COMPARISON OF CLUSTERING AND CLASSIFICATION METHODS
COMBINED WITH DIMENSION REDUCTION
USING GENE EXPRESSION DATA

by

Ahalya Sivathayalan

B. Sc. (Engineering), University of Peradeniya, Sri Lanka.

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

SCHOOL OF MATHEMATICS AND STATISTICS
OTTAWA-CARLETON INSTITUTE FOR MATHEMATICS AND STATISTICS
CARLETON UNIVERSITY
OTTAWA, ONTARIO
AUGUST 2009

© Ahalya Sivathayalan, 2009

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-60130-3
Our file Notre référence
ISBN: 978-0-494-60130-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Following the work of Dai et al. (2006), Tan et al. (2008) and Osareh and Shadgar (2009) we examine the performance of several clustering and classification algorithms, together with common dimension reduction methods, when applied to two very different microarray datasets (yeastCC and Breast Cancer).

Clustering techniques examined include two main groups - partitioning and hierarchical methods as well as Plaid Models (Lazzeroni and Owen, 2002). This study examined the partitioning methods k-means, k-medoids, and Self-Organizing Maps (SOMs), and a hierarchical method - agglomerative hierarchical with mean linkage. Plaid models, which were specifically designed to accommodate the characteristics of microarray data, were also studied.

Classification methods adaBoost, C4.5, single classification trees (both *rpart* and *tree* library from R), neural networks, support vector machines, and random forest were used in combination with two dimension reduction techniques: partial least squares and principal components analysis. The suitability of these methods in combination with dimension reduction techniques was assessed using experimental design.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Shirley.E.Mills, for her guidance and advice. She has introduced and taken me through the path of the academic research. I have learned a lot from her through our conversations.

I would like to thank Dr.J.D.Nielsen for his help with R in a timely manner, and Mr.J.Stewart and his assistant for providing me advice on remote computing.

And I would like to thank all the faculty members with whom I took courses and staff for their support and guidance throughout my years at Carleton University.

Finally I would like to thank my husband Siva, my children Janani and Ashok, and my father and mother, Mr and Mrs. Peethamparam, for their love and continued support during the time I spent at Carleton University.

TABLE OF CONTENTS

Abstract	iii
Acknowledgments	iv
Table of Contents	v
List of Tables	viii
List of Figures	x
1 Introduction	1
1.1 Motivation and background	1
1.2 Data	3
1.3 Organization of thesis	5
2 Literature Review	6
3 Clustering	13
3.1 Methodology	13
3.1.1 Dimension Reduction Techniques	13
3.1.1.1 Principal Component Analysis (PCA)	14
3.1.1.2 Independent Component Analysis (ICA)	15
3.1.2 Clustering Techniques	16
3.1.2.1 K-means	16
3.1.2.2 K-medoids	18
3.1.2.3 SOM - Self-Organizing Maps	19
3.1.2.4 Hierarchical Clustering	22
3.1.3 Plaid Models	22

3.2	Results	24
3.2.1	K-means	26
3.2.2	K-medoids	28
3.2.3	SOM	30
3.2.4	Hierarchical Clustering	32
3.2.5	Plaid Models	32
4	Classification	43
4.1	Methodology	43
4.1.1	Dimension Reduction Algorithms	44
4.1.1.1	Partial Least Squares (PLS)	44
4.1.1.2	Principal Components Analysis with varimax rotation	46
4.1.2	Classification Algorithms	47
4.1.2.1	Single Classification Trees	48
4.1.2.2	AdaBoost.M1	48
4.1.2.3	C4.5	49
4.1.2.4	Neural Networks (NNET)	51
4.1.2.5	Support Vector Machines (SVMs)	52
4.1.2.6	Random Forest	54
4.1.3	K - Nearest Neighbour	55
4.1.4	Design of Experiments	56
4.2	Results	57
4.2.1	Single Classification Trees	57
4.2.2	AdaBoost	57
4.2.3	C4.5	58
4.2.4	Neural Networks (NNET)	58
4.2.5	Support Vector Machines	67
4.2.6	Random Forest	67

4.2.7	K-Nearest Neighbours	69
4.2.8	Design of Experiments	69
5	Summary, Conclusions and Future Work	75
5.1	Summary and Conclusions	76
5.1.1	Clustering	76
5.1.2	Classification	78
5.2	Future Work	79
	Bibliography	81
	Appendix	
	A.R & SAS code	88
	B.Genes belonging to each cluster	89
	C.Detailed Results for Plaid Models	90

LIST OF TABLES

Table

3.1	Optimal number of clusters using k-means	28
3.2	Summary of number clusters using SOM	30
3.3	Top 12 genes and 3 samples in layer1 and layer 2 for yeastCC	34
3.4	Top 12 genes in layer1 and layer 2 for Breast Cancer	35
3.5	Top 3 samples in layer1 and layer 2 for Breast Cancer	35
3.6	Sample effects ($\mu + \beta_j$) in the first five layers for yeastCC	36
3.7	Sample effects ($\mu + \beta_j$) in the first five layers for Breast Cancer . . .	40
3.8	Samples/Genes belonging to more than one layers for Breast Cancer .	42
4.1	Misclassification Rate for Single Classification Tree	59
4.2	Misclassification Rate for adaBoost	59
4.3	Misclassification Rate for C4.5	59
4.4	Misclassification rates and transformed values for NNET using PCA .	59
4.5	ANOVA Table yeastCC and Breast Cancer with PCA for NNET clas- sification	60
4.6	Data*iteration Effect Sliced by data for transformed misclassification	60
4.7	NNET Misclassification rates and transformed values for yeastCC us- ing PCA	65
4.8	ANOVA Table NNET classification for yeastCC with PLS and PCA	68
4.9	Results of Tukey's HSD test for Misclassification rates for yeastCC with PLS and PCA	68
4.10	Misclassification Rate for NNET using PLS for Breast Cancer	68
4.11	Misclassification error rate for SVM	68
4.12	Misclassification Rate using Random Forest	70

4.13	Best-k using k-nn methods	70
4.14	Misclassification rates summarized	70
4.15	ANOVA Table for Overall Analysis	70
4.16	Mean of misclassification rates for data set, classification method, and dimension reduction techniques	71
4.17	Drdc*metd Effect Sliced by drdc for transformed misclassifications . .	71
4.18	Drdc*metd Effect Sliced by metd for transformed misclassification . .	71
4.19	Least Squares Means for interaction effect of drdc and method	72
5.1	Summary of plots needed for six different data sets	76

LIST OF FIGURES

Figure	
3.1	Sample plot of k-means 27
3.2	Sample plot of k-medoids 29
3.3	Sample plot of SOM for yeastCC 31
3.4	Sample plot of SOM for Breast Cancer 31
3.5	Summary of clusters at different cuts in hierarchical clustering 33
4.1	Interaction plot for data and iterations for yeastCC and Breast Cancer with PCA 61
4.2	Interaction plot for data and hidden layers for yeastCC and Breast Cancer with PCA 62
4.3	Interaction plot for iterations and hidden layers for yeastCC and Breast Cancer with PCA 63
4.4	Interaction plot of dimension reduction techniques and hidden layers for yeastcc with PCA and PLS 66
4.5	Interaction plot of dimension reduction techniques and methods 73

1. INTRODUCTION

A selection of clustering and classification techniques was used on two different microarray datasets to examine their performance and suitability. Clustering techniques were applied on the samples of genes (variables). Since the number of variables greatly exceeds the number of cases for microarray data, dimension reduction was done before applying any classification algorithms. Different classification algorithms in combination with two different dimension reduction techniques were used in the analyses that follow.

1.1 Motivation and background

Microarray data are generated in the fields of molecular biology and medicine. Biologists and medical scientists now believe that sources of presently incurable diseases are tied to differences in expressions of genes in a particular cell and that treatment directions for these presently untreatable diseases can be found by studying the differences in the gene expressions of the diseased and normal cells. Finding appropriate clustering and classification techniques for a microarray dataset with certain characteristics could pave a way to relate various gene expressions with illnesses and potentially lead to treatment methodologies to cure diseases.

As a first unsupervised way to examine microarray data, clustering techniques can be used on samples of genes to find clusters that are similar in a given dataset. This could aide in identification of groups of genes related to a certain condition. In the case of microarray data with known classes/clusters, we might determine whether classes/clusters correspond to known classes/clusters in the dataset. In this case, different classification techniques, in combination with dimension reduction techniques, can be assessed to find the procedure with the best predictive accuracy to predict the

class of a new datum.

In this thesis, two very different microarray datasets were used to demonstrate the results of clustering and classification when combined with dimension reduction. Clustering methods were applied to samples of genes of the original data, principal components using both the covariance matrix and the correlation matrix of the data, and independent components taken from the original data and the above-mentioned two different principal components of the data. Two common categories of clustering techniques were used initially to determine clusters - partitioning and hierarchical. The former includes k-means (together with the sum of squares and the Davies-Bouldin Index), k-medoids (using Manhattan and Euclidean distance in combination with average silhouette width), and the self-organizing map. These methods are discussed in detail in Chapter 3. The latter clustering technique, hierarchical clustering, with average linkage using Euclidean distance, was used on samples of the original data. In addition, the plaid model (Lazzeroni and Owen, 2002) that was specifically designed for clustering of microarray data was also used. In the literature, the above mentioned methods have been repeatedly used on different microarray datasets but without a general comparison of their performance for microarray datasets with different characteristics.

A selected number of classification techniques was also used with two specific dimension reduction techniques - Partial Least Squares (PLS) and Principal Component Analysis (PCA) - to compare misclassification error rates. Most standard statistical methods and machine learning algorithms are unable to cope with microarray data because these methods and algorithms require the number of cases in a dataset to be larger than the number of input variables. But, in the case of microarray data, the number of cases can be in the range of 50-100 and the number of variables in the range of 1000-10,000s. As a result, to be able to use these existing classification methods, original microarray data was fed into a dimension reduction

algorithm which reduces the number of input variables by filtering out irrelevant input variables or building a small number of linear or nonlinear combinations from the original set of input variables. Principal components analysis with varimax rotation and the partial least squares regression were used as feature selection/dimension reduction techniques. These were followed by different classification techniques such as C4.5, adaBoost, single classification trees (using *rpart* and *tree* add-on packages in R), Neural Networks (NNET), Support Vector Machine (SVM), and Random Forest to assess which combination provided the smallest misclassification rate. In addition, k-nearest neighbour classification was applied on samples of the original data. The above-mentioned dimension reduction methods and classification techniques are discussed in detail in Chapter 4.

1.2 Data

Genes influence human physical and behavioural development through protein synthesis. The information necessary for functioning of cells is encoded in the genes. The messages formed from the genes contain instructions for the creation of functional structures called proteins which are necessary for cell life processes. Information transfer from genes to Ribose Nucleic Acid (RNA) is called transcription, and from RNA to protein is called translation. Transcription is the information transfer process directly relevant to the Deoxyribose Nucleic Acid (DNA) microarray experiment because the quantification of the type and amount of this copied information is the data captured in the microarray experiment.

What is gene expression of microarray data? The gene expression of microarray experimental procedure contains the following steps. First, the array fabrication involves preparing the glass slide, obtaining the DNA sequences and depositing the complementary DNA (cDNA) onto the glass slide. The second step involves total RNA from the tissue samples which could be from treated and control samples or

diseased and normal tissue samples. From this, cDNA is synthesized and labelled with different fluorescent dyes, and then mixed with arrayed DNA sequences. During the process, the probes matching the gene of the printed DNA sequences will attach to the printed DNA sequences. This process is called hybridization. After this, the slides are imaged by an imaging device that makes fluorescence measurements for each dye. This measured fluorescent intensity of each gene is expected to be proportional to transcript abundance.

Our analyses are conducted on two microarray datasets downloaded from the Bioconductor website. One of them is yeast cell cycle microarray data (yeastCC), and the other is expression and CGH data on breast cancer cell lines (Neve2006 data/Breast Cancer data). The yeastCC data contains 6178 genes (variables) and 77 samples (cases). Detailed information on this dataset can be found in Spellman et al. (1998). Five (5) clusters (Spellman et al., 1998) were obtained using clustering algorithm of Eisen et al. (1999) for yeastCC. The Breast Cancer data contains 22,283 genes (variables) and 50 samples (cases) and has 3 clusters (R-Neve2006). Both datasets are given as matrices where rows represent genes and columns represent samples. For the yeastCC dataset, there were only 509 genes out of 6178 with no missing values. The missing values were imputed with column mean (sample mean) for yeast data. The Breast Cancer data has no missing values. Genes and samples are highly correlated for Breast Cancer data, but not for yeastCC. Also the Breast Cancer data has almost balanced classes¹ which are BaA (12), BaB (14), and Lu (24), on the other hand yeastCC has highly imbalanced classes which are alpha (18), cdc15 (24), cdc28 (17), clb (2), cln (2), and elu (14). Statistical software package R was used to analyse the data.

The following are websites from where the datasets were downloaded (October 2008):

¹ Number of samples in a class given in parenthesis

YeastCC Data website:

www.bioconductor.org/packages/release/data/experiment/html/yeastCC.html

Breast Cancer Data website:

<http://www.bioconductor.org/packages/release/data/experiment/html/Neve2006.html>

1.3 Organization of thesis

In Chapter 2, a literature review is presented with the focus on different clustering and classification algorithms discussed. Detailed discussion of different clustering algorithms and results are presented in Chapter 3; classification algorithms, dimension reduction techniques and results are presented in Chapter 4. Summary, conclusion and future work are given in Chapter 5.

2. LITERATURE REVIEW

Statistical analysis techniques are routinely used to better understand the nature of various phenomena and to enhance the ability to predict outcomes. However, we encounter problems when attempting to apply similar techniques to gene expression of microarray data. This type of data typically involves tens of thousands of variables but only up to hundreds cases. The significantly larger number of variables than cases (the case of $p \gg n$) and the essentially exponential increase in the required numerical effort to conduct a statistical analysis when dealing with a large number of variables makes dealing with gene expression of microarray data fairly challenging.

Clustering techniques can be used to reduce the number of cases so that similar variables (genes) can be grouped together to achieve some degree of homogeneity within the cluster and heterogeneity between clusters.

Clustering techniques are divided into two groups - partitioning and hierarchical methods (discussed in detail in Chapter 3). In this study, we used the partitioning methods such as K-means, K-medoids, and Self-Organizing Maps (SOMs), and a hierarchical method - agglomerative hierarchical with mean linkage- and the classification algorithms adaBoost (Freund, 1996; Freund and Schapire, 1996), C4.5 (Quinlan, 1992), single classification trees (Breiman et al., 1984), neural networks, support vector machines (SVM), and random forest (Breiman, 2001) along with two dimension reduction techniques, principal component analysis (PCA) and partial least squares (PLS). For the k-Nearest Neighbour (k-NN) classifier no dimension reduction techniques were used.

Among the four different K-means clustering algorithms that are implemented in R, the Hartigan and Wong algorithm (H-W algorithm) has the properties that there

is no single switch of an observation from one group to another that will decrease the objective (total variance for that cluster assignment), and it converges with a minimum number iterations. We used the H-W algorithm with the Davies-Bouldin Index (Davies and Bouldin, 1979) to obtain the optimal number clusters as smaller values of the index mean that clusters are compact and far from each other (detailed discussion in Chapter 3).

