (Gautier et al., 2004). The available algorithms in `expresso` for background correction

- **mas**: Background correction method as described in the Statistical Algorithms Description of Affymetrix (Affymetrix, 2002). For mas, array is split up into 16 rectangular zones. For each zone, the lowest 2% of probe intensities are used to calculate the background for that zone. Each probe is then adjusted based upon a weight average of the backgrounds for each region. The weights are based on the distances between the location of the probe and the centroids of 16 different regions. This method corrects both PM and MM probes.
- **rma**: This background correction method is introduced by Irizarry et al. (2003c) for correcting the PM probe intensities. PM intensities are considered as: $PM_i = bg_i + s_i$ where $bg_i$ represents background signal caused by optical noise and cross-hybridization in array $i$ and $s_i$ represents the signal in the same array. Considered background correction is $B(PM_i) \equiv E(s_i|PM_i)$ where $s_i$ is strictly imposed a positive distribution so that $B(PM_i) > 0$. The algorithm further assumes that $s_i$ is exponential and $bg_i$ is normal.

- **none**: This method returns the object unchanged.

## D.2 Normalization

Normalization remove the effects of different arrays and is an essential step to make comparison between arrays. Different normalization methods have been reviewed in Bolstad et al. (2003). Available algorithms in `expresso` for normalization are:

- **constant**: This is a scaling normalization where all arrays are scaled in a way that each has the same mean value. This is done by dividing each sample by a scaling factor. Scaling factor is obtained as the ratio between target value and the sample mean.
- **invariantset**: An implementation of the normalization used in the dChip software (Li and Wong, 2001b). Array images have different overall image brightness (intensities) (Li and Wong, 2001b). A group of arrays are normalized to a common baseline array. Baseline array is the one which has the median overall brightness. Invariantset normalization base this procedure only on probe values that are non-differentially expressed genes and the overall procedure is an iterative one to identify set of probes (invariant set). This procedure results in all arrays having similar brightness (intensities).
- **loess**: This is an extension of the Lowess normalization, widely applied to cDNA. Instead of comparing two colours as in two channel cDNA, it compares pairwise arrays and since it is pairwise comparison and it is a time consuming algorithm. This approach is based on $M$ vs. $A$ plot where $M = \log_2(x_i/x_j)$ and $A = \frac{1}{2}\log_2(x_i x_j)$ where $i$ and $j$ are the two arrays. After $M$ vs. $A$ is plotted a normalization curve $\hat{M}$ is fitted by loess, local regression method. The normalization adjustment is $M' = M - \hat{M}$ and the adjusted probe intensities are calculated as: $x'_i = 2^{A+\frac{\hat{M}}{2}}$ and $x'_j = 2^{A-\frac{\hat{M}}{2}}$. Local regression method, loess, uses a weight function that emphasise the distance of the points i.e., points closer to the point of interest have more weight (Cleveland and Devlin, 1988). The distance measure used is Euclidean distance and the weight function is:

  $$w(u) = \begin{cases} 1 - (|u^3|)^3, & |u| < 1 \\ 0, & |u| > 1 \end{cases} \tag{D.1}$$

  where $u$ is the Euclidean distance.
- **contrasts**: is also an extension of the $M$ vs. $A$ introduced by Astrand (2003). contrast first take the log of the data and then transform the basis. In the transformed basis a series of $n-1$ normalization curves are fit to $M$ vs. $A$ plot as

in loess. The data is adjusted with a smooth transformation which adjusts the normalization curve in such a way that it lies horizontally. Data is obtained by transforming back to the original basis and exponentiating.

- **qspline**: uses spline approximation, where each interval is chosen by quantiles, to normalize the arrays (Workman et al., 2002). Quantiles are the points taken at regular intervals from the sorted data in increasing order. For this purpose arrays are compared with the target array which is the geometric mean of all arrays.

- **quantiles**: The main aim of this algorithm is to make the distribution of the probe intensities for each array in a set of arrays the same and it is introduced by Bolstad et al. (2003). If quantile-quantile (qq) plot is a straight diagonal line then the distribution of the two data vectors (array in this case) are the same.

  1. Given $n$ array of length $p$, form $\mathbf{X}$ of dimension $p \times n$ where each array is a column,
  2. sort each column of $\mathbf{X}$ to get $\mathbf{X}_{sort}$,
  3. take the means across rows of $\mathbf{X}_{sort}$ and assign this mean to each element in the row to get $\mathbf{X}'_{sort}$,
  4. get $\mathbf{X}_{normalized}$ by rearranging each column of $\mathbf{X}'_{sort}$ to have the same ordering as original $X$.

  Quantile normalization is a specific case of transformation $x'_i = F^{-1}(G(x_i))$ where $G$ is the estimate of the empirical distribution of each array and $F$ is the estimation of the empirical distribution of the averaged sample.

