Sequential Dimension Reduction and Prediction Methods with High-dimensional Microarray Data

Dissertation
zur Erlangung des Grades Doktor der Naturwissenschaften (Dr. rer. nat)
an der Fakultät für Mathematik, Informatik und Statistik,
der Ludwig-Maximilians-Universität München.

vorgelegt von

Waheed Babatunde Yahya

B.Sc., M.Sc., PGDFM, MBA

April, 2009

Sequential Dimension Reduction and Prediction Methods with High-dimensional Microarray Data

Dissertation
zur Erlangung des Grades Doktor der Naturwissenschaften (Dr. rer. nat)
an der Fakultät für Mathematik, Informatik und Statistik,
der Ludwig-Maximilians-Universität München.

vorgelegt von

Waheed Babatunde Yahya

B.Sc., M.Sc., PGDFM, MBA

2009

Gutachter:	Prof. Dr. Kurt Ulm
Gutachter:	Prof. Dr. Ludwig Fahrmeir
Promotionsabschlußberatung:	24. Juni 2009

Dedication

To God, Almighty…

…and to my beloved wife, Bilikis Ajoke
…and my darling children, Sekinat, Shakirat, Riliwanulahi & Abdul-Rasheed

Acknowledgement

Firstly, I give glory to God, the Almighty, for His mercies, kindness and love on me and for making the little efforts I have put into this work a huge success.

I shall ever remain grateful to Prof. Dr. Kurt Ulm, who has been very difficult for me to be seen as my supervisor, but rather as my father for the inestimable roles he has played on my academic career. The success of this work stemmed out of his excellent sense of direction, love, and concern on my numerous academic and personal requests. I equally appreciate several contributions of my co-supervisor, Prof. Dr. Ludwig Fahrmeir, Ludwig-Maximilians University, Munich, Germany, at various stages of this research work all of which have assisted to take this thesis to its present height.

Many positive influences brought into my research works by my co-members of Prof. Dr. Ulm's research group are highly acknowledged. The concern and supports of my friend, Alexander and others, Tibor, Monika, Pia, and Bernhard are all appreciated. I also wish to express my appreciation to the director, Prof. Dr. KA Kuhn as well as other members of staff of IMSE, Technical University of Munich, Germany for their unalloyed co-operation which has made my stay in the department fruitful and rewarding.

Back in Nigeria are my fathers, Professors ET Jolayemi and BA Oyejola of the Department of Statistics, University of Ilorin, Nigeria, under the tutelage of whom I started my academic career. Their supports, prayers, encouragements and advice all the time have contributed to the success of this work.

I am equally indebted to my uncle, Prof. BL Adeleke for his prayers, guidance and encouragements which firmly prepared me for various challenges posed by this research work. I also thank Prof. IO Oshungade, the current Head of Department of Statistics, University of Ilorin, Nigeria, for his concern and prayers all the time. Also to Professors OS Adegboye, PA Osanaiye, and RI Ipinyomi, I say thank you Sirs for your prayers. To other colleagues and friends in the Department, I thank you all for your supports.

I want to appreciate the immeasurable supports and prayers of Dr. SB Adebayo and his family. Almighty God shall continue His mercies on you all.

Lest I forget, my esteemed parents and siblings, especially Abdul-Wasiu Olasunkanmi Bello, for their fervent prayers and supports over several decades all of which have assisted me through the thin and thick of my academic endeavour.

I want to specifically thank immediate members of my family all of whom my Ph.D. studies in Europe have unavoidably denied my physical presence for some years. The steadfastness, patience, prayers and supports of my beloved wife, *Hajia Bilikis Ajoke* have served as great impetus for the successful completion of this research work. And to my darling children, *Sekinat, Shakirat, Riliwanulahi and Abdul-Rasheed*, I say thank you all for your patience and prayers.

I appreciate the scholastic contributions of the authors of several literatures I consulted in the course of preparing this thesis.

Part of this research study was financed by *Deutscher Akademischer Austausch Dienst* (DAAD). I immensely thank this body for the opportunity given me to be able to carry out this research work in Germany. I particularly appreciate Mr. Michael Baumgardt, Referat 413 – Afrika – Subsahara, for his concern and prayers as well as his prompt responses to my requests always.

To all my good friends and well wishers, Drs. IO Abiala, K Rauf, AO Adejumo, GM Oyeyemi, SO Ige, Messrs. IA Akanji, MB Ganiyu, MO Adeleke, R Olopade, as well as Mr. & Dr. (Mrs.) Aderinto, Mr. & Mrs. Fatai Wilson Adeoye (Germany), and my bosom partner with whom boundary faded, Lawyer Jacob Niteleka Jaconiah (Tanzanian), I say thank you so much for your supports and prayers.

München, April 2009 *Waheed Babatunde Yahya*

Abstract

In this thesis, a novel sequential genes selection and classification (k-SS) method is proposed. The method is analogous to the classical non-linear stepwise variable selection (SVS) methods but unlike any of the SVS methods, this new method uses the *misclassification error rates* (MERs) as its search criteria for informative marker genes in any given microarray data. Here, the importance of any selected gene is determined based on its marginal contribution at improving the prediction accuracy of the classification rule. This method ensures continuous selection of more genes in as much as the improvements brought into the decision models by the selected genes are considered to be significant enough by some established test criteria. However, further gene selection terminates when none of the remaining genes is capable at improving the prediction accuracy (lowering the MER) of the current model. Therefore, our approach only seeks to select the best combination of k marker genes that are most predictive of the biological samples in any given microarray data sets.

An important feature of our new k-SS method is that the size α used by its test is not arbitrarily fixed by the user as common to some of the classical SVS methods. Rather, the value of α at which the best prediction accuracy is achieved (or the best combination of genes is selected) is determined by cross-validation.

The new k-SS classifier competes favourably with selected eight existing classification methods using eleven published microarray data sets. The k-SS classifier is very simple to apply and does not require any rigid assumption for its implementation. Another merit of this method lies in its ability to select only those genes that are of biological relevance to the existing cancer sub-groups in microarray data sets.

Lastly, we proposed a new preliminary feature selection procedure that employs the cross-validated area under the ROC curve (CVAUC) for gene selection. This method is capable at removing all the irrelevant genes at the preliminary selection stage before any standard classifier like the k-SS method is employed on the remaining data set for final optimum gene selection and classification of mRNA samples. Unlike some other data pruning methods, the new method employs the sub-sampling technique of the v-fold cross-validation to ensure consistency and efficiency of selections made at the preliminary selection stage.

Zusammenfassung

In dieser Arbeit wird eine neuartige sequentielle Geneselection und klassifikation (*k-SS*) vorgeschlagen. Die Methodik verhält sich analog zu nichtlinearen schrittweisen Variablenselektionmethoden (SVS). Im Gegensatz zu diesen benützt die neue Methode die Fehlklassifikationsrate (MER) als Suchkriterium für informative Marker-Gene in beliebigen microarray Datensätzen. Hierbei wird die Wichtigkeit eines Genes durch seinen marginalen Beitrag zur Verbesserung der Vorhersagegüte einer Klassifikationsregel bestimmt. Die Methode gewährleistet eine fortwährende Selektion weiterer Gene solange die Verbesserungen der Entscheidungsmodelle durch die ausgewählten Gene durch ein ebenfalls eingeführtes Testkriterium als signifikant genug erachtet werden. Indes endet die weitere Geneselektion sobald keines der verbleibenden Gene geeignet ist die Vorhersagegüte im aktuellen Modell zu verbessern bzw. die MER zu vermindern. Deshalb ist die Bestrebung unseres Ansatzes die beste Kombination aus k Marker-Genen, die am prädiktivsten für biologische Proben in beliebigen microarray Datensätzen sind zu selektieren.

Eine wichtige Eigenschaft unserer neuartigen k-SS Methode ist dass das Maß α, dass in ihrem Test benützt wird nicht eigenmächtig durch den Anwender bestimmt wird wie allgemein in klassischen SVS Methoden. Vielmehr wird der Wert von α, bei dem die beste Vorhersagegüte erlangt wird (oder die beste Kombination von Genen selektiert wird) durch Kreuzvalidierung bestimmt.

Der neue k-SS Klassifizierer konkurriert erfolgreich mit acht ausgewählten Klassifizierungsmethoden unter Verwendung von elf publizierten microarray Datensätzen. Der k-SS Klassifizierer ist sehr einfach anzuwenden und benötigt keine rigiden Annahmen für seine Durchführung. Ein weiterer Vorzug dieser Methode liegt in seiner Fähigkeit nur solche Gene zu selektieren, die von biologischer Relevanz bezüglich existierender Tumoruntergruppen in microarray Datensätzen sind.

Letztlich schlagen wir eine neue vorausgehende Variablenselektionsprozedur vor, die die kreuzvalidierte Fläche unter der ROC-Kurve (CVAUC) für die Genselektion benützt. Diese Methode ist fähig alle irrelevanten Gene in einem vorausgehenden Selektionsschritt zu entfernen, bevor klassische Klassifizierer wie die k-SS Methode auf dem verbleibenden Datensatz zur abschließenden, optimalen Genselektion und Klassifikation von mRNA-Proben angewendet werden. Ungleich einigen anderen pruning Methoden verwendet die neue Methode die v-fache Kreuzvalidierung als Methode zur wiederholten Stichprobenteilung um Konsistenz und Effizienz der Selektion zu einem vorausgehenden Selektionspunkt zu gewährleisten.

Preamble

A common problem in most of the microarray (cancer) studies is how to identify and select, among several available thousands, the most informative marker genes whose expression levels are predictive of clinical or other outcomes of interest. A major constraint however, is that the expression levels of all these genes are often collected on relatively few samples which makes the use of classical regression methods inappropriate for genes selection and prediction of biological samples. Several methods have been proposed in the literature to handle this task, but unfortunately, apart from procedural complexities, some of these methods like *Partial least squares, Principal component analysis* and the like only provide accurate classifiers that are often difficult to interpret. In this thesis therefore, we provide a novel but simple sequential selection procedure (*k-Sequential Selection* (*k*-SS) *method*) that efficiently selects from several thousand transcripts, the most informative *k* genes that are suitable for the prediction of biological samples. The *k*-SS procedure adopts the performance index of the average misclassification error rates (MERs) as its gene selection criteria.

The performance of the new method was evaluated and compared with eight existing standard classification methods (*Support vector machines, k-nearest neighbours, Partial least squares, Prediction analysis for microarray, Decision trees, Naïve bayes, Top scoring pair, k-Top scoring pair*) using eleven different microarray cancer data sets ten of which are publicly available. The eleventh data set is based on microarray cancer study of 43 patients with *locally advanced rectal carcinomas* (LARC) from whom 24,026 human genome U133 plus 2.0 gene-chip arrays were generated. The clinical study was carried out in the Department of Surgery, Klinikum rechts der Isar, Technical University of Munich, Munich, Germany.

Several results from this work showed that the new *k*-SS method performs efficiently well like any of the existing methods considered. In addition to this, this new approach provides stable and easily interpretable classifiers (genes) that seems to be of biological relevance to the sub-classes of tumour that are present in any given microarray data set. This obviously meets the expectations of the biologists and physicians who are not only interested in the classification of the mRNA samples into their various tumour types but also want to know the relevant informative genes that induced such classification. In addition, the k-SS method is generally simple and requires no stringent conditions for its implementation as common to some of the existing methods.

Since a typical microarray data set usually contains expression measures of both relevant and irrelevant transcripts, it has therefore become a usual practice in many microarray studies to primarily reduce the whole gene data to a manageable size of all the potentially relevant genes. This is usually done to save computation time and efforts. To this end, we proposed another new preliminary feature selection procedure that employs the cross-validated estimates of the area under the ROC curve of each observed gene for selection. This method, as a classifier-like method, improves on some of the existing methods like the *t*-statistic procedure for being capable of removing from microarray data set, only those genes that are absolutely non-predictive of the biological sub-groups of the mRNA samples. This method eliminates the risk of possible exclusion of some of the important genes at the preliminary selection stage before any standard gene selection and prediction method, like *k*-SS, could be employed on the preliminarily selected genes for further analysis. The application of the new preliminary feature selection procedure was also demonstrated using some of the microarray data sets considered in this work.

Contents

List of Figures

List of Tables

1 Background into Microarray studies

1.1 Introduction

A *gene* is a unit of *deoxyribonucleic acid* (DNA) that occupies a spot on a chromosome and helps to determine a trait in an organism. Genes are passed on from parents to child and constitute important part of what determines physical appearance and behaviour of an individual. The total amount of genes carried by individual living organism is called *genome* which in turn defines the genetic construction of the organism called *genotype*.

The existence of genes was first discovered by Gregor Mendel (1822-1884), who, in the 1860s, studied inheritance in pea plants and discovered a factor that conveys traits from parent to offspring. His various works were reported by Olby (1979).

Following Mendel's line of argument is Herman J. Muller (1951) who claimed that genes are fundamentally endowed with two basic properties: *autocatalysis* that allowed the genes to reproduce as units of transmission that connected the genotype of one generation to that of the next and *heterocatalysis* which connected the genes to the phenotype, as units involved in the expression of a particular character.

Several studies have however shown that thousands of these genes and their products (ribonucleic acid, proteins, etc.) are functioning in a complicated and orchestrated way in any living organisms which at times creates some mystery of life. The earlier traditional approach of studying one gene per experiment using radioactive detection reagents had made it difficult to understand the whole functioning processes of several thousands of genes most of which are interconnected.

Over the past few years, a new technology called DNA microarray or simply, *microarray technology* (MT) as it is often referred, Burnside *et al* (2008), was developed. This has made it possible to monitor and measure the expression levels of several thousands of genes simultaneously. By this, better understanding of the inherent relationships among various genes is accomplished.

The gene expression is the process by which *messenger ribonucleic acid* (mRNA) and protein are synthesised from the DNA template of each gene. The DNA is a nucleic acid that contains the genetic instructions used in the development and functioning of living organisms. *Ribonucleic acid* (RNA) on the other hand is a nucleic acid made from a long chain of nucleotide and structurally differs from DNA. While DNA contains deoxyribose and is double stranded, RNA contains ribose sugar and is single stranded. Messenger RNA (mRNA) is the RNA that carries information from DNA to the ribosomes which again translate the information they carry into proteins. Further details about the structural form of these two molecules can be found in Salazar *et al* (1993), Mikkola *et al* (1999), Hermann & Patel (2000), Cooper & Hausman (2004) and in many other related works.

The advent of modern methods into microarray profiling and sequencing has made it easy to generate several volumes of *complimentary* DNA (cDNA) through reverse transcription of mRNAs. It is then easy to measure the activity of thousands of genes at once and creating a global picture of cellular function. MT method, like serial analysis of gene expression (SAGE or

SuperSAGE) is commonly adopted for gene expression profiling which has made it possible to identify the cells that are actively dividing based on their mRNA functions.

Another important task after the generation of microarray data sets is to identify the genes that are differentially expressed (DE) within the mRNA samples. The DE genes are the group of genes that belong to the same functional class whose expression patterns are strong enough to classify any future mRNA samples with similar molecular features. Many statistical techniques have been proposed in many studies for proper classification of mRNA samples into their various biological sub-groups. A more flexible dimension reduction and response class prediction method is equally provided in this thesis.

However, to analyse any experimental data correctly, it is fundamental to understand the experiment that generated such data set. Therefore, in what follows, we provide some insights into the basic platforms upon which microarray data sets are usually developed.

1.2 The cDNA and Affymetrix microarrays

Microarray technology has provided us with a compelling approach that allows for simultaneous evaluation of all cellular processes at once. This has greatly assisted the process of identification of new molecular markers that could be useful in the diagnosis, prognosis, and prediction of different categories of cancers. However, there are several microarray technological platforms on which mRNA samples are processed. In all the platforms, oligonucleotide or cDNA probe sets are used for fabrication.

The common procedure especially in spotted microarray experiments is that, the DNA or oligonucleotide probes are synthesized prior to deposition on the array surface and are then robotically spotted onto glass. Thereafter, purified RNA samples are fluorescently or radioactively labelled and hybridized to the slide or membrane. In some cases, hybridization is done simultaneously with reference RNA to facilitate comparison of data across multiple experiments. After thorough washing, the raw data is obtained by laser scanning or autoradiographic imaging. At this point, the data are entered into a database and analyzed by a number of statistical methods.

Oligonucleotide is a small chain of nucleic acid residues which are used to detect the presence of larger mRNA molecules. Oligonucleotide microarray is a type of microarray technology developed at Affymetrix, Inc., California, (Affymetrix, Inc; 2001a,b). Here, short oligonucleotide sequences (20~80-mers oligos) or *peptide nucleic acid* (PNA) probes are synthesized either *in-situ* (on-chip) or by conventional synthesis onto the array surface followed by on-chip immobilization.

A particular technique due to Pease *et al* (1994) is sometimes used to produce oligonucleotide arrays. In this method, photolithographic synthesis (Agilent and Affymetrix) is performed on a silica substrate where light and light-sensitive masking agents are used to build a sequence one nucleotide at a time across the entire array.

In spotted *complementary* DNA (cDNA), Two-colour or Two-channel microarrays are typically hybridized with cDNA prepared from two samples to be compared (e.g. diseased tissue & healthy tissue) and they are labelled with two different fluorophores, Shalon *et al* (1996). Fluorophores are molecules that have fluorescent properties. The fluorescent dyes commonly used for labelling include Cy3, which has a fluorescence emission wavelength of 570 nm (corresponding to the green part of the light spectrum), and Cy5 with a fluorescence emission wavelength of 670 nm (corresponding to the red part of the light spectrum). The Cy3 and Cy5 firstly proposed by Ernst *et al* (1989) are reactive water-soluble fluorescent dyes of the *cyanine* dye family. Example of the two fluorescent colours is provided by the hit-map in *Fig 1.1* for selected transcripts from 24,026 genes measured on 43 locally advanced rectal cancer patients. The two labelled cDNA samples are then mixed and hybridized into a single microarray. This is then scanned in a microarray scanner to visualize fluorescence of the two fluorophores after excitation with a laser beam of a defined wavelength. Relative intensities of each fluorophore may then be used in ratio-based analysis to identify up-regulated and down-regulated genes, Tang *et al* (2007). *Fig 1.2* shows the schematic form of steps (not exhaustive) involved in a typical two-channel cDNA microarray experiment.

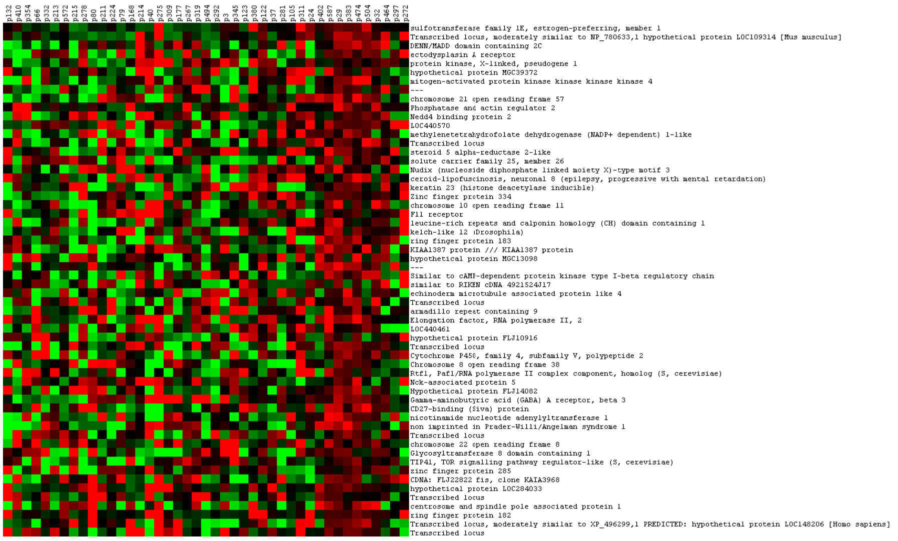


Fig 1.1: The hit-map showing the two fluorescent dyes, Cy3 (green) and Cy5 (red) indicating low and high expressions respectively of the selected transcripts among 24,026 genes observed on 43 locally advance rectal cancer patients.

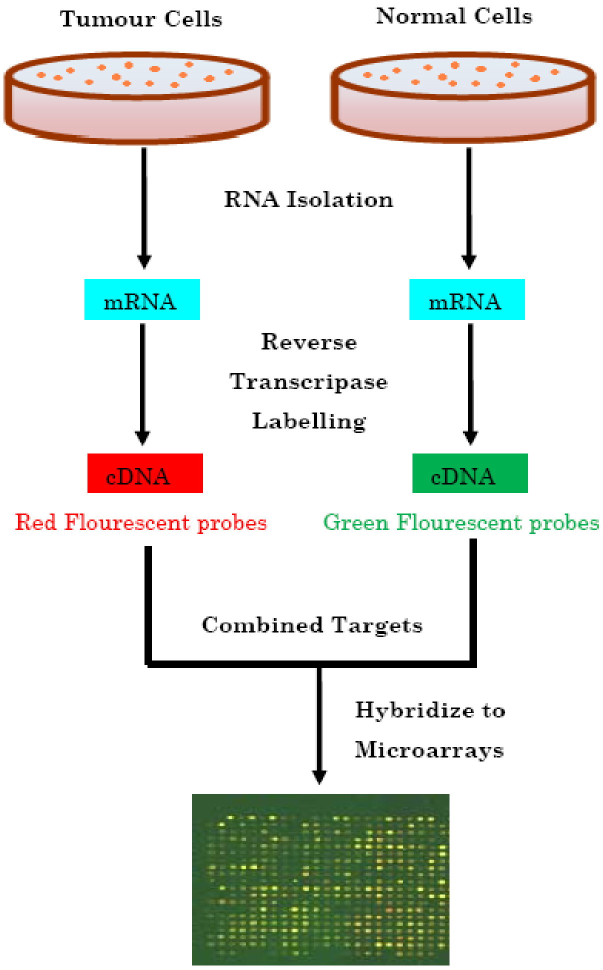


Fig 1.2: A typical two-colour spotted cDNA microarray experiment

1.3 DNA microarrays in cancer research

Cancer, sometimes called malignant neoplasm, is a complex disease in which a group of cells display certain traits of uncontrolled growth and invasion which may possibly spread (metastasize) to other parts of the body. Cancer can develop in any part of human body which eventually give rise to various kinds of cancer like *lung*, *prostate*, *breast*, *renal*, *brain*, *gastric*, *rectal*, *colon*, and *head & neck cancers* among others.

Over the past few decades, classification and diagnosis of cancer patients are based on the examination of the organs where the tumour is developed. This often resulted into the exhaustive physical and histopathological assessments of the organs that harbour the tumour. However, diagnoses are only achievable either through laboratory tests which might be too costly to bear or through surgical operations which might expose the patients to different kind of risks. In some instances, some of the test results, like autopsy can be available only after the passage of time, thus causing some delay before any diagnoses or cancer classification could be performed.

Fortunately, the advent of DNA microarray technology in the recent past has introduced dramatic changes into cancer research. With this new technology, it is possible to simultaneously analyse the expressions of several thousands of genes at once and relate their expression patterns to clinical phenotypes, Lonning *et al* (2005). By this, it is possible to identify molecular signatures whose expression patterns are capable of discriminating between infected (cancer) cells and uninfected (normal) cells. It is therefore easy to predict (diagnose) the prognostic stage (whether cancerous or normal) of all the cancer patients using the gene expression profiles without taken them through the rigour of expensive laboratory tests or surgery.

Due to high dimensional nature of microarray data typically with q genes and n biological samples, $n \ll q$, many supervised and unsupervised methods have been developed to handle dimension reduction, patterns recognition as well as prediction of biological samples using gene expression data.

The use of gene expression profiles for cancer diagnoses has been the major focus in many microarray studies. One of the most highly referred studies in this area is that of Golub *et al* (1999). In their study, the expression levels of 7129 Affymetrix gene chips generated on 72 human acute leukemia tumour subjects were used to classify the subjects into two sub-types of leukemia: *acute myeloid leukemia*

(AML) and *acute lymphoblastic leukemia* (ALL). An unsupervised class discovery method was used to identify these two classes of leukemia without *a priori* knowledge of the subjects' prognostic status. The use of gene expression data for class discovery and class prediction was firmly established in this work.

In a related study, Alizadeh *et al* (2000) used DNA microarrays to conduct a systematic characterization of gene expression in B-cell malignancies. The expression patterns of patients with *diffuse large B-cell lymphoma* (DLBCL) were studied. Hierarchical clustering with average linkage search was used on the gene expression patterns of 88 biological samples to identify two previously unidentified molecularly distinct forms of DLBCL (*germinal centre* B-like DLBCL and *in vitro* activated peripheral blood B-like DLBCL) which had gene expression patterns indicative of different stages of B-cell differentiation. They equally demonstrated that patients with the two sub-groups of tumour are susceptible to different clinical outcomes. Bhattacharjee *et al* (2001) also used hierarchical clustering method on expression patterns of *lung cancer* patients to identify patients with various kind of this cancer type that are characterized by different prognostic outcomes.

Also, Bittner *et al* (2000) used hierarchical clustering on gene expression profiles of 31 melanomas biological samples to discover identical cluster of 19 melanomas that had similar gene expression patterns. In another study, Pomeroy *et al* (2002) applied some supervised and unsupervised methods on Affymetrix oligonucleotide microarrays to distinguish between new and existing sub-classes of embryonic tumours of the central nervous system (CNS) using gene expression patterns.

In hereditary breast cancer studies, Hedenfalk *et al* (2001) used the gene expression profiles of breast cancer patients to identify 176 genes that are capable to discriminate patients with sub-types of breast cancer tumours: i.e. tumour with BRCA1 mutations and tumour with BRC2 mutations.

As application in survival studies, Nguyen & Rocke (2002c) used partial least square (PLS) components constructed from gene expression patterns of patients with locally advanced breast carcinomas as predictors in proportional hazard (PH) regression model to predict patients' survival outcomes.

A good number of classification methods have been proposed in the literature to properly classify biological samples into their respective tumour types using their gene expression profiles. The most commonly used ones include the *linear discriminant analysis* (Lee, 2004; Ye *et al*, 2004; Hastie *et al*, 2009), *classification and regression trees* (Zhang *et al*, 2001 & 2003), *logistic discriminant analysis* (Ding & Gentleman, 2004), *k-nearest neighbours* (Fix & Hodges, 1951; Cover & Hart, 1967; Giordano *et al*, 2001; Baoli *et al*, 2003), *support vector machines* (Vapnik, 1998; Christianini & Shawe-Taylor, 2000; Bennett & Campbell, 2000; Furey *et al* , 2000; Peng *et al*, 2003; Liu *et al*, 2005; Chu & Wang, 2005), *artificial neural networks* (Hertz *et al*, 1991; Ripley, 1996; Khan *et al*, 2001; Bicciato *et al*, 2003; Hastie *et al*, 2009), *boosting* (Dettling & Buhlmann, 2003) and *bagging* (Dudoit & Fridlyand, 2003) among others.