K-medoids (PAM- Partitioning Around Medoids), introduced by Kaufman and Rousseeuw (1990), was employed with Euclidean and Manhattan dissimilarity on the samples of data. To obtain optimal number of clusters, the silhouette validation technique (Rousseeuw, 1987) was used. The largest overall average silhouette width indicates the best clustering. Therefore, the number of clusters with maximum overall average silhouette width was taken as the optimal number of clusters (discussed in chapter 3).

SOMs, formulated by Kohonen (1989), would be used to reduce high dimensional data to a one- or two-dimensional grid. This algorithm is quite similar to K-means, but with a constraint that restricts the cluster centers to lie in one or two-dimensional manifold (see Chapter 3 for details).

Agglomerative hierarchical clustering with average linkage was used on the samples of the original data. In the case of agglomerative hierarchical clustering, the measure of 'closeness' has many possible definitions (e.g. single linkage, complete linkage, mean linkage etc. (see R-help file for *hclust*) when clusters are not singleton points (detailed discussion in Chapter 3).

In the literature, the above-mentioned clustering methods have been repeatedly used to cluster microarray data. K-means, hierarchical, and SOM clustering were discussed by Tibshirani et al. (1999) with the purpose of presenting a new clustering method which is expected to work on microarray data with different characteristics. Eisen et al. (1998) used agglomerative hierarchical clustering with average linkage

to cluster two different sets of gene expression of microarray data to illustrate how such methods can be useful in the analysis of gene expression data. Sorlie et al. (2001) analysed a total of 85 cDNA microarray experiments representing 78 cancers, three fibroadenomas, and four normal breast tissues using hierarchical clustering to identify tumour subtypes. Moreover, Perou et al. (1999) used hierarchical clustering to analyse gene expression patterns in human mammary epithelial cells and breast cancers. Goldstein et al. (2002) used PAM, k-means, and hierarchical clustering to highlight some of the issues that arise with use of hierarchical clustering techniques in the analysis of microarray data by looking at the effects of the above-mentioned clustering algorithms for a fixed number of clusters. In the above-mentioned clustering methods, observations must belong to one cluster only. But microarray data differs from much other data since a gene could belong to more than one function of cell process or none at all. Plaid models (Lazzeronni and Owen, 2002) were built to accommodate these characteristics of microarray data. Consequently, Plaid models were used for the purpose of comparing with traditional clustering techniques in our analysis.

In addition to determining gene clusters, there is a need to make use of labelled data to build suitable classification models to predict the class of a new datum. We used the k-NN classifier which was initially proposed by Fix and Hodges (1951) as a consistent nonparametric estimator of the maximum likelihood discriminant rule. The k-NN method has been previously used by Osareh and Shadgar (2009) to compare its performance with some other classifiers. As the nearest neighbour classifier does not perform feature selection, all the variables (whether relevant or not) are used in building the classifier. Consequently, results that we get using the k-NN method might not perform as well as others for the case of microarray data with thousands of features, most of which are likely to be uninformative. Therefore, this classifier was used on the samples of the original data to estimate the best k using leave-one

out cross-validation, ten-fold cross-validation techniques and `knn.cv` function that is implemented in R.

No comprehensive studies comparing the results of different clustering algorithms with the application of dimension reduction techniques on microarray datasets could be found in the literature. Our purpose here is to examine the performance of different clustering techniques when applied to very different microarray datasets and to find how the results vary depending on whether original, principal components or independent components of the data are used.

In the case of microarray data, dimension reduction is an important issue in classification since the number of variables is higher than the number of cases. Some classifiers like CART perform automatic feature selection and are relatively insensitive to the variable selection scheme. But for many classifiers, it is important to perform some kind of dimension reduction; otherwise performance of the classifier could degrade substantially. This can be achieved in two ways - feature selection or variable selection. When the original gene expression data is fed into a dimension reduction algorithm, if it reduces the number of input variables by **filtering out** a larger amount of irrelevant input variables, then it is called variable selection. On the other hand, if the number of variables is reduced by **building** a small number of linear or nonlinear combinations from the original set of input variables, then it is called feature selection.

Many screening procedures such as the one-gene-at-a-time approach (Dudoit et al., 2002a), ad hoc signal-to-noise statistics (Golub et al., 1999; Pomeroy et al., 2002), nonparametric Wilcoxon statistics (Dettling and Bulmann, 2002; Park et al., 2001), and p-values (Dudoit et al., 2002b) have been used previously. In this thesis we have used partial least-squares (PLS) regression (Geladi and Kowalski, 1986; Höskuldsson, 1988), and principal components (PCA) with varimax rotation as the dimensionality reduction algorithms. There exist two algorithms PLS: SIMPLS (de Jong, 1993) and

NIPLS. We use the SIMPLS algorithm that is implemented as an add-on package *pls* with option chosen as *simpls* in the R statistical software. Dai et al. (2006) previously used only one classifier (logistic discrimination) to assess its performance in combination with three different dimension reduction methods which are PLS, Sliced Inverse Regression (SIR), and PCA on two different microarray datasets. We followed the same procedure as in Tan et al. (2008) but used several classifiers on two quite different microarray data sets.

The libraries *tree* and *rpart* used to build a classifier and hence to predict the class of a new datum were based on Atkinson and Therneau (1997). Boosting (Freund and Shapire, 1996a) is a method for improving the accuracy of a learning algorithm. The boosting algorithm *adaBoost* (Freund and Shapire, 1997) which can be used in the case of binary classification and *adaBoost.M1* (Freund and Shapire, 1997) is one of the methods extending *adaboost* to the multi-class case. It has been stated in the literature (Freund and Schapire, 1996) that *adaBoost.M1* can be used with any classifier as a base classifier, but the unstable classifiers such as single classification trees tend to benefit the most (Speed, 2003). Here, we used the *adaBoost.M1* algorithm that uses classification trees as single classifiers.

C4.5 (Quinlan, 1993), which uses a divide-and-conquer approach to growing decision trees that was pioneered by Hunt et al (1966), was another classifier used.

Random forest (Breiman et al, 2001b) is built using an ensemble of classification trees. This classification algorithm has been previously used in the case of microarray data, especially to do gene selection (Díaz-Uriarte and Alvarez de Andrés, 2005). We use random forest to do classification after doing dimension reduction on the data.

The concept of a neural network (Hastie, 2001) seems to have been first formulated by Alan Turing (1948). Previously, this algorithm has been used in microarray data by others, including O'Neill and Song (2003) and Osareh and Shadgar (2009). Our analysis is very different from that of O'Neill and Song, but similar to that of Os-

areh and Shadgar. However, the latter use three different filter approaches to reduce the number of variables in the data, before using different classifiers for classification. We employ neural networks after using the dimension reduction techniques PLS and PCA. This algorithm is discussed in detail in Chapter 4.

We use support vector machines (SVMs) as the final classifier for our datasets. SVM was first introduced by Vapnik (1979, 1998) and has gained popularity recently. It has been previously used in the analysis of microarray data by Chu et al (2005), after doing a handful of different dimension reduction techniques since the number of variables in the microarray data exceeds cases. Furthermore, Brown et al. (2000) compared the performance of the SVM classifier with that of four other standard learning algorithms: Parzen windows, Fisher's linear discriminant, and two decision tree learners (C4.5, and MOC1), and concluded that SVMs perform better than the other four classifiers. Furey et al (2000) used SVMs with different kernels and a classifier based on the perceptron algorithm and concluded that they all perform similarly. Mukherjee S., et al (1998) used SVM without feature selection and showed that they achieve better results than Gloub et al (1999).

The clustering methods, classification algorithms, and dimension reduction techniques that we use in this thesis are not new and have been previously employed by many different researchers but with different purposes, in different combinations and on different microarray datasets, and none have used experimental design to discover statistically significant differences and/or interactions. There have been comparisons of several clustering (Tibshirani et al., 2000; Eisen et al., 1998; Spellman et al., 1998; Tamayo et al., 1999) or classification methods (Tan et al., 2008) for a particular microarray dataset. There have been comparisons of a single method on several microarray datasets (Eisen et al., 1998; Perou et al., 1999) or of several clustering methods on one dataset (Sorlie et al., 2001). There has been study of the use of some dimension reduction method prior to using a single classifier (Dai et al., 2006) or

several different classifiers (Osareh and Shadgar, 2009). This thesis extends the work of these latter two as well as that of Tan et al. (2008). Similar work was done by Lee et al. (2005) using different classifiers, dimension reduction techniques and datasets but with no analysis of results using experimental design. There is another similar research by Boulesteix et al. (2008) but with focus on various dimension reduction techniques and not considering the possibility of interaction effects of dimension reduction methods and classification techniques. Our purpose is to gather together these concepts in a thesis that examines the performance of several clustering and classification algorithms together with common dimension reduction methods on two very different microarray datasets.

3. CLUSTERING

3.1 Methodology

Clustering techniques such as k-means, k-medoids, SOM, and hierarchical clustering were used in this study. K-means and k-medoids were applied on samples of the original data, PCAs of the data, and ICAs of the data. The performance of these clustering techniques together with the above-mentioned dimension reduction techniques was assessed. SOM was applied on samples of the first three principal components of the data and hierarchical clustering was applied on the samples of the original data. These methods have long been used for clustering of data that comes from different fields. Most clustering methods assume that each data point belongs to only one cluster. This might not always be true for gene expression of microarray data because some genes may belong together in a cluster according to the way they were expressed for a certain biological process, while they may cluster with other genes under some other conditions. The traditional clustering methods do not allow for this. Consequently, Plaid models (Lazzeroni and Owen 2002) that accommodate this characteristic of genes were also used and we compared their performance with other traditional clustering methods.

3.1.1 Dimension Reduction Techniques

In some instances, we applied the dimension reduction techniques Principal Component Analysis and Independent Component Analysis before clustering techniques were applied.

3.1.1.1 Principal Component Analysis (PCA)

PCA is a well-known dimension reduction technique. First it constructs new variables which are linear combinations of the original ones in such a way that the new principal components are orthogonal and account for decreasing amounts of the variation in the original dataset. This is achieved by obtaining the particular linear combinations of the p predictor variables $X = [X_1, X_2, \dots, X_p]$ in such a way that

$$\begin{aligned} Y_1 &= a'_1 X \\ Y_2 &= a'_2 X \\ &\vdots \\ Y_p &= a'_p X \end{aligned}$$

where $Y = [Y_1, Y_2, \dots, Y_p]$

PCA sequentially maximizes the variance of a linear combination of the original predictor variables.

$$a_k = \arg \max_{a' a = 1} Var(Xa) \quad (3.1)$$

subject to constraint $a'_i \Sigma a_j = 0$, for all $1 \leq i \leq j$, where Σ = covariance of X . The above constraint ensures that the linear combinations $y_i = Xa_i$ are uncorrelated, i.e. $Cov(Xa_i, Xa_j) = 0, i \neq j$. The projection vectors a are obtained by eigenvalue decomposition on the covariance/correlation matrix Σ . The eigenvalue λ_i measures the variance of the i^{th} PC, and eigenvector a_i provides the loadings for the linear transformation, where $i = 1, 2, \dots, k$. Geometrically, these linear combinations represent the selection of a new coordinate system obtained by rotating the original system. The new axes represent the successive directions with maximum variability (i.e. the first PC accounts for the direction of largest variability and so on). There is the same

number of PC's as of original variables. To reduce the dimension of our data we examine a scree plot of the variance associated with each component to determine the number $k(< p)$ of important PC's. These k principal components form the new dataset (Johnson and Wichern, 2002).

3.1.1.2 Independent Component Analysis (ICA)

ICA is a variation of PCA and Factor Analysis. It differs from PCA in that the resulting components of ICA are not ordered as those of PCA. Also the components of factor analysis are modeled as Gaussian random variables, but Gaussian sources are excluded in ICA. ICA performs blind source separation assuming independence and non-Gaussianity of each original source (S). It tries to determine the source signal S from the observed mixtures X . Even though ICA starts from a factor analysis solution, it eventually looks for rotations that give independent components (S_i that are statistically independent).

The original source matrix S is considered to consist of non-Gaussian independent components. We observe mixtures $X = AS$ where the columns of S are independent components and A is a linear mixing matrix. Solving this problem results in finding an orthogonal A in such a way that components of $S = A^T X$ are independent and non-Gaussian. Maximizing the non-Gaussianity can be achieved by minimizing the sum of entropies of S . This concept was used in formulating the *fastICA* function which is available through the *fastICA* library in R which was used in our analysis. Here, non-Gaussianity is measured using approximations to neg-entropy which is robust and fast to compute (a detailed explanation is found in Hastie et al, 2001). However, the resulting components of ICA are not ordered as those of PCA.

3.1.2 Clustering Techniques

Traditional clustering methods can be grouped into two types: partitioning and hierarchical. Partitioning methods seek to optimally divide objects into a fixed number of clusters; hierarchical clustering methods produce a nested sequence of clusters. Clustering methods attempt to group data depending on a definition of similarity (or dissimilarity). The distance measures used in this thesis are as follows: Euclidean (for k-means, k-medoids, and hierarchical) and Manhattan (for k-medoids). K-means, k-medoids, and SOM belong to partitioning methods; hierarchical clustering with mean linkage belongs to hierarchical clustering.

3.1.2.1 K-means

The k-means clustering method is one of the most popular partitioning methods; it finds clusters and their centers. The technique requires that all variables be quantitative hence it can be used for gene expression of microarray data.

For a chosen number of clusters, this procedure iteratively moves the cluster centers to minimize the total within-cluster variance thus seeking to minimize the sum of squared distances from each observation to its cluster center $\hat{\mu}_k$ (Speed, 2003)

$$WSS = \sum_{k=1}^K \sum_{c(i)=k} \|x_i - \hat{\mu}_k\|^2 \quad (3.2)$$

Algorithm (Speed, 2003)

- (1) For a given number of clusters, cluster centers can be found, and then the total cluster variance (equation 3.1) is minimized with respect to $\{\hat{\mu}_1, \dots, \hat{\mu}_k\}$ yielding the means of the currently assigned clusters.
- (2) Given a current set of means $\{\hat{\mu}_1, \dots, \hat{\mu}_k\}$, (equation 3.2) is minimized by

assigning each observation to the closest cluster mean. i.e

$$C(i) = \operatorname{argmin}_{1 \leq k \leq K} \|x_i - \hat{\mu}_k\|^2$$

(3) Steps 1 and 2 are repeated until assignments do not change.

However, the initial problem is to determine the number of clusters for which to search. If there is only one cluster then every point belongs to it, resulting in a large sum of squares. On the other hand, if the number of clusters equals the number of points then every point is its own cluster and the sum of squares is zero. As a result, simply making the sum of squares a minimum is not a good way of determining how many clusters exist. We overcome this problem by using the plot of sum of squares vs number of clusters together with the plot of the Davies-Bouldin index vs number of clusters (Mills and Norminton, 2007). The Davies-Bouldin Index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation.

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{i \neq j} \left\{ \frac{S_n(c_i) + S_n(c_j)}{S(c_i, c_j)} \right\} \quad (3.3)$$

where n - number of clusters,

S_n - average distance of all objects from the cluster to their cluster center,

$S(c_i, c_j)$ - distance between clusters centers.

Therefore, the ratio is small if the clusters are compact and far from each other. As a result, the Davies-Bouldin index, which is a function of the ratio of the sum of within-cluster scatter to between-cluster separation, will have a small value for a good clustering. The corresponding number of clusters for the local minimum value of the Davies-Bouldin index (Davies and Bouldin, 1979) can be taken as an indicator of the optimum number of clusters.