- **quantiles.robust**: The only difference between quantile and quantile.robust is that, quantile robust allows the user to exclude $G$ in the above calculations (Bolstad, 2004).

## D.3 Probe specific correction

Affymetrix GeneChip technology includes mismatch probes to quantify non-specific or cross hybridization (Gautier et al., 2004). Probe specific correction is used to correct the non-specific or cross hybridization. Available algorithms in `expresso` for probe-specific correction are:

- **mas**: This is achieved by subtracting an Ideal Mismatch (IM) from Perfect Match (PM). IM is defined in a way that to prevent MisMatch (MM) to be smaller than PM. An *ideal mismatch* is defined as (Affymetrix, 2002):

$$IM_{i,j} = \begin{cases} MM_{i,j}, & MM_{i,j} < PM_{i,j} \\ \dfrac{PM_{i,j}}{2^{SB_i}} & MM_{i,j} \geq PM_{i,j} \quad \text{and} \quad SB_i > \text{contrast}\tau \\ \dfrac{PM_{i,j}}{2^{\left(\frac{\text{contrast}\tau}{1+\left(\frac{\text{contrast}\tau - SB_i}{\text{scale}\tau}\right)}\right)}} & MM_{i,j} \geq PM_{i,j} \quad \text{and} \quad SB_i \leq \text{contrast}\tau \end{cases} \tag{D.2}$$

where biweight specific background $SB_i$ for probe pair $j$ in probe set $i$ is defined as $SB_i = T_{bi}(\log_2(PM_{i,j}) - \log_2(MM_{i,j})),\ j = 1, \ldots, n_i$ and contrast$\tau$ and scale$\tau$ are set as 0.03 and 10 respectively by Affymetrix (2002). For the definition of $T_{bi}$ see *mas* of Summary method (section D.4.)

As defined by Affymetrix (2002), first one is the best case where mismatch value provides a probe-specific estimate. In the second case the estimate is not probe specific but it provides information specific to the probe set. The third case involves the least informative estimate, based only weakly on probe-set specific data.

- **pmonly**: Make no adjustment to the PM values.
- **subtractmm**: It subtracts MM from the corresponding PM. However, when PM is less than MM it returns negative values and this method can not deal with negative values. When there is negative value it outputs 'not applicable' (NA) for the probe set. This is the main drawback with this method.

## D.4 Summary method

Each gene is represented by one or more probe sets on GeneChip and summary method reports expression value for a gene for the corresponding probe-level data. Available algorithms in expresso for summary method are:

- **avgdiff**: Simply computes the average of PM and MM differences. For each probe set $n$ on each array $i$ AvgDiff is defined as:

$$AvgDiff = \frac{1}{\#A}\sum_{j \in A}(PM_j - MM_j) \tag{D.3}$$

where $A$ is a subset of probes.

- **liwong**: This is an implementation of the algorithm proposed in Li and Wong (2001a) in expresso. The model is defined as $y_{ij} = PM_{ij} - MM_{ij} = \theta_i\phi_j + \epsilon_{ij}$.

where $i$ and $j$ represent the array and probe pair respectively. $\phi_i$ is probe response parameter, $\theta_j$ is the expression on array $j$ and $\epsilon$ is random error. This model can be summarized as $y_{ij}$ = (probe effect × chip effect) + error.

- **mas**: This is one-step Tukey' s biweight algorithm as described in Hubbell et al. (2002). This algorithm is used to determine a robust average which is unaffected by outliers. First the median is defined as the center of the data and then the absolute distance for each data point to median is calculated. According to the distance of each point to the median it is decided how much each value should contribute to the average. For each data point $i$, a uniform measure of distance from the center is defined as

$$u_i = \frac{x_i - M}{5S + \epsilon} \qquad i = 1, \ldots, n \tag{D.4}$$

$M$ represents the median and $S$ is the median of the absolute distances from $M$. $\epsilon$ is a very small value (0.0001) to prevent division by zero. Following this, weights are calculated as:

$$w(u) = \begin{cases} (1-u^2)^2, & |u| \leq 1 \\ 0, & |u| > 1 \end{cases} \tag{D.5}$$

The corrected values are calculated as below:

$$T_{bi} = \frac{\sum_{i=1}^{n} w(u)x_i}{\sum_{i=1}^{n} w(u)} \tag{D.6}$$

- **medianpolish**: This implementation is introduced by Holder et al. (2001) and is a similar approach taken by Li and Wong (2001a). The main difference between the two is, medianpolish uses $y_{ij}$ = probe effect + chip effect + error. To be more precise, for probe set $k$ ($i = 1, \ldots, I_k$) and data from $j$ arrays ($j = 1, \ldots, J$) the following model is fitted : $\log_2(PM_{ij}^k) = \alpha_i^k + \beta_j^k + \epsilon_{ij}^k$. Where $\alpha_i$ is a probe effect, $\beta_j$ is the $\log_2$ of the expression value and $\epsilon_{ij}$ is the random error. The expression values obtained using this algorithm are in $\log_2$ scale.