The various microarray studies highlighted above are just a few instances among several thousands of studies hitherto being undertaken by many scientists all over the world. While some of the methods adopted are relatively simple to apply, a good number of them are characterized by rigorous procedural complexities. Nonetheless, the ever-increasing challenges in microarrays technology have made it imperative on the scientists to continuously thinking and developing more concise techniques that are suitable to address fundamental questions which often accompany new discoveries in genes expression profiling on daily basis.

Most of the studies discussed so far focused on proper classification or prediction of biological samples into difference cancer sub-classes. Another important aspect of microarray studies is the selection of the marker genes that characterized different tumour classes and responsible for the identification, prediction or diagnosis of various sub-groups of cancers. Some of the classification methods combined feature selection with class prediction while some of them only perform classification of biological samples into their various tumour categories. However, the huge numbers of data sets generated by microarray experiments have raised a lot of methodological and computational challenges in the analysis of high-dimensional genomic data.

1.4 Prior to dimension reduction and class prediction

In analysing microarray data, a number of preliminary steps need to be taken before getting to the real dimension reduction and response class prediction. We discuss the major two of such steps which centres on data normalization and preliminary gene selection.

1.4.1 *Data normalization*

In microarray studies, *normalization* is the process of identifying and removing the effects of systematic variations other than the biological differences in the measured fluorescence intensities of genes across the hybridized mRNA samples. It refers to a set of data pre-processing steps often employed to eliminate the influence of non-biological variations that might unavoidably be present in microarray data sets, so that differential expressions in genes can be truly identified.

Within the purview of cDNA microarray experiment, the expression level of each gene is measured by the ratio of two fluorescent dyes, Cy3 and Cy5 over the mRNA samples. Variations in print-tip, labelling efficiencies, spatial and hybridization specific effects, and several other scanning properties of Cy3 and Cy5 may introduce a lot of systematic variations into the observed fluorescence intensities. As a result, the actual biological differences (differential expression) inherent in a set of genes might be clouded by the effects of all the extraneous variations which may eventually lead to wrong biological decisions. Hence, the need to free microarray data sets from all these noises.

In a loose term, the process of normalizing the $n \times q$ matrix of microarray data set with n arrays and vector $\boldsymbol{X} = (X_1, \dots, X_q)$ of q genes can be viewed as transforming all the expression patterns X_{ij} of j^{th} gene across the n mRNA samples by

$$Z_{ij} = h(X_{ij}) - \frac{1}{n^{*} \in n,q} \sum\nolimits_{l \in i,j} h(X_l) \qquad (1.4.1)$$

where $h(.)$ represents the monotonically increasing Box-Cox family of transformations of X_{ij} given by

$$h(X_{ij}) = \frac{{X_{ij}}^{m} - 1}{m} \qquad (1.4.2)$$

for some constant $m > 0$. Here, $h(X_{ij}) = X_{ij}$ if $m = 1$, indicating no transformation except for shift in location, and $h(X_{ij})$ becomes the square root transformation of X_{ij} if $m = 1/2$ while it tends to the logarithm transformation as $m \rightarrow 0$. However, the gene expression patterns X_{ij} reported in most microarray data sets are already the log of the fluorescent ratios which might not require further log transformation.

Literally by (1.4.1), gene normalization across the arrays is performed by subtracting the mean expression levels of each gene from its expression level for each array while normalization across the q genes is performed by subtracting the mean expression levels of all the genes for each array from their respective individual expression levels.

Apart from the general normalization form given by (1.4.1), several other forms of normalization have been proposed to further improve the quality of microarray data before analysis could begin. Three of these approaches are discussed below.

i) Intensity-dependent normalization

Yang *et al* (2002) suggested the use of *intensity-dependent normalization* which is based on the *locally weighted regression* (LOWESS)(Cleveland, 1979; 1981) smoothing of the MA-plot. Let X_{ij}^{G} and X_{ij}^{R} denote the green and red intensities of expressions of gene j on i mRNA samples, $i = 1, \dots, n$, $j = 1, \dots, q$, as observed from the fluorescent dyes, Cy3 and Cy5 respectively. What is often reported as the gene expressions are the ratios X_{ij}^{R}/X_{ij}^{G} or log-ratios $log(X_{ij}^{R}/X_{ij}^{G})$ of the fluorescent dyes. Conventionally, we denote the log intensity ratios by

$$M = log(X_{ij}^{R}/X_{ij}^{G}) = logX_{ij}^{R} - logX_{ij}^{G} \qquad (1.4.3)$$

and the average log intensity of the two colours by

$$\widetilde{M} = log\sqrt{X_{ij}^{R} \times X_{ij}^{G}} = \frac{1}{2}\left(logX_{ij}^{R} + logX_{ij}^{G}\right) \qquad (1.4.4)$$

The plot of M against $\widetilde{M}$ is call the MA-plot which gives a 45^0 rotation and rescaling of the plot of $logX_{ij}^{R}$ against $logX_{ij}^{G}$, Dudoit *et al* (2002), Huber *et al*(2005). A fit of LOWESS function $l(\widetilde{M})$ of the average intensity $\widetilde{M}$ is then obtained and this is used to normalize M by computing the difference $M - l(\widetilde{M})$. Thus, the general normalization form in (1.4.1) becomes $Z_{ij} = M - l(\widetilde{M})$. This normalization type is design to remove extraneous colour effects that may be induced by different pin tips. More details about this approach could be found in Lee (2004), Huber *et al*(2002) and many other related studies.

ii) Rank-Invariant genes normalization

The *rank-invariant* method as proposed by Tseng *et al* (2001) as a non-linear normalization method considers a microarray experiment in which two differentially expressed specimens are separately labelled with green (Cy3) and red (Cy5) flours and co-hybridized to the same slide. Unlike in the intensity-dependent normalization in which all the genes are used to determine normalization factor, here, a sub-set of genes that are biologically assumed not to be differentially expressed in the two specimens are selected for normalization. Thus, a particular gene X_j is used for normalization if the ranks of its green and red intensities are similar up to a threshold value d and the rank of its average intensities is not among the highest $q - l$ ranks or lowest l ranks for any choosing constant q and l. These statements are given by

$$\left|Rank\left(X_j^R\right) - Rank\left(X_j^G\right)\right| < d \qquad (1.4.5)$$

$$l < \left|Rank\left\{\frac{\left(X_j^R\right)+\left(X_j^G\right)}{2}\right\}\right| < q - l \qquad (1.4.6)$$

iii) Global normalization

Another widely adopted genes normalization approach is the global normalization method which uses *analysis of variance* (ANOVA) model introduced by Kerr *et al* (2000). This procedure assumes linear normalization factor and incorporates both main and/or interaction effects of these factors into the ANOVA models. The global normalization model is given by

$$log\left(X_{ictj}\right) = \mu + \alpha_i + \delta_c + \tau_t + q_j + (\alpha q)_{ij} + (\tau q)_{tj} + \varepsilon_{ictj} \quad (1.4.7)$$

where $log\left(X_{ictj}\right)$ is the logarithm of the gene expression measure of gene j over cDNA array i, dye c, and tissue sample type t. Parameters μ is the overall population average log-expression (average signal), α_i represents the effect of i^{th} array, δ_c is the effect of c^{th} dye, τ_t is the effect of t^{th} tissue type, q_j is the effect of j^{th} gene, $(\alpha q)_{ij}$ is the interaction effect of i^{th} array and j^{th} gene, $(\tau q)_{tj}$ is the interaction effect of t^{th} tissue type and j^{th} gene while ε_{ictj} is an independent and identically distributed error term. This approach has been employed in many other related studies (Lee *et al*, 200; Wolfinger *et al*, 2001; Lee, 2004; etc.).

There are many other variants of normalization procedures apart from the three provided above (see Smyth *et al*, 2002; Smyth & Speed, 2003; Huber *et al*, 2003; Steinhoff & Vingron, 2006; etc.). The choice of any of the method depends on the nature of microarray data set being investigated.

A particular important aspect of normalization is data *standardization*. It is all about standardizing microarray data so that each array has zero mean and unit variance. It is a scale adjustment measure that prevents the expression measures in a particular array to dominate the overall average expression, Yang *et al* (2001).

1.4.2 ***Preliminary feature selection***

A typical microarray data set is characterized by having several thousands of q genes measured on relatively small number n of biological samples with $n < q$. Several experimental microarray studies (Botstein & Risch, 2003; Su *et al*, 2002; etc.) have revealed that very few numbers of these numerous genes are differentially expressed (DE) and might actually be relevant to the clinical status of the biological samples. Therefore, our objective here is to perform a primary selection of potentially relevant q^*genes from all the available q genes such that all the $q - q^*$ non-predictive (irrelevant) genes are removed prior to proper analysis. The reasons for this are two-fold: One is to save a lot of computation time and efforts while analysing the data. If the $q - q^*$ 'useless' genes are not removed before any dimension reduction and/or class prediction is performed, a good classifier will still filter them out during the analysis proper, but at a huge cost of analysis time. To avoid this therefore, it is proper to filter all the apparently irrelevant genes before proper analysis could begin. The second reason which is not too far from the first one is to minimize unnecessary 'noise' in the data before proper analysis could commence. In a nutshell, a good preliminary gene selection is expected to prevent undue influence of the irrelevant genes on prediction.

Among the preliminary feature selection methods commonly adopted in the literature are the *p-value* method (Golub *et al*, 1999), the *Wilcoxon-Mann-Whitney rank sum test* (Thomas *et al*, 2001), the *student-t test* or its equivalent; the *Welch test* (Nguyen & Rocke, 2002a; Rimkus *et al*, 2008) and the *Wilks' lambda score* (Dillon & Goldstein, 1984; Johnson & Wichern, 1992; Hwang *et al*, 2002) among others.

Generally speaking, no single method can efficiently be suitable to handle all kinds of microarray data sets. The choice of method to adopt at times may depend on the nature of the data or the taste of the investigator. The common denominator is to ensure that the method adopted retains all the potential differentially expressed genes among the primarily selected q^* genes.

We shall discuss the procedure of the student-t test as used in this thesis and later in Chapter 2, we propose another flexible classifier-like preliminary feature selection method – the *AUC feature selection method*- which has not been given much attention in the literature. The reasons for this shall be provided later.

It is intuitively reasonable to ask that, why seeking for further dimension reduction methods when some of the methods adopted for preliminary feature selection can perform similar function? The answers to this are two-folds. First, after the preliminary gene selection where $q - q^*$ non-DE genes are pruned out, the remaining potentially relevant q^*genes selected might still be more than what is optimally suitable for good prediction. In other words, not all the preliminarily selected q^*genes would still be suitable for good classification of mRNA samples into their respective cancer sub-classes. Hence, there is need to evolve a more robust method that would further extract the most relevant and informative k genes ($k < q^*$) from the preliminarily selected q^* genes. The second but less important reason is that, the number of q^* genes selected might still be more than n, the number of biological samples. This would again render the use of any standard regression methods practically impossible for response class prediction due to the violation of non-singularity condition of the design matrix of the predictors.

Feature selection by Student-t statistic

By Student-t statistic approach, each of the measured genes X_j, $j = 1, \dots, q$, are divided into two, X_{0j} and X_{1j} based on the response class categories (0,1) with corresponding sample sizes n_0 and n_1 respectively. The equality of the group means $\bar{X}_{0j}$ and $\bar{X}_{1j}$ is examined via the t-statistic

$$t_s = \frac{\bar{X}_{1j}-\bar{X}_{0j}}{\sqrt{\left(\frac{(n_0-1)S_0^2+(n_1-1)S_1^2}{n_0+n_1-2}\right)\times\left(\frac{n_0+n_1}{n_0 n_1}\right)}} \tag{1.4.8}$$

or its equivalent, the Welch test (Welch, 1947) that gives an approximate solution to Behrens-Fisher problem (correcting for unequal variances within each class) given by

$$t_w = \frac{\bar{X}_{1j}-\bar{X}_{0j}}{\sqrt{\frac{S_0^2}{n_0}+\frac{S_1^2}{n_1}}} \tag{1.4.9}$$

with modified degree of freedom

$$v = \frac{\left(\frac{S_0^2}{n_0}+\frac{S_1^2}{n_1}\right)^2}{\left(\frac{S_0^4}{n_0^2(n_0-1)}+\frac{S_1^4}{n_1^2(n_1-1)}\right)}$$

where, for each gene j, $\bar{X}_{yj}$, S_y^2 and n_y is the mean, the variance and the sample size for subject class y, $y = 0, 1$ respectively. The Welch approximation is often preferred in many microarray studies. The estimates of the t_w or t_s above is computed for all the q genes and for each computation a high positive t-score corresponds to high expression in favour of class 1 while the least negative t-score corresponds to high expression for class 0. The absolute values of all the t-scores are taken and subsequently sorted in descending order to identify the top q^* genes base on the estimated t (t_w or t_s) values. The cut-point for the selection of the top q^* genes from the ordered list is determined either by a pre-specified implied p-value, p^* or its critical value equivalent for the upper tail of the student-t distribution. For instance, selection of all genes whose p-values are less than or equal to $p^* = 0.001$ may be desirable. This would be equivalent to selecting all genes whose critical values, $|\hat{t}_s|$ or $|\hat{t}_w|$ values, are greater than or equal to $t_{0.999,\, n_0+n_1-2}$ or $t_{0.999,\ v}$ respectively. The higher the value of p^* chosen (i.e. as $p^* \rightarrow 1$ or as the chosen cut-point $t_\alpha \rightarrow 0$) the higher the chance of retaining more genes and vice-versa.

While using Student t-test for preliminary feature selection in this work, we have allowed our choice of cut-point p^* to be dictated by the underlying features of the various data being analysed. Our study here have shown that, it is wrong to fix a general cut-off point, say $p^* = 0.001$, as a benchmark for all microarray data sets as done in many studies irrespective of the nature of the data under study. The value of p^* used for a particular microarray data might not be suitable for another data, hence the need to consider the peculiar features of each data as a guide for selecting the cut-off points.

1.5 Research motivation and objectives

The advent of DNA microarray technology has made it possible to simultaneously study the expression profiles of several thousand of genes on a given number of mRNA samples. This has helped the researchers to have a clear understanding of different kinds of diseases like heart diseases, mental illness, infectious disease and of course, the cancer varieties. In cancer research for instance, the evolution of microarray technology has made it possible for molecular biologists and physicians to classify various sub-classes of cancer types on the basis of the patterns of gene activity in the tumour cells. This strongly underscores the biological relationship between the gene expression profiles and various sub-classes of cancer types.

In a more statistical term, let us consider a DNA microarray experiment that generated expression data on q genes $\boldsymbol{X} = (X_1, \dots, X_q)$ for n mRNA samples where response of interest represented by $Y_i, i = 1, \dots, n$, is recorded for each sample. Response variable Y_i may be binary or categorical, especially if the response of interest is the cancer tumour sub-group as in leukemia study of Golub *et al* (1999), in which case, $Y_i = 0$ for acute lymphoblastic leukemia (ALL) while $Y_i = 1$ for acute myeloid leukemia (AML). When the tumour sub-groups are more than two, typical of the molecular cancer study by Ramaswamy *et al* (2001), then the outcome variable Y_i may be given by the set $\{Y_i\} = \{0, 1, 2, \dots, \mathbb{y}\}$. Also, variable Y_i may be continuous denoting a desired continuous clinical outcome like blood pressure readings, x-rays' results, laboratory tests' results and so on.

It should be noted that both $\boldsymbol{X}$ and Y_i represent random samples from a given population of interest and it is often desirable in microarray studies to use the observed sample data on expression measures of q genes X_j, $j = 1, \dots, q$ and observed response Y_i to make inference about the population or future subjects. Specifically, a common goal is to use information on the observed data $(\widehat{\boldsymbol{X}}, \widehat{Y}_i)$ to predict independent future subject $n_* \notin \{n\}$ in the population.

Typically, microarray data sets are characterized by having very few number of experimental mRNA samples, often less than 100, on which expression levels of several thousands of genes are simultaneously being observed. Hence, the situation where $n \ll q$ is a common scenario in genomic analysis. Therefore, to predict the clinical/tumour status of future subjects n_*, a functional relationship between $\boldsymbol{X}$ and Y_i of the form $Y_i = g(\boldsymbol{X\beta};\, \varepsilon)$ may be desirable for any link function $g(.)$. If the relationship is linear, then, the task is to fit the model

$$Y_i = \boldsymbol{X\beta} + \varepsilon \qquad (1.5.1)$$

But with the condition that $n \ll q$, obviously, the linear model (1.5.1) cannot be estimated using the classical least square (LS) method. The reason for this is that, the $q \times q$ variance-covariance (design) matrix $\boldsymbol{X}^T\boldsymbol{X}$ would be singular (non-invertible).

Several attempts directed at circumventing this common dimensionality problem in microarray data resulted to the development of many supervised and unsupervised techniques for dimension reduction and tumour classification in several microarray studies. Among the earlier methods developed for response class prediction include the *support vector machines* (SVM), *k-nearest neighbours* (k-NN), *principal component analysis* (PCA), *sliced inverse regression* (SIR) and the much celebrated approach of the *partial least squares regression* (PLSR) among many others. While some of these methods (e.g. SVM, k-NN etc.) are mainly design for response class prediction, few other ones (e.g. PCA, PLS etc.) are shrinkage techniques that are only meant for dimension reduction of original q genes to a small number of k gene components, $k < n$, using the expression patterns of all the q genes. For both PCA and PLS techniques for instance, tumour classifications are only possible through the use of other standard discriminant methods like linear, logistic or quadratic discriminant analyses on the k gene components constructed.

Expectedly, some of the existing methods perform accurate classification of tumour classes using the observed gene expression profiles, but unfortunately the classifiers they provided are often difficult to interpret in relation to the tumour sub-classes they predicted. For instance, the partial least squares (PLS) procedures can only reduce the entire q genes to a few number of k gene orthogonal components, say, $Z_1, \dots, Z_k$, $k < n$ using the expression measures of all the original q genes. The constructed k components are then being used as predictors in replacement of the original q genes, $X_1, \dots, X_q$, in regression model (1.5.1) to predict the tumour categories of any future biological subjects n_* (see Nguyen & Rocke, 2002a-d; Rosipal & Kräme, 2006; Rimkus *et al,* 2008; etc.). Although, PLS method has been reputed to provide accurate predictions especially when suitable cross-validation method is employed, but regrettably in most cases, the k components it constructed for prediction are not easily tend to direct biological interpretations in relation to the response groups they predicted. This has made it imperative to evolve a separate procedure that could actually identify and select the most relevant gene combinations that are actually related to different tumour categories. Obviously, this important goal are difficult to accomplished using the factor loadings of the constructed *k* PLS components as suggested in some studies, Barker & Rayens (2003), Ding & Gentleman (2004).

One of the most important advantages of DNA microarray technology lies in gene discovery. Due to the dynamic nature of general hormone systems of individual organism, molecular biologists and physicians are not only interested in proper identification and prediction (diagnosis) of different categories of tumour types, but rather, they are now more interested about knowing those human transcripts (genes) that are responsible for each of the identified tumour conditions. Identification of these relevant transcripts would immensely help in the development of appropriate therapeutic measures (drug discovery). This could be further useful to pharmacogenomists in determining the relationship between therapeutic responses to drugs and the genetic profiles of patients. However, all these important benefits may be difficult to achieve if appropriate statistical techniques that are capable to select the most relevant and informative marker genes among several available thousands are not developed. Again, it is obvious that the latent components constructed by PLS or PCA technique might not be suitable to address this problem. It is based on this premise that the study carried out in this thesis is conceived.

The prime goal of this work therefore, is to develop a new flexible dual-purpose approach that would efficiently identify and select the most relevant gene chips that are informative enough to predict the various tumour conditions of mRNA subjects in any given microarray data set.

Our method shall be evaluated on some of the existing microarray data sets while its general performance relative to those provided by some of the few selected existing methods (PLS, SVM, k-NN, etc.) shall be examined.

The high-dimensional nature of microarray data sets has made preliminary feature selection a desirable task before further analysis like final optimal gene selection and classification are performed. Due to this end, we shall review some of the existing preliminary feature selection methods and provide yet another approach that would efficiently handle features selections at the preliminary stage. This becomes necessary because the prediction performance of any classification rules largely depends on the crop of genes selected for analyses at the preliminary selection stage.

1.6 Main research contributions

The main contributions of this research work include, but not limited to the following:

- We developed a dual-purpose flexible method that simultaneously performs informative genes selection and classifies mRNA samples into their respective biological groups using the sub-set of genes selected irrespective of the dimension of the microarray data involve.
- Our new method is capable at selecting those genes that are of biological relevance to the tumour conditions of the mRNA subjects in any given microarray data sets. This, we hope, shall be helpful in the determination of appropriate therapeutic measures for the treatment of various cancer sub-groups.

- We equally proposed a new classifier-like preliminary feature selection method that is capable at reducing the huge number of genes in any microarray data set to a manageable size by selecting all the potentially discriminative marker genes for further analysis by any standard gene selection and/or classification method. The new approach eliminates the risk of leaving out some of the important genes at the preliminary selection stage.
- In addition to all these, this research work avails us the opportunity to thoroughly review the fundamental basis of some of the existing classification techniques and offer useful contributions, suggestions and recommendations based on our experience in this study.

1.7 **Outline of the Thesis**

The rest of this thesis is arranged as follows. We presented our newly proposed sequential dimension reduction and prediction method in Chapter two including a review of various performance indices that are used to assess the efficiency of the proposed method. This is followed by introducing a new versatile preliminary feature selection procedure. We conclude this chapter by presenting an overview of some of the existing classification methods as employed in this thesis. Several simulation studies carried out and few applications of our proposed classifier are provided in Chapter three while its applications on real microarray data sets are presented in Chapter four. Chapter five presents the summary of our results, necessary conclusions and suggestions for future studies.

2 The *k*-Sequential Selection (*k*-SS) method

2.1 Introduction

The characteristic feature of a typical microarray data set has posed a lot of challenges to statisticians and experimental biologists due to high dimensional nature of such data. A typical microarray data set consists of q transcripts and response class information on n subjects with $q >> n$. In most cases, the number of transcripts measured on each biological subject ranges between 1,000 to more than 50,000 transcripts while the available experimental unit may fall below 100. Hence, the need to evolve a robust method that will be capable to identify and select from the cloud of several thousand of observed genes, the most relevant informative genes for the prediction of biological sample. This is particularly important to the biologists and physicians who are interested to know which genes have correlated expression levels with the biological samples for determination of proper therapeutic measures among other intents. We therefore present in this work, a novel but flexible approach that is capable at selecting the most relevant gene sets as well as providing accurate prediction of the tumour sub-groups of biological samples in any given genomic data. We have used some of the existing microarray data sets to demonstrate the application of our method. Nonetheless, this new approach can be applied, for instance to proteomic, chemometrics or any other data sets in which high-dimensionality is a common scenario.

Consider a total of N subjects that belong to two different population groups Ω_1 and Ω_2, $\Omega_1, \Omega_2 \in \{\Omega\}$. Let a random sample of size n be drawn from population N with n_1 from population Ω_1 and n_2 from population Ω_2, $n_1 + n_2 = n$ and let the q-dimensional vector $\boldsymbol{X} = (X_1, \dots, X_q)$, $\boldsymbol{X} \in \Re^{n \times q}$ denotes the expression levels of q genes simultaneously measured on n biological samples with $n < q$ as earlier presented in Section 1.5. Under the classical regression settings, the primary goal is to establish the association between predictor vector $\boldsymbol{X}$ and a (continuous or categorical) response variable $Y_i \in \Re$, $i = 1, \dots, n$, of the form

$$Y_i = g(\boldsymbol{X\beta};\ \varepsilon) \tag{2.1.1}$$

for some link function $g(.)$. Within the framework of this study, we define the response variable Y_i by

$$Y_i = \begin{cases} 0, \text{if } y \in \Omega_1 \\ 1, \text{if } y \in \Omega_2 \end{cases} \tag{2.1.2}$$

for any realization y of Y_i in Ω. This literally indicates a binary response group (0,1) for all the n biological samples, which by far, is the most common in many microarray studies. While implementation of our proposed method shall be demonstrated extensively on dichotomous response class microarray data sets, the extension of its application to multi-categorical response cases shall be equally discussed.

The definition of the outcome variable Y_i in (2.1.2) implies that any given subject in n is labelled 1 if it has a particular characteristic of interest of those in group $\Omega_2 \in \Omega$ and a given subject is label 0 if it possesses the features of those in group $\Omega_1 \in \Omega$. In microarray cancer studies for instance, the characteristic of interest may be patients having particular cancer tumour types labelled 1 if tumourous, and labelled 0 if the subject is normal. This particular instance existed in many studies (Alon *et al* 1999, Singh *et al* 2002, Stuart *et al* 2004, Welsh *et al* 2001, Ramaswamy *et al* 2001, *etc*.). In survival analysis studies however, such characteristic of interest may be the survival outcome of the patients after a given follow-up period with $Y_i = 1$ for death outcome and $Y_i = 0$ if the patient is still alive at the end of the study (censored).

Assuming a linear form of the link function $g(.)$ in (2.1.1), obviously it is impossible to apply the usual least square method to establish the linear relationship between $\boldsymbol{X}$ and Y_i due to dimensionality constraint imposed with $n \ll q$ as remarked in Section 1.5. Our major goal in this thesis therefore, is to design a classification rule based on variable pair (Y_i, X_{ij}), $i = 1, \dots, n$, $j = 1, \dots, q$, that will use subset $\boldsymbol{x}$ of the measured gene expressions $\boldsymbol{X}$ to correctly predict/classify any independent future subjects into either of the two biological groups $Y_i = y$, $y \in \{0,1\}$.

Most often, it is difficult to get independent samples to test the accuracy of any developed classification rule. The usual practice is to randomly partition the original sample size n into training/design sample, n_{TR} and test sample, n_{TE} using a suitable ratio. The classifiers are usually built using n_{TR} while the goodness of the classifiers is assessed on the test set n_{TE}. Some splitting ratios 2:1, 4:1 and 9:1 in favour of the training and test data respectively have been suggested in some studies (Dudoit *et al*, 2002; Lee *et al*, 2005; etc.). However, a common practice in most studies is to train the classifier with large proportion of the original data while its goodness is assessed using the remaining left-out sample.