There should also be a significant decrease in the sum of squares when the optimal number of clusters is reached. As a result, the number of clusters corresponding to the kink in the sum of squares curve and to a low Davies-Bouldin index can be taken as an indication of the optimal number of clusters.

One problem with k-means clustering is that it uses squared Euclidean distance as a dissimilarity measure and hence is not robust against outliers. For this reason k-medoids clustering, that does not have the above mentioned shortcoming, was also done. A second problem with k-means clustering is that, as the number of clusters changes, cluster membership can change in arbitrary ways. To overcome this problem, hierarchical clustering which does not possess this characteristic was also used.

3.1.2.2 K-medoids

The medoid of a cluster of points is the point with smallest average dissimilarity to all other points. This calculation is expensive, requiring all pair-wise dissimilarities within each cluster. However, the advantage is that the k-medoids algorithm does not use the original data but the dissimilarity measures (which can be calculated using Euclidean or Manhattan distance). This causes k-medoids to be more robust to outliers than k-means, where the squared Euclidean distance places the highest influence on the largest distances.

Algorithm (Hastie et al., 2001):

- (1) For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster:

$$\mathbf{i}_k^* = \underset{C(i)=k}{\operatorname{argmin}} \sum_{C(i')=k} D(x_i, x_{i'})$$

Then $m_k = x_{i_k^*}$, $k = 1, 2, \dots, K$ are the current estimates of the cluster center.

- (2) Given a current set of cluster centers $m_1, \dots, m_k$, minimize the total error by assigning each observation to the closest(current) cluster center:

$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} D(x_i, m_k)$$

- (3) Repeat steps 1 and 2 until the allocations do not change.

The partitioning around medoids (PAM) (Kaufman and Rausseeuw, 1990) algorithm, which first finds an initial set of medoids and then swaps points so that no single switch of an observation with a medoid will decrease the objective, was used in the analysis of our data. To obtain the optimal number of clusters for the sample, the silhouette validation method can be used as a measure of goodness for k-medoids just like the Davies-Bouldin index was used for k-means. To construct silhouettes $S(i)$ (Rousseeuw.P.J., 1987) the following formula is used:

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (3.4)$$

where a_i - average dissimilarity of i^{th} object to all other objects in the same cluster. b_i - minimum of average dissimilarity of i^{th} object to all objects in other cluster (closest cluster).

The overall average silhouette width for the entire plot is simply the average of the S_i for all objects in the whole dataset. The largest overall silhouette width gives the best clustering. Therefore, the number of clusters with largest overall average silhouette width is taken as the optimal number of clusters. Silhouette average width was calculated for initial number of clusters ranging from 2 to 16, and the corresponding number of clusters that gives the maximum average silhouette width was taken as the optimal number of clusters.

3.1.2.3 SOM - Self-Organizing Maps

SOM can be viewed as a constrained version of k-means clustering. Observations that are initially in high-dimension are mapped down to a two-dimensional coordinate system. K prototype vectors $m_j \in R^p$ (where p is the dimension of the data space - called *input* space) are scattered at random or initialized to lie in the two-dimensional principal component plane of the data in the input space. These prototype vectors are placed on a two-dimensional hexagonal-grid with numbering

starting in the lower left corner. Each of these K prototypes are parameterized with respect to an integer co-ordinate pair $l_j \in Q_1 \times Q_2$, and $Q_1 = \{1, 2, \dots, q_1\}$, similarly Q_2 , and $K = q_1 \cdot q_2$.

The K prototype vectors are also placed on a two-dimensional grid (called the *output space*) with numbering starting in the lower left corner. We used a hexagonal grid although a rectangular grid may also be used.

Each of the K prototypes can be parametrized with respect to an integer coordinate pair $l_j \in Q_1 \times Q_2$ where $Q_1 = \{1, 2, \dots, q_1\}$ and $K = q_1 \cdot q_2$. For some purposes, they are also labelled sequentially on reading from left to right moving up the rows. We find the closest prototype m_b (referred to as the Best Matching Unit - BMU) to x_i (an observation) in Euclidean distance in R^p and then move all neighbours m_j of m_b toward x_i , using (Mills and Norminton, 2007).

$$m_j \leftarrow m_j + \alpha(t) h(\|r_j - r_b\|) (x_i - x_j) \quad (3.5)$$

where α - learning rate (typically α is decreased from 1 to 0 over a few thousand iterations)

r - distance threshold (which is decreased from starting value R to 1 over a few thousand iterations)

$\|r_j - r_b\|$ - distance between two prototype vectors.

Note: The neighbours of m_b are defined to be all m_j that are *close* in the **output** space (less than some threshold or neighbourhood radius σ). The distance is defined in the space $Q_1 \times Q_2$ of integer topological coordinates of the prototypes, rather than in the feature space R^p . Updating moves the prototypes closer to the data (in the input space) and maintains a smooth two-dimensional spatial relationship between the prototypes. Typically $\alpha(t)$ is decreased from 1.0 to 0.0 and σ is decreased from some starting value to 1 over a few thousand iterations. The neighbourhood function $h(\|r_j - r_b\|)$ is used to give more weight to prototypes m_j that are closer to m_b than to those further away. If we take σ small enough so that each neighbourhood contains

only one prototype vector, the spatial connection between prototypes is lost and SOM is equivalent to k-means clustering (Mills and Norminton, 2007).

The first three principal components (representing 30% of the variability for yeastCC and 94% of the variability for Breast Cancer data) of the covariance matrix of the sample were used for clustering using SOM, thus the input space was three-dimensional. Components were standardized so that the values fell between 0 and 1. This standardization was needed in order that each of the first three principal components could be outputted as the three primary colours (RGB). This representation helped in obtaining the optimal number of clusters in the final configuration for each sampling procedure since the values from each principal component could be represented based on the intensity of colours to which that component was assigned. Prototype vectors of size 66 were randomly sampled from the principal components so a 11 by 6 hexagonal grid was chosen as the output space. First, the initial configuration was plotted, and after several iterations the final configuration was plotted to find the number of clusters. A similar procedure was done for the first three principal components obtained using the correlation matrix of the sample.

We used principal components obtained using both the covariance and correlation matrices of the data since they behave differently depending on the way variables were measured. Using the covariance matrix implies that only differences due to different means are removed from the original variables but the variances are not considered equal hence variables with larger variances will have a greater impact on the weights of the PCAs. If the variables studied are measured using the same scale, then using the covariance matrix to obtain the PCAs is encouraged so that the differences in the variances of the variables are preserved. If different scales are present, the correlation matrix is used to obtain the PCAs since in that way the original variables are all standardized to unit variance. To allow for all possibilities, we used both covariance and correlation matrices of the data to obtain the PCAs.

3.1.2.4 Hierarchical Clustering

There are two types of hierarchical clustering: divisive and agglomerative. Divisive clustering has not been studied nearly extensively as agglomerative method but a potential advantage of divisive over agglomerative is when there is a need to partition the data into several relatively small clusters. As our interest was not focused there, agglomerative clustering was used in our analysis.

Agglomerative clustering can be done in three ways - single linkage, complete linkage, and average linkage. Clusters produced by single linkage can violate the 'compactness' property that all observations within each cluster tend to be similar to one another; clusters produced by complete linkage violate the 'closeness' property whereby observations assigned to a cluster can be much closer to members of other clusters than they are to some members of their own cluster. Agglomerative using average linkage represents a compromise between using single linkage and complete linkage and hence was used for our data analysis. The dissimilarity measure used was Euclidean; the average dissimilarity was calculated using the following equation (Hastie et al., 2001; Speed, 2003).

$$d_{AB}(A, B) = \frac{1}{N_A N_B} \sum_{i \in A} \sum_{j \in B} d_{ij} \quad (3.6)$$

where N_A and N_B - number of observations in each group

3.1.3 Plaid Models

The above-mentioned traditional clustering methods were not designed to take into account the special characteristics of gene expression of microarray data. In these methods, each observation belongs to at least one cluster and to no more than one cluster, but in Plaid models (Lazzeroni and Owen, 2002), observations are allowed to be in more than one layer (cluster) or none at all. Allowing this possibility is

important, because a gene could participate in more than one function of cell process or none. Certain groups of genes may belong together in a cluster according to the way they expressed when a certain function is performed but at the same time, they might cluster with other genes for a different function. Plaid models allow clusters to overlap and allow a cluster of genes to be defined with respect to only a subset of samples, not necessarily with respect to all of them.

Let Y_{ij} = gene expression level for gene i and sample j , for $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, n$. The general Plaid model has the following form (Lazzeroni and Owen, 2002):

$$Y_{ij} \approx \mu_0 + \sum_{k=1}^K (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} K_{jk} \quad (3.7)$$

where k = each of the layers in the model.

K = total number of layers in the model.

μ_k = constant expression level within the layer.

$\rho_{ik} = 1$ if gene i is in the layer k and zero otherwise.

$K_{jk} = 1$ if sample j is in the layer k and zero otherwise.

α_{ik}, β_{jk} are structural parameters.

If α_{ik} and β_{jk} are used, then imposing the corresponding constraints $\sum_i \rho_{ik} \alpha_{ik} = 0$ and $\sum_j \kappa_{jk} \beta_{jk} = 0$ avoid over-parameterization.

Three different conditions can be imposed on this model.

- (1) The conditions $\sum_k \rho_{ik} = 1$ for all i , and $\sum_k \kappa_{jk} = 1$ for all j , mean that every gene and every sample is in exactly one cluster.
- (2) The conditions $\sum_k \rho_{ik} \geq 2$ for some i , or $\sum_k \kappa_{jk} \geq 2$ for some j results in some genes or samples being in more than one cluster.
- (3) If there are some genes or samples do not fit into any clusters, then the following conditions are implied: $\sum_k \rho_{ik} = 0$ for some i , or $\sum_k \kappa_{jk} = 0$ for

some j .

Imposing condition (1) results in traditional clustering; the other two satisfy the general approach for the clustering of gene expression of microarray data.

Some interpretations of the layers:

- If $\rho_{ik} = 1$ for all i , but κ_{jk} is not 1 for all j , then layer k describes a cluster of samples.
- $\kappa_{jk} = 1$ for all j , but ρ_{ik} is not 1 for all i , implies that layer k describes a cluster of genes.
- The values of α_{ik} and β_{jk} provide orderings of the effects of layer k upon the genes and samples.
- Genes with larger values of $|\mu_k + \alpha_{ik}|$ are affected more greatly under the conditions of layer k , compared to genes with smaller values. These effects are greater for samples with larger values of $|\mu_k + \beta_{jk}|$.
- If $\mu_k + \alpha_{ik}$ is positive, then the corresponding gene is up-regulated; if negative, then it is down-regulated within layer k .

(This model is discussed in detail with examples in Lazzeroni and Owen (2002)).

3.2 Results

Clustering techniques were applied on the samples of genes of size 1000 from each of the six different data sets obtained for both yeastCC and Breast Cancer so that the performance of k-means and k-medoids can be compared among these data sets and between different clustering methods. Sampling 20 times was adequate to see obvious high frequency for a certain number of clusters for most of the data sets in the case of both yeastCC and Breast Cancer. However, if there was no obvious

high frequency for a particular number of clusters based on using 20 samples, then sampling was increased to 100.

Dimension reduction techniques PCA (see Chapter 3.1.1.1) and ICA (see Chapter 3.1.1.2) were used to obtain these six different datasets. To reduce the data dimension, we retained the number of components that explained at least 80% of the variability in the original data. Since ICA components are not ordered, we used independent components obtained from principal components of the data. About 30 PCs were obtained in the case of yeastCC but 3 were enough for Breast Cancer to capture at least 80% of the variability. Therefore, ICA was used as a re-representation of the data especially in the case of yeastCC.

The following are the six different datasets obtained:

- Original data (Ori-dat)
- First 27 principal components for yeastCC and 3 principal components for Breast Cancer obtained using the correlation matrix of the corresponding data (PCA-Cor)
- First 33 principal components for yeastCC and 3 principal components for Breast Cancer obtained using the covariance matrix of the corresponding data (PCA-Cov)
- First 30 independent components for yeastCC and 3 independent components obtained using the corresponding original data (ICA-Ori)
- 33 independent components obtained from the first 33 principal components got using the covariance matrix of the data for yeastCC and 3 for Breast Cancer. (ICA-PCov)
- 27 independent components obtained from the first 27 principal components obtained using the correlation matrix of the data for yeastCC and 3 for Breast

Cancer. (ICA-PCor)

R-codes for the following analysis are in Appendix-A

3.2.1 K-means

The *k*-means clustering technique was applied on the samples from these six different data sets. Sum of squares and Davies-Bouldin Index were calculated for number of clusters ranging from 2-14. For each sample obtained from each of the above-mentioned data sets (see R-code in Chapter A.1.1), plots of sum of squared error (SS) vs number of clusters, and Davies-Bouldin Index (DBI) vs number of clusters were constructed. Using these two plots, the optimal number of clusters for the corresponding sample was determined (e.g. see Figure 3.1 obtained for a sample from yeastCC). The optimal number of clusters was indicated by the number of clusters corresponding to the kink in the curve for the plot of SS vs number of clusters, and by the cluster number corresponding to the local/global minimum point of the curve in the plot of DBI vs number of clusters. If these two values coincide then that value was taken as the optimal number of clusters. In the case of the yeast dataset on which Figure 3.1 is based, the global minimum for the DBI plot is 12. However, for the SS plot it is 4; that value corresponds to a local minimum in the DBI plot. The optimal number of clusters was taken as 4 for this particular sample (see detailed discussion in Chapter 3.1.2.1). In the case of the Breast Cancer dataset, the global minimum of the plot of the Davies-Bouldin Index vs number of clusters seems to coincide with that indicated by the SS plot and give the optimal number of clusters when samples were taken from the PCA based on the correlation matrix of the data.

The above-mentioned procedure was repeated 20 times for all six different data sets for each of yeastCC and Breast Cancer. The most frequent value was considered to be the optimal number of clusters for that dimension reduction method and data.

Figure 3.1: Sample plot of k-means

When there was no obvious most frequent value, the sampling procedure was repeated up to 100 times. This was done for the case of the original yeastCC data. Frequencies of the value three, four, and five remained almost the same even with this extended sampling. In all other cases, there was an obvious modal value. Table 3.1 presents the results for the yeastCC and Breast Cancer datasets.

DimensionReduction	YeastCC	BreastCancer
Ori-dat	4(0.4*)	3(0.5)
PCA-Cor	4(0.6)	3(0.6)
PCA-Cov	4(0.6)	4(0.55)
ICA-Ori	4(0.4)	3(0.75)
ICA-Pcor	3(0.35)	3(0.85)
ICA-Pcov	4(0.35)	3(0.75)

Table 3.1: Optimal number of clusters using k-means

* - Proportion given in parenthesis

From our analysis the optimal number of clusters for yeastCC is 4, and for Breast Cancer 3. The genes that belonged to each cluster are given in Appendix-B.

3.2.2 K-medoids

K-medoids clustering was done using both Euclidean and Manhattan dissimilarity measures. Twenty samples of size 1000 were taken from each of six different datasets. Average silhouette width vs number of clusters was plotted for number of clusters ranging from 2 to 16 (e.g. see Figure 3.2 for yeastCC). The number of clusters corresponding to the maximum average silhouette width was taken as the optimal number of clusters for that particular sample (see details in Chapter 3.1.2.2). In Figure 3.2 the optimal number of clusters is 5. The procedure was repeated 20 times, with the modal value taken as the optimal number of clusters for that data using this clustering method.