- **playerout**: This algorithm is developed by Lazaridis et al. (2002) and it is a non-parametric approach used to determine weights. Oligos are considered as players and the performance of each oligo depends on how well that oligo estimate the expression value of a gene. $y_{ij}$ representing the average intensity of each oligo, the estimated expression value for sample $\theta_j$ is given by $\phi_i y_{ij}$ and $\phi$ here corresponds to the weight. The main aim here is to calculate the set of parameters $\phi_i$ such that a sum of squares loss over oligo and microarray instances is minimized. The expression value is then calculated as the weighted average.

# Bibliography

Affymetrix. Statistical Algorithms Description Document. *Technical report*, 2002.

S. E. Ahnert, K. Willbrand, F. C. S. Brown, and T. M. A. Fink. Unbiased pattern detection in microarray data series. *Bioinformatics*, 22(12):1471–1476, 2006.

T. Akutsu and S. Miyano. Selecting informative genes for cancer classification using gene expression data. *Computational and Statistical Approaches to Genomics (Wei Zhang, Ilya Shmulevich (ed.))*, 2006.

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. Hudson Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403: 503–511, 2000.

D. B. Allison, X. Cui, G. P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nat Rev Genet*, 7(1):55, 2006.

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96(12):6745–6750, 1999.

O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, 97(18):10101–10106, 2000.

M. Astrand. Contrast Normalization of Oligonucleotide Arrays. *J Comput Biol*, 10(1): 95–102, 2003.

A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

Y. Barash, E. Dehan, M. Krupsky, W. Franklin, M. Geraci, N. Friedman, and N. Kaminski. Comparative analysis of algorithms for signal quantitation from oligonucleotide microarrays. *Bioinformatics*, 20(6):839–846, 2004.

K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano. Reverse engineering of regulatory networks in human B cells. *Nat Genet*, 37(4):382–390, 2005.

D. Bhojwani, H. Kang, N. P. Moskowitz, D.-J. Min, H. Lee, J. W. Potter, G. Davidson, C. L. Willman, M. J. Borowitz, I. Belitskaya-Levy, S. P. Hunger, E. A. Raetz, and W. L. Carroll. Biologic pathways associated with relapse in childhood acute lymphoblastic leukemia: a Children's Oncology Group study. *Blood*, 108(2):711–717, 2006.

S. Bilke, T. Breslin, and M. Sigvardsson. Probabilistic estimation of microarray data reliability and underlying gene expression. *BMC Bioinformatics*, 4(1):40, 2003.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, USA, 2006.

B. J. Boersma, M. Reimers, M. Yi, J. A. Ludwig, B. T. Luke, R. M. Stephens, H. G. Yfantis, D. H. Lee, J. N. Weinstein, and S. Ambs. A stromal gene signature associated with inflammatory breast cancer. *Int J Cancer*, 122(6):1324–1332, 2007.

B. Bolstad. affy: Built-in Processing Methods, 2004.

B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res*, 8(11):1202–1215, 1998.

A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Letters*, 480(1):17–24, 2000.

M. L. Brewer. Development of a Spectral Clustering Method for the Analysis of Molecular Data Sets. *J Chem Inf Model*, 47(5):1727–1733, 2007.

C. S. Brown, P. C. Goodwin, and P. K. Sorger. Image metrics in the statistical analysis of DNA microarray data. *PNAS*, 98:16, 2001.

M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, Jr. Ares, M., and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS*, 97(1):262–267, 2000.

T. A. Brown. *Genomes*. Bios Scientific Publishers, Oxford-UK, 1999. ISBN 1 85996 201 7.

W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. In *UAI*. AUAI Press Arlington, Virginia, United States, 2004.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

H. C. Causton, J. Quackenbush, and A. Brazma. *Microarray gene expression analysis: a beginner's guide*. Blackwell, 108 Cowley road, Oxford, OX4 1JF, UK, 2004.

H. C. Causton, B. Ren, S. S. Koh, C. T. Harbison, E. Kanin, E. G. Jennings, T. I. Lee, H. L. True, E. S. Lander, and R. A. Young. Remodeling of Yeast Genome Expression in Response to Environmental Changes. *Mol Biol Cell*, 12(2):323–337, 2001.

S. Choe, M. Boutros, A. Michelson, G. Church, and M. Halfon. Preferred analysis methods for affymetrix genechips revealed by a wholly defined control dataset. *Genome Biol*, 6(2):R16, 2005. ISSN 1465-6906.

T. H. Chung, M. Brun, and S. Kim. Quantization of global gene expression data. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 187–192. IEEE Computer Society Washington, DC, USA, 2006.