The adverse effects associated with the partitioning of the already small biological sample n into training and test sets for classifiers' construction and assessment have been reported in many studies, e.g. see Bura & Pfeiffer (2003), Molinaro *et al* (2005), Boulesteix *et al*

(2008) and several others. A common argument is that, classifier that is constructed with a fraction of already small sample size n might underestimate the misclassification error rate (MER). In other words, a classifier that is trained with a relatively large number of subjects is likely to provide more accurate and stable results than the one trained with smaller sample. In view of this fact, we have adopted a splitting ratio of 19:1 for $n_{TR}:n_{TE}$ in this thesis. This literally translates to using 95% of original n subjects to build our classifier and using the remaining 5% as external data to evaluate the performance of the classifier. The justification of our choice is discussed in Section 3.5. With this partitioning ratio, sufficient part of the original data is used to construct the classifiers which considerably improved prediction results as shall be seen later.

To further ensure generalization and stability of results, several replicates of the original data sets are generated at the construction and evaluation stages of our classifier using sub-sampling technique of Monte Carlo Cross Validation (MCCV) (Dudoit *et al*, 2002), Bootstrap (Efron & Gong 1983), and Bootstrap .632+ (Efron and Tibshirani, 1997). The details of these sampling methods as adopted in this thesis are provided in Section 2.5.

Since the variable selection and class prediction method proposed in this thesis adapts the estimation procedures of *logistic regression* method, in the next two sections therefore, we briefly provide the basic theoretical background into the *generalized linear models* (GLM) and *logistic discriminant* (LD) analysis.

2.2 **Generalized Linear Models (GLMs)**

Under the classical *linear regression models* (LRMs), the relationship between the response variable $Y_i \in \Re^{n\times 1},\ i = 1, \dots, n$, and a set of predictors $\boldsymbol{X} = (X_1, \ldots, X_q)$, $\boldsymbol{X} \in \Re^{n \times q}$ in the form $Y = \boldsymbol{X\beta} + \varepsilon$, $\boldsymbol{\beta} = (\boldsymbol{\beta_1}, \ldots, \boldsymbol{\beta_q})^{\boldsymbol{T}}$, is usually established by assuming Gaussian distribution with constant variance, σ^2 for both Y and the error component, ε with each of them having means $\hat{Y} = \boldsymbol{X\hat{\beta}}$ and zero respectively.

When the outcome variable Y is not Gaussian, but rather dichotomous with distinct class labels $(0, 1)$, then, the Gaussian distribution cannot be assumed for Y. This implies that the linear regression model $Y = \boldsymbol{X\beta} + \varepsilon$ cannot be fitted on Y because the range of the conditional expectation $\hat{Y} = E(Y|\boldsymbol{X})$ is no loger bounded between zero and one.

The *generalized linear model* (GLM), first developed by John Nelder & Robert Wedderburn in 1972, provides a flexible generalization of the linear regression concepts which unifies various other statistical models including linear, logistic, Poisson and many other regression models with or without Gaussian responses under one framework. This led to the development of general algorithms for the maximum likelihood estimation (MLE) of all the models' parameters.

In GLM, each response variable Y is assumed to come from a particular member of the *exponential family of distributions* (EFD) with a probability distribution $f_Y(y_i; \theta_i, \omega)$, $\theta_i, \omega \in \Theta$. The form of this distribution is given by

$$f_Y(y_i; \theta_i, \omega) = exp\left\{\frac{[y_i\theta_i - b(\theta_i)]}{a(\omega)} + c(y_i, \theta_i)\right\} \qquad (2.2.1)$$

where, $a(.)$, $b(.)$, $c(.)$ are known functions that take the form of y_i. For each form of y_i, θ_i is the natural parameter. The dispersion function $a(\omega)$ is sometimes written as $a(\omega) = \frac{\omega}{w_i}$ where ω is the dispersion parameter which is constant for all observations and w_i is a prior weight meant to correct for the violation of unequal variances which might arise contrary to the constancy of ω already assumed.

The EDF family include among others, the Gaussian, Binomial, Poisson, Exponential and Gamma distributions.

Consider a general regression function $Y = g(\boldsymbol{X\beta};\, \varepsilon)$ as defined in (2.1.1). Within the framework of GLM, the relationship between the random component, $\mu = E(Y|\boldsymbol{X})$ and the systematic component $\eta = \boldsymbol{X\beta}$, a linear combination of the predictors, is specified by a linear or non-linear monotonic and differentiable link function $\eta = g(\mu)$. This link is a function of response variable Y which enables the relationship between Y and vector of predictors $\boldsymbol{X}$ to be linear in parameter $\boldsymbol{\beta}$. Dropping subscript i from θ_i for simplicity, it then follows from EFD in (2.2.1) that

$$E(Y|\boldsymbol{X}) = \mu = b'(\theta) \tag{2.2.2}$$

$$Var(Y|\boldsymbol{X}) = b''(\theta)a(\omega) \tag{2.2.3}$$

where $b'(\theta)$ and $b''(\theta)$ are the first and second derivatives of $b(\theta)$ respectively.

The special case of the link function which concerns us here is the *logit link* when response variable Y is Bernoulli distributed with $Y =$ 0 or 1. Here, the link function is given by $g(\mu) = log\left[\frac{p(Y=1|\boldsymbol{X})}{p(Y=0|\boldsymbol{X})}\right] = \boldsymbol{X\beta}$. More details on this are provided in the next section. Other forms of GLMs as applied into different fields can be found in Bliss (1935), Berkson (1944), Cox (1972), Finney (1972), Kleinbaum & Kupper (1978), Draper & Smith (1981), McCullagh & Nelder (1989) and many others.

2.3 The basics in logistic discriminant analysis

The concepts of the logistic regression analysis are the primary basis for the construction of *logistic discriminant* (LD) analysis technique. Ripley (1996), Dudoit *et al* (2002) and several other authors have argued at different times in favour of using LD analysis for class prediction purposes. Their unanimous conclusion is that LD analysis provides a more direct and unambiguous way of estimating the posterior probabilities $p(Y = y|\boldsymbol{X})$ that are used in the construction of *logistic discriminant* (LD) rules. It has been equally reported that LD procedure tends to more easy generalization than some of the other classifiers like *linear discriminant analysis* (LDA) and *quadratic discriminant analysis* (QDA), Dudoit *et al* (2002).

Suppose we consider a set of n biological samples belonging to two outcome groups (0,1) according to response variable Y as defined in (2.1.2). Let $\boldsymbol{x} = (X_1, \ldots, X_k)$ be the subset of measured q genes $\boldsymbol{X} = (X_1, \ldots, X_q)$, $\boldsymbol{X} \epsilon \Re^{n \times q}$, $k < q$, selected using a suitable variable selection method for predicting the response group Y.

Suppose that all the n samples represent independent and identically distributed random samples from an unknown distribution Ψ over $\mathbb{X} \times \mathbb{Y} \in \Re$, $\mathbb{X}$ and $\mathbb{Y}$ being the feature space of $\boldsymbol{x}$ and Y respectively. Without loss of generality therefore, the LD rule $\varphi(\boldsymbol{x})$ to be constructed can be seen as the mapping of $\mathbb{X}$ into the real line $\mathbb{Y}$ i.e. $\varphi(\boldsymbol{x}): \mathbb{X} \rightarrow \mathbb{Y}$ (for continuous response variable Y) or as the partitioning of the feature space $\mathbb{X}$ into y disjoint and exhaustive groups $\mathbb{X}(y)$ of $\mathbb{Y}$ (for categorical response variable Y), $y = 0, 1, \ldots, \mathbb{y}$. For binary response class, $\mathbb{y} = 1$. Therefore, the predicted response class $\hat{Y}$ by classification rule $\varphi(\boldsymbol{x})$ based on the observed feature $\boldsymbol{x}$ can be denoted by $\hat{Y} = \hat{\varphi}(\boldsymbol{x})$.

Since response variable Y can only assume value 0 or 1 it follows that Y is Bernoulli distributed with parameter $\pi(\boldsymbol{x})$. The equivalent form of the linear model $Y = E(Y|\boldsymbol{x}) + \varepsilon$ under this condition is given by

$$Y = \pi(\boldsymbol{x}) + \varepsilon \qquad (2.3.1)$$

which implies that

$$\varepsilon = Y - \pi(\boldsymbol{x}) \qquad (2.3.2)$$

Thus, from (2.3.2) it is obvious that $E(\varepsilon) = 0$ and $Var(\varepsilon) = \pi(\boldsymbol{x})[1 - \pi(\boldsymbol{x})]$. This shows that under the regression form in (2.3.1), the error term ε, though has zero mean but do not have constant variance (σ^2 as in Gaussian model) but rather, an heteroscendastic form that depends on the values of $\boldsymbol{x}$. A specific form of $\pi(\boldsymbol{x})$ is the logistic regression function given by

$$p(Y = 1|\boldsymbol{x}) = \pi(\boldsymbol{x}) = \frac{exp\,(\boldsymbol{x\beta})}{1+exp\,(\boldsymbol{x\beta})} \qquad (2.3.3)$$

The quantity that transforms $\pi(\boldsymbol{x})$ as a linear function of $\boldsymbol{X}$ and $\boldsymbol{\beta}$ is the *logit link* $\eta(\boldsymbol{x})$ as described in Section 2.2 and is given by

$$\eta(\boldsymbol{x}) = ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \boldsymbol{x\beta} \qquad (2.3.4)$$

Thus, when Y has two groups (0,1), the link function $\eta(\boldsymbol{x})$ is the natural logarithm of the ratio of conditional probability $p(Y = 1|\boldsymbol{x})$ and $p(Y = 0|\boldsymbol{x}) = 1 - p(Y = 1|\boldsymbol{x})$. That is, $\eta(\boldsymbol{x}) = ln\left[\frac{p(Y=1|x)}{p(Y=0|x)}\right] = \boldsymbol{x\beta}$.

Now, given any n biological samples with dichotomous class group Y and a vector of observed predictors (*genes*) $\boldsymbol{x}$, the parameter vector $\boldsymbol{\beta}$ of the logistic regression model (2.3.3) can then be estimated iteratively using the *iterative weighted least squares* as implemented in the Newton-Raphson algorithm (Anderson *et al*, 1993). This is the

GLM procedure for a regression model with binary (Bernoulli) outcome variable Y.

After the fit of the logit model (2.3.4) as described above, the next task is to construct a *logistic discriminant* (LD) rule, $\varphi(\boldsymbol{x})$ that would be used to predict the response class Y_0 of any independent external subjects n_{TE}. By procedure of LD rule, the response class predictions are made using the estimated conditional probability $\hat{p}(Y = y|\boldsymbol{x})$, $y \in \{0, 1\}$. The predicted class of any subject is then given by $\hat{y} = I\big(\hat{p}(y|\boldsymbol{x}) > 1 - \hat{p}(y|\boldsymbol{x})\big)$ where $I(.)$ is an indicator function that is 1 if its argument is true and 0 otherwise. Thus, subject i would be classified by rule $\varphi(\boldsymbol{x})$ into class $y \in Y$ if it has the highest estimated posterior probability $\hat{p}(y|\boldsymbol{x})$ of being in that class. Therefore, the connection between $\varphi(\boldsymbol{x})$ and $\hat{p}(y|\boldsymbol{x})$ could be stated as

$$\varphi(\boldsymbol{x}) = \ argmax_y\, \hat{p}(y|\boldsymbol{x}) \qquad (2.3.5)$$

The predicted conditional probabilities $\hat{p}(y|\boldsymbol{x})$ may be formally converted to the predicted class labels $\hat{y} \in \{0, 1\}$ for each subject by choosing a cut-point c, $0 < c < 1$, which finally yield the following classifications;

$$\varphi(\boldsymbol{x}) = \begin{cases} 1, & \text{if}\, \hat{p}(1|\boldsymbol{x}) \ \geq \ c \\ 0, & \text{if}\, \hat{p}(0|\boldsymbol{x}) \ < \ c \end{cases} \qquad (2.3.6)$$

By (2.3.6), a subject would be classified into response class 1 if its estimated posterior probability $\hat{p}(1|\boldsymbol{x}) \geq c$ and into class 0 if otherwise.

If the sample class prior probabilities $\hat{p}_y = n(y)/n$, $n(y)$ being the number of class y subjects in the sample, $y = 0, 1$, are very close to 0.5, the choice of 0.5 for value of c has been found more appropriate. But if one of these priors is very close to 1, then, it is recommended to use the estimated prior probability of class 0 as the cut point, c. Nonetheless, the general practice, which we equally adopted here is to use $c = 0.5$ (Efron,1975; O'Gorman & Woolson, 1991; Pohar *et al*, 2004; etc.). The close connection between *logistic discriminant* (LD) analysis and *linear discriminant analysis* (LDA) has been equally reported as a factor that favours the choice of 0.5 cut-point in logistic discriminant analysis (Efron,1975; Hosmer & Lemeshow, 1989).

If the class conditional density of $\boldsymbol{x}$ given Y, $p(\boldsymbol{x}|y)$, $y = 0,1$, is multivariate Gaussian with mean μ_y and constant variance-covariance matrix Σ, i.e., $\boldsymbol{x}|y \sim N(\mu_y,\ \Sigma)$, then, with known class prior probabilities $\hat{p}_y$, the posterior probability of Y given $\boldsymbol{x}$, $p(y|\boldsymbol{x})$, is provided by *Bayes theorem* as $p(y|\boldsymbol{x}) = \frac{p(y)p(x|y)}{p(x)}$. With known class priors $p(1) = \hat{p}_1$ and $p(0) = \hat{p}_0$ of the subjects' groups $y \in \{0,1\}$, the logarithm of the ratio $\frac{p(1|\boldsymbol{x})}{p(0|\boldsymbol{x})}$ is given by $ln\left\{\frac{p(Y=1|\boldsymbol{x})}{p(Y=0|\boldsymbol{x})}\right\} = ln\left\{\frac{\hat{p}_1 p(\boldsymbol{x}|Y=1)}{\hat{p}_0 p(\boldsymbol{x}|Y=0)}\right\}$. This implies that,

$$ln\left\{\frac{p(Y=1|\boldsymbol{x})}{p(Y=0|\boldsymbol{x})}\right\} = ln\left\{\frac{\frac{\hat{p}_1}{(2\pi)^{\frac{q}{2}}|\Sigma|^{\frac{1}{2}}}exp\left[-\frac{1}{2}(\boldsymbol{x}-\mu_1)^T\Sigma^{-1}(\boldsymbol{x}-\mu_1)\right]}{\frac{\hat{p}_0}{(2\pi)^{\frac{q}{2}}|\Sigma|^{\frac{1}{2}}}exp\left[-\frac{1}{2}(\boldsymbol{x}-\mu_0)^T\Sigma^{-1}(\boldsymbol{x}-\mu_0)\right]}\right\} \tag{2.3.7}$$

which reduces to

$$ln\left\{\frac{p(Y=1|\boldsymbol{x})}{p(Y=0|\boldsymbol{x})}\right\} = \boldsymbol{x}\Sigma^{-1}(\mu_1 - \mu_0) - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + ln\frac{\hat{p}_1}{\hat{p}_0} \tag{2.3.8}$$

$$\rightarrow \qquad ln\left\{\frac{p(Y=1|\boldsymbol{x})}{p(Y=0|\boldsymbol{x})}\right\} \cong \boldsymbol{x\beta} \tag{2.3.9}$$

This is the same as the logit model given in (2.3.4).

The procedure (2.3.7) through (2.3.9) simply provides alternative way of constructing logistic discriminant (LD) function especially when predictor vector $\boldsymbol{x}$ has multivariate Gaussian density.

However, it can be easily shown (Cornfield, 1962; Lachenbruch, 1975; Hosmer & Lemeshow, 1989) that the estimates of the k parameters $\boldsymbol{\beta} = (\beta_{\mathbf{1}}, \dots, \beta_{\boldsymbol{k}})^{\boldsymbol{T}}$ of the model (2.3.9) can be obtained non-iteratively from LD functions in (2.3.8) as follows:

$$\hat{\beta}_{\mathbf{1}} = ln\frac{\hat{p}_1}{\hat{p}_0} - 0.5(\mu_1 - \mu_0)^T\Sigma^{-1}(\mu_1 + \mu_0) \qquad (2.3.10)$$

$$\widehat{\boldsymbol{\beta}}^{k-1} = (\mu_1 - \mu_0)^T\Sigma^{-1} \qquad (2.3.11)$$

where $\hat{\beta}_{\mathbf{1}}$ is the estimate of the constant parameter (intercept) of logistic regression model (2.3.3) and $\widehat{\boldsymbol{\beta}}^{k-1} = (\hat{\beta}_{\mathbf{2}}, \dots, \hat{\beta}_{\boldsymbol{k}})$ are the estimates of the remaining $k-1$ parameters. All the parameters are obtained by substituting the estimators of Σ and μ_y, $y = 0{,}1$, into (2.3.10) and (2.3.11). Thus, for subject group Y, μ_y is estimated by the mean of predictors X_{yj}, $j = 1, 2, \dots, k$, as $\hat{\mu}_y = \bar{X}_{yj}$ and covariance Σ is estimated by the estimate of the pooled sample variance-covariance defined by subjects group Y as

$$\hat{\Sigma} = \frac{(n_0-1)\boldsymbol{S}_0+(n_1-1)\boldsymbol{S}_1}{n_0+n_1-1} \qquad (2.3.12)$$

where $\boldsymbol{S}_y$ is the $k \times k$ unbiased estimator of the sub-groups variance-covariances computed for each subjects' groups as defined by Y.

The discriminant function estimators given above may be bias, especially when normality condition does not hold for the predictors. It may however, be adopted for preliminary analysis after which the final parameter estimates can be obtained using a more robust *maximum likelihood estimation* (MLE) as implemented in Newton-Raphson algorithm or any other suitable iterative procedure as earlier discussed.

2.4 The *k*-SS set-up

2.4.1 *The need for k-SS technique*

The new method proposed in this thesis is a comprehensive but flexible dual-purpose gene selection technique which simultaneously performs dimension reduction, informative genes selection and accurate classification of biological samples into their respective tumour sub-classes in any given high-dimensional microarray data. This new procedure is analogous to the non-linear stepwise variable selection technique under the classical logistic regression settings. The prime objective is to develop a robust variable selection approach that will provide flexible but efficient models that are suitable for proper prediction of biological samples in any given genomic data sets. Our procedure would select the most informative predictors (genes) from the cloud of several available thousand of genes based on some fixed decision rules.

The variable selection procedure of the *stepwise logistic regression* (SLR) for instance, as implemented in some statistical packages [e.g. SAS® (SAS institute Inc., 1995), SPSS 12.0 (Chicago, IL), STATA/SE 8.0 (Stata Corporation, Texas, USA)] is purely based on two parameters: SLENTRY, p_e which is the significant level specified for any variable to enter the model and SLSTAY, p_s which is the significant level for a variable selected to remain in the model. A major flaw of the SLR method is that the values of both p_e and p_s are determined arbitrarily by the investigator the choice of which may, of course, vary from one person to another. Hence, the whole procedure under this set-up is not too far from a trial and error exercise. Nonetheless, the SLR approach has been successfully adopted in many studies (Stevens *et al*, 1992; Seligman & Pullinger, 1996; Valenzuela *et al,* 1997; etc.) and is still in use till date.

Apart from SLR method, several other approaches have been proposed purposely to shrink the number of predictors in any regression set-up. For instance, a shrinkage method, the *least absolute shrinkage and selection operator* (LASSO), proposed by Robert Tibshirani (1996) uses quadratic programming technique to minimize the *residual sum of squares* subject to the sum of absolute value of the coefficients being less than a predetermined constant. In other words, LASSO method provides the estimate of parameters $\hat{\boldsymbol{\beta}} = \arg min \left\{\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{q}\beta_j X_{ij}\right)^2\right\}$ subject to the constraint that $\sum_{j=1}^{q}|\beta_j| \leq t$. Here, the value of t, the tuning parameter, is usually fixed by the user, which, like the choice of p_e and p_s under the SLR method, might vary from one investigator to another. Similar arguments hold for the use of *non-negative Garrote* method due to Breiman (1993) for features selection.

Another method reported in Zucknick *et al* (2008) is the univariate filtering method that equally adapts the *logistic regression* approach concept in its implementation. In this approach, the *logit model* is fitted to each of the gene variable X_j, $j = 1, \dots, q$, separately and the gene effects, $\frac{\hat{\beta}_j}{s.e(\hat{\beta}_j)}$ is computed where $\hat{\beta}_j$ and $s.e(\hat{\beta}_j)$ is the estimated regression coefficient and its standard error for gene X_j respectively. The best set of q^* genes, $q^* < q$, with the largest absolute effects $\frac{|\hat{\beta}_j|}{s.e(\hat{\beta}_j)}$ are then selected using arbitrarily chosen cut-point $\hat{\beta}_0$.

It has been established in many studies that the use of the default significant level $\alpha = 0.05$ or less for p_e in the implementation of SLR method may yield a highly sensitive selection criteria that might result into the exclusion of some of the important variables from the model (Bendel & Afifi, 1977; Hosmer & Lemeshow, 1989; Shtatland *et al* 2000, etc). On the other hand, if the value of p_e is set too high, the resulting model might be loaded with noise due to the presence of both needed and unwanted variables in the model (Hosmer & Lemeshow, 1989). Therefore, there is need to strike a balance between the selection of not too sensitive and not too conservative values for both p_e and p_s.

In an attempt to solve this problem, Hosmer & Lemeshow (1989) advocated the choice of p_e (i.e. α) between 0.15 to 0.25 and further suggested a choice of $p_s > p_e$ for any given value of p_e within this range. However, this submission sharply contradicts what was proposed by SAS institute Inc., 1995, page 51, in which a value relatively smaller than 0.05 is suggested for p_e. Specifically, it was remarked that the choice of $p_e < 0.05$ could be a better choice if the sole objective of performing variable selection is to describe and interpret the data under investigation. These differing positions notwithstanding, what is common to all the submissions is that the choice of p_e and p_s are highly subjective and are at the discretion of the investigator.

Shtatland *et al* (2000) proposed alternative approach; *output delivery system* (ODS) to the SLR implementation. Their approach uses both *Akaike information criterion* (AIC), Akaike (1974, 1983) and *Schwarz information criterion* (SIC), Schwarz (1978) for variable selection. Here again, any arbitrary values very close to 1 are suggested for both p_e and p_s in the implementation of their method.

Basically, two main objectives are desirable while performing variable selections which might apparently result to the development of two different models in any given regression problems. One might be to develop a parsimonious model (with fewer predictors) that best describes and interprets the data at hand. To select variables for this kind of model, the values of p_e to chose may range from 0.001 to 0.05, Shtatland *et al* (2000), as equally recommended by SAS institute Inc., 1995. Secondly, another objective might be to have a robust model that best predicts the response class. For this type of model, the use of default significant level $\alpha = 0.05$ or less for p_e might not be suitable, the reason why any value between 0.15 and 0.25 was suggested for α (p_e) by Hosmer & Lemeshow (1989). Considering the above two possible models, it is clear that more variables are likely to be selected under the latter than the former. This clearly suggests that, a single regression model might not be capable enough to provide both the best fit and best prediction of the response class at the same time. A good regression model that fits (describes) a data very well might poorly predict the response class (Hosmer & Lemeshow, 1989).

In any microarray studies however, two important objectives are always intended. One is to identify and select the few marker genes whose expression patterns are related to the various cancer tumour status of the biological subjects under study. In other words, it is mostly intended to identify those genes whose expression levels could, for instance, accelerate the discovery of key biological processes for proper therapeutic measures among other things. The next is to correctly classify the subjects into their respective biological groups (e.g. cancerous or normal) based on the expression levels of the marker genes already identified and selected. This usually serves as a measure to screen the mRNA samples for early detection of cancer or other tumour types before it metastasize to other neighbouring cells. The major tasks in this thesis are therefore targeted at achieving these two cardinal objectives by

i) identifying and selecting the most relevant marker genes that are related to the biological properties of the tissue samples.

ii) classifying the RNA samples properly into their respective tumour classes based on the selected marker genes.

Therefore, the sequential variable selection procedure we proposed here is basically aimed at building models not just for data description or interpretation but also for accurate prediction of tumour conditions of the biological samples. Our new method shall strive to optimize both the variable selection and response class prediction processes by ensuring that the criteria set for achieving the best optimal prediction model are not subjectively imposed by the investigator as common to most of the existing methods.

2.4.2 The *k-SS set-up in details*

Let the q-dimensional vector $\boldsymbol{X} = \left(X_1, \dots, X_q\right)$, $\boldsymbol{X} \epsilon \Re^{n \times q}$ of measured q genes on n biological samples with two outcome groups $Y \in \{0,1\}$ be as defined under Section 2.1. Our task in this thesis is to develop a k-sequential selection and prediction (k-SS) rule $\varphi(\boldsymbol{x})$ that would select the most informative k genes subset $\boldsymbol{x} = (X_1, \dots, X_k)$ from $\boldsymbol{X}$, $k < q$, to predict the binary response classes {0,1} of any future (external) subjects $n_* \notin \{n\}$

As discussed in Chapter one, Section 1.4.2, a preliminary selection of q^* genes, $q^* < q$, may be necessary to filter out the irrelevant genes from the whole q genes to a manageable size number, q^* before the final selection of the most informative k marker genes are made for classification purposes. This concept shall be revisited in Section 2.5 where we propose a new preliminary gene selection procedure based on *cross-validated area under the receiver operating characteristics curve* (CVAUC). However, both q^* and q may be used interchangeably in this thesis to mean a large set of genes from which the selection of k informative marker genes is desirable.