The optimal number of clusters obtained using both the Euclidean and Manhattan measures of distance are 2 for both yeastCC and Breast Cancer data regardless

Figure 3.2: Sample plot of k-medoids

of dimension reduction technique used.

3.2.3 SOM

The first three principal/independent components of the data were used in the analysis using SOM. A very high proportion of variability (94%) in the Breast Cancer dataset was represented using relatively few PCs (3) but the same is not true for yeastCC (only 30% of the variability was explained by the first three components). There are five different datasets used in this clustering analysis.

- The first three principal components obtained using the correlation matrix of the data (PCA-Cor3).
- The first three principal components using the covariance matrix of the data (PCA-Cov3).
- The first three independent components based on using original data (ICA-ori3).
- The first three independent components obtained using the above first category principal components (ICA-PCor3).
- The first three independent components obtained using the above second category principal components (ICA-Pcov3).

DimensionReduction	YeastCC	BreastCancer
PCA-Cor3	4(0.45*)	2(1.0)
PCA-Cov3	5(0.5)	1(1.0)
ICA-Ori3	5(0.75)	4(0.65)
ICA-Pcor3	5(0.45)	4(0.6)
ICA-Pcov3	5(0.65)	4(0.5)

Table 3.2: Summary of number clusters using SOM

Figure 3.3: Sample plot of SOM for yeastCC

Figure 3.4: Sample plot of SOM for Breast Cancer

Analysis was done by sampling 20 times on each of the above-mentioned data sets obtained using dimension reduction techniques on both yeastCC and Breast Cancer. Figure 3.3 shows output obtained for this analysis using principal components from the correlation matrix of the data for yeastCC. The number of clusters in Figure 3.3 is 4. A similar plot for Breast Cancer is in Figure 3.4 where the number of clusters is 2. Table 3.2 summarizes the results for all of the above categories.

3.2.4 Hierarchical Clustering

Agglomerative hierarchical clustering with average linkage was applied on samples of size 500 taken from the original data. The Euclidean dissimilarity measure was used. This procedure was repeated on 50 random samples of both yeastCC and Breast Cancer. Both dendrogram and heatmap were obtained for our data sets. The cuts made to get the number of clusters for yeastCC are in the range of Euclidean distances of 6-7.75 and 7.75-9, and for Breast Cancer in the range of Euclidean distances of 15-18 and 18-20.

For yeastCC, the most frequent number of clusters between 6-7.75 was 9; between 7.75-9 it was 5. For Breast Cancer, the most frequent number of clusters between 15-18 is 5; between 18-20 is 3. For both of our datasets, two sets of cluster values resulted. The results are shown in Figure 3.5.

For both yeastCC and Breast Cancer datasets, some genes ended up in a single gene cluster (see Appendix-B.2).

3.2.5 Plaid Models

Analysis using Plaid models (Lazzeronni and Owen, 2002) was done using their software and *Plaid* User's Guide (Owen, 2000) (detailed discussion in Chapter 3.1.3). The value of α_{ik} for the first 12 genes and the value of β_{jk} for the first 3 samples are shown for the first 2 layers for yeastCC in Table 3.3. Tables 3.4 and 3.5 show

Figure 3.5: Summary of clusters at different cuts in hierarchical clustering

corresponding results for Breast Cancer. (Results are available for the first 25 genes and 5 samples in the first five layers, and appear as Appendix-C).

$\mu = 0.165$	Layer 1	K-mean	$\mu = 0.088$	Layer 2	K-mean
α_i	Gene	Cluster#	α_i	Gene	Cluster#
0.965	YBR072W	2	0.949	YKR039W	4
0.862	YER150W	2	0.632	YKL096W	1
0.804	YFL012W	2	0.588	YHR137W	4
0.804	YCR007C	2	0.525	YML030W	4
0.739	YHR087W	4	0.525	YLR081W	1
0.710	YPL130W	2	0.516	YLR131C	4
0.702	YDR461W	2	0.498	YLR304C	4
0.640	YJL038C	2	0.494	YML100W	1
0.635	YDR273W	2	0.492	YJR048W	4
0.624	YFL051C	2	0.492	YLR079W	4
0.621	YGL204C	2	0.479	YFR053C	1
0.620	YGL032C	2	0.477	YOR332W	4
β_j	sample	class	β_j	sample	class
0.408	elu0	elu	0.248	cdc15.150	cdc15
0.404	cdc15.240	cdc15	0.230	cln3.2	cln
0.269	cdc15.220	cdc15	0.220	cln3.1	cln

Table 3.3: Top 12 genes and 3 samples in layer1 and layer 2 for yeastCC

The main difference between Plaid Models and traditional clustering is that a gene can be in more than one cluster/layer in Plaid Models. For Plaid models applied to yeastCC, when comparison was done for the top 25 genes and 5 samples that exist in the first 5 layers, results show that gene YML100W is in layer 1 and 2, and gene YLR131C is in layer 2 and 3; also sample cln3.1 belongs to layer 2 and 5, and samples cdc15.30 and cdc15.50 belong to both layer 3 and 5. That means that genes YML100W and YLR131C and the samples cln3.1, cdc15.30 and cdc15.50 exist in more than one cluster. Genes that belonged to *each layer* were compared to genes that belonged to *each cluster obtained using k-means clustering* (Appendix-B). In the case of yeastCC, 88% of layer 1 genes belonged to cluster 2 (of k-means); layer 2 genes belonged to all clusters except cluster 2; layer 3 genes belonged to cluster 1 and 3; 92% of layer 4 genes belonged to cluster 4; 84% of layer 5 genes belonged to

$\mu=0.165$	Layer 1	K-mean	$\mu=0.088$	Layer 2	K-mean
α_i	Gene	clus#	α_i	Gene	clus#
7.273	201492_s_at	2	1.760	220468_at	3
7.224	AFFX-hum_alu_at	2	1.748	204422_s_at	3
7.190	217398_x_at	2	1.732	220432_s_at	3
7.188	212581_x_at	2	1.689	218541_s_at	3
7.145	204892_x_at	2	1.688	206100_at	3
7.116	203107_x_at	2	1.683	214862_x_at	3
7.104	221798_x_at	2	1.679	205222_at	3
7.080	206559_x_at	2	1.678	211361_s_at	3
7.064	AFFX-r2-P1-cre-3_at	2	1.670	218602_s_at	3
7.061	213453_x_at	2	1.664	215172_at	3
7.031	201049_s_at	2	1.663	210286_s_at	3
6.970	211296_x_at	2	1.661	220609_at	3

Table 3.4: Top 12 genes in layer1 and layer 2 for Breast Cancer

β_j	sample	β_j	sample
0.138	016_HCC202.U133A.CEL	0.526	08.SUM225.CEL
0.105	CC_MCF10A.U133A.CEL	0.406	035_ZR75-1.U133A.CEL
0.085	035_ZR75-1.U133A.CEL	0.4.5	037_SUM-44PE.U133A.CEL

Table 3.5: Top 3 samples in layer1 and layer 2 for Breast Cancer

cluster 3. For Breast Cancer, 100% of genes in layer 1 belonged to cluster 2; 100% of layer 2 genes to cluster 3; but layer 3, 4, and 5 genes belonged to all clusters.

Sample effects for the first five layers are summarized for yeastCC data in Table 3.6 and for Breast Cancer in Table 3.7. The results in Table 3.6 do not show any significant results unlike the results in Table 3.7. The reason could be that there were many missing values which were imputed by sample mean for yeastCC. Values of the sample mean may not reflect the real characteristics of a gene for the samples in the case of yeastCC. The results in Table 3.7 show that layer 1 has significant sample effects from all of the samples. Layer 2 has significant sample effects from samples that come from the classes Lu and BaA, though not as significant as for layer 1. There are a few samples from class BaB that also have significant sample effects on layer 2. Layer 3 has sample effects only from all samples of class Lu, and layer 4 only from all samples of class BaB. The above results for the Breast Cancer dataset give the confidence that Plaid models can be used to classify a new sample with unknown class. If the new sample belonged to layer 3, we can say that it belonged to the class of Lu; if it belonged to layer 4, then to the class of BaB; if it belonged to layers other than 3 or 4, then we can say that it belonged to the class of BaA. Table 3.8 shows genes and samples belonging to more than one layer (for Breast Cancer), obtained using results of the top 25 genes and top 5 samples from the first 5 layers.

Table 3.6: sample effects ($\mu + \beta_j$) in the first five layers
for yeastCC

sample/layer	class	1	2	3	4	5
cln3.1	cln	-0.060	0.133	-0.262	-0.098	0.268
cln3.2	cln	-0.64	0.143	-0.389	0.073	0.117
Continued on next page						

Table 3.6 – continued from previous page

sample/layer	class	1	2	3	4	5
clb2.2	clb	-	-0.276	-0.152	-	-
clb2.1	clb	-	-0.113	0.044	-	-
alpha0	alpha	-	-	-	-0.006	-
alpha7	alpha	0.294	-0.270	-	-0.046	-
alpha14	alpha	0.194	-0.098	-	-0.018	-
alpha21	alpha	0.060	-0.031	-	-0.051	0.002
alpha28	alpha	-	-	-	-	0.113
alpha35	alpha	-	-	-	-	-
alpha42	alpha	-	-	0.093	-	-
alpha49	alpha	-	-	-0.026	-	-
alpha56	alpha	-	-	0.049	-	-
alpha63	alpha	-	-	-	-	-
alpha70	alpha	-	-	-	0.028	-
alpha77	alpha	-	-	-	-0.052	-0.011
alpha84	alpha	-	-	-	-	0.071
alpha91	alpha	-	-	-0.109	-	-0.109
alpha98	alpha	-	-	-	-	-
alpha105	alpha	-	-	-0.043	-	-
alpha112	alpha	-	-	0.006	-	-
alpha119	alpha	-	-	-0.032	-	-
cdc15.10	cdc15	-	-	0.411	-0.49	-
cdc15.30	cdc15	-	-	0.432	-0.526	0.135
cdc15.50	cdc15	-	-	0.248	-0.39	0.18
Continued on next page						

Table 3.6 – continued from previous page

sample/layer	class	1	2	3	4	5
cdc15.70	cdc15	-	-0.024	0.047	-	-
cdc15.80	cdc15	0.192	-	-0.325	-	0.065
cdc15.90	cdc15	-	-0.205	0.094	-	-
cdc15.100	cdc15	0.316	-	-0.223	-	-
cdc15.110	cdc15	-	-0.142	0.112	-	-
cdc15.120	cdc15	0.27	-	-0.394	-	-
cdc15.130	cdc15	-	-0.026	-	0.035	-
cdc15.140	cdc15	0.13	-	-	-0.126	-0.2
cdc15.150	cdc15	-	0.160	-	0.012	0.113
cdc15.160	cdc15	0.145	-	-	-	-0.244
cdc15.170	cdc15	-	0.040	-	-	0.074
cdc15.180	cdc15	0.380	-	-0.282	-	-0.014
cdc15.190	cdc15	-	-0.233	0.135	-	-
cdc15.200	cdc15	0.233	-	-0.309	-	-
cdc15.210	cdc15	-	-0.212	0.12	-	-
cdc15.220	cdc15	0.433	-	-0.380	-	-
cdc15.230	cdc15	-	-	-	0.106	-
cdc15.240	cdc15	0.569	-	-	0.040	-0.217
cdc15.250	cdc15	-	-0.317	-	0.288	-
cdc15.270	cdc15	-	-0.208	-	0.201	-
cdc15.290	cdc15	-	-0.317	-	0.364	-
cdc28.0	cdc28	0.286	-	-	-0.16	-
cdc28.10	cdc28	0.075	-	-	-0.242	-
cdc28.20	cdc28	-0.236	-	-	-	0.235
Continued on next page						

Table 3.6 – continued from previous page

sample/layer	class	1	2	3	4	5
cdc28.30	cdc28	-0.085	-	-0.080	-	0.216
cdc28.40	cdc28	-	-	-0.081	-	0.129
cdc28.50	cdc28	-	-	-0.101	-	-
cdc28.60	cdc28	-	-	-0.07	-	-
cdc28.70	cdc28	-	-	-0.074	-	-
cdc28.80	cdc28	-	-	-0.045	-	-
cdc28.90	cdc28	-	0.048	-	0.207	-
cdc28.100	cdc28	-	0.037	-	0.011	-
cdc28.110	cdc28	-	0.046	-	0.073	-0.097
cdc28.120	cdc28	-	-	-	-	-
cdc28.130	cdc28	-	0.052	-	-	-
cdc28.140	cdc28	-	-	-	-	-
cdc28.150	cdc28	-	0.008	-	-	-
cdc28.160	cdc28	-	-0.038	-	0.224	-
elu0	elu	0.573	-0.607	-	0.103	-
elu30	elu	-	-	-	-0.099	-
elu60	elu	-	-	-	-0.088	-
elu90	elu	-	-	-	-	-
elu120	elu	-	-	0.126	-	0.054
elu150	elu	-	-	0.062	-	0.019
elu180	elu	-	-	-0.025	-	0.048
elu210	elu	-	-	-	-	0.091
elu240	elu	-	0.077	-	-	-
elu270	elu	-	-	0.018	-	0.003
Continued on next page						

Table 3.6 – continued from previous page

sample/layer	class	1	2	3	4	5
elu300	elu	-	-	0.038	-	-0.021
elu330	elu	-	0.025	-	-	-
elu360	elu	-	-0.110	-	-	-
elu390	elu	-	-	-	-	-

Table 3.7: sample effects ($\mu + \beta_j$) in the first five layers
for Breast Cancer

sample/layer	class	1	2	3	4	5
08.600MPE.U133A.CEL	Lu	5.911	1.642	-0.622	-	-
01.AU-565.CEL	Lu	5.856	1.575	-0.319	-	0.105
01.BT474.U133A.CEL	Lu	5.920	1.619	-0.607	-	-
11.BT483.U133A.CEL	Lu	5.850	1.632	-0.541	-	0.301
044.CAMA-1.U133A.CEL	Lu	5.914	1.366	-0.473	-	-
020.HCC1007.U133A.CEL	Lu	5.885	1.580	-0.488	-	0.109
018.HCC1428.U133A.CEL	Lu	5.910	1.430	-0.484	-	-
016.HCC202.U133A.CEL	Lu	5.991	1.512	-0.870	-	-
021.HCC2185.U133A.CEL	Lu	5.873	1.677	-0.377	-	0.065
09.LY2.U133A.CEL	Lu	5.878	1.234	-0.381	-	-
02.MCF7.U133A.CEL	Lu	5.889	1.332	-0.412	-	-
024.MDA-MB134-VI.U133A.CEL	Lu	5.884	1.352	-0.360	-	-
08.MDA-MB361.U133A.CEL	Lu	5.859	1.637	-0.551	-	0.293
030.MDA-MB415.U133A.CEL	Lu	5.878	1.558	-0.404	-	0.081
Continued on next page						