W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *J Am Stat Assoc*, 83(403):596–610, 1988.

S. Cooper. Is whole-culture synchronization biology's perpetual-motion machine? *Trends Biotechnol*, 22(6):266–269, 2004.

S. Cooper and K. Shedden. Microarray analysis of gene expression during the cell cycle. *Cell Chromosome*, 2(1):1, 2003.

L. M. Cope, R. A. Irizarry, H. A. Jaffee, Z. Wu, and T. P. Speed. A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20(3):323–331, 2004.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

F. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

N. Cristianini and J. Shawe-Taylor. *An Introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK, 2000. ISBN 0 521 78010 5.

T. Czechowski, R. P. Bari, M. Stitt, W. R. Scheible, and M. K. Udvardi. Real-time RT-PCR profiling of over 1400 Arabidopsis transcription factors: unprecedented sensitivity reveals novel root-and shoot-specific genes. *Plant J*, 38(2):366–379, 2004.

U. de Lichtenberg, R. Wernersson, T. S. Jensen, H. B. Nielsen, A. Fausboll, P. Schmidt, F. B. Hansen, S. Knudsen, and S. Brunak. New weakly expressed cell cycle-regulated genes in yeast. *Yeast*, 22(15):1191–1201, 2005.

J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.

M. Dettling. BagBoosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004.

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(6849):822–826, 2001.

B. Di Camillo, F. Sanchez-Cabo, G. Toffolo, S. Nair, Z. Trajanoski, and C. Cobelli. A quantization method based on threshold optimization for microarray short time series. *BMC Bioinformatics*, 6(Suppl 4):S11, 2005.

J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pages 194–202, 1995.

S. Draghici, P. Khatri, A.C. Eklund, and Z. Szallasi. Reliability and reproducibility issues in DNA microarray measurements. *Trends Genet*, 22(2):101–109, 2006.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, USA, 2001. ISBN 0-41-05669-3.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyse expression data. *J Comput Biol*, 7:601–620, 2000.

G. N. Fuller, C. Mircean, I. Tabus, E. Taylor, R. Sawaya, J. M. Bruner, I. Shmulevich, and W. Zhang. Molecular voting for glioma classification reflecting heterogeneity in the continuum of cancer progression. *Oncol Rep*, 14(3):651, 2005.

L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry. affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315, 2004.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, USA, 1989. ISBN 0-8018-3772-3.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531–537, 1999.

G. J. Gordon, R. V. Jensen, L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno. Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Res*, 62(17):4963–4967, 2002.

J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Scholkopf. Covariate shift by kernel mean matching. *Springer: Dataset shift in machine learning, Quionero-Candela, J. and Sugiyama, M. and Schwaighofer, A. and Lawrence, N.D. (ed.)*, pages 131–160, 2009.

S. Gruvberger, M. Ringnér, Y. Chen, S. Panavally, L. H. Saal, A. Borg, M. Ferno, C. Peterson, and P.S. Meltzer. Estrogen Receptor Status in Breast Cancer Is Associated with Remarkably Distinct Gene Expression Patterns. *Cancer Res*, 61(16):5979–5984, 2001.

S. R. Gunn. Support vector machines for classification and regression. Technical report, University of Southampton, 1998.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

D. J. Hand and R. J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

D. J. Higham, G. Kalna, and M. Kibble. Spectral clustering and its use in bioinformatics. *J Comput Appl Math*, 204(1):25–37, 2007. ISSN 0377-0427.

D. Holder, R.F. Raubertas, and V.B. Pikounis. Statistical analysis of high density oligonucleotide arrays: A safer approach. In *Proceedings of the Annual Meeting of the American Statistical Association*, 2001.

J. D. Holliday, N. Salim, M. Whittle, and P. Willett. Analysis and display of the size dependence of chemical similarity coefficients. *J Chem Inf Comput Sci*, 43(3):819 – 828, 2003.

A. J. Holloway, R. K. van Laar, R. W. Tothill, D. D. L. Bowtell, et al. Options availablefrom start to finishfor obtaining data from DNA microarrays II. *Nature Genetics*, 32(supp):481–489, 2002.

S. D. Hooper, S. Boue, R. Krause, L. J. Jensen, C. E. Mason, M. Ghanim, K. P. White, E. E. M. Furlong, and P. Bork. Identification of tightly regulated groups of genes during drosophila melanogaster embryogenesis. *Mol Syst Biol*, 3, January 2007.

E. Huang, S. H. Cheng, H. Dressman, J. Pittman, M. Tsou, C. Horng, A. Bild, E. S. Iversen, M. Liao, C. Chen, M. West, J. R. Nevins, and A. T. Huang. Gene expression predictors of breast cancer outcomes. *Lancet*, 361:1590–1596, 2003.