We begin by dividing randomly, the original sample size n into training set n_{TR} and test set n_{TE} as described earlier. This is followed by fitting univariate *generalized linear models* (GLMs) $glm_1, \dots, glm_q$ with *logit link* (i.e. $logit(\pi(X_j)) = \alpha + \beta_j X_j$, $j = 1, \dots, q$) on each of the q genes (variables) using the training set n_{TR} and constructing classification rules $\varphi(X_j) = argmax_y \, \hat{p}(y|X_j)$ for each gene X_j, $j = 1, \dots, q$, and predict the two class labels $\{0,1\}$ of the (external) test sample n_{TE} via the following classification scheme;

$$\hat{\varphi}_i(X_j) = \begin{cases} 1, & \text{if } \hat{p}_i(1|X_j) \geq 0.5 \\ 0, & \text{if } \hat{p}_i(0|X_j) < 0.5 \end{cases}, \; i = 1,2, \dots, n_{TE}. \qquad (2.4.0)$$

For each of the true response class y_i, $i = 1, \dots, n_{TE}$, of the test sample predicted by $\hat{\varphi}_i(X_j)$, $j = 1, \dots, q$, the risk (error) of misclassifying any subject is estimated through the *loss function* $L\{\varphi_i(X_j), Y_i\}$. We shall digress a little here to provide a brief discussion on the prediction error rate's estimators.

The *true error* of misclassification by rule $\varphi(X_j)$ is usually defined by

$$\vartheta_j = E_{XY \sim \Psi}[L\{\varphi_i(X_j), Y_i\}]$$

$$\rightarrow \; \vartheta_j = E_{XY \sim \Psi}\left[I_{\{\varphi_i(X_j) \neq Y_i\}}\right], \; 0 \leq \vartheta_j \leq 1, \qquad (2.4.1)$$

where $I_{\{.\}}$ is an indicator function with a value of 1 if its argument is true and 0 if otherwise. Since the joint distribution, Ψ of $\boldsymbol{x}$ and Y in (2.4.1) is not known, the true error (conditioning on both $\boldsymbol{x}$ and Y) cannot be determined directly. The usual practice is to estimate ϑ_j by its *empirical risk* using observed finite independent sample, in this case, the test sample n_{TE}. This is computed by

$$\hat{\vartheta}_j = \frac{1}{n_{TE}} \sum_{i=1}^{n_{TE}} \left[I_{\{\hat{\varphi}_i(X_j) \neq Y_i\}} \right] \tag{2.4.2}$$

and it measures the proportion of the subjects in the test sample that are incorrectly classified by classification rule $\varphi_j(X_j)$ (Efron and Tibshirani, 1997). We shall therefore, call $\hat{\vartheta}_j$ the *misclassification error rate* (MER) and in a later section, we are going to present two other variants of the MER's estimators; *the brier score* which considers the discrepancies between the true class labels and the estimated conditional (*posterior*) probabilities $\hat{p}(y|\boldsymbol{x})$, $y = 0, 1$, of subjects belonging to that class and *the logarithmic scores* which equally uses $log\{\hat{p}(y|\boldsymbol{x})\}$ in its error rate estimation.

Generally, the empirical error rate of classification rule $\varphi(\boldsymbol{x})$ constructed using any subset of measured feature $\boldsymbol{x}$ is given by

$$\hat{\vartheta} = \frac{1}{n_{TE}} \sum_{i=1}^{n_{TE}} \left[I_{\{\hat{\varphi}_i(x) \neq Y_i\}} \right] \tag{2.4.3}$$

where indicator function $I_{\{.\}}$ is as defined in (2.4.1), $0 \leq \hat{\vartheta} \leq 1$, $\hat{\varphi}_i(\boldsymbol{x}), Y_i \in \{0,1\}$.

Using the MER concepts and its estimator as presented above, the response class predictions by discriminant rules $\varphi_i(X_1), \dots, \varphi_i(X_q)$ produced a set of q MERs $\hat{\vartheta}_1, \dots, \hat{\vartheta}_q$, one for each predictor (gene) X_j, $j = 1, \dots, q$. From each prediction made by $\varphi_i(X_j)$, a 2×2 *confusion matrix*, typical of the one given in *Table 2.1* can be constructed. The confusion matrix cross-classifies the predicted response class (predicted by classification rule $\varphi(\boldsymbol{x})$) by the observed true class labels, the confusion being in the off-diagonal cells. This matrix enables us to see at a glance, in the main and off-diagonals, the number of subjects that are correctly and incorrectly classified by rule $\varphi(\boldsymbol{x})$ respectively. From this matrix, several performance indices can be estimated to assess the goodness of the classifier. For instance, from *Table 2.1*, MER can be simply estimated by $(b+c)/(a+b+c+d)$.

		True Class (T)		
		1	0	Marginal Total
Predicted class (P) by $\varphi(X)$	1	a	b	$a+b$
	0	c	d	$c+d$
	Marginal Total	$a+c$	$b+d$	$a+b+c+d$

Table 2.1: A typical confusion matrix showing the cross-classification of subjects by their true class labels T and predicted class labels (P) by classification rule $\varphi(\boldsymbol{x})$

A number of re-sampling techniques are commonly adopted in the literature to eliminate bias from the estimated prediction error rates. This is termed *cross-validation* (CV) and it starts by drawing randomly, sub-samples of the training set n_{TR} from the original n samples R number of times (*with or without replacement*). The classification rules are constructed on n_{TR} while the response categories of the remaining test samples n_{TE} are predicted using the constructed classification rules for each successive sample drawn over R repetitions. A set of R MERs $\hat{\vartheta}_{1j}, \dots, \hat{\vartheta}_{Rj}$, are then computed for each gene variable X_j after which the q average MERs $\hat{\hat{\vartheta}}_1, \hat{\hat{\vartheta}}_2, \dots, \hat{\hat{\vartheta}}_q$ are estimated for all the q gene variables $X_1, \dots, X_q$ respectively. The average MERs $\hat{\hat{\vartheta}}_j$, $j = 1, \dots, q$ now become the cross-validated MERs and their estimate are expected to be more efficient than the MERs

$\hat{\vartheta}_j$ which are estimated based on a single sample. A typical table of matrix of the MERs $\hat{\vartheta}_{rj}$ provided by classifier $\varphi(\boldsymbol{x})$ at different repetitions for each gene X_j is presented in *Table 2.2*. Detail discussions on various cross-validation methods are provided in Section 2.7.

	Genes X_j			
Repetitions	X_1	X_2	...	X_q
	Misclassification error rates (MERs) $\hat{\vartheta}_{rj}$			
1	$\hat{\vartheta}_{11}$	$\hat{\vartheta}_{12}$	...	$\hat{\vartheta}_{1q}$
2	$\hat{\vartheta}_{21}$	$\hat{\vartheta}_{22}$	...	$\hat{\vartheta}_{2q}$
⋮	⋮	⋮	...	⋮
R	$\hat{\vartheta}_{R1}$	$\hat{\vartheta}_{R2}$	...	$\hat{\vartheta}_{Rq}$
Mean MERs	$\hat{\hat{\vartheta}}_1$	$\hat{\hat{\vartheta}}_2$	...	$\hat{\hat{\vartheta}}_q$

Table 2.2: A typical table of matrix of misclassification error rates (MERs) provided by classification rule $\varphi(\boldsymbol{x})$ for each gene X_j at different repetitions.

At this stage, all the q gene variables might be ordered in order of their prediction performance based on their respective average MER values. Suppose we allow the sequence $\hat{\hat{\vartheta}}_{(1)}, \hat{\hat{\vartheta}}_{(2)}, \dots, \hat{\hat{\vartheta}}_{(q)}$ be the observed order of the above observed q average MERs satisfying the condition that $\hat{\hat{\vartheta}}_{(1)} < \hat{\hat{\vartheta}}_{(2)} < \cdots < \hat{\hat{\vartheta}}_{(q)}$. Based on this ordered average MERs we let the corresponding order of all the original q genes be given by $X_{(1)}, X_{(2)}, \dots, X_{(q)}$. By this representation, gene $X_{(1)}$ with estimated mean MER $\hat{\hat{\vartheta}}_{(1)}$ becomes the best gene followed by the second best $X_{(2)}$ with respective mean MER estimate $\hat{\hat{\vartheta}}_{(2)}$ and so on. However, if we define $X^{m_1} = X_{(1)}$ and $\hat{\hat{\vartheta}}_{(1)} = \hat{\hat{\vartheta}}^{m_1}$, then superscript m_1 would indicate that gene X^{m_1} is the first gene with minimum average MER contribution to be selected into our prediction model. Thus, $\hat{\hat{\vartheta}}_{(1)} = \hat{\hat{\vartheta}}^{m_1} = min\left(\hat{\hat{\vartheta}}_{(1)}, \hat{\hat{\vartheta}}_{(2)}, \dots, \hat{\hat{\vartheta}}_{(q)}\right)$.

Under the conventional stepwise variable selection procedure, the importance of any variable to enter the model is judged by an arbitrarily selected implied significance level p_e against which the respective p-values of the estimated likelihood ratio statistics are compared. Under this new proposal however, individual genes and their combinations are judged to be suitable for inclusion into the model based on their predictive strength of the response classes. This we simply assessed through their estimated MER values. By this criterion, the marginal contribution of each selected gene at reducing the prediction error rate of the successive models is examined. If this marginal contribution is significant enough based on some test criteria (*to be developed*), the selected gene is retained in the model, but if otherwise, it is not selected. The significant level(s) α at which the best set of genes are selected is determined through internal cross-validation and is not to be subjectively fixed by the investigator. At the end of the whole exercise, the combination of genes that yielded the minimum overall estimated average MER value among the family of all possible gene combinations in the data is chosen as the best by our method.

In a nutshell, our sequential selection procedure begins at *step 0* with the selection of gene X^{m_1}, being the gene that yielded the minimum mean MER $\hat{\bar{\vartheta}}^{m_1}$ among all the q genes. To determine whether any of the remaining $q-1$ genes is important once the gene X^{m_1} is in the model, we construct $q-1$ classification rules $\varphi^{m_{1(2)}}(\boldsymbol{x})$, $\varphi^{m_{1(3)}}(\boldsymbol{x})$, … , $\varphi^{m_{1(q)}}(\boldsymbol{x})$ on the respective gene pairs $X^{m_1}\,X_{(2)}$, $X^{m_1}\,X_{(3)}$, … , $X^{m_1}\,X_{(q)}$ according to the same scheme given in (2.4.0). Based on the constructed $q-1$ prediction rules, the response classes of the test sample n_{TE} are predicted and with the use of suitable cross-validation technique the respective average MERs $\hat{\bar{\vartheta}}^{m_{1(2)}}, \hat{\bar{\vartheta}}^{m_{1(3)}}, \dots, \hat{\bar{\vartheta}}^{m_{1(q)}}$ are computed.

Let $X^{m_1}X^{m_2} \in \left\{X^{m_1}\,X_{(2)}, X^{m_1}\,X_{(3)}, \dots, X^{m_1}\,X_{(q)}\right\}$ be the gene pair that yielded the minimum average MER defined by $\hat{\hat{\vartheta}}^{m_1m_2} = min\left(\hat{\hat{\vartheta}}^{m_{1(2)}}, \hat{\hat{\vartheta}}^{m_{1(3)}}, \dots, \hat{\hat{\vartheta}}^{m_{1(q)}}\right)$. Therefore, at *step 1*, gene X^{m_2} is chosen for possible consideration into our prediction model for being the gene that contributed to the estimated minimum mean MER $\hat{\hat{\vartheta}}^{m_1m_2}$ out of the remaining $q-1$ genes. Like m_1, subscript m_2 in the above representations also indicates that gene X^{m_2} is the second gene, with minimum average MER contribution, desirable for consideration into our prediction model. Thus, gene X^{m_2} becomes the next best gene candidate suitable for selection into the model provided it satisfies certain test criteria.

Without loss of generality therefore, for any set of sequentially selected genes $X^{m_1}X^{m_2}\dots X^{m_{j+1}}$, the last gene $X^{m_{j+1}}$ is the next best $(j+1)^{th}$ gene to be considered into the model among all the remaining $q-j$ genes at the j^{th} selection step. Therefore, gene $X^{m_{j+1}}$ is the gene that has the highest contribution at reducing the average prediction error rate of the preceding model that uses j set of genes $X^{m_1}X^{m_2}\dots X^{m_j}$. This gene selection procedure shall continue for all the possible combination of genes for which their marginal contributions into the successive model(s) are significant as established by our test criteria. Further gene selection processes only terminate when none of the remaining (left-out) genes is capable at improving the prediction strength of the current model.

We presented in *Table 2.3*, the schematic representation of the MER computations required while searching for the second best gene, X^{m_2} to be included with X^{m_1} in the classification model.

Test sample	*Repetitions* $(\boldsymbol{R})$	*Sequence of genes selection*				
		X^{m_1}	$X^{m_1}\,X_{(2)}$	$X^{m_1}\,X_{(3)}$	…	$X^{m_1}\,X_{(q)}$
		Misclassification Error rates (MERs)				
$\boldsymbol{n_{TE}}$	1	$\hat{\vartheta}_1^{m_1}$	$\hat{\vartheta}_1^{m_{1(2)}}$	$\hat{\vartheta}_1^{m_{1(3)}}$	…	$\hat{\vartheta}_1^{m_{1(q)}}$
$\boldsymbol{n_{TE}}$	2	$\hat{\vartheta}_2^{m_1}$	$\hat{\vartheta}_2^{m_{1(2)}}$	$\hat{\vartheta}_2^{m_{1(3)}}$	…	$\hat{\vartheta}_2^{m_{1(q)}}$
$\boldsymbol{n_{TE}}$	3	$\hat{\vartheta}_3^{m_1}$	$\hat{\vartheta}_3^{m_{1(2)}}$	$\hat{\vartheta}_3^{m_{1(3)}}$	…	$\hat{\vartheta}_3^{m_{1(q)}}$
⋮	⋮	⋮	⋮	⋮	…	⋮
$\boldsymbol{n_{TE}}$	R	$\hat{\vartheta}_R^{m_1}$	$\hat{\vartheta}_R^{m_{1(2)}}$	$\hat{\vartheta}_R^{m_{1(3)}}$	…	$\hat{\vartheta}_R^{m_{1(q)}}$
Average MERs		$\hat{\hat{\vartheta}}^{m_1}$	$\hat{\hat{\vartheta}}^{m_{1(2)}}$	$\hat{\hat{\vartheta}}^{m_{1(3)}}$	…	$\hat{\hat{\vartheta}}^{m_{1(q)}}$

Table 2.3: The schematic representation of the MER computations required while searching for the second best gene to be added to the first selected best gene X^{m_1} into classification the model.

The next step is to determine the significance of the marginal contribution of gene X^{m_2} into the new classification rule $\varphi(X^{m_1}, X^{m_2})$ (*later defined as* $\varphi^{m_1,m_2}(\boldsymbol{x})$) with an average MER of $\hat{\hat{\vartheta}}^{m_1 m_2}$ over the previous rule $\varphi(X^{m_1})$ (*later defined as* $\varphi^{m_1}(\boldsymbol{x})$) with an average MER performance of $\hat{\hat{\vartheta}}^{m_1}$ based on some test criteria (*to be developed*). If this marginal contribution is significant as established by such test criteria, then gene X^{m_2} stays in the model and the search for the next best gene, say gene X^{m_3}, to be added with genes X^{m_1}, X^{m_2} in the model would begin.

The marginal improvement of the current classification rule $\varphi^{m_1,m_2}(\boldsymbol{x})$ over the preceding rule $\varphi^{m_1}(\boldsymbol{x})$ is determined by the difference between $\hat{\hat{\vartheta}}^{m_1,m_2}$ and $\hat{\hat{\vartheta}}^{m_1}$, their respective average MERs. However, two forms of such mean MER differences exist which eventually returned similar results as would be established later. These are denoted by $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$ and $\hat{\delta}_{1^2} = \hat{\hat{\vartheta}}^{m_1,m_2} - \hat{\hat{\vartheta}}^{m_1}$. Appropriate test procedures shall be constructed for the two formulations in what follows.

Let the population mean MERs of the estimated empirical mean MERs $\hat{\hat{\vartheta}}^{m_1}$ and $\hat{\hat{\vartheta}}^{m_1,m_2}$ be represented by $\mu_{\vartheta}^{m_1}$ and $\mu_{\vartheta}^{m_1 m_2}$ respectively.

Then, the following one directional hypothesis of difference are desirable;

$$\left.\begin{array}{lc} & H_{011}: \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} \leq 0 \ \ vs. \ \ H_{a11}: \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} > 0 \\ & \rightarrow \qquad H_{011}: \delta_{1^1} \leq 0 \ vs. \ H_{a11}: \delta_{1^1} > 0 \\ \text{or} & H_{012}: \mu_{\vartheta}^{m_1,m_2} - \mu_{\vartheta}^{m_1} \geq \ 0 \ vs. \ \ H_{a12}: \mu_{\vartheta}^{m_1,m_2} - \mu_{\vartheta}^{m_1} < 0 \\ & \rightarrow \qquad H_{012}: \delta_{1^2} \geq 0 \ vs. \ H_{a12}: \delta_{1^2} < 0 \end{array}\right\} \qquad (2.4.4)$$

where $\delta_{1^1} = \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2}$ and $\delta_{1^2} = \mu_{\vartheta}^{m_1,m_2} - \mu_{\vartheta}^{m_1}$ with their respective unbiased estimators given by $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$ and $\hat{\delta}_{1^2} = \hat{\hat{\vartheta}}^{m_1,m_2} - \hat{\hat{\vartheta}}^{m_1}$. This hypotheses sets shall be used later to illustrate the basic steps involved in the sequential gene selection method we proposed in this thesis. But before we go into that, it is necessary to establish the sampling distribution of $\hat{\hat{\vartheta}}^{m_1}$ (or more generally $\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}$) and that of $\hat{\delta}_{1^1}$ (or $\hat{\delta}_{j^1}$) under some special cross-validation techniques as used in the construction of our test procedure for testing (2.4.4). The sampling distribution of $\hat{\delta}_{1^2}$ (or $\hat{\delta}_{j^2}$) takes the same form as that of $\hat{\delta}_{1^1}$.

Let the class label $Y_{i0} \in \{0,1\}$ of all the subjects in the test sample n_{TE} be as earlier defined in (2.1.2). It then follows that classification rule $\varphi^{m_1}(\boldsymbol{x})$, for instance, can correctly (if $\varphi^{m_1}(\boldsymbol{x}) = Y_{i0}$) or incorrectly (if $\varphi^{m_1}(\boldsymbol{x}) \neq Y_{i0}$) classify any subject in sample n_{TE} as being a class 0 or 1 subject. For those cases for which $\varphi^{m_1}(\boldsymbol{x}) \neq Y_{i0}$, let the rule $\varphi^{m_1}(\boldsymbol{x})$ has a chance ϑ^{m_1}, $0 \leq \vartheta^{m_1} \leq 1$, of misclassifying any subject in the population containing test sample n_{TE} into either being a class 0 or 1 subject. It then follows that the classification function $\varphi^{m_1}(\boldsymbol{x})$ is a random variable having a Bernoulli process ϑ^{m_1}. The probability mass function of $\varphi^{m_1}(\boldsymbol{x})$ for a single subject classification is given by

$$p(\varphi^{m_1}(\boldsymbol{x}) = \varphi^{m_1} | \vartheta^{m_1}) = (\vartheta^{m_1})^{\varphi^{m_1}}(1 - \vartheta^{m_1})^{1-\varphi^{m_1}}, \;\; \varphi^{m_1} = 0{,}1. \quad (2.4.5)$$

The distribution of the sum, $\Phi(\boldsymbol{x}) = \sum_{i=1}^{n_{TE}} \varphi_i^{m_1}(\boldsymbol{x})$ over the entire test sample n_{TE} is given by

$$p(\Phi(\boldsymbol{x}) = \Phi | \vartheta^{m_1}) = \binom{n_{TE}}{\Phi}(\vartheta^{m_1})^{\Phi}(1 - \vartheta^{m_1})^{n_{TE} - \Phi}, \; \Phi = 1, \ldots, n_{TE}. (2.4.6)$$

The unbiased estimator of ϑ^{m_1} is $\hat{\vartheta}^{m1} = \frac{\sum_{i=1}^{n_{TE}} \varphi_i^{m_1}(x)}{n_{TE}}$ which simply equals to the empirical error rate, as given in (2.4.3), of wrongly classifying any subject in the test sample n_{TE} by classification rule $\varphi^{m_1}(\boldsymbol{x})$ but presently using only one gene X^{m_1}. From sampling distribution of $\hat{\vartheta}^{m_1}$ it follows that

$$E\left(\hat{\vartheta}^{m_1}\right) = \frac{\sum_{i=1}^{n_{TE}} E[\hat{\varphi}_i^{m_1}(x)]}{n_{TE}} = \vartheta^{m_1} \qquad (2.4.7)$$

$$\sigma^2\left(\hat{\vartheta}^{m_1}\right) = \frac{\sum_{i=1}^{n_{TE}} \sigma^2[\hat{\varphi}_i^{m_1}(x)]}{n_{TE}^2} = \frac{\vartheta^{m_1}(1 - \vartheta^{m_1})}{n_{TE}} \qquad (2.4.8)$$

and by *central limit theorem* (CLT) we simply have that

$$Z = \frac{\hat{\vartheta}^{m_1} - E\left(\hat{\vartheta}^{m_1}\right)}{\sqrt{\sigma^2\left(\hat{\vartheta}^{m_1}\right)}} \sim N(0{,}1) \qquad (2.4.9)$$

It should be recalled that when any of the cross-validation techniques (MCCV or bootstrap) is used, a set of R estimates of average MERs $\hat{\vartheta}_1^{m_1} = \frac{\sum_{i=1}^{n_{TE}} \hat{\varphi}_{i1}^{m_1}(x)}{n_{TE}}$, $\hat{\vartheta}_2^{m_1} = \frac{\sum_{i=1}^{n_{TE}} \hat{\varphi}_{i2}^{m_1}(x)}{n_{TE}}$, … , $\hat{\vartheta}_R^{m_1} = \frac{\sum_{i=1}^{n_{TE}} \hat{\varphi}_{iR}^{m_1}(x)}{n_{TE}}$ would be computed, one for each of the classification rules $\varphi_1^{m_1}(\boldsymbol{x})$, $\varphi_2^{m_1}(\boldsymbol{x}), \ldots, \varphi_R^{m_1}(\boldsymbol{x})$ that were constructed over all the R repeatedly drawn random samples of size n_{TE} from the original sample size n. By this, the response class of a total of $n_{TE} \times R$ future subjects would be predicted. Hence, the sampling distribution of the mean prediction error rate $\hat{\bar{\vartheta}}^{m_1} = \frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_r^{m_1}$ according to (2.4.7), (2.4.8) and (2.4.9) is as follows;

$$E\left(\hat{\bar{\vartheta}}^{m_1}\right) = \frac{1}{R}\sum_{r=1}^{R} E\left(\hat{\vartheta}_r^{m_1}\right) = \frac{1}{R}\sum_{r=1}^{R}\vartheta_r^{m_1} = \mu_\vartheta^{m_1}$$

$$\rightarrow \qquad E\left(\hat{\bar{\vartheta}}^{m_1}\right) = \mu_\vartheta^{m_1} \qquad (2.4.10)$$

$$\sigma^2\left(\hat{\bar{\vartheta}}^{m_1}\right) = \frac{1}{R^2}\sum_{r=1}^{R}\sigma^2\left(\hat{\vartheta}_r^{m_1}\right) = \frac{1}{R^2 \times n_{TE}}\sum_{r=1}^{R}\vartheta_r^{m_1}\left(1-\vartheta_r^{m_1}\right) \qquad (2.4.11)$$

Also, by CLT we have that

$$\bar{Z}_1 = \frac{\hat{\bar{\vartheta}}^{m_1} - E\left(\hat{\bar{\vartheta}}^{m_1}\right)}{\sqrt{\sigma^2\left(\hat{\bar{\vartheta}}^{m_1}\right)}} \sim N(0,1) \qquad (2.4.12)$$

Similarly, for the mean misclassification error rate $\hat{\bar{\vartheta}}^{m_1,m_2}$ estimated by classification rule $\varphi^{m_1,m_2}(\boldsymbol{x})$ using the gene pair X^{m_1}, X^{m_2}, we shall have that

$$\mathrm{E}\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right) = \mu_\vartheta^{m_1,m_2} \qquad (2.4.13)$$

$$\sigma^2\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right) = \frac{1}{R^2 \times n_{TE}}\sum_{r=1}^{R}\vartheta_r^{m_1,m_2}\left(1-\vartheta_r^{m_1,m_2}\right) \qquad (2.4.14)$$

and also that

$$\bar{Z}_2 = \frac{\hat{\bar{\vartheta}}^{m_1,m_2} - E\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right)}{\sqrt{\sigma^2\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right)}} \sim N(0,1). \qquad (2.4.15)$$

Without loss of generality therefore, the mean prediction error rate $\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j}$, $j = 1, 2, \dots, q$, computed by classification rule $\varphi^{m_1,m_2,\dots,m_j}(\boldsymbol{x})$ using the set of j genes $X^{m_1}, X^{m_2}, \dots, X^{m_j}$ would have the following distributional properties;

$$\mathrm{E}\left(\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j}\right) = \mu_\vartheta^{m_1,m_2,\dots,m_j} \qquad (2.4.16)$$

$$\sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right) = \frac{1}{R^2\times n_{TE}}\sum_{r=1}^{R}\vartheta_r^{m_1,m_2,\dots,m_j}\left(1-\vartheta_r^{m_1,m_2,\dots,m_j}\right) \qquad (2.4.17)$$

and $$\bar{Z}_j = \frac{\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - E\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right)}{\sqrt{\sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right)}} \sim N(0,1). \qquad (2.4.18)$$

All the above sampling distributions of the mean MERs work perfectly under the MCCV sub-sampling scheme. If the cross-validation by bootstrapping is to be used, little modification has to be effected. We shall only present the sampling distribution of the average MER estimates for the bootstrap.632+ scheme (Efron & Tibshirani, 1997) as used in this thesis.