Table 3.7 – continued from previous page

sample/layer	class	1	2	3	4	5
06.SKBR3.U133A.CEL	Lu	5.896	1.474	-0.510	-	-
041.SUM-185PE.U133A.CEL	Lu	5.866	1.650	-0.582	-	0.244
037.SUM-44PE.U133A.CEL	Lu	5.935	1.692	-0.664	-	-
038.SUM-52PE.U133A.CEL	Lu	5.843	1.444	-0.452	-	0.341
036.T47D.U133A.CEL	Lu	5.908	1.404	-0.458	-	0.517
09.UACC812.CEL	Lu	5.787	1.639	-0.440	-	-
035.ZR75-1.U133A.CEL	Lu	5.937	1.693	-0.640	-	-
034.ZR75-30.U133A.CEL	Lu	5.927	1.593	-0.616	-	-
ZR-75-B.CEL	Lu	5.857	1.446	-0.235	-	-
10.BT20.U133A.CEL	BaA	5.820	1.214	-	-	-
013.HCC1143.U133A.CEL	BaA	5.823	1.314	-	-	-
04.HCC1187.CEL	BaA	5.856	1.338	-	-	-0.306
014.HCC1569.U133A.CEL	BaA	5.832	1.037	-	-	-
022.HCC1937.U133A.CEL	BaA	5.799	1.540	-	-	0.067
015.HCC1954.U133A.CEL	BaA	5.819	1.212	-	-	-
019.HCC2157.U133A.CEL	BaA	5.819	1.223	-	-	-
017.HCC3153.U133A.CEL	BaA	5.817	1.288	-	-	-
02.HCC70.CEL	BaA	5.769	1.587	-	-	0.219
11.MDAMB-468.U133A.CEL	BaA	5.778	1.484	-	-	0.059
042.SUM-190PT.U133A.CEL	BaA	5.802	1.437	-	-	0.078
08.SUM225.CEL	BaA	5.721	1.813	-	-	0.461
05.BT549.U133A.CEL	BaB	5.795	0.584	-	0.088	-
07.HBL100.U133A.CEL	BaB	5.719	0.609	-	0.255	-
Continued on next page						

Table 3.7 – continued from previous page

sample/layer	class	1	2	3	4	5
05.HCC1500.CEL	BaB	5.911	1.244	-	-0.317	0.299
03.HCCC38.U133A.CEL	BaB	5.815	0.976	-	0.074	0.017
025_HS578T.U133A.CEL	BaB	5.750	0.490	-	0.272	-
CC_MCF10A_U133A.CEL	BaB	5.957	1.040	-	-0.452	0.467
07.MCF-12A.CEL	BaB	5.932	1.340	-	-0.473	0.511
023_MDA-MB157.U133A.CEL	BaB	5.779	0.606	-	0.163	-
10.MDAMB-231.U133A.CEL	BaB	5.877	0.770	-	-0.133	-
031_MDA-MB435.U133A.CEL	BaB	5.744	0.546	-	0.220	-
032_MDA-MB436.U133A.CEL	BaB	5.828	0.710	-	0.048	-
043_SUM-1315-MO2.U133A.CEL	BaB	5.760	0.454	-	0.241	-0.028
039_SUM-149PT.U133A.CEL	BaB	5.875	0.963	-	-0.042	-
040_SUM-159PT.U133A.CEL	BaB	5.873	0.687	-	-0.033	-

sample	class	layers
CC_MCF10A_U133A.CEL	BaB	1,5
035_ZR75-1.U133A.CEL	BaA	1,2
037_SUM-44PE.U133A.CEL	Lu	1,2
07.MCF-12A.CEL	BaB	1,5
08.SUM225.CEL	BaA	2,5
021_HCC2185.U133A.CEL	Lu	2,3
gene	-	layers
202391_at	-	3,4

Table 3.8: Samples/Genes belonging to more than one layers for Breast Cancer

4. CLASSIFICATION

4.1 Methodology

Classification is known as supervised learning, where for a given dataset the classes are known (i.e. the data are labelled) and the purpose is to formulate a basis from the learning set. This information is then used to build a classifier that can be applied to predict the class of future unlabelled observations. The performance of a classifier needs to be assessed on a given dataset before using it to predict the class of a new datum. To do so, data with known classes are divided into two groups known as the training set (we used 90% of the data) and the test set (we used 10% of the data). The training set is used to build a classifier. Using this classifier, the classes of the test set are predicted and subsequently the misclassification error rate is calculated with the preferred classifier being the one with the lowest misclassification error rate. On this basis, our study examines the suitability of the following classifiers on our two microarray datasets: Single classification trees (both *rpart* and *tree*), *adaBoost*, *C4.5*, Neural networks, Support Vector Machines, Random Forest and k-nearest neighbour.

However, in applying these standard procedures to microarray data, we encounter a problem. The above-mentioned classification procedures work well as long as the number of cases exceeds the number of variables. In the case of microarray data, the number of genes (variables) is in the thousands or tens of thousands, and the number of samples (cases) is only up to the hundreds. As a result, many of these standard classifiers do not work on these datasets. This can be dealt with in several ways - by selecting a subset of genes which are considered by experts to be important, by using purely (statistical) dimension reduction techniques, or by using a combination of the above two strategies. In this thesis, we have chosen not to restrict the genes to those chosen by experts but instead to use two more objective dimension

reduction techniques, Partial Least Squares (PLS) and Principal Components Analysis (PCA) with varimax rotation, to reduce the dimension of the dataset, in order to permit the application of these standard classifiers.

4.1.1 Dimension Reduction Algorithms

The number of variables far exceeds the number of cases for our two datasets (detail in Chapter 1.2). As a result, PLS and PCA with varimax rotation were separately used to reduce the number of variables.

4.1.1.1 Partial Least Squares (PLS)

PLS constructs components in such a way that covariance between the response variable y and the original predictor variable X is maximized.

$$w_i = \arg \max_{w'w=1} Cov(Xw, y) \quad (4.1)$$

subject to $t_i^T t_j = 0$, where $i \neq j$, and $t_k = w_k^T x$. PLS obtains the vectors of optimal weights w_i ($i = 1, \dots, K$) to form a small number of components that best predict the response variable y . To derive the components, $[t_1, t_2, \dots, t_k]$, PLS decomposes X and y to produce a bilinear representation of the data (Dai et al., 2006):

$$X = t_1 w_1 + t_2 w_2 + \dots + t_k w_k + E \quad (4.2)$$

and

$$y = t_1 q_1 + t_1 q_2 + \dots + t_k w_k + F \quad (4.3)$$

where $\mathbf{w}$'s are vectors of weights for constructing the PLS components $t = w^T x$, q 's are scalars, and E and F are the residuals. The values of $\mathbf{w}$ and $\mathbf{q}$ are estimated by regression. The first PLS component t_1 is estimated based on the covariance between

X and y. Every other component t_i is computed using the residuals of X and y from the previous step, which accounts for the variation left by the previous components (Dai et al., 2006). Therefore, the PLS components are uncorrelated and ordered. Zhu and Barnes (1995) have shown that the simple iteration algorithm for PLS regression is essentially identical with the SIMPLS algorithm. We used the add-on package *pls* in R with the option chosen as SIMPLS (R-code in Appendix-A).

The SIMPLS algorithm (de Jong, 1993) is as follows:

For each $h = 1, \dots, c$ where $A_0 = X'Y, M_0 = X'X, C_0 = I$, and c given,

- Compute q_h , the dominant eigen vector of $A_h' A_h$
- $w_h = A_h q_h, c_h = w_h' M_h w_h, w_h = w_h / \sqrt{c_h}$, and store w_h into W as a column.
- $p_h = M_h w_h$ and store p_h into P as a column.
- $q_h = A_h' w_h$, and store q_h into Q as a column.
- $v_h = c_h p_h$, and $v_h = v_h / \|v_h\|$
- $c_{h+1} = c_h - v_h v_h'$ and $M_{h+1} = M_h - p_h p_h'$
- $A_{h+1} = c_h A_h$

T of the SIMPLS is computed as $T = XW$ and B for the regression of Y on X is computed as $B = WQ'$ (www.statsoft.com/textbook/, date:Feb2009)

Before applying PLS on a given data set, 100 pairs of test and training sets were created for both yeastCC and Breast Cancer. The total number of cases was randomly divided into ten groups by sampling without replacement. These groups were taken as test sets and the corresponding remaining cases for each test set were taken as the training pair for that test set. One hundred pairs of test and training sets were created by repeating the above procedure ten times. This procedure was followed for both the yeastCC and Breast Cancer datasets. Each column of the training and

the test set was normalized to have mean zero and variance one. This transformation eliminated bias introduced by differences in scales of the several variables. The PLS model was built by feeding the original training set into the SIMPLS algorithm. The test set was obtained by feeding each instance of the test set data into the PLS model built using the training set.

The number of components is a free parameter that needs to be estimated; the optimal number of components that gives the highest predictive accuracy is needed. Following Tan et al. (2008) the maximum number of components used is four (4) times the square-root of number of cases in a dataset. Therefore, the dimension was reduced to 36 for yeastCC and to 30 for Breast Cancer for all pairs of training and test sets. The number of components tested was from 2 to either 30 or 36 (depending upon the dataset used) in steps of 2 to reduce the computational cost/time. Misclassification rates were computed using the leave-one-out cross-validation method for a selected classification technique using the training set. The number of components corresponding to the minimum misclassification rate was taken as the optimal number of components for that training set. This procedure was repeated on all 100 training sets. Then the most frequent value was taken as the optimal number of components for that classification method.

For a selected classification technique, the number of components of all training and test sets was reduced to the value of the optimal number. Then misclassification rates were computed for the 100 pairs and averaged to give a final misclassification rate for a selected classification method combined with PLS dimension reduction.

4.1.1.2 Principal Components Analysis with varimax rotation

The principal components of a dataset are obtained as explained in Chapter 3.1.1.1 using the correlation matrix of the data. Varimax rotation was done on these k number of principal components. The varimax criterion focuses on simplifying the

columns of the factor matrix. This method maximizes the sum of variances of required loadings of the factor matrix, consequently the maximum possible simplification is achieved if there are only ones and zeros in a single column. This tends to simplify interpretability of the resulting component.

First the correlation matrix was obtained for each of our datasets. Eigenvalues and eigenvectors were obtained using singular value decomposition of the correlation matrix. The plot of eigenvalue vs component number helped determine the number of components that explained most of the variability in the corresponding dataset. The first 72 components for yeastCC and the first 49 for Breast Cancer explained more than 99 percent of the variability in the dataset. The PCs were obtained by multiplying the original data by the corresponding number of eigenvectors. A procedure, similar to that followed in the case of PLS dimension reduction to create 100 pairs of test and training sets, was repeated after PCA dimension reduction with varimax rotation was done on both yeastCC and Breast Cancer datasets. The corresponding classification models were built using the training set; misclassification rates were computed using the corresponding test sets. The average of these 100 misclassification rates was taken as the misclassification rate for that classification technique when combined with PCA dimension reduction.

4.1.2 Classification Algorithms

Following either PLS or PCA dimension reduction, we used the following classification methods in our analysis: single classification trees, adaBoost, C4.5, Neural Networks (NNET), Support Vector Machines (SVMs), and Random Forest. We briefly describe each below.

4.1.2.1 Single Classification Trees

Here the basic idea was to build a tree by finding binary splits on variables that separate the data in the best way. Each split produces a left child and right child. A split that decreases the total impurity of the child node as compared to the parent node is considered to be the best split. Therefore, all available variables are tested for the best individual split value and then the variable that has the best is selected. The procedure is recursive. The cases at the left and right nodes are assessed as above to do further splits. A minimum size for leaves is used to avoid over-fitting (Mills and Norminton, 2007). Detailed discussion can be found in Hastie et al (2001). One common measure to determine the impurity is the deviance or entropy. The impurity at leaf t is given by,

$$I_t = -2 \sum_{j \in J} n_{tj} \log p_{j/t}$$

$$\text{where } p_{j/t} = \frac{p_{tj}}{p_t}$$

and the deviance at a node T is given by,

$$I(T) = \sum_{\text{leaf } t \in T} I_t$$

where p_{tj} - probability that a case reaches leaf t and that it is of class j . p_t - probability of reaching leaf t .

We used the libraries *tree* and *rpart* in R. Default settings, which are recursive partition and deviance/entropy as split criterion, were used in this study.

4.1.2.2 AdaBoost.M1

AdaBoost.M1 is a simple generalization of adaBoost for more than two classes. It is a procedure that combines the outputs of many weak classifiers to produce better results. (A classifier is said to be a weak classifier if it has error rate slightly better than random guessing.) Boosting sequentially applies the weak classification algorithm to repeatedly modified versions of the data, producing a sequence of weak classifiers $G_m(x)$, $m = 1, 2, \dots, M$ (Hastie et al., 2001). As iterations proceed, each

successive classifier is forced to concentrate on those training observations that are misclassified by previous classifiers in the sequence. The predictions from all of them are combined through a weighted majority vote to produce the final prediction.

The following is the algorithm for AdaBoost.M1 (Hastie et al., 2001):

- (1) Initialize the observation weights $w_i = 1/N, i = 1, 2, \dots, N$.
- (2) For $m = 1 \dots M$:
 - Fit a classifier $G_m(x)$ to the training data using the weights w_i .
 - Compute $err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}$
 - Compute $\alpha_m = 1/2 (\ln((1 - err_m) / err_m))$
 - Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \dots, N$.
- (3) Output $G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]$

For our analysis, we used the function *adaboost.M1* in the *adabag* library with default settings. Classification trees was used as the single classifier (weak learner) in *adaboost.M1*.

4.1.2.3 C4.5

C4.5 is a decision tree algorithm introduced by Quinlan (1992). Among decision tree algorithms, C4.5 and its predecessor ID3 are the most popular. They adopt a divide-and-conquer approach to construct decision trees. C4.5 is capable of producing trees with varying numbers of branches per node. Particularly in the cases of categorical variables, it produces a separate branch for each value of the categorical attribute.

Decision tree algorithms begin with a set of cases and create a tree data structure that can be used to classify new cases. Each case is described by a set of

attributes which can have numeric or symbolic values. Each training case contains a label representing the name of a class. Each internal node of a decision tree contains a test (rule) which is used to decide what branch to follow from that node. If the test is true, the case will proceed down the left branch; if not, the case proceeds down the right branch. C4.5 and its predecessor use formulae based on information theory to evaluate the goodness of a test; in particular they choose a test that extracts the maximum amount of information from a set of cases. The leaf nodes contain class labels instead of tests. In classification mode, when a test case reaches a leaf node, C4.5 classifies it using the label stored there.

Modifications to handle unknown or missing values are as follows: Cases with unknown values are ignored when computing the information content; information gain (details in Chapter 2 of Quinlan, 1992) for a variable X is then multiplied by the fraction of cases for which the value of X is known. If the variable X had missing values for a large fraction of cases, then information gained by testing X at a node will be quite small. The default splitting criterion used by C4.5 is gain ratio which produces very bushy decision trees. Quinlan, (1992) provides further detail on these matters.

Information gain is given by (Ruggieri, 2002),

$$gain = info(T) - \sum_{i=1}^s \frac{|T_i|}{|T|} \times info(T_i)$$

where

$$info(T) = - \sum_{j=1}^{NClass} \frac{freq(C_j, T)}{|T|} \times \log_2 \left(\frac{freq(C_j, T)}{|T|} \right)$$

where T = the set of cases associated at the node.

$freq(C_i, T)$ = weighted frequency is computed as number of cases in T whose class is C_i for $i \in [1, NClass]$

s - number of children at a decision node.