S. Huang. Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *J Mol Med*, 77(6):469–480, 1999.

E. Hubbell, W. M. Liu, and R. Mei. Robust estimators for expression analysis. *Bioinformatics*, 18(12):1585–1592, 2002.

D. A. Hume. Probability in transcriptional regulation and its implications for leukocyte differentiation and inducible gene expression. *Blood*, 96(7):2323, 2000.

R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*, 31(4):e15, 2003a.

R. A. Irizarry, L. Gautier, and L. Cope. An R package for analyses of Affymetrix oligonucleotide arrays. *Springer: The analysis of gene expression data: methods and software(Parmigiani, G. and Garrett, E. S. and Irizarry, R. A. and Zeger, S. L. (ed.))*, pages 102–119, 2003b.

R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003c.

R. A. Irizarry, Z. Wu, and H. A. Jaffee. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, 22(7):789–794, 2006.

V. Iyer and K. Struhl. Absolute mRNA levels and transcriptional initiation rates in Saccharomyces cerevisiae. *PNAS*, 93(11):5208–5212, 1996.

V. R. Iyer, M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. F. Lee, J. M. Trent, L. M. Staudt, J. Jr. Hudson, M. S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P. O. Brown. The Transcriptional Program in the Response of Human Fibroblasts to Serum. *Science*, 283(5398):83–87, 1999.

A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel. Fully automatic quantification of microarray image data, 2002.

T. Joachims. Making large-scale svm learning practical. advances in kernel methods - support vector learning. *MIT-Press (B. Schlkopf and C. Burges and A. Smola (ed.))*, 1999.

C. Jones, A. Mackay, A. Grigoriadis, A. Cossu, J. S. Reis-Filho, L. Fulford, T. Dexter, S. Davies, K. Bulmer, E. Ford, S. Parry, M. Budroni, G. Palmieri, A. M. Neville, M. J. O'Hare, and S. R. Lakhani. Expression Profiling of Purified Normal Human Luminal and Myoepithelial Breast Cells: Identification of Novel Prognostic Markers for Breast Cancer. *Cancer Res*, 64(9):3037–3045, 2004.

C. Kendziorski, R. A. Irizarry, K. S. Chen, J. D. Haag, and M. N. Gould. On the utility of pooling biological samples in microarray experiments. *PNAS*, 102(12):4252, 2005.

K. Kim, D. H. Ki, H. Jeung, H. C. Chung, and S. Y. Rha. Improving the prediction accuracy in classification using the combined data sets by ranks of gene expressions. *BMC Bioinformatics*, 9(283), 2008.

S. Kim, E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, and J. M. Trent. General nonlinear framework for the analysis of gene interaction via multivariate expression arrays. *J Biomed Opt*, 5:411, 2000a.

S. Kim, E. R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J.M. Trent, and M. Bittner. Multivariate Measurement of Gene Expression Relationships. *Genomics*, 67(2):201–209, 2000b.

S. Kim, E. R. Dougherty, J. Whitmore, E. Suh, and M. Bittner. Cellular contexts from gene expression profile. In *IEEE International Workshop on Genomic Signal Processing and Statistics, Newport, RI*, 2005.

V. A. Kuznetsov, G. D. Knott, and R. F. Bonner. General statistics of stochastic process of gene expression in eukaryotic cells. *Genetics*, 161(3):1321–1332, 2002.

M. T. Landi, T. Dracheva, M. Rotunno, J. D. Figueroa, H. Liu, A. Dasgupta, F. E. Mann, J. Fukuoka, M. Hames, A. W. Bergen, et al. Gene Expression Signature of Cigarette Smoking and Its Role in Lung Adenocarcinoma Development and Survival. *PLoS ONE*, 3(2), 2008.

E. N. Lazaridis, D. Sinibaldi, G. Bloom, S. Mane, and R. Jove. A simple method to improve probe set estimates from oligonucleotide arrays. *Math Biosci*, 176(1):53–58, 2002.

M. L. T. Lee, F. C. Kuo, G. A. Whitmore, and J. Sklar. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *PNAS*, 97(18):9834–9839, 2000.

J. M. Levsky, S. M. Shenoy, R. C. Pezo, and R. H. Singer. Single-Cell Gene Expression Profiling. *Science*, 297(5582):836, 2002.

J. M. Levsky and R. H. Singer. Gene expression and the myth of the average cell. *Trends Cell Biol*, 13(1):4–6, 2003.

C. Li and W. H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *PNAS*, 98(1):31–36, 2001a.

C. Li and W.H. Wong. Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol*, 2(8):0032–1, 2001b.

W. Liu, R. Mei, X. Di, TB Ryder, E. Hubbell, S. Dee, TA Webster, CA Harrington, M. Ho, J. Baid, et al. Analysis of high density expression microarrays with signed-rank call algorithms. *Bioinformatics*, 18(12):1593–1599, 2002.