The estimator of the average MER employed by classification rule $\varphi^{m_1,m_2,\dots,m_j}(\boldsymbol{x})$ using a set of genes $X^{m_1}, X^{m_2}, \dots, X^{m_j}$ according to the bootstrap.632+ sub-sampling scheme is given by

$$\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j} = 0.632 * \frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_{r.test}^{m_1,m_2,\dots,m_j} + 0.368 * \frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_{r.train}^{m_1,m_2,\dots,m_j}$$

From the above estimator, the following results are trivial;

$$\mathrm{E}\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right) = 0.632 * \frac{1}{R}\sum_{r=1}^{R}\vartheta_{r.test}^{m_1,m_2,\dots,m_j} + 0.368 * \frac{1}{R}\sum_{r=1}^{R}\vartheta_{r.train}^{m_1,m_2,\dots,m_j}$$

$$\rightarrow \quad \mathrm{E}\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right) = 0.632 * \mu_{\vartheta.test}^{m_1,m_2,\dots,m_j} + 0.368 * \mu_{\vartheta.train}^{m_1,m_2,\dots,m_j} \qquad (2.4.19)$$

Also, $$\sigma^2\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right) = (0.632)^2 * \frac{1}{R^2}\sum_{r=1}^{R}\sigma^2\left(\hat{\vartheta}_{r.test}^{m_1,m_2,\dots,m_j}\right) + (0.368)^2 * \frac{1}{R^2}\sum_{r=1}^{R}\sigma^2\left(\hat{\vartheta}_{r.train}^{m_1,m_2,\dots,m_j}\right)$$

$$\rightarrow \sigma^2\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right) = (0.632)^2 * \frac{1}{R^2\times n_{TE}}\sum_{r=1}^{R}\vartheta_{r.test}^{m_1,m_2,\dots,m_j}\left(1-\vartheta_{r.test}^{m_1,m_2,\dots,m_j}\right) + (0.368)^2 * \frac{1}{R^2\times n_{TR}}\sum_{r=1}^{R}\vartheta_{r.train}^{m_1,m_2,\dots,m_j}\left(1-\vartheta_{r.train}^{m_1,m_2,\dots,m_j}\right) \qquad (2.4.20)$$

and similarly we have that

$$\bar{Z}_{j.bootstrap} = \frac{\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j} - E\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right)}{\sqrt{\sigma^2\left(\hat{\hat{\vartheta}}_{bootstrap}^{m_1,m_2,\dots,m_j}\right)}} \sim N(0,1). \qquad (2.4.21)$$

where $n_{TR} = n$, the bootstrap sample. Further details on bootstrap.632+ MER estimator are provided in Section 2.5.

Now, let us consider the unbiased estimator of $\delta_{1^1} = \mu_\vartheta^{m_1} - \mu_\vartheta^{m_1,m_2}$ given by $\hat{\delta}_{1^1} = \hat{\bar{\vartheta}}^{m_1} - \hat{\bar{\vartheta}}^{m_1,m_2}$ as defined under one directional hypothesis set (2.4.4). It then follows that

$$E(\hat{\delta}_{1^1}) = E\left(\hat{\bar{\vartheta}}^{m_1}\right) - \mathrm{E}\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right) = \mu_\vartheta^{m_1} - \mu_\vartheta^{m_1,m_2} \qquad (2.4.22)$$

and if we consider any possible association between $\hat{\bar{\vartheta}}^{m_1}$ and $\hat{\bar{\vartheta}}^{m_1,m_2}$ since both of them are estimated using the cross-validated random samples $n_{TE} \times R$ that are generated from original sample size n, then, the variance of $\hat{\delta}_{1^1}$ could be estimated by

$$\sigma^2(\hat{\delta}_{1^1}) = \sigma^2\left(\hat{\bar{\vartheta}}^{m_1}\right) + \sigma^2\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right) - 2cov\left(\hat{\bar{\vartheta}}^{m_1,m_2}, \hat{\bar{\vartheta}}^{m_1}\right) \quad (2.4.23)$$

where $cov\left(\hat{\bar{\vartheta}}^{m_1,m_2}, \hat{\bar{\vartheta}}^{m1}\right)$ is the covariance estimate that accounts for any possible association that may exist between the two empirical average MERs. This could be simply estimated by

$$cov\left(\hat{\bar{\vartheta}}^{m_1,m_2}, \hat{\bar{\vartheta}}^{m1}\right) = \hat{\rho}\left(\hat{\bar{\vartheta}}^{m_1,m_2}, \hat{\bar{\vartheta}}^{m_1}\right) * \sqrt{\sigma^2\left(\hat{\bar{\vartheta}}^{m_1,m_2}\right) * \sigma^2\left(\hat{\bar{\vartheta}}^{m_1}\right)}, \quad (2.4.25)$$

where $\hat{\rho}\left(\hat{\bar{\vartheta}}^{m_1,m_2}, \hat{\bar{\vartheta}}^{m_1}\right)$ is the Pearson correlation coefficient estimate between $\hat{\bar{\vartheta}}^{m_1}$ and $\hat{\bar{\vartheta}}^{m_1,m_2}$.

If Gaussian distribution is assumed for random variable $\hat{\delta}_{1^1} = \hat{\bar{\vartheta}}^{m_1} - \hat{\bar{\vartheta}}^{m_1,m_2}$, the difference of two successive MERs estimated at step 1, then, it follows that $\hat{\delta}_{1^1} \sim N\left(E(\hat{\delta}_{1^1}), \sigma^2(\hat{\delta}_{1^1})\right)$ and consequently, we shall have that $Z_{\hat{\delta}_{1^1}} = \frac{\hat{\delta}_{1^1} - E(\hat{\delta}_{1^1})}{\sqrt{\sigma^2(\hat{\delta}_{1^1})}} \sim N(0,1)$.

More generally, for any observed pair of empirical average MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}$ and $\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}$, $j = 1, 2, \ldots, q-1$, for which Gaussian distribution is assumed for the difference of successive pair of average MERs $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}$ or $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}$, that is, $\hat{\delta}_{j^s} \sim N\left(\delta_{j^s}, \sigma^2(\hat{\delta}_{j^s})\right)$, $s = 1,2$, it is obvious that

$$\left.\begin{aligned} \delta_{j^1} = E(\hat{\delta}_{j^1}) = \mu_{\vartheta}^{m_1,m_2,\ldots,m_j} - \mu_{\vartheta}^{m_1,m_2,\ldots,m_{j+1}} \\ \delta_{j^2} = E(\hat{\delta}_{j^2}) = \mu_{\vartheta}^{m_1,m_2,\ldots,m_{j+1}} - \mu_{\vartheta}^{m_1,m_2,\ldots,m_j} \end{aligned}\right\} \qquad (2.4.26)$$

and with $\hat{\delta}_{j^s} = \pm\left(\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}\right)$ for $s = 1$ or 2, we shall have that

$$\sigma^2(\hat{\delta}_{j^s}) = \sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}\right) + \sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}\right)$$

$$\pm 2cov\left(\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}, \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}\right) \qquad (2.4.27)$$

Hence, the assumption that

$$Z_{\hat{\delta}_{j^s}} = \frac{\hat{\delta}_{j^s} - \delta_{j^s}}{\sqrt{\sigma^2(\hat{\delta}_{j^s})}} \sim N(0,1) \qquad (2.4.28)$$

equally holds.

However, when considering the differences between two successive pair of bootstrap MERs, the modifications effected on the bootstrap .632+ MER estimator as provided in equations (2.4.19) to (2.4.21) need to be incorporated.

In what follows, we present the procedure for testing the general form of the hypothesis set in (2.4.4) over successive j average MER differences. Its optimality properties shall also be discussed. The simple case of two gene selection as considered by hypothesis set (2.4.4) shall be illustrated at the end of this section.

Let X^{m_1}, $X^{m_1}X^{m_2}$, $X^{m_1}X^{m_2}X^{m_3}$, … , $X^{m_1}X^{m_2}X^{m_3} \ldots X^{m_q}$ be the sequence of selected gene combinations by respective classification rules $\varphi^{m_1}(\boldsymbol{x})$, $\varphi^{m_1,m_2}(\boldsymbol{x})$, $\varphi^{m_1,m_2,m_3}(\boldsymbol{x})$,…, $\varphi^{m_1,m_2,m_3,\ldots,m_q}(\boldsymbol{x})$ based on their marginal contributions at reducing the average MERs in successive models with the last classifier using all the q genes. The corresponding average cross-validated MERs produced by the above sets of gene combinations are given by $\hat{\hat{\vartheta}}^{m_1}$, $\hat{\hat{\vartheta}}^{m_1,m_2}$, $\hat{\hat{\vartheta}}^{m_1,m_2,m_3}$, … ,$\hat{\hat{\vartheta}}^{m_1,m_2,m_3,\ldots,m_q}$ respectively. However, the prediction accuracy of each successive classification rule is expected to improve as additional genes are selected into the model. Therefore, the following order of the estimated mean MERs is expected for all the selection steps at which additional genes are selected for prediction:

$$\hat{\hat{\vartheta}}^{m_1} > \hat{\hat{\vartheta}}^{m_1,m_2} > \hat{\hat{\vartheta}}^{m_1,m_2,m_3} >, \ldots , > \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_q} \qquad (2.4.29)$$

If the complete ordered form of average MERs in (2.4.29) is observed by our new classifier in any given microarray data set, it simply indicates that the best prediction model with the least (optimum) average MER $\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_q}$ uses all the q genes. However, this is not practically feasible, because the apparent improvement in prediction accuracies due to successive inclusion of additional genes would vanish at a particular selection step. When such a step is reached, the inclusion of additional gene(s) would either brings no further improvement in prediction accuracy into the current model or worsen the prediction performance of the previous model. Our proposed classification rule here therefore seeks to determine the optimal gene selection level at which the best prediction accuracy would be achieved.

If we consider the difference between the j^{th} and $(j+1)^{th}$ average MERs as indexed by $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ or $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$, $j = 1, \dots, q-1$, using the q expected order of performance formulated in (2.4.29), then we shall have two ways by which the $q-1$ mean MER differences can be formulated. We present these two formulations as $\widehat{\boldsymbol{\delta}}_{\mathbf{1}} = \left(\hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}, \dots, \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{q-1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_q}\right)$ and $\widehat{\boldsymbol{\delta}}_{\mathbf{2}} = \left(\hat{\hat{\vartheta}}^{m_1,m_2} - \hat{\hat{\vartheta}}^{m_1}, \dots, \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_q} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{q-1}}\right)$. The estimators of the two vectors $\widehat{\boldsymbol{\delta}}_{\mathbf{1}}$ and $\widehat{\boldsymbol{\delta}}_{\mathbf{2}}$ are identical except for the sign differences. These two formulations are again presented in *Table 2.4*. We shall develop the test procedures that will handle the two formulations for our gene selection problem. The two vectors may therefore be represented in terms of $\hat{\delta}_{j^s}, s = 1,2$, as

$$\widehat{\boldsymbol{\delta}}_{\mathbf{1}} = \left(\hat{\delta}_{1^1}, \hat{\delta}_{2^1}, \dots, \hat{\delta}_{(q-1)^1}\right) \qquad (2.4.30)$$

$$\widehat{\boldsymbol{\delta}}_{\mathbf{2}} = \left(\hat{\delta}_{1^2}, \hat{\delta}_{2^2}, \dots, \hat{\delta}_{(q-1)^2}\right) \qquad (2.4.31)$$

Mean MERs	$j = 1$	...	$j = q-1$
$\hat{\delta}_{j^1} = \hat{\hat{\boldsymbol{\vartheta}}}^{\boldsymbol{m_1,m_2,\dots,m_j}} - \hat{\hat{\boldsymbol{\vartheta}}}^{\boldsymbol{m_1,m_2,\dots,m_{j+1}}}$	$\hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$	...	$\hat{\hat{\vartheta}}^{m_1,m_2,m_3,\dots,m_{q-1}} - \hat{\hat{\vartheta}}^{m_1,m_2,m_3,\dots,m_q}$
$\hat{\delta}_{j^2} = \hat{\hat{\boldsymbol{\vartheta}}}^{\boldsymbol{m_1,m_2,\dots,m_{j+1}}} - \hat{\hat{\boldsymbol{\vartheta}}}^{\boldsymbol{m_1,m_2,\dots,m_j}}$	$\hat{\hat{\vartheta}}^{m_1,m_2} - \hat{\hat{\vartheta}}^{m_1}$	...	$\hat{\hat{\vartheta}}^{m_1,m_2,m_3,\dots,m_q} - \hat{\hat{\vartheta}}^{m_1,m_2,m_3,\dots,m_{q-1}}$

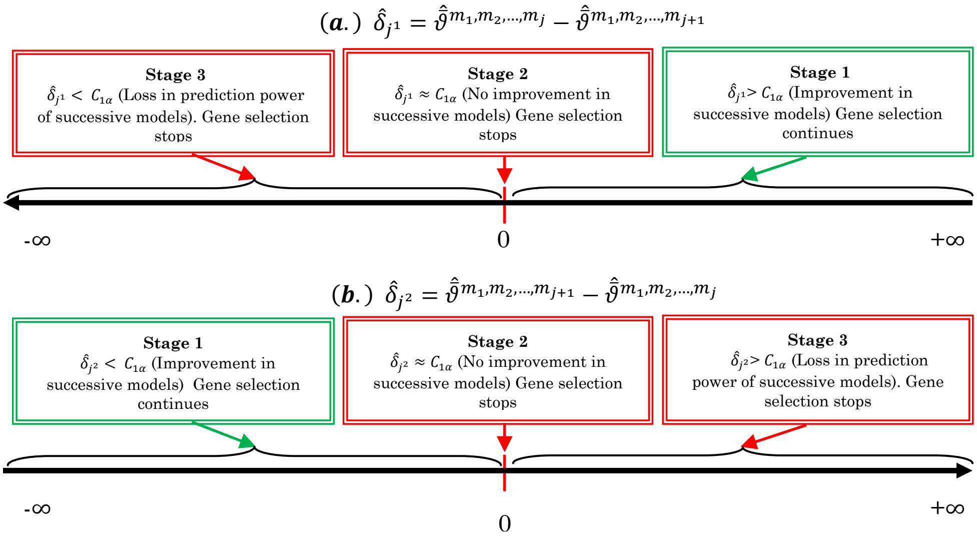
Table 2.4: Table of the two average MER difference formulations $\hat{\delta}_{j^s} = \pm(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}})$ *for s = 1 or 2 respectively at any two successive gene selection steps j and* $j+1$.

It should be noted that, the expected order of mean MERs in (2.4.29) does not necessarily suggest that the respective minimum mean MER pair differences as given in (2.4.30) and (2.4.31) would also followed that unique order. The implementation of the k sequential selection procedure (k-SS) we proposed under the two minimum mean MER difference formulations in (2.4.30) and (2.4.31) are presented in what follows.

- *The k-SS procedures under the* $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ *difference formulations*

For any two successive selection steps j and $j+1$, let $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$, $j = 1,\dots,q-1$ be the vector of minimum mean MER differences as presented in (2.4.30). Better improvements in successive models are expected as additional genes are being selected into the models. Thus, at any two successive selection steps j and $j+1$ at which additional gene is selected, positive values of $\hat{\delta}_{j^1}$'s would be observed in as much as the inequality $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} > C_{1\alpha}$ is maintained for some critical value $C_{1\alpha}$ of the k-SS test procedure to be determined. This is the *stage 1* of our sequential selection procedure. Improvement in prediction performance as observed at *stage 1* shall continue until the second selection stage, *stage 2*, is reached at which the marginal improvements in successive models begin to diminish. At this stage, the estimated average minimum mean MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ would be approaching that of $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ an indication that the current model (with additional one gene) is no more having significant marginal gain in terms of better prediction accuracy over the preceding model since $\hat{\delta}_{j^1} \rightarrow 0$.

At the last selection stage, *stage 3*, considerable losses in prediction accuracy of the succeeding models are expected as more genes are selected. This selection stage is characterized by having the estimated $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}{}'s > \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}{}'s$ which consequently implies that the $\hat{\delta}_{j^1}{}'s < 0$. Nonetheless, the optimal gene selection is expected at any of the last two selection stages (*stage 2* or *stage 3*) at which further selection of additional genes into the model would yield no improvement in model's prediction performance. The moment such stage is reached, further gene selection stops. The schematic illustration of the three basic selection stages as described above with respect to the $\hat{\delta}_{js}$ formulation is presented in *Fig 2.1a* for $s = 1$.


($\boldsymbol{a}$.) $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$

Stage 3
$\hat{\delta}_{j^1} < C_{1\alpha}$ (Loss in prediction power of successive models). Gene selection stops
Stage 2
$\hat{\delta}_{j^1} \approx C_{1\alpha}$ (No improvement in successive models) Gene selection stops
Stage 1
$\hat{\delta}_{j^1} > C_{1\alpha}$ (Improvement in successive models) Gene selection continues
-∞
0
+∞

($\boldsymbol{b}$.) $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$

Stage 1
$\hat{\delta}_{j^2} < C_{1\alpha}$ (Improvement in successive models) Gene selection continues
Stage 2
$\hat{\delta}_{j^2} \approx C_{1\alpha}$ (No improvement in successive models) Gene selection stops
Stage 3
$\hat{\delta}_{j^2} > C_{1\alpha}$ (Loss in prediction power of successive models). Gene selection stops
-∞
0
+∞

Fig 2.1: The schematic representations of the three stages of gene selection processes by the newly proposed k-sequential selection (k-SS) method under the two minimum mean MER differences $\boldsymbol{a}.)\,\widehat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ *and* $\boldsymbol{b}.)\,\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ *at any two successive gene selection steps* j *and* $j+1, j = 1,\dots,q-1$. *The* $c_{1\alpha}$ *represents some critical value of the k-SS test procedure.*

For any two successive j^{th} and $(j+1)^{th}$ gene selection steps, the appropriate general one directional hypothesis test required to justify the selection of additional gene at step j is given by

$$H_{01j}\colon \mu_{\vartheta}^{m_1,m_2,\dots,m_j} - \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} \leq 0 \quad vs.\ H_{a1j}: \mu_{\vartheta}^{m_1,m_2,\dots,m_j} - \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} > 0$$

$$\rightarrow \qquad H_{01j}: \delta_{j^1} \leq 0 \ vs.\ H_{a1j}: \delta_{j^1} > 0,\ j = 1,\dots,q-1 \qquad (2.4.32)$$

where $\delta_{j^1} = \mu_{\vartheta}^{m_1,m_2,\dots,m_j} - \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$. Obviously, the unbiased estimator of δ_{j^1} is given by $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$. However, the test hypothesis set (2.4.32) is the general form of the one directional hypothesis test (2.4.4).

If H_{01j} is accepted in the test hypothesis set (2.4.32) for any successive j^{th} and $(j+1)^{th}$ pair of steps, this is an indication that the selection of additional one gene into the preceding j^{th} model that yielded the mean MER $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ is no longer necessary because

i) no further improvement in prediction accuracy is achieved from the current $(j+1)^{th}$ model despite the selection of additional one gene into the preceding j^{th} model if $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ and that

ii) the misclassification error rate of the current $(j+1)^{th}$ model is further worsened if one more gene is included in the j^{th} model for which $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} < \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$, as represented by the null hypothesis H_{01j} where $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ and $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ are the average MER of the j^{th}(preceding) and $(j+1)^{th}$ (current) models respectively.

Therefore in a loose term, at any two successive gene selection steps j^{th} and $(j+1)^{th}$ the performance difference $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} > C_{1\alpha}$ need to be satisfied to guarantee the inclusion of additional one more gene into the preceding j^{th} model, for some critical value $C_{1\alpha} \in \mathbb{R}$. This literally translates to stopping the selection of additional gene at step j if $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} \leq C_{1\alpha}$.

To construct a formal statistical test for hypothesis set (2.4.32), let

$$Z_{\hat{\boldsymbol{\delta}}_1} = \left(\frac{\hat{\delta}_{1^1} - E(\hat{\delta}_{1^1})}{\sqrt{\sigma^2(\hat{\delta}_{1^1})}}, \frac{\hat{\delta}_{2^1} - E(\hat{\delta}_{2^1})}{\sqrt{\sigma^2(\hat{\delta}_{2^1})}}, \dots, \frac{\hat{\delta}_{(q-1)^1} - E(\hat{\delta}_{(q-1)^1})}{\sqrt{\sigma^2(\hat{\delta}_{(q-1)^1})}} \right) \quad (2.4.33)$$

be the vector of test statistics for testing the set of j one directional hypothesis in (2.4.32), $j = 1, \dots, q-1$. According to (2.4.28), each of the test statistics $Z_{\hat{\delta}_{j1}} = \frac{\hat{\delta}_{j1} - E(\hat{\delta}_{j1})}{\sqrt{\sigma^2(\hat{\delta}_{j1})}} \in Z_{\hat{\boldsymbol{\delta}}_1}$ in (2.4.33) is assumed to have a standard Gaussian distribution. It then follows that vector $Z_{\hat{\boldsymbol{\delta}}_{\mathbf{1}}} = \left(Z_{\hat{\delta}_{1^1}}, Z_{\hat{\delta}_{2^1}}, \dots, Z_{\hat{\delta}_{(q-1)^1}}\right)$ of the test statistics could be assumed to have a multivariate standard Gaussian distribution with $(q-1) \times (q-1)$ unit variance-covariance matrix Σ. It should be noted that, we only assumed Gaussian distribution for $Z_{\hat{\delta}_{j^1}}$ or $\hat{\delta}_{j^1}$, its true theoretical distribution (if different from Gaussian) shall be determined at a later part of this work.

Nonetheless, under the null hypothesis H_{01j}, $E(\hat{\delta}_{j^1}) = \delta_{j^1} = 0$, and by our earlier distributional assumption on $\hat{\delta}_{j^s}$, $s = 1,2$, we have that $\hat{\delta}_{j^1} \widetilde{_{asympt}} N\left(0, \sigma^2(\hat{\delta}_{j^1})\right)$ and that $Z_{\hat{\delta}_{j^1}} = \frac{\hat{\delta}_{j^1}}{\sqrt{\sigma^2(\hat{\delta}_{j^1})}} \widetilde{_{asympt}} N(0,1)$. It then follows that each successive pair of mean MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ and $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ computed at j^{th} and $(j+1)^{th}$ steps could be tested sequentially using the test statistic $\frac{\hat{\delta}_{j^1}}{\sqrt{\sigma^2(\hat{\delta}_{j^1})}}$. Therefore, the decision rules for such sequential test could be stated as follows;

i) Stop the selection of additional one gene into the j^{th} model (accept H_{01j} at the j^{th} step) if

$$Z_{\hat{\delta}_{j^1}} = \frac{\hat{\delta}_{j^1}}{\sqrt{\sigma^2(\hat{\delta}_{j^1})}} \leq C_\alpha^1 \qquad (2.4.34)$$

ii) Select additional one gene into the j^{th} model (accept H_{a1j} at the j^{th} step) if

$$Z_{\hat{\delta}_{j^1}} = \frac{\hat{\delta}_{j^1}}{\sqrt{\sigma^2(\hat{\delta}_{j^1})}} > C_\alpha^1 \qquad (2.4.35)$$

where C_α^1 is the critical value of the percentage point of the hypothesized distribution (e.g. Gaussian, etc.) of $Z_{\hat{\delta}_{j^1}}$at a significance level α to be determined.

Equivalently, the above decision rules (2.4.34) and (2.4.35) can be re-stated respectively as follows;

iii) Stop the selection of additional one gene into the j^{th} model (accept H_{01j} at the j^{th} step) if

$$\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} \leq C_\alpha^1 \sqrt{\sigma^2\left(\hat{\delta}_{j^1}\right)} \qquad (2.4.36)$$

iv) Select additional one gene into the j^{th} model (accept H_{a1j} at the j^{th} step) if

$$\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} > C_\alpha^1 \sqrt{\sigma^2\left(\hat{\delta}_{j^1}\right)} \qquad (2.4.37)$$

Using the decision rules (2.4.36) and (2.4.37), the new critical value $C_\alpha^1 \sqrt{\sigma^2\left(\hat{\delta}_{j^1}\right)}$ of $\hat{\delta}_{j^1}$ directly substitutes for $C_{1\alpha}$ as used earlier. Therefore, at the j^{th} selection step, the decision is to stop the selection of additional one gene into the j^{th} model if the inequality in (2.4.36) is satisfied while the selection of additional one gene is accepted if the inequality (2.4.37) is satisfied.

An important aspect of this new test procedure is that for any hypothesized distribution of our test statistics $Z_{\hat{\delta}_{j^1}}$ or $\hat{\delta}_{j^1}$, the value of the significance level α used by the test which consequently determines the size of the critical values C_α^1 or $C_\alpha^1 \sqrt{\sigma^2\left(\hat{\delta}_{j^1}\right)}$is not pre-determined by us as often the case with some variable selection methods (see our comments on SLR method and some other variable selection techniques as earlier discussed under this chapter). In other words, the size α of our sequential test procedure at which optimal sub-set of genes are selected is determined through internal cross-validation and not arbitrarily fixed, for instance, to 0.05 or something else by the investigator. Our procedure seeks to perform gene selections and response class predictions over all possible range of values of significance level α within the interval [0,1]. That value (range of values) of α between 0 and 1 at which the decision rule (2.4.36) is satisfied and for which the optimal (best) prediction accuracy is achieved becomes the size of α of our test. Consequently, the selected $k = j$ gene(s), $j = 1, \dots, q-1$ at which further gene selection terminates becomes the needed optimal informative k genes suitable for classifying the mRNA subjects into their appropriate the tumour sub-groups. More details on this shall be provided in Chapter 3.

It should be recalled that each of the estimated average MERs $\hat{\bar{\vartheta}}^{m_1}$, $\hat{\bar{\vartheta}}^{m_1,m_2}$, $\hat{\bar{\vartheta}}^{m_1,m_2,m_3}$, … ,$\hat{\bar{\vartheta}}^{m_1,m_2,m_3,\dots,m_q}$ is a minimum statistic estimate computed at each gene selection steps. This literally implies that at any given successive j^{th} and $(j+1)^{\text{th}}$ pair of gene selection steps, $j = 1, \dots, q-1$, the statistic $\hat{\delta}_{j^1} = \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ or $\hat{\delta}_{j^2} = \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j}$ is a difference between two observed minimum mean MERs $\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j}$ and $\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ obtained at j^{th} and $(j+1)^{th}$ steps respectively.

Although, Gaussian distribution has been earlier assumed for the estimators $\hat{\delta}_{j^s} = \pm(\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}})$, $s = 1$ or 2, their true distribution might be different from Gaussian due to the fact that their realizations are the differences of two minimum statistics. Therefore, in testing the null hypothesis $H_{01j}\colon \mu_{\vartheta}^{m_1,m_2,\dots,m_j} - \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} \leq 0$ in (2.4.32), we suspected that the test statistic $Z_{\hat{\delta}_{j^1}} = \frac{\hat{\delta}_{j^1}}{\sqrt{\sigma^2(\hat{\delta}_{j^1})}}$ constructed for the test might not follow a standard Gaussian distribution as would have been expected under the null. If our suspicion is correct, then, the use of the critical values of the percentage point of the standard normal distribution $\mathrm{Z}_{1-\alpha}$ for C_{α}^{1} in (2.4.34) to (2.4.37) for the test might not be appropriate. Base on this suspicion, it is necessary to determine the true distribution of the difference $\hat{\delta}_{j^1}$ or $Z_{\hat{\delta}_{j^1}}$ whose quantile values could be suitably determine as the correct value of $C_{\alpha}^{1}\sqrt{\sigma^2\left(\hat{\delta}_{j^1}\right)}$ or C_{α}^{1} as appropriate.