Information gain ratio of the splitting $T_1, \dots, T_s$, which is the ratio of information gain to its split information is given by,

$$Split(T) = - \sum_{i=1}^s \frac{|T_i|}{|T|} \times \log_2 \left(P \frac{|T_i|}{|T|} \right)$$

C4.5 uses pruning to avoid over-fitting. Its pruning method is based on estimating the error rate at each node of the decision tree. If replacing a subtree (at a certain node) with a leaf node produces a decision tree with lower error rate, then the subtree at that node is replaced with a leaf node.

This classification tree algorithm runs the fastest compared to all the other classification algorithms we used and it is easy to implement. The add-on package *RWeka* in R was used to apply the function `J48` for our study.

4.1.2.4 Neutral Networks (NNET)

The term neutral network consists of a large class of models and learning methods. These can be considered as two-stage classification models which contain units (nodes) called a hidden layer in the middle of the network (Hastie et al., 2002). A hidden layer with at least one unit is required if a class boundary is curved. For a range of complexity, we chose 2, 5, and 10 hidden units with a varying number of iterations (in the range of 10 to 900) of the neural net. We examined the effect on misclassification rate of the number of hidden layers and number of iterations.

Neural networks have unknown parameters called weights. To fit the model to the training data, values of the weights need to be selected. The algorithm never moves when the weights are chosen to be zero. On the other hand, large weights often result in poor solutions. Weights are usually chosen to be random values near zero so that the model starts out as linear and becomes nonlinear as the weight increases (Hastie et al., 2002). The weights were chosen to be in the range of (-0.1, 0.1) for our analysis.

The magnitudes of weights decrease as the weight decay value increases. Increasing weight decay results in smoother boundary curves (Mills and Nominton, 2007). As a result, having the value of weight decay other than zero helped avoid

over-fitting and having not too large a weight decay improved the prediction error.

We have used the default setting for *nnet* R-function which uses the squared-error as entropy. Typically, we do not want the global minimizer of the sum-of-squared error function, as this is likely to be an over-fit function. Over-fitting was avoided indirectly by early stopping (explained in detail in Hastie et al., 2001).

Hidden layers of size 2, 5, and 10 were used for both of our datasets. The numbers of iterations used were 100, 300, 500, 700, and 900 for both datasets, when PCA with varimax rotation was used. The above mentioned iterations were used for yeastCC data with PLS dimension reduction, but the number of iterations 10, 30, 50, 70, and 100 were used for Breast Cancer with PLS dimension reduction. Dataset characteristics such as balanced classes and highly correlated genes may account for the fact that a very small number of iterations was adequate to obtain minimum misclassification rates for the case of Breast Cancer with PLS. The library called *nnet* in R was used in our analysis to utilize the function *nnet*.

4.1.2.5 Support Vector Machines (SVMs)

Support Vector Machines were introduced by Vapnik (1979, 1998). SVMs were initially designed to deal with binary classification problems; they can be generalized to deal with multi-class problems by considering several binary problems simultaneously using a one-against-all approach.

SVM looks for the best hyperplane in order to separate classes in the learning set. The hyperplane could be in the original space or a higher-dimensional space. As a result, there are linear support vector machines and nonlinear support vector machines. Linear support vector machines can be grouped into the linear separable case and the linear nonseparable case.

If the training data consists of N pairs $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, with $x_i \in \mathbb{R}^p$. Define a hyperplane by $\{x : f(x) = x^T \beta + \beta_0 = 0\}$ where β is a unit vector.

Support vector criterion for separated data can be expressed as follows:

$$\min_{\beta, \beta_0} \|\beta\| \quad (4.4)$$

subject to $y_i (x_i^T \beta + \beta_0) \geq 1, i = 1, \dots, N$

Support classifier defined for the non-separable case is as follows:

$$\min \|\beta\| \text{ subject to } \begin{cases} y_i (x_i^T \beta + \beta_0) \geq 1 - \xi_i \forall i, \\ \xi_i \geq 0, \sum \xi_i \leq \text{constant}. \end{cases}$$

The above two equations are quadratic with linear inequality constraints, hence they are convex optimization problem. Computationally above equation can be re-expressed in the equivalent form

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + \gamma \sum_{i=1}^N \xi_i \quad (4.5)$$

subject to $\xi_i \geq 0, y_i (x_i^T \beta + \beta_0) \geq 1 - \xi_i \forall i$

where γ replaces the constant for non-separable case; the separable case corresponds to $\gamma = \infty$

In the linear case, support vector training amounts to maximizing the dual (Speed, 2003).

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

with respect to the α_i , subject to constraints,

if linear separable case

$$\alpha_i \geq 0$$

if linear nonseparable case

$$0 \leq \alpha_i \leq \gamma$$

and

$$\sum_i \alpha_i y_i = 0$$

for all i . The solution is given by

$$\beta = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

where S is the set of support vectors. Detailed discussion can be found in Hastie et al (2001), and Speed (2003).

Often misclassification rates are minimized if nonlinear class boundaries are used. However, mapping data into a higher dimensional space and then reducing it to a linear case can lead to simple solutions. Generally, linear boundaries in the enlarged space achieve better training-class separation and translate to nonlinear boundaries in the original space.

We have used the *e1071* add-on package in R. There are four different kernels in `svm` function in R - linear, polynomial of varying degrees, radial basis, and sigmoid. We used the kernel that gave the minimum misclassification rate for the corresponding dataset.

4.1.2.6 Random Forest

Random Forest grows many single classification trees. A new object from an input vector is sent down each of the trees in the forest. Each tree classifies that object and the final classification is determined by majority vote. (The idea of Random Forest is applicable not only to trees but also to any other classifier.) Each tree in the forest is grown as follows (Breiman and Cutler, 2008):

- Let the number of cases in the training set be n . Sample n cases at random with replacement from the original data. This sample becomes the training set for growing the tree.
- Let G is the number of input variables. Select g variables at random out of the G , and use the best split on these g to split the node.

- Each tree is grown to the largest extent possible *without* pruning.

The only adjustable parameter to which Random Forests is sensitive is the number of input variables g . A tree with a low error rate is considered to be a strong classifier. Increasing g increases both the correlation between any two trees and the strength. However, increasing the correlation increases the forest error rate while increasing the strength decreases forest error rate. Thus we seek an optimal value of g to minimize the forest error rate. The default value of g is taken to be the square-root of the number of variables in the *randomForest* function (R help-file). Function *randomForest* with default settings available in the add-on package *randomforest* in R was used for this analysis.

4.1.3 K - Nearest Neighbour

K-NN methods are based on a distance function (i.e our research used Euclidean distance in R). This method classifies a new observation on the basis of the learning set by finding the k closest observations in the learning set and predicts the class by majority vote (i.e choose the class that is most common among those k neighbours). However, the choice of k , the number of neighbours, has to be estimated for the relevant data. K-NN classifiers are often criticized because of their heavy computing time and storage requirements for large learning sets (Speed, 2003).

The number of neighbours can have a large impact on performance of the classifier and needs to be chosen carefully. Leave-one-out cross-validation was done to select a suitable number of neighbours. For every choice of k , each observation in the learning set was treated in turn as if its class were unknown and it was classified using the nearest neighbour rule. The classification of the learning set observations was then compared to the truth to produce the cross-validation error rate and the k corresponding to the smallest cross-validation error was retained.

4.1.4 Design of Experiments

To examine whether results would differ significantly based on the microarray dataset used, two different microarray datasets (yeastCC and Breast Cancer) were used in our analyses. Dimension reduction techniques PLS and PCA were applied on both datasets prior to using any classification technique. Classification methods compared in this research were adaBoost, C4.5, single classification tree using both *rpart* and *tree*, neural network, and support vector machines. Misclassification rate was used to assess suitability of these methods for these two datasets. These misclassification rates were summarized as proportions and analyzed to determine statistically significant differences. Following Mosteller and Youtz (1961) these proportions were transformed to stabilize the variance approximately and to make proportions nearly normal.

$$p = \begin{cases} 1/4n & \text{if } y = 0; \\ y/n & \text{if } 1 \leq y \leq (n-1); \\ 1 - 1/4n & \text{if } y = n. \end{cases}$$

where y - number of misclassifications, n - total number of trials

$$\theta = \arcsin(\sqrt{p}) 180/\Pi$$

where $\Pi = 3.14159$

Among all of the above classification techniques, the misclassification rate for neural networks (NNET) can be improved significantly by altering the parameter values. This was investigated further by varying the number of hidden layers and iterations used to find a minimum classification rate for NNET. The numbers of hidden layers used were two (2), five (5), and ten (10). For the two datasets (yeastCC and Breast Cancer) with PCA dimension reduction and for yeastCC with PLS dimension reduction, the numbers of iterations used were 100, 300, 500, 700, and 900. However, for Breast Cancer data with PLS dimension reduction, results did not require

that large a number of iterations; 10, 30, 50, 70, and 100 iterations were adequate to obtain a minimum misclassification error. With PCA dimension reduction and using the two datasets as blocks, we examined the effects due to number of nodes in the hidden layer, number of iterations, and the interaction of these two factors using a factorial treatment structure. Also, using just the yeastCC dataset, we compared results under each of the two dimension reduction methods PCA and PLS. SAS-codes for the above analysis are in Appendix-A.

4.2 Results

As explained in Chapter 4.1.1.1 and Chapter 4.1.1.2, 100 pairs of test and training datasets were formulated in the cases of both PLS and PCA dimension reduction. The misclassification rate was calculated for each of the 100 datasets using various classification methods; then the average misclassification rate was obtained for each classification technique. This was done for both yeastCC and Breast Cancer datasets. R-codes are in Appendix-A

4.2.1 Single Classification Trees

Two add-on libraries available in R were used in this analysis, as explained in Chapter 4.1.2.1. Table 4.1 summarizes results for both dimension reduction methods. Regardless of the dimension reduction technique used, the misclassification rate was lower for Breast Cancer than for yeastCC. Also, dimension reduction technique PCA gave better results than PLS for both yeastCC and Breast Cancer datasets.

4.2.2 AdaBoost

Table 4.2 shows the average misclassification rate for the adaBoost technique (discussed in detail in Chapter 4.1.2.2). AdaBoost is an ensemble classification technique which uses single classification tree as a base classifier. AdaBoost improved the

misclassification rate for both yeastCC and Breast Cancer when PCA was used as a dimension reduction technique but not when PLS dimension reduction was used.

4.2.3 C4.5

C4.5 is the only classification technique that gave minimum misclassification rates regardless of the dimension reduction technique used or the data set being analyzed. Table 4.3 presents average misclassification rates under C4.5 (detailed discussion in Chapter 4.1.2.3).

4.2.4 Neutral Networks (NNET)

Neutral Networks have two parameters - the number of hidden layers and the number of iterations - which may be altered to improve misclassification rate. We used three levels each for the number of hidden layers (hl) and the number of iterations (iter) to examine the effect on misclassification rate. (see detailed discussion in Chapter 4.1.2.4). Table 4.4 summarizes misclassification rates for yeastCC and Breast Cancer with PCA dimension reduction technique for varying numbers of hidden layers and iterations (Note: *values within brackets are transformed values of the misclassification rates).

Analysis was conducted taking the two datasets (yeastCC and Breast Cancer, each with PCA dimension reduction) as blocks, and number of hidden layers together with number of iterations as treatments, using a factorial structure. Hidden layers had 3 levels: 2, 5, and 10; number of iterations had 5 levels: 100, 300, 500, 700, and 900. Analysis of the misclassification rates was done on the transformed data, as explained in Chapter 4.1.4. As the transformed values of misclassification rates are proportional to the misclassification rates, comparison among iterations, hidden layers and data sets can be done with transformed values. The model adequacies on the transformed data were satisfied.

Single Tree-rpart	PCA	PLS
YeastCC	0.3828	0.9007
BreastCancer	0.1780	0.3760
Single Tree-tree	PCA	PLS
YeastCC	0.3443	0.9013
BreastCancer	0.1980	0.2560

Table 4.1: Misclassification Rate for Single Classification Tree

adaBoost	PCA	PLS
YeastCC	0.0960	0.9057
BreastCancer	0.0930	0.3760

Table 4.2: Misclassification Rate for adaBoost

C4.5	PCA	PLS
YeastCC	0.0057	0.0086
BreastCancer	0.0	0.0

Table 4.3: Misclassification Rate for C4.5

	YeastCC PCA			Breast Cancer PCA		
	hl2	hl5	hl10	hl2	hl5	hl10
iter100	0.9001 (71.57*)	0.9591 (78.33)	0.961 (78.61)	0.3758 (37.81)	0.2134 (27.51)	0.158 (23.42)
iter300	0.9066 (72.2)	0.9503 (77.12)	0.967 (79.53)	0.3126 (34.0)	0.1254 (20.74)	0.0814 (16.58)
iter500	0.9008 (71.64)	0.9464 (76.61)	0.9871 (83.48)	0.282 (32.07)	0.0926 (17.72)	0.0784 (16.26)
iter700	0.9157 (73.12)	0.946 (76.56)	0.983 (82.51)	0.2733 (31.52)	0.0898 (17.44)	0.076 (16)
iter900	0.924 (74.0)	0.9566 (77.98)	0.983 (82.51)	0.276 (31.7)	0.1 (18.43)	0.076 (16.0)

Table 4.4: Misclassification rates and transformed values for NNET using PCA

Source	DF	Sum of Squares	Mean Square	F value	Pr > F
Model	21	22322.39	1062.97	1608.68	<0.0001
Error	8	5.29	0.66		
Corrected Total	29	22327.68			
Source	DF	Type III SS	Mean Square	F value	Pr > F
dat	1	21256.60	21256.60	32169.3	<0.0001
iter	4	46.24	11.56	17.5	0.0005
hl	2	98.10	49.05	74.23	<0.0001
dat*iter	4	94.87	23.72	35.90	<0.0001
dat*hl	2	807.70	403.85	611.18	<0.0001
iter*hl	8	18.86	2.36	3.57	0.0454

Table 4.5: ANOVA Table yeastCC and Breast Cancer with PCA for NNET classification

dat	p-value
Breast Cancer	<0.0001
YeatCC	0.0789

Table 4.6: Data*iteration Effect Sliced by data for transformed misclassification

The SAS System

Figure 4.1: Interaction plot for data and iterations for yeastCC and Breast Cancer with PCA

Figure 4.2: Interaction plot for data and hidden layers for yeastCC and Breast Cancer with PCA

The SAS System

Figure 4.3: Interaction plot for iterations and hidden layers for yeastCC and Breast Cancer with PCA

The following deductions can be made using Tables 4.5, 4.6, and Figures 4.1, 4.2, 4.3 for $\alpha = 0.05$.