X. Liu. *Microarray Data Analysis using Probabilistic Methods.* PhD thesis, University of Manchester, 2006.

X. Liu, K. Lin, B. Andersen, and M. Rattray. Including probe-level uncertainty in model-based gene expression clustering. *BMC Bioinformatics*, 8(1):98, 2007. ISSN 1471-2105.

X. Liu, M. Milo, N. D. Lawrence, and M. Rattray. A tractable probabilistic model for Affymetrix probe-level analysis across multiple chips. *Bioinformatics*, 21(18):3637–3644, 2005.

X. Liu, M. Milo, N. D. Lawrence, and M. Rattray. Probe-level measurement error improves accuracy in detecting differential gene expression. *Bioinformatics*, 22(17): 2107–2113, 2006.

S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Norton, et al. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol*, 14(13):1675–1680, 1996.

D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and DNA arrays. *Nature*, 405(6788):827–836, June 2000. ISSN 0028-0836.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, pages 281–297. Univ. of Calif. Press, 1967.

MAQC. The MicroArray Quality Control (MAQC) project shows inter-and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol*, 24:1151–1161, 2006.

J. H. Martens, J. Kzhyshkowska, M. Falkowski-Hansen, K. Schledzewski, A. Gratchev, U. Mansmann, C. Schmuttermaier, E. Dippel, W. Koenen, F. Riedel, M. Sankala, K. Tryggvason, L. Kobzik, G. Moldenhauer, B Arnold, and S. Goerdt. Differential expression of a gene signature for scavenger/lectin receptors by endothelial cells and macrophages in human lymph node sinuses, the primary sites of regional metastasis. *J Pathol*, 208(4):574, 2006.

F. Millenaar, J. Okyere, S. May, M. van Zanten, L. Voesenek, and A. Peeters. How to decide? different methods of calculating gene expression from short oligonucleotide array data will give different results. *BMC Bioinformatics*, 7(1):137, 2006. ISSN 1471-2105.

M. Milo, A. Fazeli, M. Niranjan, and N. D. Lawrence. A probabilistic model for the extraction of expression levels from oligonucleotide arrays. *Biochem Soc Trans*, 31(Pt 6):1510–1512, 2003.

C. Mircean, I. Tabus, J. Astola, T. Kobayashi, H. Shiku, M. Yamaguchi, I. Shmulevich, and W. Zhang. Quantization and similarity measure selection for discrimination of lymphoma subtypes under k-nearest neighbor classification. In *Proceedings of SPIE*, volume 5328, pages 6–17, 2004.

C. Mircean, I. Tabus, and J. T. Astola. Quantization and distance function selecton for discrimination of tumors using gene expression data. In *Proceedings of SPIE*, volume 4623, page 1. SPIE, 2002.

I. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer, 2002.

R. Narsai, K.A. Howell, A.H. Millar, N. O'Toole, I. Small, and J. Whelan. Genome-wide analysis of mRNA decay rates and their determinants in Arabidopsis thaliana. *Plant Cell*, 19(11):3418, 2007.

B. Noble. *Applied linear algebra*. Prentice-Hall, USA, 1969.

R. Pal, A. Datta, Jr Fornace, A. J., M. L. Bittner, and E. R. Dougherty. Boolean relationships among genes responsive to ionizing radiation in the NCI 60 ACDS. *Bioinformatics*, 21(8):1542–1549, 2005.

P. J. Park, M. Pagano, and M. Bonetti. A nonparametric scoring algorithm for identifying informative genes from microarray data. *Pac Symp Biocomput*, 52:63, 2001.

R. Pearson. A comprehensive re-analysis of the Golden Spike data: Towards a benchmark for differential expression methods. *BMC Bioinformatics*, 9(1):164, 2008.

R. Pearson, X. Liu, G. Sanguinetti, M. Milo, N. Lawrence, and M. Rattray. puma: a Bioconductor package for propagating uncertainty in microarray analysis. *BMC Bioinformatics*, 10(1):211, 2009.

D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(suppl1):S215–224, 2001.

P. Perona and W. Freeman. A factorization approach to grouping. *Lecture Notes in Computer Science*, 1406:655–670, 1998.

A. Ploner, L. Miller, P. Hall, J. Bergh, and Y. Pawitan. Correlation test to assess low-level processing of high-density oligonucleotide microarray data. *BMC Bioinformatics*, 6(1):80, 2005. ISSN 1471-2105.

G. Potamias, L. Koumakis, and V. Moustakis. Gene selection via discretized gene-expression profiles and greedy feature-elimination. *LNAI*, 3025:256–266, 2004.

L. X. Qin, R. Beyer, F. Hudson, N. Linford, D. Morris, and K. Kerr. Evaluation of methods for oligonucleotide array data via quantitative real-time pcr. *BMC Bioinformatics*, 7(1):23, 2006. ISSN 1471-2105.