We shall use the gene selection results at steps 0 and 1 to illustrate the procedures that leads to the determination the distribution of $\hat{\delta}_{j^1}$. At step 0, gene $X^{m_1} \in \left\{X_{(1)}, X_{(2)}, \dots, X_{(q)}\right\}$ that yielded the minimum mean MER $\hat{\hat{\vartheta}}^{m_1} = min\left(\hat{\hat{\vartheta}}_{(1)}, \hat{\hat{\vartheta}}_{(2)}, \dots, \hat{\hat{\vartheta}}_{(q)}\right)$ among the set of ordered mean MERs $\hat{\hat{\vartheta}}_{(1)}, \hat{\hat{\vartheta}}_{(2)}, \dots, \hat{\hat{\vartheta}}_{(q)}$ is selected with $\hat{\hat{\vartheta}}^{m_1} = \hat{\hat{\vartheta}}_{(1)}$. Therefore, for $j = 1, \dots, q$, let $\hat{\hat{\vartheta}}_{(j)}$ has unknown density function $f_{\hat{\hat{\vartheta}}_{(j)}}(\xi_0)$. Then, from the distribution of ordered statistics, it is very easy to establish the density function of $\hat{\hat{\vartheta}}^{m_1}$ as

$$f_{\hat{\hat{\vartheta}}^{m_1}}(s_0) = q[1 - F_{\hat{\hat{\vartheta}}^{m_j}}(\xi_0)]^{q-1} f_{\hat{\hat{\vartheta}}_{(j)}}(\xi_0) \qquad (2.4.38)$$

where $F_{\hat{\hat{\vartheta}}_{(j)}}(\xi_0) = \int_{-\infty}^{\xi_0} f_{\hat{\hat{\vartheta}}_{(j)}}(u)du$.

Similarly, at step 1, our sequential procedure selected the gene pair $X^{m_1}X^{m_2} \in \left\{X^{m_1}\,X_{(2)}, X^{m_1}\,X_{(3)}, \dots, X^{m_1}\,X_{(q)}\right\}$ that yielded the minimum mean MERs $\hat{\hat{\vartheta}}^{m_1,m_2} = min\left(\hat{\hat{\vartheta}}^{m_{1(2)}}, \hat{\hat{\vartheta}}^{m_{1(3)}}, \dots, \hat{\hat{\vartheta}}^{m_{1(q)}}\right)$ among the set of $q-1$ mean MERs $\hat{\hat{\vartheta}}^{m_{1(2)}}, \hat{\hat{\vartheta}}^{m_{1(3)}}, \dots, \hat{\hat{\vartheta}}^{m_{1(q)}}$. Let the ordered statistics of the $q-1$ mean MER sequence $\hat{\hat{\vartheta}}^{m_{1(2)}}, \hat{\hat{\vartheta}}^{m_{1(3)}}, \dots, \hat{\hat{\vartheta}}^{m_{1(q)}}$ be given by $\hat{\hat{\vartheta}}_{(12)}, \hat{\hat{\vartheta}}_{(13)}, \dots, \hat{\hat{\vartheta}}_{(1q)}$ respectively with $\hat{\hat{\vartheta}}^{m_1,m_2} = \hat{\hat{\vartheta}}_{(12)}$. Also let the unknown density function of each $\hat{\hat{\vartheta}}_{(1j)}$, $j = 2, \dots, q$, be given by

$f_{\hat{\hat{\vartheta}}_{(1j)}}(\xi_1)$. Then, it can be easily verified that the density function of $\hat{\hat{\vartheta}}^{m_1,m_2}$ is of the form

$$f_{\hat{\hat{\vartheta}}^{m_1,m_2}}(s_1) = (q-1)[1 - F_{\hat{\hat{\vartheta}}_{(1j)}}(\xi_1)]^{q-2} f_{\hat{\hat{\vartheta}}_{(1j)}}(\xi_1) \qquad (2.4.39)$$

Given that the difference of the two minimum average MERs $\hat{\hat{\vartheta}}^{m_1}$ and $\hat{\hat{\vartheta}}^{m_1,m_2}$ is $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$, then, the distribution of $\hat{\delta}_{1^1}$ is desired from which the critical value C_α^1 of our one directional hypothesis tests (2.4.4) and by extension, that of the general test in (2.4.32) can be determined.

If we represent the joint density of $\hat{\hat{\vartheta}}^{m_1}$ and $\hat{\hat{\vartheta}}^{m_1,m_2}$ by $f_{\hat{\hat{\vartheta}}^{m_1},\hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2})$, then, the distribution of $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$ can be determined as follows;

Let the distribution function of $\hat{\delta}_{1^1}$ be given by

$$F_{\hat{\delta}_{1^1}}(\hat{\delta}) = P(\hat{\delta}_{1^1} \le \hat{\delta}) = P\left(\hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} \le \hat{\delta}\right)$$

$$F_{\hat{\delta}_{1^1}}(\hat{\delta}) = \iint_{\hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} \le \hat{\delta}} f_{\hat{\hat{\vartheta}}^{m_1},\hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2}) d(\hat{\hat{\vartheta}}^{m_1}) d\,(\hat{\hat{\vartheta}}^{m_1,m_2})$$

$$\equiv \qquad F_{\hat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\hat{\hat{\vartheta}}^{m_1} - \hat{\delta}} f_{\hat{\hat{\vartheta}}^{m_1},\hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2})\, d(\hat{\hat{\vartheta}}^{m_1,m_2})\right] d(\hat{\hat{\vartheta}}^{m_1}) \qquad (2.4.40)$$

If we substitute $\hat{\hat{\vartheta}}^{m_1} - v$ for $\hat{\hat{\vartheta}}^{m_1,m_2}$ in (2.4.40) for any arbitrary variable v, then we shall have that,

$$F_{\hat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\hat{\delta}} f_{\hat{\hat{\vartheta}}^{m_1},\hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1} - v)\, d(v)\right] d(\hat{\hat{\vartheta}}^{m_1}) \qquad (2.4.41)$$

Similarly, if $\hat{\hat{\vartheta}}^{m_1,m_2} + v$ is substituted for $\hat{\hat{\vartheta}}^{m_1}$ in (2.4.40), we have

$$F_{\hat{\delta}_{1^1}}(\hat{\delta}) \equiv \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\hat{\delta}} f_{\hat{\hat{\vartheta}}^{m_1},\hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1,m_2} + v, \hat{\hat{\vartheta}}^{m_1,m_2})\, d(v)\right] d(\hat{\hat{\vartheta}}^{m_1,m_2}) \quad (2.4.42)$$

The representations (2.4.41) and (2.4.42) are the expressions for the distribution function of $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$. To obtain the density function of $\hat{\delta}_{1^1}$, $f_{\hat{\delta}_{1^1}}(\hat{\delta})$, we simply take the derivative of the distribution function in (2.4.41) and (2.4.42) respectively. Therefore, from (2.4.41) we shall have that;

$$f_{\hat{\delta}_{1^1}}(\hat{\delta}) = \frac{dF_{\hat{\delta}_{1^1}}(\widehat{\delta})}{d\widehat{\delta}} = \frac{d}{d\widehat{\delta}}\left\{\int_{-\infty}^{\hat{\delta}}\left[\int_{-\infty}^{\infty} f_{\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1} - v)\, d(\hat{\hat{\vartheta}}^{m_1})\right] d(v)\right\}$$

while from (2.4.42) we shall have that;

$$f_{\hat{\delta}_{1^1}}(\hat{\delta}) = \frac{dF_{\hat{\delta}_{1^1}}(\widehat{\delta})}{d\widehat{\delta}} = \frac{d}{d\widehat{\delta}}\left\{\int_{-\infty}^{\hat{\delta}}\left[\int_{-\infty}^{\infty} f_{\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1,m_2} + v, \hat{\hat{\vartheta}}^{m_1,m_2})\, d(\hat{\hat{\vartheta}}^{m_1,m_2})\right] d(v)\right\}$$

These consequently yield the two forms of $f_{\hat{\delta}_{1^1}}(\hat{\delta})$ given by

$$f_{\hat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} f_{\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1} - v) d(\hat{\hat{\vartheta}}^{m_1}) \qquad (2.4.43)$$

$$f_{\hat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} f_{\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2}}(\hat{\hat{\vartheta}}^{m_1,m_2} + v, \hat{\hat{\vartheta}}^{m_1,m_2})\, d(\hat{\hat{\vartheta}}^{m_1,m_2}) \qquad (2.4.44)$$

respectively.

If Gaussian densities with means $\mu_{\vartheta}^{m_1}$ & $\mu_{\vartheta}^{m_1,m_2}$ and variances $\sigma_1 = \sigma^2\left(\hat{\hat{\vartheta}}^{m_1}\right)$ & $\sigma_2 = \sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2}\right)$ are as initially assumed for the distribution of both $\hat{\hat{\vartheta}}^{m_1}$ & $\hat{\hat{\vartheta}}^{m_1,m_2}$ respectively hold, then, the density function $f_{\hat{\delta}_{1^1}}(\hat{\delta})$ in (2.4.43) can be expressed in terms of the joint density function of both $\hat{\hat{\vartheta}}^{m_1}$ and $\hat{\hat{\vartheta}}^{m_1,m_2}$ as

$$f_{\widehat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} exp\left[-\frac{z_1}{2(1-\rho^2)}\right] d\left(\hat{\hat{\vartheta}}^{m_1}\right) \qquad (2.4.45)$$

where $z_1 = \frac{\left(\hat{\hat{\vartheta}}^{m_1} - \mu_{\vartheta}^{m_1}\right)^2}{\sigma_1} + \frac{\left(\hat{\hat{\vartheta}}^{m_1} - v - \mu_{\vartheta}^{m_1,m_2}\right)^2}{\sigma_2} - \frac{2\rho\left(\hat{\hat{\vartheta}}^{m_1} - \mu_{\vartheta}^{m_1}\right)\left(\hat{\hat{\vartheta}}^{m_1} - v - \mu_{\vartheta}^{m_1,m_2}\right)}{\sigma_1\sigma_2}$

and $\rho = corr(\hat{\hat{\vartheta}}^{m_1}, \hat{\hat{\vartheta}}^{m_1,m_2})$. Also, from (2.4.44), the equivalent form of $f_{\widehat{\delta}_{1^1}}(\hat{\delta})$ as in (2.4.45) can be established in terms of $\hat{\hat{\vartheta}}^{m_1,m_2}$ as

$$f_{\widehat{\delta}_{1^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} exp\left[-\frac{z_2}{2(1-\rho^2)}\right] d\left(\hat{\hat{\vartheta}}^{m_1,m_2}\right) \qquad (2.4.46)$$

where

$$z_2 = \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2}+v-\mu_{\vartheta}^{m_1}\right)^2}{\sigma_1} + \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2}-\mu_{\vartheta}^{m_1,m_2}\right)^2}{\sigma_2} - \frac{2\rho\left(\hat{\hat{\vartheta}}^{m_1,m_2}+v-\mu_{\vartheta}^{m_1}\right)\left(\hat{\hat{\vartheta}}^{m_1,m_2}-\mu_{\vartheta}^{m_1,m_2}\right)}{\sigma_1\sigma_2}$$

More generally, for any pair of minimum average MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ and $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ having Gaussian densities with respective means $\mu_{\vartheta}^{m_1,m_2,\dots,m_j}$ and $\mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$ and variances $\sigma_1 = \sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right)$ and $\sigma_2 = \sigma^2\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}\right)$, the density function of the difference $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ could be obtained from the marginal density functions of both $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ and $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ as

$$f_{\hat{\delta}_{j^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} exp\left[-\frac{z_{11}}{2(1-\rho^2)}\right] d\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right) \quad (2.4.47)$$

or equivalently as

$$f_{\hat{\delta}_{j^1}}(\hat{\delta}) = \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} exp\left[-\frac{z_{12}}{2(1-\rho^2)}\right] d\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}\right) \quad (2.4.48)$$

where $\rho = corr(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}, \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}})$, z_{11} and z_{12} are respectively given as

$$z_{11} = \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}-\mu_{\vartheta}^{m_1,m_2,\dots,m_j}\right)^2}{\sigma_1} + \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}-v-\mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}\right)^2}{\sigma_2}$$

$$- \frac{2\rho\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}-\mu_{\vartheta}^{m_1,m_2,\dots,m_j}\right)\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}-v-\mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}\right)}{\sigma_1\sigma_2}$$

and $$z_{12} = \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}+v-\mu_{\vartheta}^{m_1,m_2,\dots,m_j}\right)^2}{\sigma_1} + \frac{\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}-\mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}\right)^2}{\sigma_2}$$

$$- \frac{2\rho\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}+v-\mu_{\vartheta}^{m_1,m_2,\dots,m_j}\right)\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}-\mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}\right)}{\sigma_1\sigma_2}$$

However, since we have suspected earlier that the Gaussian density might not be appropriate as the distribution of $\hat{\delta}_{j^1}$, the true density function, $f_{\hat{\delta}_{j^1}}(\hat{\delta})$ in (2.4.47) or (2.4.48) of $\hat{\delta}_{j^1}$ estimator would be determined through simulation studies in Chapter 3 in line with the set-up of our proposed sequential test procedures. The quantile values of the true theoretical density of $\hat{\delta}_{j^1}$ (or $\hat{\delta}_{j^2}$) to be determined shall then be true critical value C_{α}^{1} (or C_{α}^{2}) of our test.

- *The k-SS procedures under the* $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ *difference formulations*

In a similar manner, if the differences of the successive pairs of minimum mean MERs $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ as presented in equation (2.4.31) and *Table 2.4* are used to construct our k-SS method, the same results and conclusion as obtained under the $\hat{\delta}_{j^1}$ formulations would be obtained. Under the $\hat{\delta}_{j^2}$ formulation however, negative values of $\hat{\delta}_{j^2}$'s would be observed at all selection steps for which $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} < \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$, denoting the *stage 1* of the sequential selection stages. At this stage, the prediction power of the succeeding models would continue to improve. At *stage 2* however, the situation for which $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} \to \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ would exist, implying that the difference $\hat{\delta}_{j^2} \to 0$. Thus, no significant improvements in successive models in terms of prediction accuracies would be expected at this selection stage. Finally, at *stage 3*, it is expected that $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} > \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ so that $\hat{\delta}_{j^2} > 0$. Considerable losses in prediction accuracies of successive models would be recorded at this stage. The schematic presentation of these three selection stages under the $\hat{\delta}_{j^2}$ formulation of our k-SS method is presented in *Fig 2.1b*.

Now, if we consider the difference formulation $\hat{\delta}_{j^2}$, the appropriate one directional hypothesis of interest would be of the form

$$H_{02j}\colon \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} - \mu_{\vartheta}^{m_1,m_2,\dots,m_j} \geq 0 \; vs.\; H_{a2j}\colon \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} - \mu_{\vartheta}^{m_1,m_2,\dots,m_j} < 0$$

$$\rightarrow \qquad H_{02j}\colon \delta_{j^2} \geq 0 \;\; vs.\;\; H_{a2j}\colon \delta_{j^2} < 0, j = 1, \dots, q-1 \qquad (2.4.49)$$

where $\delta_{j^2} = \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}} - \mu_{\vartheta}^{m_1,m_2,\dots,m_j}$ with its unbiased estimator given by $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$. As defined in (2.4.43), vector $Z_{\hat{\boldsymbol{\delta}}_{\mathbf{2}}} = \left(Z_{\hat{\delta}_{j^2}}\right)$, for $j = 1, \dots, q-1$, is the vector of the test statistics for testing the $q-1$ hypothesis set (2.4.49), where $Z_{\hat{\delta}_{j^2}} = \frac{\hat{\delta}_{j^2} - E\left(\hat{\delta}_{j^2}\right)}{\sqrt{\sigma^2\left(\hat{\delta}_{j^2}\right)}}$, $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$, and $E\left(\hat{\delta}_{j^2}\right) = 0$ under H_{02j}. The decision rules with respect to $\hat{\delta}_{j^2}$ formulation are as follows;

i) Stop the selection of additional one gene into the j^{th} model (accept H_{02j}) at the j^{th} step if

$$\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} \geq C_{\alpha}^{2}\sqrt{\sigma^2\left(\hat{\delta}_{j^2}\right)} \qquad (2.4.50)$$

ii) Select additional one gene into the j^{th} model (accept H_{a2j}) at the j^{th} step if

$$\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} < C_{\alpha}^{2}\sqrt{\sigma^2\left(\hat{\delta}_{j^2}\right)} \qquad (2.4.51)$$

where the critical values C_{α}^{2} for the test shall equally be determined through cross-validation using the theoretical distribution of $\hat{\delta}_{j^2}$ or $Z_{\hat{\delta}_{j^2}}$. The true distribution of $\hat{\delta}_{j^2}$ is similar to that of $\hat{\delta}_{j^1}$. Necessary details on this are provided in Chapter 3.

It should however be noted that the use of either of the hypothesis test (2.4.32) or (2.4.49) would yield the same selection and classification results. All these are demonstrated in the next chapter.

In summary, when H_{01j} or H_{02j} is accepted using decision rules (2.4.34)-(2.4.37) or (2.4.50)-(2.4.51) depending on whether hypothesis set (2.4.32) or (2.4.49) is used respectively at any particular j step, further gene selection into the j^{th} model stops and the $k = j$ genes selected at that point becomes the optimal informative genes. If on the other hand, H_{a1j} or H_{a2j} is accepted, additional one gene is added at step j after which the search for the next best gene begins. A single algorithm that captures the whole k-SS procedures is presented in Section 3.2. Nonetheless, we present clearly in what follows, the basic steps required in the implementation of our k-SS method. We shall provide illustrations using the hypothesis set (2.4.4) designed for only two gene selection steps.

Here, we shall revert to the use of our initial notations in which gene X^{m_1} is the first gene to be selected at Step 0 being the gene that yielded the minimum mean MER $\hat{\hat{\vartheta}}^{m_1}$ among the ordered sequence of the original q genes, $X_{(1)}$, $X_{(2)}, \dots, X_{(q)}$ and the gene pair $X^{m_1}X^{m_2}$ is the set of genes that yielded the minimum mean MER $\hat{\hat{\vartheta}}^{m_1 m_2}$ among the $q-1$ sequence of gene pairs $X^{m_1}\, X_{(2)}, X^{m_1}\, X_{(3)}, \dots, X^{m_1}\, X_{(q)}$. Here, we shall test whether the inclusion of additional gene X^{m_2} into the preceding classification model that contains only gene X^{m_1} improves or worsen the prediction strength of the current model through the average minimum MERs difference $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1 m_2}$ or $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1 m_2} - \hat{\hat{\vartheta}}^{m_1}$. If this difference is not significant based on appropriate decision rule (2.4.37) or (2.4.51) depending on whether test statistic $\hat{\delta}_{j^1}$ or $\hat{\delta}_{j^2}$is used, it simply shows that the marginal contribution of gene X^{m_2} at improving the current model is not significant. Then, further gene selection stops and the model containing only gene X^{m_1} becomes the best optimal model. On the other hand, if its contribution is significant according to decision rule (2.4.36) or (2.4.50), the new selected gene X^{m_2} would be retained with X^{m_1} in the model while the search for the next best gene to be added with X^{m_1} X^{m_2} would begin.

This sequential selection steps continues until none of the remaining genes could satisfy the decision criteria (2.4.37) or (2.4.51) that allows the selection of additional genes into the model. The k genes selected, $k < q$, at which no additional genes can be selected into the model becomes the required optimal k informative genes and the response class predictions provided by such set of genes becomes the optimal prediction.

Backward checks

It is suspected that at each gene selection step where new gene is selected into the model, it might be possible for some of the previously selected genes not to be useful again for prediction given that a new gene is now in the model. Based on this suspicion, we perform backward checks on each of the previously selected genes whenever a new gene is selected. The procedure is straight forward, if a new gene is selected into the model and an average MER, say $\hat{\hat{\vartheta}}^{full}$ is computed for the full model, then each of the previously selected gene is removed from the model and a new model is fitted using all other genes except the removed gene. An average MER, say $\hat{\hat{\vartheta}}^{remove}$ is computed for each model without the removed gene. If $\hat{\hat{\vartheta}}^{remove} > \hat{\hat{\vartheta}}^{full}$, it simply suggests that the removed gene is important in the model and should be retained. But if $\hat{\hat{\vartheta}}^{remove} \leq \hat{\hat{\vartheta}}^{full}$, then the removed gene is not useful again in the model and it is permanently removed from the model. Generally, the number of backward checks, denoted by n_{BC}, to be performed at each gene selection step j for which $k = j + 1$ genes have been selected is $n_{BC} = k - 1$

Our newly proposed k-sequential gene selection (k-SS) method is implemented using R statistical package (http://www.r-project.org/) and the R code we developed for its implementation is presented in Appendix B.1. The R code that performs the backward checks is also provided in Appendix B.2.

The dimension reduction, informative gene selection and response class prediction procedures as executed by our new k-SS method for binary response class can be generalized to a polytomous class prediction with true class categories $y = 0, 1, \ldots, \mathbb{y}$ $(y > 1)$ using any of the following approaches:

- *Pair-wise coupling*: This approach is adapted from Hastie & Tibshirani (1998) and it begins by constructing a separate binary k-SS classifier for each of the distinct pair of classes $y', y'' \in \mathbb{y}$, $y' \neq y''$. For any microarray data set that contains a fixed response class $\mathbb{y} > 1$, a total of $\mathbb{y}(\mathbb{y} - 1)/2$ distinct binary k-SS classifiers would be constructed with each of them predicting a class member in $\mathbb{y}$. At the end, the results of all the classifiers are combined and final decision is made by majority voting. The class category with the highest votes would be chosen as the predicted class for each subject. This approach is also called *One-vs-One-scheme* (Tan *et al*, 2005) or Round Robin Ensemble (Furnkranz, 2002).
- *One-vs-Others scheme*: For a polytomous response class $y = \{0, 1, \ldots, \mathbb{y}\}$ in which the class members follow some natural ordering, the k-SS classifier can be constructed to distinguish a reference class $y^* \in \mathbb{y}$ from all other class labels. By this, all

other complementary classes are put into one group. The log of the ratio of the posterior probabilities used in the logit model would be of the form $ln\left[\frac{p(y^*|\boldsymbol{X})}{\sum_y^{\mathbb{y}-1} p(y|\boldsymbol{X})}\right]$. Other variants of this approach can be found in Hand (1997), Speed (2003), Dudoit *et al*(2002) and some other related works.

2.5 Assessment of the *k*-SS classifier

As remarked earlier, the goodness of classification rule $\varphi(\boldsymbol{x})$ is generally assessed through a discrepancy function $L\{Y_0, \varphi(\boldsymbol{x})\}$ called the *loss function*, where $Y_0 = y_i, i = 1, \dots, n_{TE}$, is the true class labels (0,1) of any independent n_{TE} subjects that are predicted by $\varphi(\boldsymbol{x})$. From now on, φ and φ_j shall be used to represent $\varphi(\boldsymbol{x})$ and $\varphi_j(X_j)$ respectively, dropping both $\boldsymbol{x}$ and X_j for simplicity. For instance, the loss function $L\{Y_0, \varphi(\boldsymbol{x})\}$ shall become $L(Y_0, \varphi)$.

As demonstrated in the previous section, the main concern while assessing any classification function is to find that rule φ that minimizes the loss function $L(Y_0, \varphi)$. The concept of *0-1 loss function* as commonly used is to describe a situation where φ correctly or incorrectly predicts each of the n_{TE} subjects. In this case, the respective loss is 0 or 1 for any subject that is correctly or incorrectly predicted by rule φ. That is, the loss is $L(\hat{Y}_0 = 1, \hat{\varphi} = 1) = L(\hat{Y}_0 = 0, \hat{\varphi} = 0) = 0$ for correct prediction and is $L(\hat{Y}_0 = 1, \hat{\varphi} = 0) = L(\hat{Y}_0 = 0, \hat{\varphi} = 1) = 1$ for incorrect prediction.

However, the loss function may be given in terms of absolute or square error loss functions. An *absolute error loss function* is defined by

$$L(Y_0,\ \varphi) = |\,Y_0 - \varphi| \qquad (2.5.1)$$

while the *square error loss function* is given by

$$L(Y_0,\ \varphi) = (Y_0 - \varphi)^2 \qquad (2.5.2)$$

The expected loss of using rule φ to classify all the n_{TE} subjects is then given by the *risk function*

$$r(\varphi) = \ E_{\boldsymbol{x}}[L(Y_0,\ \varphi)] \qquad (2.5.3)$$

$$\rightarrow \qquad r(\varphi) = \ \sum_{i=1}^{n_0} L(y_i,\ \hat{\varphi}_i) p(\boldsymbol{x}|y_i) p\,, y = 0, 1. \qquad (2.5.4)$$

But since the true density function $p(\boldsymbol{x}|y_i)$ in (2.5.4) is not known, the risk $r(\varphi)$ is usually estimated from the sample by

$$\hat{\vartheta} = \hat{r}(\varphi) = \frac{1}{n_{TE}} \sum_{i=1}^{n_0} |y_i - \ \hat{\varphi}_i| \qquad (2.5.5)$$

if absolute error loss function is used, or by

$$\hat{\vartheta} = \hat{r}(\varphi) = \frac{1}{n_{TE}} \sum_{i=1}^{n_0} (y_i - \ \hat{\varphi}_i)^2 \qquad (2.5.6)$$

if the square error loss function is used.

The risk estimator given by (2.5.5) is the equivalent form of the *empirical misclassification error rate* (MER) given by (2.4.3).

Among other estimators of prediction error rate suggested in the literature are the *brier or quadratic score* and *logarithmic score*.

The brier score, proposed by Brier (1950), is the average deviation between the predicted probabilities $\hat{p}(1|\boldsymbol{x})$ that a set of subjects belong to particular response class and the true subjects classes. The brier score simply replaces the predicted class labels $\hat{\varphi}_i$ with the predicted class probabilities $\hat{p}_i(1|\boldsymbol{x})$ in the square error loss function definition of the MER in (2.5.6) that the subjects belong to the predicted classes. This is given by

$$\hat{\vartheta}_{Brier} = \frac{1}{n_{TE}} \sum_{i=1}^{n_{TE}} (y_i - \hat{p}_i(1|\boldsymbol{x}))^2 \tag{2.5.7}$$

where $0 \leq \hat{\vartheta}_{Brier} \leq 1$.