- (1) Table 4.5 gives the analysis of variance results. As the p-values for main effects and interaction effects are less than α , they are all statistically significant.
- (2) Tukey's Honest Significance Difference (HSD) test shows that there is a significant difference due to number of iterations (with number of iterations of 100 significantly differing from all other numbers of iterations) and due to number of hidden layers (with 2 hidden layers significantly differing from 5 and 10.) (Results are available but not presented here).
- (3) Table 4.6 presents the dataset-iteration interaction effect sliced by dataset. Iteration effect in the Breast Cancer dataset is significant (p-value $< 0.0001 < \alpha = 0.05$). Iteration effect for yeastCC is slightly significant ($0.05 < \text{p-value} = 0.08 < 0.1$). The interaction effect plot for data and iteration is given in Figure 4.1.
- (4) Hidden layer effect is significant for both Breast Cancer and yeastCC datasets with p-values being < 0.0001 (results not shown here). However, the plot (Figure 4.2) of interaction effect of dataset and hidden layers shows that increasing the number of hidden layers improves the misclassification rates for Breast Cancer, but worsens it for yeastCC. A minimum misclassification rate was obtained for yeastCC with 2 hidden layers, and for Breast Cancer data with 10 hidden layers.
- (5) Figure 4.3 gives the transformed misclassification rates for iteration effect on all hidden layers. Misclassification rate (and transformed values) for yeastCC is higher at a particular number of hidden layers compared to Breast Cancer. (In Figure 4.3, there are two same colour ticks for hl10. The tick correspond-

ing to the high value is for yeastCC and the low value is for Breast Cancer. The same holds for hl2 and hl5.) Minimum misclassification rate was obtained for yeastCC using 2 hidden layers and iterations other than 700 and 900 (Figure 4.3), but for Breast Cancer using 10 hidden layers and iterations other than 100.

For neural networks, using the above deductions and Table 4.4 for yeastCC with PCA dimension reduction, the final misclassification rate was chosen at 2 hidden layers and 500 iterations; for Breast Cancer with PCA dimension reduction, the final misclassification rate was chosen at 10 hidden layers and 700 iterations.

A similar analysis, changing the number of hidden layers and iterations, was made for yeastCC under two different dimension reduction techniques (drdc) PLS and PCA. The two different dimension reduction methods were considered to be blocks. Number of iterations was taken at 5 levels and hidden layers at 3 levels. Analysis was again done on transformed misclassification rates. The model adequacies were satisfied. The analysis of variance table for NNET for yeastCC when PLS or PCA was used is given in Table 4.8.

	YeastCC PCA			YeastCC PLS		
	hl2	hl5	hl10	hl2	hl5	hl10
iter100	0.9001 (71.57*)	0.9591 (78.33)	0.961 (78.61)	0.281 (32.01)	0.2063 (27.01)	0.2064 (27.02)
iter300	0.9066 (72.2)	0.9503 (77.12)	0.967 (79.53)	0.2418 (29.45)	0.2056 (26.96)	0.2126 (27.46)
iter500	0.9008 (71.64)	0.9464 (76.61)	0.9871 (83.48)	0.2464 (29.76)	0.2056 (26.96)	0.2121 (27.42)
iter700	0.9157 (73.12)	0.946 (76.56)	0.983 (82.51)	0.2393 (29.29)	0.2054 (26.95)	0.2126 (27.46)
iter900	0.924 (74.0)	0.9566 (77.98)	0.983 (82.51)	0.238 (29.2)	0.2054 (26.95)	0.2071 (27.07)

Table 4.7: NNET Misclassification rates and transformed values for yeastCC using PCA

Figure 4.4: Interaction plot of dimension reduction techniques and hidden layers for yeastcc with PCA and PLS

The number of iterations and its interaction with other factors or blocks are insignificant. The HSD test shows that there is a significant difference between the means of PLS and PCA, and between the means of hidden layer of size 10 and the others, but insignificant differences between the means for different numbers of iterations. The results are presented in Table 4.9.

Figure 4.4 shows that neural networks with PCA gave minimum misclassification with 2 hidden layers; with PLS it was minimum with 5 hidden layers. Overall minimum misclassification rate for yeastCC with PCA was obtained for 2 hidden layers and 500 iterations (Table 4.7); for yeastCC with PLS, the minimum was obtained for 5 hidden layers and 700 iterations (Table 4.7).

Neural networks applied to Breast Cancer with PLS dimension reduction achieved minimum misclassification with 5 hidden layers and 30 iterations (Table 4.10).

4.2.5 Support Vector Machines

SVM misclassification error rates vary as the type of kernel used is altered. Linear, polynomial of degree 3 and degree 4, radial, and sigmoid kernels were used to find the kernel that gives the minimum error rate (Chapter 4.2.5). Table 4.11 summarizes these error rates.

YeastCC with PCA dimension reduction had the same misclassification rate regardless of the kernel used. Minimum misclassification was obtained for yeastCC with PLS using kernel of polynomial of degree 4; for Breast Cancer with PCA using kernel of polynomial of degree 4; and for Breast Cancer with PLS, using sigmoid kernel.

4.2.6 Random Forest

Random Forest did not work on the yeastCC data because of the presence of classes with an extremely small number of cases, but it did work on the Breast Cancer

Source	DF	Sum of Squares	Mean Square	F value	Pr > F
Model	21	18239.970	868.570	680.62	<0.0001
Error	8	10.209	1.276		
Corrected Total	29	18250.178			
Source	DF	Type III SS	Mean Square	F value	Pr > F
drdc	1	17997.701	17997.701	14103.2	<0.0001
hl	2	50.108	25.054	19.60	<0.0001
drdc*hl	2	171.692	85.846	67.16	<0.0001

Table 4.8: ANOVA Table NNET classification for yeastCC with PLS and PCA

Tukey Grouping	Mean	Dimension Reduction Method
A	0.950	pca
B	0.221	pls
Tukey Grouping	Mean	Iterations
A	0.640	900
A	0.632	700
A	0.632	500
A	0.623	300
A	0.628	100
Tukey Grouping	Mean	Hidden Layers
A	0.660	10
B	0.623	5
B	0.608	2

Table 4.9: Results of Tukey's HSD test for Misclassification rates for yeastCC with PLS and PCA

BreastCancer	iter10	iter30	iter50	iter70	iter100
hl2	0.0884	0.00158	0	0	0
hl5	0.0546	0	0	0	0
hl10	0.0506	0	0	0	0

Table 4.10: Misclassification Rate for NNET using PLS for Breast Cancer

	YeastCC		Breast Cancer	
	PCA	PLS	PCA	PLS
linear	0.6671	0.8986	0.328	0.078
poly3	0.6671	0.9686	0.592	0.52
poly4	0.6671	0.66	0.328	0.52
radial	0.6671	0.8986	0.498	0.044
sigmoid	0.6671	0.9986	0.616	0.008

Table 4.11: Misclassification error rate for SVM

dataset. Table 4.12 summarizes the result for Breast Cancer.

4.2.7 K-Nearest Neighbours

In the case of yeastCC data, the best k was found to be 1. In the case of 1-nn classification, each training point is a prototype. Because it uses the training point that is closest to the query point, the bias of the 1-nn estimate is often low but the variance is high. The decision boundary of 1-nn is very irregular compared to others. The summary of the results obtained by following the methods explained in Chapter 4.1.3 is shown in Table 4.13.

4.2.8 Design of Experiments

An overall analysis of the classification results using a design of experiment analysis is presented here. Table 4.14 summarizes the misclassification rates; their transformed values shown within parentheses.

Table 4.15 presents the analysis of variance Table for overall analysis for this study. The model used was as explained in Chapter 4.1.4.

The following deductions can be made using Tables 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, and Figure 4.5.

- (1) The model (Table 4.15) was significant with $p\text{-value } (0.0057) < \alpha (0.05)$. Dataset and classification method were found to be significant, but not dimension reduction technique. However, the interaction effect of dimension reduction and classification method was significant with $p\text{-value of } 0.0386$ (Table 4.15) and hence dimension reduction was left in the model.
- (2) Table 4.16 presents Duncan's Multiple Range base on $\alpha = 0.05$ for classification methods, and Tukey grouping for dimension reduction techniques used (YeastCC and Breast Cancer datasets are blocks). Classification methods

RandomForest	PCA	PLS
Breast Cancer	0.044	0.0454
YeastCC	n/a	n/a

Table 4.12: Misclassification Rate using Random Forest

K-NN	knn.cv	LOO	ten-fold
Breast Cancer	3	3	1
YeastCC	1	1	1

Table 4.13: Best-k using k-nn methods

	yeastCC		Breast Cancer	
	PLS	PCA	PLS	PCA
adaBoost	0.9057 (72.12)	0.096 (18.05)	0.376 (37.82)	0.093 (17.76)
c4.5	0.0086 (5.32)	0.0057 (4.33)	0 (2.87)	0 (2.87)
rpart	0.9007 (71.63)	0.3828 (38.22)	0.376 (37.82)	0.178 (24.95)
tree	0.9013 (71.69)	0.3443 (35.93)	0.256 (30.4)	0.178 (24.95)
nnet	0.2054 (26.95)	0.9008 (71.64)	0 (2.87)	0.076 (16)
svm	0.66 (54.33)	0.667 (54.76)	0.044 (12.11)	0.328 (34.94)

Table 4.14: Misclassification rates summarized

Source	DF	Sum of Squares	Mean Square	F value	Pr > F
Model	12	11012.02	917.67	5.08	0.0057
Error	11	1987.0	180.64		
Corrected Total	23	12999.01			
Source	DF	TypeIII SS	Mean Square	F value	Pr > F
dat	1	3422.24	3422.24	18.95	0.0012
metd	5	4191.29	838.26	4.64	0.0160
drdc	1	231.57	231.57	1.28	0.282
drdc*metd	5	3166.91	633.38	3.51	0.0386

Table 4.15: ANOVA Table for Overall Analysis

Duncan's grouping	classification method	Mean
A	classification tree:rpart	0.468
A	classification tree:tree	0.426
A	support vector machine	0.367
A	adaBoost	0.353
A	neural network	0.240
B	c45	0.004
Tukey grouping	dimension reduction method	Mean
A	PLS	0.327
A	PCA	0.231

Table 4.16: Mean of misclassification rates for data set, classification method, and dimension reduction techniques

drdc	p-value
PCA	0.0744
PLS	0.0096

Table 4.17: Drdc*metd Effect Sliced by drdc for transformed misclassifications

metd	p-value
adaBoost	0.019
c45	0.971
NNET	0.055
rpart	0.113
SVM	0.285
tree	0.154

Table 4.18: Drdc*metd Effect Sliced by metd for transformed misclassification

drdc	method	LSMean
PCA	adaBoost	0.094
PCA	c45	0.004
PCA	NNET	0.480
PCA	rpart	0.274
PCA	SVM	0.500
PCA	tree	0.257
PLS	adaBoost	0.670
PLS	c45	0.005
PLS	NNET	0.066
PLS	rpart	0.666
PLS	SVM	0.246
PLS	tree	0.605

Table 4.19: Least Squares Means for interaction effect of drdc and method

Figure 4.5: Interaction plot of dimension reduction techniques and methods

such as single classification tree (both *rpart* and *tree*), support vector machine, adaBoost, and neural network fell into one group; C4.5 into another. Dimension reduction techniques PCA and PLS do not produce significantly different results.

- (3) Table 4.17 presents the interaction effect of dimension reduction and classification method sliced by dimension reduction. As the p-value for PLS is 0.0096 ($< \alpha$), means of the classification methods differ significantly in the case of PLS. The p-value for PCA is 0.0744, indicating means of classification methods differ somewhat in the case of PCA.
- (4) Table 4.18 presents the interaction effect of dimension reduction and classification method sliced by classification method. Only adaBoost has p-value $< \alpha$. Therefore, means of the dimension reduction methods significantly differ in the case of adaBoost and there is some difference for NNET also). They do not differ significantly for all other methods.
- (5) Table 4.19 shows the mean for classification method in combination with the dimension reduction techniques used. Figure 4.5 provides the same information graphically. AdaBoost and single classification tree (both *rpart* and *tree*) give minimum misclassification rates with PCA; NNET and SVM work better with PLS; C4.5 works really well with both PCA and PLS.
- (6) Further research is needed to make conclusions about the significance of a likely three-way interaction - the interaction of dataset, dimension reduction technique, and classification method.

5. SUMMARY, CONCLUSIONS AND FUTURE WORK

The literature on microarray analysis contains attempts to improve or to find better performing traditional clustering or classification methods by employing them in combination with some dimension reduction techniques on a *single* microarray dataset (Tibshirani et al., 1999; O'Neill and Song, 2003). Other studies have evaluated the performance of a *particular* classification method in combination with different dimension reduction techniques on one or more microarray datasets (Dai et al, 2006; Nguyen and Rocke, 2002) or have compared a few classification methods in combination with a *selected* dimension reduction method (O'Neill and Song, 2003; Osareh and Shadgar, 2009). However, to date no research has examined *several* clustering and classification techniques combined with a *few* dimension reduction methods on microarray datasets having very different characteristics. The research in this thesis provides a more comprehensive way of examining the effectiveness of these methodologies. Our research extends the work of Dai et al. (2006) and Tan et al. (2008) as well as that of Osareh and Shadgar (2009) and provides an examination of the performance of several clustering and classification algorithms, together with two common dimension reduction methods (partial least squares PLS and principal components PCA) when applied to two very different microarray datasets (yeastCC and Breast Cancer).

Several clustering techniques - k-means, k-medoids, SOM, and agglomerative hierarchical clustering with mean linkage - were applied to samples of the original data, principal components and independent components of the datasets. Plaid models designed specifically for microarray data were also used. Classification techniques adaBoost, C4.5, single classification trees (both *rpart* and *tree*), neural network, support vector machines, and Random Forest were applied after the application

of dimension reduction via PCA and PLS. To see if there was any combination of methodologies that would perform consistently well, we examined the results for two microarray datasets that differ in many statistical characteristics.

5.1 Summary and Conclusions

5.1.1 Clustering

K-means clustering was done on the six different data sets (see Chapter 3.2) in the case of yeastCC and Breast Cancer.

The optimal number of clusters was obtained using the number of clusters corresponding to the kink in the sum of squares plot **and** the local minimum in the Davies-Bouldin Index plot for all of the above mentioned data sets for yeastCC.

Optimal number of clusters was obtained using **only** the global minimum of the DBI plot when clustering was done using principal components based on both the correlation and covariance matrices of the Breast Cancer dataset though the results were more accurate in the case of PCAs obtained using the correlation matrix of the dataset. The same procedures that were followed in the case of yeastCC were used to obtain optimal number of clusters for all the other four datasets obtained from the Breast Cancer dataset.

DimensionReduction	YeastCC	BreastCancer
Ori-dat	SS, DBI plots(4*)	SS, DBI plots(3)
PCA-Cor	SS, DBI plots(4)	DBI plot(3)
PCA-Cov	SS, DBI plots(4)	DBI plot(4)
ICA-Ori	SS, DBI plots(4)	SS, DBI plots(3)
ICA-Pcor	SS, DBI plots(3)	SS, DBI plots(3)
ICA-Pcov	SS, DBI plots(4)	SS, DBI plots(3)

Table 5.1: Summary of plots needed for six different data sets

* - Number of clusters given in parenthesis

The number of clusters for yeastCC is five (5) and for Breast Cancer is three

(3) (Chapter 1.2). The results were more accurate for Breast Cancer data (Table 5.1) than for yeastCC.

The optimal number of clusters turned out to be two (2) for all six (6) different data sets for both yeastCC and Breast Cancer, when clustering was done using k-medoids using both Euclidean and Manhattan dissimilarity. It appears that the k-medoids method using these dissimilarity measures is not sensitive enough for finding all the clusters in this kind of data.

When SOM clustering was done using principal components obtained from the correlation matrix of yeastCC resulted in four (4) clusters; and using the covariance matrix in five (5) clusters were found. Five (5) clusters were also obtained when the three different independent components (Chapter 3.2.1.3) of the data were used (Table 3.2). Hence for yeastCC, the expected number of clusters (5) was obtained for all situations except for principal components based on correlation matrix.