L. X. Qin, K. F. Kerr, and Contributing Members of the Toxicogenomics Research Consortium. Empirical evaluation of data transformations and ranking statistics for microarray analysis. *Nucl. Acids Res.*, 32(18):5471–5479, 2004.

J. Quackenbush. Computational analysis of microarray data. *Nat Rev Genet*, 2(6): 418–427, 2001.

J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32(supp):496–501, 2002.

A. Raj, C. S. Peskin, D. Tranchina, D. Y. Vargas, and S. Tyagi. Stochastic mRNA synthesis in mammalian cells. *PLoS Biol*, 4(10):e309, 2006.

S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. 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 tumor gene expression signatures. *PNAS*, 98(26):15149–15154, 2001.

M. Rattray, X. Liu, G. Sanguinetti, M. Milo, and N. D. Lawrence. Propagating uncertainty in microarray data analysis. *Brief Bioinform*, 7(1):37–47, 2006.

G. Rustici, J. Mata, K. Kivinen, P. Lio, C. J. Penkett, G. Burns, J. Hayles, A. Brazma, P. Nurse, and J. Bahler. Periodic gene expresiion program of the fission yeast cell cycle. *Nat Genet*, 36(8):809 – 817, 2004.

P. Ruusuvuori, O. Yli-Harja, C. Sima, and E.R. Dougherty. Classification of quantized small sample data. In *Genomic Signal Processing and Statistics, 2006. GENSIPS'06. IEEE International Workshop on*, pages 93–94, 2006.

D. Sahoo, D. L. Dill, R. Tibshirani, and S. K. Plevritis. Extracting binary signals from microarray time-course data. *Nucleic Acids Res*, 35(11):3705–12, 2007.

G. Sanguinetti, M. Milo, M. Rattray, and N. D. Lawrence. Accounting for probe-level noise in principal component analysis of microarray data. *Bioinformatics*, 21(19): 3748–3754, 2005.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235): 467–470, 1995.

B. Scholkopf and A. J. Smola. *Learning with kernels*. MIT press Cambridge, Mass, 2002.

N. Schonrock, V. Exner, A. Probst, W. Gruissem, and L. Hennig. Functional Genomic Analysis of CAF-1 Mutants in Arabidopsis thaliana. *J Biol Chem*, 281(14):9560–9568, 2006.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

K. Shedden, W. Chen, R. Kuick, D. Ghosh, J. Macdonald, K. Cho, T. Giordano, S. Gruber, E. Fearon, J. Taylor, and S. Hanash. Comparison of seven methods for producing Affymetrix expression scores based on False Discovery Rates in disease profiling data. *BMC Bioinformatics*, 6(1):26, 2005.

J. Shendure. The beginning of the end for microarrays? *Nat Methods*, 5:585–587, 2008.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

L. Shi, W. D. Jones, R. V. Jensen, R. D. Wolfinger, E. S. Kawasaki, D. Herman, L. Guo, F. M. Goodsaid, and W. Tong. Reply to MAQC papers over the cracks. *Nat Biotechnol*, 25:28–29, 2007.

I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2): 261–274, 2002a.

I. Shmulevich, E.R. Dougherty, and W. Zhang. From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks. *PROCEEDINGS-IEEE*, 90(11): 1778–1792, 2002b.

I. Shmulevich and W. Zhang. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565, 2002.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, 1(2):203–209, 2002.

V. A. Smith, E. D. Jarvis, and A. J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18(suppl-1):S216–224, 2002.

J. Somers, C. Smith, M. Donnison, D. N. Wells, H. Henderson, L. McLeay, and P. L. Pfeffer. Gene expression profiling of individual bovine nuclear transfer blastocysts. *Reproduction*, 131(6):1073–1084, 2006.

P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Mol Biol Cell*, 9(12):3273–3297, 1998.

A. Statnikov, L. Wang, and C. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9(1):319, 2008.

K. F. Storch, O. Lipan, I. Leykin, N. Viswanathan, F. C. Davis, W. H. Wong, and C. J. Weitz. Extensive and divergent circadian gene expression in liver and heart. *Nature*, 417(6884):78–83, 2002.

L. J. Su, C. W. Chang, Y. C. Wu, K. C. Chen, C. J. Lin, S. C. Liang, C. H. Lin, J. Whang-Peng, S. L. Hsu, C. H. Chen, and C. Y. Huang. Selection of ddx5 as a novel internal control for q-rt-pcr from microarray data using a block bootstrap re-sampling scheme. *BMC Genomics*, 8(1):140, 2007.

S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(suppl1):i359–368, 2005.

T. T. Tanimoto. An elementary mathematical theory of classification and prediction. *IBM Internal Report*, 1958.

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS*, 99(10):6567, 2002.