The logarithmic or informational score has been equally reported as a reliable measure of performance of classifiers (Hand, 1997; Witten & Frank, 2000). Like brier score, it also uses the predicted probabilities $\hat{p}_i(1|\boldsymbol{x})$ in its assessment. Its definition for a two-class prediction is given by

$$\hat{\vartheta}_{log} = -\frac{1}{n_{TE}} \sum_{i=1}^{n_{TE}} \{y_i log[\hat{p}_i(1|\boldsymbol{x})] + (1 - y_i) log[1 - \hat{p}_i(1|\boldsymbol{x})]\} \tag{2.5.8}$$

where $0 \leq \hat{\vartheta}_{log} < \infty$. Like both the MER and the brier scores, a small value of the log score equally shows a better performance of the classifier. What distinguishes the log score index from the other two scores is that it produces a set of general and uncalibrated scores that are not bounded between 0 and 1.

Nonetheless, both the MER and brier scores are part of assessment measures adopted to evaluate the performance of our new k-SS classifier.

Apart from MER, brier or logarithm scores, there are some other performance measures under the pseudo name of similarity indices as well as the receiver operating characteristics (ROC) analysis that are equally appropriate to assess the goodness of a classification rule.

The similarity indices

The most prominent similarity indices among others are the *Jaccard index* (Jaccard, 1901), *Dice-Sørensen index* (Dice, 1945), *Ochiai index* (Ochiai, 1957) and the *Simple Matching index* (Sokal & Michener, 1958). Some other variants to these four can be found in Simpson (1960), Hazel (1970), Sokal & Sneath (1973) and many others. It has been reported, Zucknick *et al* (2008), that both Dice-Sørensen and Ochiai indexes are simple modification of Jaccard index. Expectedly, these three indices tend to similar results' interpretation. Therefore, we shall only consider the *Jaccard index* being the most popular among the three measures.

The *simple matching index*, as would be seen shortly, is just the complement of the misclassification error rate (MER) given by (2.13), (2.14), and (2.63), which we have adopted in the construction of our k-SS classifier. In an unambiguous term, SMI = 1-MER. Therefore, the SMI shall not be given any separate treatment here again.

More generally, using the 2×2 confusion matrix in *Table 2.1*, the following similarity indices can be estimated as follows;

- *Jaccard index* (JI) is an asymmetric similarity measure between two classifiers (subjects' true class grouping and classification by k-SS classifier) which attaches more importance to the correct or incorrect classification of subjects with outcome of interest (group 1 subjects). It is estimated by

$$\rho_J(T,P) = \frac{n(T \cap P)}{n(T \cup P)} = \frac{a}{a+b+c} \qquad (2.5.9)$$

- *Dice-Sørensen index*: $\rho_{D-S}(T,P) = \frac{2a}{2a+b+c}$
- *Ochiai index*: $\rho_O(T,P) = \frac{a}{\sqrt{a+b} * \sqrt{a+c}}$
- *Simple matching index* (SMI): $\rho_{SM}(T,P) = \frac{a+d}{a+b+c+d}$

Like any other performance measures adopted in this work, the estimates of the *Jaccard index*, as will be reported later, are the cross-validated estimates based on the respective subsampling scheme adopted for estimation. The R code that computes the JI are already part of the main code we developed for the implementation of our k-SS method as provided in Appendix B.1. Therefore, the cross-validated estimates of the JI indices shall be part of our k-SS results' outputs.

The receiver operating characteristic (ROC) analysis

The ROC analysis is an integral part of measures commonly adopted to assess the worth of any classification rule. It was originally developed by Egan (1975) for analysis of radar images in signal detection theory. Its procedure was later adapted into the screening of diagnostic tests to aid medical decisions (Swets, 1988; Zou, 2002; Shapiro & Brutlag, 2004; etc.). This has helped to determine whether a particular patient will benefit from a given treatment or not. The extension of ROC analysis to assess the performance of classifiers has been reported in Swets *et al* (2000), Fawcett (2006) and many other related studies.

The excellent use of ROC analysis lies in the construction and uses of the ROC curve and the area under the curve (AUC). The ROC curve is a useful tool to describe the performance of a classifier (or diagnostic test) that discriminates between normal (healthy) and cancerous (diseased) subjects based on variable(s) measured on continuous scale. In other words, both the ROC curve and the area under the curve (AUC) are measures of ranking of the quality of a classifier.

Suppose the expression level of gene X_j is measured on n subjects with two outcome groups 1 (for tumour subjects) and 0 (for normal subjects). Let X_{1j} and X_{0j} (X_{1j},$X_{0j} \in X_j$) denote the expression levels of n_1 and n_0 subjects in groups 1 and 0 respectively, $n_1 + n_0 = n$. Necessarily, X_{1j} measures are assumed to be greater that X_{0j} if gene

X_j is to discriminated the response group n_1 from n_0. What AUC then does is to rank all the n subjects based on their gene expression levels and compute the probability of correct ranking of any randomly selected (tumour, normal) subject pair given by (Green & Swets, 1966),

$$A = p(X_{1j} > X_{0j}), \ \ 0 \leq A \leq 1. \qquad (2.5.10)$$

This is the true area under the ROC curve and its estimate can be obtained in different ways. We highlighted below, four of the methods by which AUC can be computed as equally being reported in Hanley & McNeil (1983):

- The trapezoidal rule, Morrison (2005), Fawcett (2006).
- The output from Dorfman & Alf maximum likelihood estimation program, Dorfman & Alf (1969).
- Plot of the original data on binomial graph paper and compute the AUC area from the slope and intercept of the plot by $\hat{A} = p(Z \ \leq \ \hat{Z}_A)$, where $\hat{Z}_A = \frac{intercept}{\sqrt{1+slope^2}}$, and $Z \sim N(0{,}1)$, Swets (1979).
- The use of the Wilcoxon-Mann-Whitney test statistics approximation, Bamber (1975).

After the AUC estimate $\hat{A}_j$ has been computed for each gene X_j, the (null) hypothesis test that X_j is not capable to discriminate between any two subjects' groups can be tested. This is given by,

$$H_0 : p\left(X_{1j} > X_{0j}\right) \leq 0.5 \quad \text{vs.} \quad H_a : p\left(X_{1j} > X_{0j}\right) > 0.5 \qquad (2.5.11)$$

The value of $\hat{A}_j$ very close to 1 will provide evidence to support that X_j is a good discriminator of the two subjects' groups (accepting H_a) while a value of $\hat{A}_j$ very close to 0.5 or less will suggest otherwise (accepting H_0). One can arrive at any of these two decisions using the $100(1-\alpha)\%$ confidence interval of $\hat{A}_j$ given by $\hat{A}_j \pm Z_\alpha s.e(\hat{A}_j)$ where $s.e(\hat{A}_j)$ is the standard error of $\hat{A}_j$ as defined in Section 2.6, and Z_α is the percentiles of the standard normal distribution at a specified Type I error, α.

The plot of the ROC curve can be obtained for each gene X_j to visualize the performance of each of them as reported by their AUC estimates, $\hat{A}_j$. The ROC curve is a 2-dimensional plot of sensitivity of the classifier against 1-specificity. The sensitivity, sometimes called the true positive (TP) rate or recall is plotted on the y-axis while 1-specificity, also called the false positive (FP) /false alarm (FA) rate is plotted on the x-axis. In other words, the sensitivity of a classifier φ is given by the probability $p(\varphi = 1|Y = 1)$ while its specificity is estimated as $p(\varphi = 0|Y = 0)$. The ROC curve however, shows the trade-off between the benefits (TP) and the costs (FP) of a classification or ranking rule φ. Some of the metrics used to compute the sensitivity, specificity and other related measures are presented in the confusion matrix in *Table 2.5*.

		True Class (*T*)		
		1	0	Marginal Total
Predicted class (*P*) by *φ*	1	TP	FP	$TP + FP$
	0	FN	TN	$FN + TN$
	Marginal Total	$TP + FN$	$FP + TN$	

Table 2.5: Confusion matrix showing common performance metrics calculated from it.

Along the column of the confusion matrix is the true class label of the outcome variable Y for the two biological sub-groupings of mRNA samples and along the row are the predicted classes of these subjects by the classifier $\varphi(\boldsymbol{x})$. The cell entries TP, FP, FN, TN represent the *true positive*, *false positive*, *false negative* and *true negative* respectively. Therefore, given the observed biological groups $y \in \{0,1\}$ of the test sample n_{TE} and the predicted response class $\hat{\varphi} \in \{0,1\}$ as provided by k-SS classifier $\varphi(\boldsymbol{x})$, the following performance measures can be computed from the confusion matrix in *Table 2.5* among others:

- Sensitivity $= \frac{TP}{TP+FN} = \frac{\sum_{i=1}^{n_{TE}} I(y_i=1;\, \hat{\varphi}_i=1)}{\sum_{i=1}^{n_{TE}} I(y_i=1)}$
- Specificity $= \frac{TN}{FP+TN} = \frac{\sum_{i=1}^{n_{TE}} I(y_i=0;\, \hat{\varphi}_i=0)}{\sum_{i=1}^{n_{TE}} I(y_i=0)}$
- Positive predictive value $= \frac{TP}{TP+FP} = \frac{\sum_{i=1}^{n_{TE}} I(y_i=1;\, \hat{\varphi}_i=1)}{\sum_{i=1}^{n_{TE}} I(\hat{\varphi}_i=1)}$
- Negative predictive value $= \frac{TN}{TN+FN} = \frac{\sum_{i=1}^{n_{TE}} I(y_i=0;\, \hat{\varphi}_i=0)}{\sum_{i=1}^{n_{TE}} I(\hat{\varphi}_i=0)}$

where $I(.)$ is an indicator function whose value is 1 if its argument is true and 0 otherwise. The positive predictive value (PPV) measures the precision of the classifier. It shows the proportion of the true class 1 (tumour) subjects that are correctly classified into that class among those that were classified as class 1 subjects by classifier $\varphi(\boldsymbol{x})$. Similarly, the negative predictive value gives the proportion of group 0 (healthy) subjects that are correctly classified into that group among the subjects classified as group 0 subjects.

The estimates of all the above performance measures are obtained as cross-validated estimates for each of the k-SS classifiers constructed. The `R` codes we developed to compute all the cross-validated performance measures are already incorporated into the main `R` codes we developed for the construction of our k-SS classifier as given in Appendix B.1.

To construct the ROC curves for the k-SS classifiers, all the test samples $n_{TE1}, \dots, n_{TER}$ generated by MCCV or bootstrap over R

random partitions are merged into one large sample $n_{TE}^{*} = (n_{TE1}, \dots, n_{TER})$. The true class labels $y \in \{0,1\}$ and the predicted probabilities $\hat{p}(y|\boldsymbol{x}) \in [0,1]$ of belonging to any of the y classes estimated for each subject in n_{TE}^{*} are observed. These two values are then passed into our algorithm to generate the cross-validated ROC (CVROC) curves for each of the k-SS classifiers. More details on various ways to construct a typical ROC curve are provided by Fawcett (2006).

A flexible procedure for generating ROC curve in `R` as implemented in the `ROCR` library (`library(ROCR)`) by Sing *et al* (2005) was adapted into our main `R` codes (see Appendix B.1) to generate the CVROC curves for our k-SS classifier.

A particular variant of the ROC curve which we do not consider in this thesis is the *ordinal dominance curve* (ODC) proposed by Bamber (1975). The ODC is obtained by reversing the axes of the ROC curve. By this, a plot of specificity (on the y-axis) against 1-sensitivity (on the x-axis) produces a typical ODC curve. More details on this could be found in Hsieh & Turnbull(1996).

2.6 The AUC preliminary feature selection method

A new preliminary feature selection procedure we introduce in this work is based on the concepts and criteria of the area under the ROC curve (AUC). The importance of the ROC curve as a good measure of performance of a classification or ranking rule has been reported in many works as discussed in Section 2.3.1. The exact relationship between the empirical prediction error rate (PER) and the estimated area beneath the ROC curve (AUC) has been established by Cortes & Mohri (2004). In their study, they established that if the empirical PER of a given ranking function, say $\varphi(X)$, is given by ϑ, then, the average estimated AUC over all possible rankings of subjects corresponding to $\varphi(X)$ could be approximated by $1 - \vartheta$ especially when the two class probabilities $\hat{p}_0 = \frac{n_0}{n}$ and $\hat{p}_1 = \frac{n_1}{n}$ are very close to each other. This argument particularly underscores the relevance of the AUC as another efficient measure to assess the goodness of classification or ranking rules. Therefore, the preliminary selection we are proposing here using AUC criteria could be seen as a classifier-like preliminary feature selection method.

The reasons for proposing this new preliminary selection method are two-fold. The fact remains that there are no unique standard criteria for determining which genes to be selected at the preliminary selection stage while working with most of the preliminary feature selection methods. This is very true of the t-test approach as presented in Chapter 1 Section 1.4.2. For example, the choice of the cut-point p^* or its t-statistic ($\hat{t}_s$ or $\hat{t}_w$) equivalent the under this approach is at the discretion of the investigators. Due to the absence of standard way of choosing such cut-point, it is not uncommon for different analysts to select different number and types of transcripts at preliminary selection stage for analysis under this method.

Secondly, the common practice of using all the available mRNA sample size n while performing preliminary feature selection without leaving out certain proportion of the sample for cross-validation has been criticized to be capable of increasing the prediction bias of classification rules (Ambroise & McLachlan, 2002). This might consequently result to poor gene selection at the preliminary stage. This line of argument was equally corroborated by Ioannidis (2005) and recently by Boulesteix *et al* (2008). Hence, there is need to evolve a preliminary feature selection procedure, like the one proposed here, that will allow for easy cross-validation through via external (independent) test samples.

Consider a set of q genes, $\boldsymbol{X} = (X_1, \dots, X_q)$, whose expression levels are measured on two groups $Y_i \in \{0,1\}$ of n biological subjects as previously described in relevant sections. The main goal here is to perform a preliminary (primary) selection of potentially relevant q^*genes from all the available q genes such that all the $q - q^*$ non-predictive genes are removed prior to model construction proper. The reasons for this are two-fold: One is to save a lot of computation time and efforts while carrying out the analysis. If the $q - q^*$ 'unwanted' genes are not removed before any dimension reduction and prediction exercise is performed, a good classifier will still filter them out during the analysis proper, but at a huge cost of analysis time. To avoid this therefore, it is proper to filter them out before proper classifiers construction could begin. The second reason that is not too far from the first one is to reduce noise from the data before proper analysis could commence. This is to avoid undue influence of the irrelevant genes on classification results.

Our procedure starts by partitioning the entire sample size n into training sample, n_{TR} and test sample, n_{TE}. This is followed by fitting univariate logit model, $logit\big(\pi(X_j)\ \big) = \alpha + \beta_j X_j$, $j = 1, \dots, q$, on each of the q genes using the training sample, n_{TR}. Next is to use the fitted model to estimate the predicted class probabilities, $\hat{p}_i(Y_i = y|X_j)$, $i = 1, \dots, n_{TE}$, (probability of subjects belonging to class y), for each subject in the left out test sample, n_{TE}. This is followed by cross-validation using sub-sampling scheme of v-fold-cross-validation, the concepts of which shall be discussed fully in Section 2.7. By this choice of cross-validation method, the entire sample size n is divided into a number of equal fold v with each of the v fold serving as the test sample at each sample selection. The remaining $v - 1$ is then used to build the *logit* model. This method has the advantage of ensuring that all the observations are being used as both the training and test samples at different time. Thereafter, both the predicted probabilities $\hat{p}_i(Y_i = y|X_j)$ and the true class labels $y \in \{0,1\}$ for each subject in the test sample n_{TE} as observed from the fitted model for each gene X_j are used to construct the cross-validated ROC (CVROC) curve from which the respective area under the curve (AUC) would be estimated.

Let the estimated AUC for each gene X_j using the test sample n_{TE} be denoted by $\hat{A}_{X_j}$ and let $\hat{\bar{A}}_{X_j}$ be the respective average AUC obtained over the entire v fold. To establish the significance or otherwise of the estimated average AUC $\hat{\bar{A}}_{X_j}$ for each gene, we simply test one directional hypothesis set given in (2.6.1) for each $\hat{\bar{A}}_{X_j}$. By this, we construct and tested a total of q hypothesis set of the form

$$H_{0j}: p(X_{1j} > X_{0j}) \leq 0.5 \quad \text{vs.} \quad H_{aj}: p(X_{1j} > X_{0j}) > 0.5\ ,\ j = 1, \dots, q,$$

This could be equivalently written in terms of the average AUC, $\bar{A}_{X_j}$ for the population as

$$H_{0j}: \bar{A}_{X_j} \leq 0.5 \quad \text{vs.} \quad H_{aj}: \bar{A}_{X_j} > 0.5 \qquad (2.6.1)$$

Since the estimated AUC, $\hat{A}_{X_j}$ has a Gaussian distribution, Hanley & McNeil (1982), it then becomes easier to develop a test procedure for the hypothesis set in (2.6.1) as follows;

$$\hat{A}_{X_j} \sim N(\mu_A, \sigma_A^2) \leftrightarrow \hat{\bar{A}}_{X_j} \sim N(\mu_{\bar{A}}, \sigma_{\bar{A}}^2)$$

$$\rightarrow \qquad \hat{\hat{Z}}_{X_j} = \frac{\hat{\hat{A}}_{X_j} - \mu_{\bar{A}}}{\sqrt{\sigma^2_{\bar{A}}}} \sim N(0,1) \qquad (2.6.2)$$

where $\mu_{\bar{A}}$ and $\sigma^2_{\bar{A}}$ are the mean and variance of $\hat{\hat{A}}_{X_j}$ respectively. If we adapt Bamber's estimator of standard error of the AUC, Bamber (1975), $\sigma^2_{\bar{A}}$ could be estimated by

$$\sigma^2_{\bar{A}} = \frac{\hat{\hat{A}}_{X_j}\left(1-\hat{\hat{A}}_{X_j}\right)+(n_1-1)\left(\hat{P}_{jx^+x^+x^-}-\hat{\hat{A}}^2_{X_j}\right)+(n_0-1)(\hat{P}_{jx^+x^-x^-}-\hat{\hat{A}}^2_{X_j})}{n_1 n_0} \qquad (2.6.3)$$

where $\hat{P}_{jx^+x^+x^-}$ is defined as the probability that a classifier ranks any two randomly chosen tumour subjects higher than a normal subject and $\hat{P}_{jx^+x^-x^-}$ is the probability that a classifier ranks two randomly chosen normal subjects lower that a tumour subject. These two probabilities can be estimated by adapting the statistics proposed by Hanley & McNeil (1982) for which

$$\hat{P}_{jx^+x^+x^-} = \frac{\hat{\hat{A}}_{X_j}}{\left(2-\hat{\hat{A}}_{X_j}\right)} \text{ and } \hat{P}_{jx^+x^-x^-} = \frac{2\left(\hat{\hat{A}}_{X_j}\right)^2}{\left(1+\hat{\hat{A}}_{X_j}\right)} \qquad (2.6.4)$$

For any pre-specified level of significance α, the apparent decision rule for the test hypothesis in (2.6.1) is to reject the null, H_{0j} in favour of H_a if $\hat{\hat{Z}}_{X_j} = \frac{\hat{\hat{A}}_{X_j} - \mu_{\bar{A}}}{\sqrt{\sigma^2_{\bar{A}}}} \geq Z_{1-\alpha}$. This can be equivalently re-constructed as; reject H_{0j} in favour of H_{aj} if

$$\hat{\hat{A}}_{X_j} \geq \mu_{\bar{A}} + Z_{1-\alpha}\sqrt{\sigma^2_{\bar{A}}} \qquad (2.6.5)$$

Under H_{0j}, $E\left(\hat{\hat{A}}_{X_j}\right) = \mu_{\bar{A}} = 0.5$, then, H_{0j} is rejected in favour of H_{aj} if

$$\hat{\hat{A}}_{X_j} \geq 0.5 + Z_{1-\alpha}\sqrt{\sigma^2_{\bar{A}}} \qquad (2.6.6)$$

and for any preliminary feature selection, the decision rule is to select that gene X_j whose estimated average AUC value $\hat{\bar{A}}_{X_j}$ satisfies the inequality in (2.6.6).

It should be noted that $\hat{\bar{A}}_{X_j} = 0.5$ corresponds to AUC area that lies on the 45^0 diagonal line of a typical ROC plane as shown in *Fig 2.2*. Any gene whose AUC value revolves around the diagonal, as the case with gene OIP106 in *Fig 2.2*, does not possess any useful information to correctly predict (rank) the response group. Such gene lacks any good predictive power and should be dropped. In a nutshell, any gene whose AUC value is greater than 0.5 by $Z_{1-\alpha}$ of its standard error would be selected primarily by this method for further analysis, where $Z_{1-\alpha}$ is the quantile of the standard Gaussian density obtained at significance level α. The size of α for this test could be any of the conventional default values in the range $\alpha \in (0, 0.05]$.

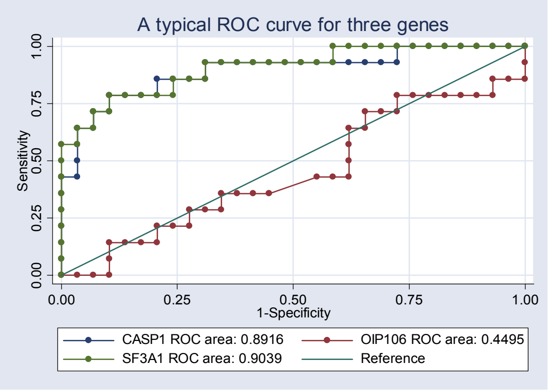


Fig 2.2: A typical ROC curve for three (CASP1, SF3A1, OIP106) of the 24,026 genes in the rectal cancer microarray data. While the two genes, CASP1 and SF3A1 are informative as shown by their ROC curves being far away from the diagonal reference line with their respective high AUC estimates of 0.8916 and 0.9039, gene OIP106 contains no information to be able to predict the response group, hence, its own ROC curve revolves round or below the diagonal reverence line with relatively small AUC estimate of 0.4495.

Using this procedure, a total of q^* potential discriminating genes would be selected at the preliminary gene selection stage with extremely small chance of leaving out any of the potentially good genes from further analysis.

The new preliminary feature selection method proposed here is implemented using `R` statistical package. The sub-sampling technique of v-fold-cross-validation is adopted in the implementation of this method and the `R` codes that implement the procedure is presented in Appendix B.3. Due to the huge number of gene variables involved (usually in thousands) in microarray data sets, any choice of fold v between 2 to 10 would be suitable for the test. The application of this new preliminary feature selection method is demonstrated in Chapter 3 in relation to our new k-SS method. The k-SS algorithms under the two sub-sampling scheme of MCCV and bootdtrap.632+ for which the new AUC preliminary feature selection is incorporated are provided in Appendix B.5 and B.7 respectively.

2.7 **Cross-validation techniques in brief**

In any typical microarray data, the number of available biological samples is usually very small. Since genes selections, biological sample predictions and all other performance measures are based on these small samples, it is therefore possible for the estimated results to be bias. As a result of this, it is important to device some estimation procedures that would ensure that the results obtained from the small sample would be a good representation of the population, thereby removing any form of bias from the estimators. For instance, the empirical *prediction error rate* (PER), $\hat{\vartheta}_{emp(PER)}$ estimated by classification rule $\varphi(\boldsymbol{x})$ using n sample is expected to be close to the unseen true PER, $\vartheta_{true(PER)}$ for the entire population. The difference between the expected value of the PER estimator $\hat{\vartheta}_{emp(PER)}$ and the true PER value from the population is called the bias of

$\hat{\vartheta}_{emp(PER)}$. That is, $\hat{\vartheta}_{bias} = E\big(\hat{\vartheta}_{emp(PER)}\big) - \vartheta_{true(PER)}$. If the bias, $\hat{\vartheta}_{bias}$ is zero, it implies that $E\big(\hat{\vartheta}_{emp(PER)}\big) = \vartheta_{true(PER)}$ an indication that the estimator $\hat{\vartheta}_{emp(PER)}$ is a good estimator of the population parameter $\vartheta_{true(PER)}$. Hence, $\hat{\vartheta}_{emp(PER)}$ becomes an unbiased estimator of $\vartheta_{true(PER)}$. But a large value of bias indicates that $\hat{\vartheta}_{emp(PER)}$ is not a good estimator of the population parameter and its results might not be suitable for generalization.