For Breast Cancer with SOM clustering, principal components obtained using the covariance matrix of the data gave one (1) cluster; based on the correlation matrix of the data, it gave two (2) clusters. Using the SOM on Breast Cancer, unlike other researchers (Spellman et al., 1998, R-Neve2006), we never obtained three (3) clusters.

In the case of hierarchical clustering, five (5) clusters were obtained when the cut was made in the range of Euclidean distance of 7.75 and 9 in the dendrogram obtained for yeastCC; for Breast Cancer, the number of clusters was three (3) when the cut was in the range of Euclidean distance of 18 and 20 (Figure 3.5). These results correspond to the findings of other researchers (Spellman et al., 1998 and R-Neve2006).

Using Plaid Models, the top 25 genes and 5 samples that belong to the first 5 layers/clusters were obtained for both of our data sets. Sample effects (Table 3.7) for each layer group the samples according to their classes for two out of three classes (i.e layer 3 contained all and only samples from class Lu, and layer 4 all and only

samples from class BaB) for Breast Cancer, but this is not the case for yeastCC. This may be due to the characteristics of the dataset - yeastCC had many missing values and highly imbalanced classes.

Results for clustering methods showed that using only the Davies-Bouldin plot was adequate for obtaining the optimal number of clusters when k-means clustering was done on the samples of principal components obtained using the correlation and covariance matrices of the Breast Cancer data but the sum of squares plot is also needed for all the other data sets. K-medoids did not provide satisfactory results for either of the two datasets. SOM gave better results for yeastCC than for Breast Cancer. On the other hand, Plaid Models gave better results in the case of Breast Cancer.

5.1.2 Classification

Classification was done using adaBoost (Freund, 1996; Freund and Schapire, 1996), C4.5 (Quinlan, 1992), single classification tree (*rpart* and *tree*) (Breiman et al., 1984), SVM, NNET, and Random Forest (Breiman, 2001) after doing dimension reduction using PCA and PLS, We also foundd the best k for k-nearest neighbours

Table 4.19 shows that single classification tree (using both *rpart* and *tree* library from R) and adaBoost, for which the base classifier is single classification tree, provide lower misclassification rates (on average) when used with PCA dimension reduction. For NNET and SVM, misclassification rates improved when used with PLS dimension reduction. C4.5 works well with both PLS and PCA. Since yeastCC has classes with an extremely small number of cases (i.e. 2), Random Forest failed to work on yeastCC. As a result, the results of Random Forest were not included in the final analysis but it is important to note the deficiency of this method for this situation.

Even though dataset (the two data sets used in this study differ very widely in their characteristics) and classification method effects are significant (Table 4.15),

interaction between dataset and either classification method or dimension reduction technique turned out to be statistically insignificant. However, our analysis showed that the interaction between classification method and dimension reduction technique was significant and that lower misclassification rates can be obtained by choosing an appropriate combination of dimension reduction technique and classification method (i.e AdaBoost and single classification tree (both *rpart* and *tree*) give minimum misclassification rates with PCA; NNET and SVM work better with PLS; C4.5 works really well with both PCA and PLS). Results using, adaBoost differed significantly and those of neural networks to a lesser extent depending on dimension reduction method used. Under PLS, the results for classification methods differed but under PCA the differences were less. Further research is needed to decide on the significance of the three-way interaction (dataset, dimension reduction technique and classification method).

5.2 Future Work

Two data sets which differ in many statistical characteristics were used in this study. Breast Cancer has no missing values, but yeastCC has at least one value missing in 5669 genes out of 6178. Missing values were imputed by sample mean but there is no evidence that this is the best choice. Further research is needed to determine the best possible way to impute the missing values so that the discrepancies created by imputation can be overcome.

Furthermore, we used the sample size of 1000 for clustering. This sample size was considered adequate, but further confirmation of this would add confidence to the findings reported herein. Further studies assessing the effect of sample size on the results would be useful in this regard.

In the case of classification, we applied PCA to the entire data set in order to apply dimension reduction, and then created 100 pairs of test and training sets for our

classification. For the case of PLS dimension reduction, we first created 100 pairs of test and training sets, and then the dimension reduction was done using PLS. Further study is needed to see how our results would change if PLS dimension reduction were applied before creating 100 pairs of data sets. The practical problem for this kind of analysis is the need for computing power and time. Using some of the most powerful computers available to us, some of our computations still required two or more days to complete. With computing power constantly increasing one could expect this to be substantially reduced in time.

BIBLIOGRAPHY

- [1] **Atkinson,E.J. and Therneau,T.M.**, An Introduction to Recursive Partitioning Using the RPART Routines, **Technical Report 61**, Mayo Clinic, **Section of Statistics, 1997.**
- [2] **Boulesteix.A.-L., Strobl.C, Augustin.T., Daumer.M.** (2008), Evaluating Microarray-based Classifiers: An Overview, **Cancer Informatics 2008:6 77-97**
- [3] **Breiman,L.,and Cutler,A.**, Random Forests, www.stat.berkeley.edu/~breiman/RandomForest/
- [4] **Breiman,L., Friedman,J.H., Olshen,R.A., and Stone,C.J.** (1984), Classification and Regression Trees, **Wadsworth**
- [5] **Brown.M.P.S., et al**(2000), Knowledge-based analysis of microarray gene expression data by using support vector machines, **PNAS, Vol.97, No.1,262-267**
- [6] **Buttrey, S.E. and C.Karo** (2002), Using k-nearest neighbor classification in the leaves of a tree, **Technical Report**, Nval Postgraduate School, Monterey, CA
- [7] **Chu.F., and Wang.L.**, (2005), Applications of Support Vector Machines to cancer classification with microarray data, **International Journal of Neutral Systems, Vol.15,No.6, 475-484**
- [8] **Dettling,M., and Bulmann.P.**,(2002), How to use Boosting with Tumor Classification with Gene Expression Data, **Res.Rep. 103,Sem.f.Statistik,ETHZurich**

- [9] Dai, J.J., Lieu, L., and Rocke, D. (2006), Dimension Reduction for Classification with Gene Expression Microarray Data, **Statistical Applications in Genetics and Molecular Biology**, Vol.5, Issue 1, Article 6
- [10] Davies, D.L and Bouldin, D.W (1979), A cluster separation measure, **IEEE Trans. Pattern Anal. Machine Intell.** 1(4). 224-227
- [11] Díaz-Uriarte.R., and Alvarez de Andrés.S. (2006), Gene selection and classification of microarray data using random forest, **Bioinformatics** 2006, 7:3
- [12] de Jong, S., (1993), SIMPLS: an alternative approach to partial least squares regression, **Chemometrics and Intelligent Laboratory Systems** Vol:18(3), 251-263
- [13] Dudoit,S., J.P.Shaffer, and J.C.Boldrick (2002a); Multiple hypothesis testing in microarray experiments, **Technical Report 110, Division of Biostatistics, University of California at Berkeley**
- [14] Dudoit et al., (2002b), Statistical Methods for identifying differentially expressed genes in replicated cDNA microarray experiments, **Statistica Sinica**, 12(1):111-139, 2002b
- [15] Eisen, M., Spellman,P., Brown, P.and Botstein, D.(1998), Cluter analysis and display of genome wide expression patterns, **Proc. Nat. Acad. Sci** 95, 14863-14868
- [16] Friedman, J.H, (1994), Flexible metric nearest neighbor classifcaion, **Dept of Statistics, Technical Report, Stanford University, Stanford, CA**

- [17] **Fix, E. and J.Hodges (1951)**, Discriminatory analysis, nonparametric discrimination: consistency properties, **Technical Report, Randolph Field, Texas: U.S Air Force School of Aviation Medicine**
- [18] **Forgy, E.W.(1965)**, Cluster analysis of multivariate data: Efficiency vs interpretability of classifications, **Biometrics**, **21**, 768-769
- [19] **FREUND.Y., AND SCHAPIRE.R.E., (1996)**, Experiments with New Boosting Algorithm, **Machine Learning (1996)**
- [20] **Freund.Y., and Schapire.R.E., (1996a)**, A decision-theoretic generalization of on-line learning and an application to boosting, www.research.att.com/orgs/ssr/people/{yoav,schapire}
- [21] **Freund.Y., and Schapire.R.E., (1997)**, A decision-theoretic generalization of on-line learning and an application to boosting, **Journal of Computer and Systems Sciences** **55**, 119-139 (1997)
- [22] **Furey.T.S., et al (2000)**, Support vector machine classification and validation of cancer tissue samples using microarray expression data, **Bioinformatics**, **Vol.16 no.10**, 906-914
- [23] **Geladi, P., Kowalski, B. (1986)**, Partial least-squares regression: a tutorial, **Analytical Chimica Acta** **185**, 1-17
- [24] **Golub, T.R. et al (1999)**, Molecular classification of cancer: class discovery and prediction by gene expression monitoring, **Science**, **286**:531-537
- [25] **Goldstein.D.R., Ghosh.D., and Conlon.E.M. (2002)**, Statistical issues in clustering of gene expression data, **Statistica Sinica** **12(2002)**, 219-240
- [26] **Hartigan, J.A and Wong, M.A (1979)**, A K-means clustering algorithm, **Applied Statistics**, **28**, 100-108

- [27] Hastie, T. and R.Tibshirani (1996b), Classification by pairwise coupling, Technical Report, Dept of Statistics, Stanford University
- [28] Hastie,T., R.Tibshirani, and J.H.Friedman (2001), The Elements of Statitical Learning: Data Mining, Inference, and Prediction, New York: Springer-Verlag
- [29] Höskuldsson, A. (1988), PLS regression methods, Journal of Chemometrics 2(3), 211-228
- [30] Hunt, E.B., Marin, J., and Stone, P.J.(1966) Experiments in Induction, New York: Academic Press
- [31] Johnson.R.A., and Wichern.D.W (2002) Applied Multivariate Statistical Analysis, Prentice Hall, New Jersey
- [32] Kaufman and Rousseeuw (1990), Findings Groups in Data: An Introduction to Cluster Analysis, Wiley, New York
- [33] Kohonen, T.(1989), Self-Organization and Associative Memory (3rd edition), Springer-Verlag, Berlin.
- [34] Lazzeronni,L., and Owen,A. (2002), Plaid Models for Gene Expression Data, Statistica Sinica, 12(2002), 61-86
- [35] Lee.J.W., Lee.J.B., Park.M., and Song.S.H. (2005), An Extensive comparison of recent classification tools applied to microarray data, Computational Statistics and Data Analysis 48 (2005) 869-885
- [36] Lloyd, S.P. (1957,1982), Least squares quantization in PCM, Technical Note, Bell Laboratories, Published in 1982 in IEEE Transactions on Information Theory 28, 128-137

- [37] **MacQueen, J. (1967)**, Some methods for classification and analysis of multivariate observations, **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability**, eds L.M. LeCam and J.Neyman, 1, pp.281-297, Berleley, CA: University of California Press.
- [38] **Mills,S.E., and Normington** , Datamining lecture Notes,
- [39] **Mosteller,F., and Youtz,C.**, Tables of the Freeman-Tukey Transformations for the Binomial and Poisson Distributions, **Biometrika**, Vol.48, No.3/4(Dec 1961). pp.433-440
- [40] **Mukherjee et al (1998)**, Support vector machine classification of microarray data, A.I. Memo No.1677, C.B.C.L paper No.182
- [41] **O'Neill,C.M., and Song,L.**, (2001), Neutral network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect, **Bioinformatics**, 4:13
- [42] **Osareh, A., and Shadgar,B.**, (2009), Classification and Diagnostic Prediction of Cancers Using Gene Microarray Data Analysis, **Journal of Applied Sciences** 9(3):459-468
- [43] **Park.P.J., M.Pagano, and M.Bonetti**, (2001), A nonparametric scoring algorithm for identifying informativwe genes from microarray data, **Pacific. Symp. on Biocomputing**, 6:52-63, 2001
- [44] **Perou et al (1999)**, Distinctive Gene Expression Patterns in Human Mammary Epithelial Cells and Breat Cancers, **Proceedings of teh National Academy of Sciences of the USA**, Vol.96, No.16 pp 9212-9217

- [45] **Pomeroy et al (2002)**, Prediction of central nervous system embryonal tumour outcome based on gene expression, **Nature**, **415(24):436-442** (and supplementary information)
- [46] **Quinlan, J.R. (1992)**, C4.5: Programs for Machine Learning, **Morgan Kaufmann, San Mateo, CA, U.S.A**
- [47] **R-Neve2006**, R Command - summary(print(pData(neveRMAMatch)))
- [48] **Rousseeuw, P.J. (1987)**, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, **Journal of Computational and Applied Mathematics**. **20**, 53-65
- [49] **Sorlie,T., Perou,C.M., Tibshirani,R., Aas,T. et al (2001)**, Gene Expression Patterns of Breast Carcinomas distinguish tumor subclasses with clinical implications, **Proceedings of the National Academy of Sciences of the USA**, vol.98, No.19, pp 10869-10874
- [50] **Speed, T. (2003)**, Statistical Analysis of Gene Expression of Microarray Data, **Chapman and Hall/CRC**
- [51] **Spellman,P.T., Sherlock.G., Zhang.M.Q., Iyer.V.R., Anders.K., Eisen.M.B., Brown.P.O., Botstein.D., and Futcher.B. (1998)**, Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization, **Molecular Biology of the Cell**, vol. **9**, 3273-3297, December(1998)
- [52] **Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S., and Golub,T.R. (1998)**, Interpreting patterns of gene expression with self-organizing maps: methods and applicaiton to hematopoietic differentiation, **Proceedings of the National Academy of Science of USA**, Vol.96,pp2907-2912

- [53] **Tan,P.J., David L. Dowe and Trevor I. Dix**, Building classification models from microarray data with tree-based classification algorithms, **Proc. 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Queenslan, December 2007**
- [54] **Tibshirani.R., Trevor Hastie, Mike Eisen, Doug Ross, David Botstein, and Pat Brown (1999)**, Clustering methods for the analysis fo DNA microarray data, **Tech report Oct 1999**
- [55] **Tibshirani.R, Walther.G, and Hastie.T (2000)**, Estimating the number of clusters in a data set via the gap statistic, **J.R.Statist.Soc. B(2001) 63, Part 2, pp.411-423**
- [56] **Tibshirani.R, Walther.G, and Hastie. T (June2001)**, Estimating the nubmer of clusters in a data set via the gap statistic, **Journal of the Royal Statistical Society: Series B (Statistical Methodology): vol63, Issue2,P411**
- [57] **Vapnik.V., (1979)**, Estimation of dependences based on empirical data (in russian), **Nauka, Moskow**
- [58] **Vapnik.V., (1998)**, Statistical Learning Theory, First ed., **New York: John Wiley and Sons**
- [59] **Zhu,E. and Ramon M.Barnes (1995)**, A simple Iteration Algorithm for PLS Regression, **Journal of Chemometrics, Vol.9, 363-372**

A. R & SAS CODE

Available at

<http://www.math.carleton.ca/~smills/ahalya/thesis/Appendix-A.pdf>

B. GENES BELONGING TO EACH CLUSTER

Available at

<http://www.math.carleton.ca/~smills/ahalya/thesis/Appendix-B.pdf>

C. DETAILED RESULTS FOR PLAID MODELS

Available at

<http://www.math.carleton.ca/~smills/ahalya/thesis/Appendix-C.pdf>