I. Tirosh, A. Weinberger, D. Bezalel, M. Kaganovich, and N. Barkai. On the relation between promoter divergence and gene expression evolution. *Mol Syst Biol*, 4(1), 2008.

M. M. Tomayko, S. M. Anderson, C. E. Brayton, S. Sadanand, N. C. Steinel, T. W. Behrens, and M. J. Shlomchik. Systematic Comparison of Gene Expression between Murine Memory and Naive B Cells Demonstrates That Memory B Cells Have Unique Signaling Capabilities. *J Immunol*, 181(1):27, 2008.

A. Torrente, M. Kapushesky, and A. Brazma. A new algorithm for comparing and visualizing relationships between hierarchical and flat gene expression data clusterings. *Bioinformatics*, 21(21):3993–3999, 2005.

D. Tritchler, S. Fallah, and J. Beyene. A spectral clustering method for microarray data. *Comput Stat Data Anal*, 49(1):63–76, 2005.

M. W. B. Trotter. *Support Vector Machines for Drug Discovery.* PhD thesis, University College London, UK, 2006.

V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, page 91062498, 2001.

A. R. van Erkel and P. M. Th. Pattynama. Receiver operating characteristic (roc) analysis: Basic principles and applications in radiology. *Eur J Radiol*, 27(2):88 – 94, 1998. ISSN 0720-048X.

V. N. Vapnik. *Statistical learning theory.* Wiley, New York, 1998.

S. S. Vukkadapu, J. M. Belli, K. Ishii, A. G. Jegga, J. J. Hutton, B. J. Aronow, and J. D. Katz. Dynamic interaction between t cell-mediated beta-cell damage and beta-cell repair in the run up to autoimmune diabetes of the nod mouse. *Physiol Genomics*, 21(2):201–211, 2005.

T. A. Wallace, R. L. Prueitt, M. Yi, T. M. Howe, J. W. Gillespie, H. G. Yfantis, R. M. Stephens, N. E. Caporaso, C. A. Loffredo, and S. Ambs. Tumor immunobiological differences in prostate cancer between African-American and European-American men. *Cancer Res*, 68(3):927–936, 2008.

Y. Wang, C.L. Liu, J.D. Storey, R.J. Tibshirani, D. Herschlag, and P.O. Brown. Precision and functional specificity in mRNA decay. *PNAS*, 99(9):5860, 2002.

P. Warnat, R. Eils, and B. Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6 (1):265, 2005.

J. B. Welsh, L. M. Sapinoso, A. I. Su, S. G. Kern, J. Wang-Rodriguez, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res*, 61(16):5974–5978, 2001.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Jr. Olson, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, 98(20):11462–11467, 2001.

K. Willbrand, F. Radvanyi, J. P. Nadal, J. P. Thiery, and T. M. A. Fink. Identifying genes from up-down properties of microarray expression series. *Bioinformatics*, 21 (20):3859–3864, 2005.

P. Willett. Similarity-based virtual screening using 2d fingerprints. *Drug Discov Today*, 11(23/24):1046–1053, 2006.

P. Willett, J. M. Barnard, and G. M. Downs. Chemical similarity searching. *J Chem Inf Comput Sci*, 38(6):983–996, 1998.

C. Workman, L. J. Jensen, H. Jarmer, R. Berka, L. Gautier, H. B. Nielsen, H. H. Saxild, C. Nielsen, S. Brunak, and S. Knudsen. A new non-linear normalization method for reducing variability in DNA microarry experiments. *Genome Biol*, 3(9):1–0048, 2002.

E. P. Xing and R. M. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(suppl1):S306–315, 2001.

E. Yang, E. van Nimwegen, M. Zavolan, N. Rajewsky, M. Schroeder, M. Magnasco, and J. E. Darnell. Decay rates of human mRNAs: correlation with functional characteristics and sequence attributes. *Genome Res*, 13(8):1863–1872, 2003.

Y. H. Yang, M. J. Buckley, and T. P. Speed. Analysis of cDNA microarray images. *Brief Bioinform*, 2(4):341–349, 2001.

S. Yegnasubramanian, M. C. Haffner, Y. Zhang, B. Gurel, T. C. Cornish, Z. Wu, R. A. Irizarry, J. Morgan, J. Hicks, T. L. DeWeese, W. B. Isaacs, G. S. Bova, A. M. De Marzo, and W. G. Nelson. DNA hypomethylation arises later in prostate cancer progression than cpg island hypermethylation and contributes to metastatic tumor heterogeneity. *Cancer Res*, 68(21):8954–8967, 2008.

X. Zhou, X. Wang, and E. R. Dougherty. Binarization of microarray data on the basis of a mixture model. *Mol Cancer Ther*, 2(7):679–684, 2003.

M. J. Zilliox and R. A. Irizarry. A gene expression bar code for microarray data. *Nat Methods*, 4(11):911–913, 2007.

# Index