One of the popular short cuts at removing bias from an estimator is through cross-validation techniques first introduced by Seymour Geisser (1993) with additional discussions on his works by Berry (2005). By cross-validation approach, the original sample size n is partitioned into subsets such that the analysis is initially performed on a single subset of n called the training sample, while the other subset(s), called the test sample(s) are retained for subsequent use in confirming and validating the results from previous analysis. Several forms of this method are available in the literature. The most prominent ones are discussed in what follows.

i.) *Holdout method*

By this method, the original n sample is splitted randomly into two, n_{TR}, n_{TE}, with $n_{TR} + n_{TE} = n$. One part (n_{TR}) is used to train the classifier while the second part (n_{TE}) is held out to test the goodness of the classifier. This is sometimes called *out-of-bag* method. In practice, it is customary to holdout 1/3 of n (n_{TE}) for testing and the remaining 2/3 of n (n_{TR}) for training, McLachlan (1992). The empirical prediction error rate is computed over the test sample n_{TE} by

$$\hat{\vartheta}_{holdout} = \frac{1}{n_{TE}}\sum_{i=1}^{n_{TE}}\left[I_{\{\hat{\varphi}(\boldsymbol{X}_i)\neq Y_{i0}\}}\right] \tag{2.7.1}$$

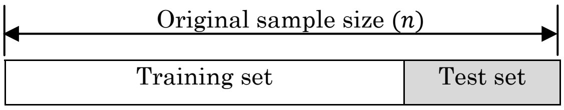


Fig 2.3: Schematic representation of the sample splitting under the Holdout cross-validation method

The schematic representation of the sample split under this method is presented in *Fig 2.3*. This method poses no computational burden. Its major disadvantage apart from small sample size problem is that the sample used as the training or test sample might not be representative of the original sample. It is possible to miss out all members of a certain class in a training or test set. Therefore, whatever error rate reported might be misleading.

ii.) *Monte Carlo cross validation (MCCV)*

The MCCV method sometimes called random subsampling is one of the cross-validation techniques proposed to overcome the limitations of the holdout method. The approach is to repeat the process of taken random sub-samples of training set, n_{TR} and test set, n_{TE} from the original sample size n several number of (R) times (e.g. 50, 100, 500, 1000 or 10000 repetitions) without replacement. At each random split, classifier is learned on the training set while its goodness is assessed on the test set via prediction error rate $\hat{\vartheta}_r = \frac{1}{n_{TE}}\sum_{i=1}^{n_{TE}}\left[I_{\{\hat{\varphi}_i(\boldsymbol{X})\neq Y_{i0}\}}\right]$ which is computed at each r repetition, $r = 1,2,\ldots,R$. The different prediction error rates over the entire R repetitions are then averaged to yield an overall average prediction error rate. That is

$$\hat{\hat{\vartheta}}_R = \frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_r \qquad (2.7.2)$$

However, the maximum number of subsamples of test set n_{TE} that can be drawn from n without replacement is $R = \frac{n(n-1)\ldots(n-n_{TE}+1)}{n_{TE}!}$. This approach has been widely adopted in many works (Xu & Liang, 2001; Dudoit *et al*, 2002; Xu *et al*, 2004; Lee *et al*, 2005; Du *et al*, 2006; Zucknick *et al*, 2008; etc.) due to its reliability and results' consistencies. The supremacy of MCCV over the *leave-one-out* cross-validation method (discussed below) was equally reported in Xu *et al*(2004). However, the MCCV approach is computationally demanding unlike the holdout method. A schematic representation of subsampling stages under MCCV is given by *Fig 2.4.*

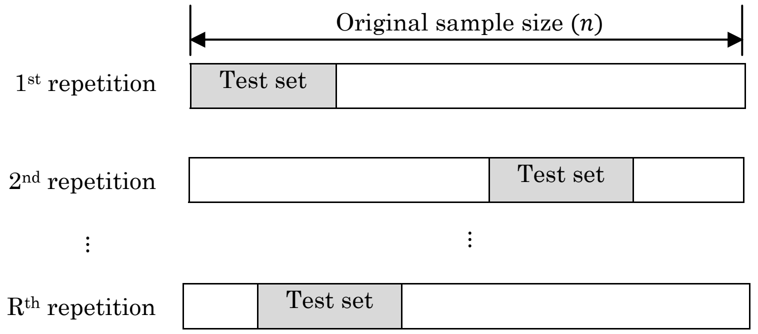


Fig 2.4: Schematic representation of the random sub-sampling for cross-validation under the MCCV method

iii.) *v-fold-cross-validation*

In this method, the n sample is divided into a number of mutually exclusive equal subsamples of fixed fold, v. Each fold is used for testing while the remaining $v-1$ folds are used for training. This exercise is repeated v times such that each of the v test samples is used once. The prediction error rate $\hat{\vartheta}_v = \frac{1}{n_{TE}}\sum_{i=1}^{n_{TE}}\left[I_{\{\hat{\varphi}(X_i)\neq Y_{i0}\}}\right]$ is computed at each fold and the average of all the prediction error rates, averaged over v, is computed as the true prediction error rate. Thus, we have that

$$\hat{\hat{\vartheta}}_v = \frac{1}{v}\sum_{v=1}^{v}\hat{\vartheta}_v \qquad (2.7.3)$$

A major challenge of this method is the determination of the best number of fold to be adopted. However, ten-fold cross-validation has been suggested in many studies as a standard way of measuring the misclassification error rate using this approach, Witten & Frank (2000), Molinaro *et al*(2005). Advantage of this approach is that one is sure that all the original samples are used for both classifier construction and its assessment. Nonetheless, the estimated prediction error rate may be associated with high variance due to the smallness of the sample size. A schematic representation of this subsampling procedure with v = 3 is given by *Fig 2.5.*

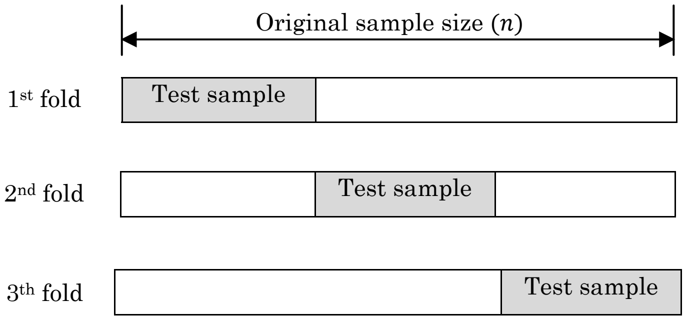


Fig 2.5: Schematic representation of the v-fold cross-validation method with v = 3

iv.) Leave-one-out cross-validation (LOOCV)

The LOOCV is an extreme case of v-fold cross-validation with $v = n$. Here, each subject in the sample is left out and the remaining $n - 1$subjects are used to learn the classifier. The left out sample in turn is used to test the goodness of the classifier. This exercise is performed n times to ensure that each subject has been used in the construction and validation of the classifier. *Fig 2.6* gives its schematic form at each evaluation. The prediction error is obtained for each left out sample and the average for all the n samples is taken as the empirical prediction error rate. That is

$$\hat{\vartheta}_{loocv} = \frac{1}{n}\sum_{i=1}^{n}\left[I_{\{\hat{\varphi}(X_i)\neq Y_{i0}\}}\right] \qquad (2.7.4)$$

where indicator function $I_{\{.\}}$ is as defined in (2.4.3).

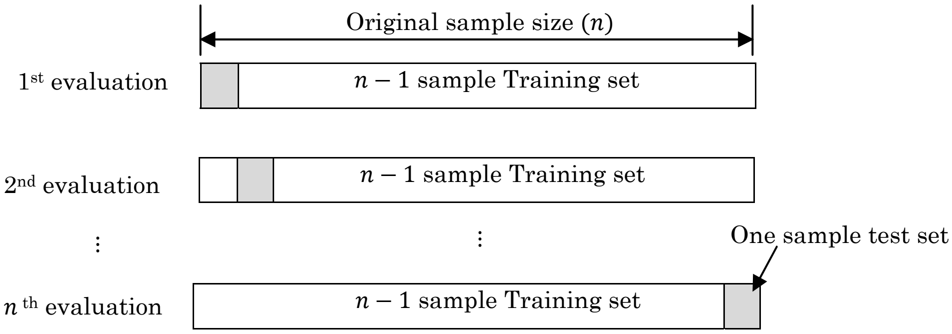

Original sample size (n)
1st evaluation
$n-1$ sample Training set
2nd evaluation
$n-1$ sample Training set
⋮
⋮
One sample test set
n th evaluation
$n-1$ sample Training set

Fig 2.6: Schematic representation of the Leave-one-out cross-validation method

The advantage of this method is that it returns low bias for prediction error rate since almost all the sample size is used to train. Like in the v-fold method, the LOOCV is equally associated with high variance of the prediction error rate. Nonetheless, it has been described as an elegant cross-validation measure suitable for eliminating bias from an estimator provided that the original sample size n is a true representation of the targeted population. This method has received a wider application in many research studies due to its simplicity, (Nguyen & Rocke, 2002a; Man *et al*, 2004; Boulesteix, 2004; Statnikov *et al*, 2005; etc.).

v.) *Bootstrap*

The bootstrap method is based on sampling with replacement. All the n subjects is sampled n times with replacement to give another ‘new’ n data set. The new n sample now becomes the training set and the original n sample is the test set. Since sampling is done with replacement, there is tendency to have some observations repeated in the new sample while some may not be sampled at all from the original sample. Therefore, the unsampled subjects in the original data become the test set by implication. A particular variant to this general bootstrapping is the bootstrap.632+ (Efron & Gong, 1983; Efron & Tibshirani,1997; etc.). The idea behind this new modification is that each subject in the original n sample has a probability $\frac{1}{n}$ of being selected into the new sample and $(1-\frac{1}{n})$ of not being selected. Since the samples are drawn n times with replacement, the chance that a subject is not selected into the new sample is then $\left(1-\frac{1}{n}\right)^n \approx \frac{1}{e} = 0.368$. Thus, for n random bootstrap sampling, about 36.8% of n will not be selected into the new data set (the training set). It shows that only about $1-\left(1-\frac{1}{n}\right)^n \approx 0.632$ of n would be in the training set while the remaining 0.368 of n would be in the test set, hence, the term bootstrap.632+. Suppose we define $\hat{\vartheta}_{train}$ as the re-substitution prediction error rate computed over the training set and $\hat{\vartheta}_{test}$ as the bootstrap prediction error rate computed over the test set. The empirical prediction error rate for bootstrap.632+ scheme is given (Efron & Tibshirani,1997; Gerds & Schumacher, 2007; Binder & Schumacher, 2008) by

$$\hat{\vartheta}_{boot} = 0.632 * \hat{\vartheta}_{test} + 0.368 * \hat{\vartheta}_{train} \qquad (2.7.5)$$

The entire bootstrap procedures are then repeated R number of times as in MCCV, and respective average prediction error rates $\hat{\hat{\vartheta}}_{test} = 0.632\frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_{r.test}$ and $\hat{\hat{\vartheta}}_{train} = 0.368\frac{1}{R}\sum_{r=1}^{R}\hat{\vartheta}_{r.train}$ are computed. These estimators are then used to compute the overall average prediction error rate, $\hat{\hat{\vartheta}}_{boot}$ for bootstrap.632+. Thus, we have that

$$\hat{\hat{\vartheta}}_{boot} = \frac{1}{R}\sum_{r=1}^{R}\left(0.632 * \hat{\vartheta}_{r.test} + 0.368 * \hat{\vartheta}_{r.train}\right) \quad (2.7.6)$$

Out of all these cross-validation techniques, the methods of MCCV and bootstrap are adopted in this thesis for the implementation of our proposed k-SS classifier.

2.8 Overview of some other classification methods

In this section we provide brief overview of three of the existing state-of-the art classification methods as considered in this thesis. The three methods discussed here are the *Support vector machines* (SVM), *k-nearest neighbours* (k-NN), and *Partial least squares* (PLS) methods. The theoretical background of other classification methods considered in this thesis can be found in the relevant literatures. The relative performance of all the methods as compared to the prediction results provided by our new k-SS classifier are discussed in later chapters.

2.8.1 *Support Vector Machines (SVM)*

Support vector machines (SVM) is one of the state-of-the art techniques developed in the field of statistical learning theory and pattern recognition. The original SVM algorithm was pioneered in Russia by Vapnik and his co-workers in the early sixties (Vapnik & Lerner, 1963; Vapnik & Chervonenkis, 1964; etc.) after which several modifications were incorporated into the original theory (see Vapnik & Chervonenkis, 1974; Vapnik, 1982; 1995; & 1998). The SVM method has become increasingly popular among the kernel based methods as an excellent tool in response group classification, regression and statistical pattern recognition. Because of the huge contributions of Vapnik and Chervonenkis to the present form of the SVM methodology, the SVM theory is now been referred to as the *Vapnik-Chervonenkis (VC) theory*. We have adopted SVM methodologies in this work mainly for the prediction/classification of mRNA samples into their respective biological groups using various microarray data sets. In what follows therefore, we present a brief theoretical background of the SVM procedure for classification.

There are several forms of SVM algorithms available in the literature, see McCormick (1983), Vapnik (1995), Cortes & Vapnik (1995), Smola (1998), Smola & Schölkopf (2004), Lee (2004) and a host of others. However, we shall present the SVM procedures of Burges (1998) and Lee (2004) which essentially are adaptations of the original algorithm of Vapnik (1995).

Let $\boldsymbol{t_r} = \{(x_{1j}, y_1), (x_{2j}, y_2), \dots, (x_{n_{TR}j}, y_{n_{TR}})\}$, $j = 1, \dots, q$, be the training set of n_{TR} biological samples with the corresponding test sample n_{TE} defined by $\boldsymbol{t_e} = \{(x_{1j}, y_1), (x_{2j}, y_2), \dots, (x_{n_{TE}j}, y_{n_{TE}})\}$, $n = n_{TR} + n_{TE}$. Each y_i, $i = 1, \dots, n$, is the true class label that correspond to the observed x_{ij} genes expression levels. For simplicity, we shall use the variable pair $(\boldsymbol{x}_i, y_i)$ to denote the input vector $\boldsymbol{x}_i$ of observed gene expression profiles on i biological sample with response class label y_i. With little modification of the definition of the response groups given in (2.1.2), we assume that both the training and test data sets come from only two response classes Ω_1 and Ω_2 but with $y_i = 1$ if subject i comes from class Ω_2 and $y_i = -1$ if the i subject comes from class Ω_1 with both classes remained as defined under Section 2.1. The goal in SVM methods is to find a decision function of the form

$$h(\boldsymbol{x}_i) = sgn(\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b) \qquad (2.8.1)$$

that would classify any unseen subject in the test sample n_{TE} into their respective class labels $y_i \in \{-1,1\}$, where $\boldsymbol{w}$ is a vector of weights with Euclidean norm $\|\boldsymbol{w}\| = \langle \boldsymbol{w}.\boldsymbol{w} \rangle^{1/2} = 1$ with b being the bias. The quantity $\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle$ is the inner product of vectors $\boldsymbol{w}$ and $\boldsymbol{x}_i$ defined as $\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle = \boldsymbol{w}'\boldsymbol{x}_i$. Suppose we define a hyperplane $H_0 \in H$, simply called the separating hyperplane, that separates the training samples into the two existing response class labels $(-1,1)$. If the two response groups Ω_1 and Ω_2 of subjects that make up the training sample are linearly separable, then we can define the maximal distance of the separating hyperplane H_0 from the closest positive sample $(y_i = 1)$ by d_+units and its respective maximal distance from the closest negative sample $(y_i = -1)$ by d_-units. If the two maximal distances are the same, that is, $d_+ = d_- = d$, then the two sample groups are $2d$units apart. The task in SVM procedure therefore, is to find the weight vector $\boldsymbol{w}$ and bias b that will maximize the distance d. In a linearly separable sample, the SVM algorithm seeks for the separating hyperplane with the maximal margin (distance) d. This essentially results to the following optimization problem using (2.8.1);

$$max_{\boldsymbol{w},b}\, d \qquad (2.8.2)$$

subject to the conditions that;

$$\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b \geq d, \text{ if } y_i = 1 \qquad (2.8.3)$$

$$\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b \leq -d, \text{ if } y_i = -1 \qquad (2.8.4)$$

with $\boldsymbol{w}$ having a unit norm $\|\boldsymbol{w}\| = 1$. Therefore, for any given linearly separable set of training data, we define a maximal margin hyperplane $H_1 \in H$ for which the equality $\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b = d$ in (2.8.3) holds and maximal margin hyperplane $H_{-1} \in H$ for which the equality $\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b = -d$ in (2.8.4) also holds. All vectors $\boldsymbol{x}_i$ for which these two equalities are satisfied are called *support vectors* and the solutions of the optimization problem depend only on these vectors and not on the entire dimension of the training set. In other words, support vectors are those points $\boldsymbol{x}_i$ that lie on the two maximal margin hyperplanes H_1 and H_{-1}. Thus, a subject would be classified into group $y_i = 1$ if the condition (2.8.3) is satisfied and into group $y_i = -1$ if condition (2.8.4) is satisfied. This concept is geometrically illustrated in *Fig 2.7.*

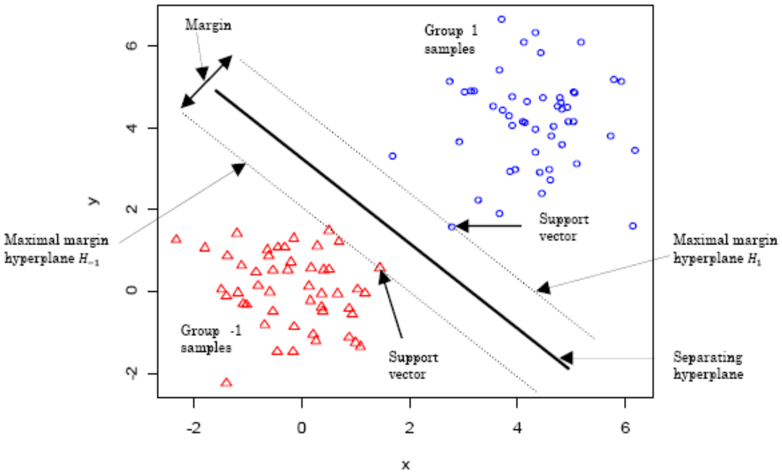


Fig 2.7: The figure showing the typical separating hyperplane and the maximal margin hyperplanes for the linearly separable subjects with two distinct subject groups. This is an example of linear SVM classification function given by equation (2.8.1). The support vectors lie on the margins.

If the two constraints in (2.8.3) and (2.8.4) are multiplied by their respective class labels and the weight vector $\boldsymbol{w}$ is divided by its norm $||\boldsymbol{w}||$ we shall have a single constraint of the form

$$\frac{1}{||\boldsymbol{w}||} y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \ b] \geq d, \forall_i, i = 1, \dots, n \qquad (2.8.5)$$

Since the two maximal margin hyperplanes H_1 and H_{-1} have the same normal (parallel), it shows that there exist a pair of hyperplanes in H that will provide the maximum margin between the two subject groups in the training set. This can be achieved by setting $d = \frac{1}{||\boldsymbol{w}||}$. Therefore, maximizing the value of d as given in

(2.8.2) is equivalent to minimizing the value of $\|\boldsymbol{w}\|$. Hence, the whole problem becomes that of looking for the weight vector $\boldsymbol{w}$ and bias b that minimizes $\|\boldsymbol{w}\|$. Thus, the optimization problem in (2.8.2) shall become that of;

$$min_{\boldsymbol{w},b}\|\boldsymbol{w}\| \qquad (2.8.6)$$

subject to the constraint that

$$y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + \ b] \geq 1, \forall_i, \, i = 1, \dots, n \qquad (2.8.7)$$

Under the new formulation of (2.8.6), all points $\boldsymbol{x}_i$ with margins $y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + \ b] = 1$ are now the support vectors.

In a situation where the training set $\boldsymbol{t_r}$ contains linearly but non-separable group members, then, it may be necessary to introduce the *slack variables* ξ_i to the constraints in (2.8.7). This is analogous to the *soft margin loss function* due to Bennett & Mangasarian (1992) which was later employed into SVM by Cortes & Vapnik (1995). The whole idea is to allow for some misclassification errors and the value ξ_i represents the amount by which the prediction function $h(\boldsymbol{x}_i)$ classifies subjects into the wrong side of the margin, Hastie *et al* (2009). Thus, the whole optimization problem in (2.8.6) then becomes that of

$$min_{\boldsymbol{w},b}\|\boldsymbol{w}\|, \text{subject to } \ y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + \ b] \geq 1 - \xi_i, \forall_i, \, i = 1, \dots, n \qquad (2.8.8)$$

with additional condition that $\xi_i > 0$, $\forall_i$, and that $\sum_{i=1}^{n} \xi_i = \xi$, for some fixed constant ξ.

The Lagrangian formulation of the above optimization problem is often preferred for easy generalization of the SVM procedures to pure non-linear separating data sets. This is done by constructing a

Lagrange function to be minimized from the objective function in (2.8.6) which we called the *primal* objective function of the form

$$\mathcal{L}_p \coloneqq \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + b] + \sum_{i=1}^{n} \alpha_i \qquad (2.8.9)$$

subject to the constraint that $\alpha_i \geq 0$, $i = 1, \dots, n$. The α_i are the Lagrange multipliers on each of the inequality constraints in (2.8.7) or (2.8.8). After little algebra, the *dual* form of the convex optimization problem (2.8.9) is obtained (Burges, 1998; Lee, 2004) as

$$\mathcal{L}_d(\alpha_i) \coloneqq \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{w}.\boldsymbol{x}_i\rangle \qquad (2.8.10)$$

This function is to be maximized subject to the conditions that $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. The Karush-Kuhn-Tucker (KKT) condition (Karush, 1939, Kuhn & Tucker, 1951) that

$$\alpha_i\{y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + b] - 1\} = 0 \;\; \forall_i, \, i = 1, \dots, n \qquad (2.8.11)$$

is often adopted to provide the estimate of b. From KKT condition above, it is very easy to verify that only few of the α_i's, say α_i^*, are non-zero at the optimal solution level and they are those α_i^*'s for which the margin $y_i[\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + b] = 1$. Hence, the vector $\boldsymbol{w}^*$ that defines the optimal maximal separating hyperplane has non-zero weights for the support vectors and can be easily obtained as

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \, \boldsymbol{x}_i \qquad (2.8.12)$$

More details on this can be found in Burges (1998), Bennett & Campbell (2000) and Lee (2004).

The classification function $h(\boldsymbol{x})$ in terms of the optimal separating hyperplane is now of the form

$$h(\boldsymbol{x}) = sign[\langle \boldsymbol{w}^*.\boldsymbol{x}\rangle + b^*]$$

$$\rightarrow \quad h(\boldsymbol{x}) = sign[\sum_{i=1}^{n} \alpha_i^* y_i \langle \boldsymbol{x}_i . \boldsymbol{x} \rangle + \ b^*] \qquad (2.8.13)$$

More generally, SVM algorithms embed data vector $(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ from the input space $\mathfrak{R}$ into the high-dimensional feature space $\mathcal{F}$ through the use of kernel functions $K(.,.)$. Given any non-linear mapping $\emptyset$ that embeds input vector $(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ into the feature space $\mathcal{F}$, kernel $K(.,.)$ has the following representation;

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \ \langle \emptyset(\boldsymbol{x}_i) . \emptyset(\boldsymbol{x}_{i'}) \rangle \qquad (2.8.14)$$

where $\boldsymbol{x}_i, \boldsymbol{x}_{i'} \in \mathfrak{R}$ and $\emptyset(\boldsymbol{x}_i), \emptyset(\boldsymbol{x}_{i'}) \ \in \mathcal{F}$. This implies that points $\boldsymbol{x}_i, \boldsymbol{x}_{i'}$ in the input space $\mathfrak{R}$ correspond to the points $\emptyset(\boldsymbol{x}_i), \emptyset(\boldsymbol{x}_{i'})$ in the feature space $\mathcal{F}$. The kernel representation allows efficient computation of the inner product directly in the feature space which saves a lot of rigorous data embedding and computational burden in the input space. The SVM method using kernel function separates the training data in the feature space by a hyperplane defined by the type of kernel function adopted. The kernel representation of the classification function $h(\boldsymbol{x})$ is of the form

$$h(\boldsymbol{x}) = sign[\sum_{i=1}^{n} \alpha_i^* y_i \, K(\boldsymbol{x}_i . \boldsymbol{x}) + \ b^*] \qquad (2.8.15)$$

The four types of kernels mostly adopted are the linear, polynomial, radial basis function and sigmoid kernels. The functional forms of these kernels are presented below:

- *Linear*: $K(\boldsymbol{x}_i . \boldsymbol{x}) = \langle \boldsymbol{x}_i . \boldsymbol{x} \rangle$
- *Polynomial*: $K(\boldsymbol{x}_i . \boldsymbol{x}) = [\langle \gamma \boldsymbol{x}_i . \boldsymbol{x} + c \rangle]^p$
- *Radial basis function (RBF)*: $K(\boldsymbol{x}_i . \boldsymbol{x}) = exp(-\gamma |\boldsymbol{x}_i - \boldsymbol{x}|^2)$
- *Sigmoid*: $K(\boldsymbol{x}_i . \boldsymbol{x}) = tanh\langle \gamma \boldsymbol{x}_i . \boldsymbol{x} + c \rangle$

The linear kernel corresponds to the single *inner product function* used by the linearly separable case as presented in (2.8.1) through (2.8.13). Both γ and c are the parameters used to determine the respective kernel functions while p is the number of degree used in polynomial kernel.

Like any other classification methods, the prediction accuracy of the SVM method over the test sample $\boldsymbol{t_e}$ is assessed through empirical misclassification rate analogous to the MER estimators given in (2.4.2) and (2.4.3). This is defined over the test sample n_{TE} by

$$\hat{\vartheta}_{SVM} = \frac{1}{2n_{TE}} \sum_{i=1}^{n_{TE}} \left| y_i - \hat{h}(\boldsymbol{x}_i) \right| \qquad (2.8.16)$$

where $y_i \in (-1,1)$ is the observed class labels and $\hat{h}(\boldsymbol{x}_i) \in (-1,1)$ is the predicted class label by SVM classifier $h(\boldsymbol{x})$ for i subject.

The SVM procedures for response class prediction are implemented in `R` statistical package under the `e1071` library. This we have adopted for analysis under the SVM implementations in this thesis.

2.8.2 *k-Nearest Neighbours* (*k*-NN)

The *k-nearest neighbours* (k-NN) is a supervised learning algorithm where the predictions of future test samples are determined based on the majority of nearest neighbours' category closest to them. It is the simplest form of classification procedure that has been adopted in many studies, (Zhang & Srihari, 2002; Baoli *et al*, 2003; Kuramochi & Karypis, 2005; Shang & Shen, 2005; etc.). It does not require any rigorous model to fit. For any given test data point, we only need to determine the number k of subjects in the training samples that are closest to that test data point. The classification is done through the use of simple majority votes of the classified categories.

More formally, let us consider a set of training sample $\boldsymbol{t_r} = \{(x_{1j}, y_1), (x_{2j}, y_2), \ldots, (x_{n_{TR}j}, y_{n_{TR}})\}$, $j = 1, \ldots, q$, on which the expression levels of q genes were measured. We assumed that the response group has binary category $y \in \{0,1\}$. Now, to predict/classify each member in the test sample $\boldsymbol{t_e} = \{(x_{1j}, y_1), (x_{2j}, y_2), \dots, (x_{n_{TE}j}, y_{n_{TE}})\}$, the k-NN algorithm begins by calculating the minimum distance of each test subjects from their corresponding training subjects and determine the k-nearest neighbours by ranks. The simple majority of these k-nearest neighbours become the prediction of the respective test samples. The similarity measure commonly used to measure the distance between the training and test sample is the Euclidean distance measure. The misclassification error rate (MER) for k-NN algorithm is calculated using the estimator given in (2.4.3) as used by our k-SS method.

2.8.3 ***Partial Least Squares*** **(PLS)**

The *partial least squares* (PLS) method is one of the old data reduction methods originally pioneered by Harald Wold (Wold, 1966, 1973, 1983, etc.). It has been adopted by chemometricians and other researchers for various purposes over many years, (Volmer *et al*, 1993; Holland *et al*, 1998; Naik & Tsai, 2000; etc.). The typical nature of microarray data in which it is often the interest to classify very few biological samples into their respective tumour groups using expression profiles of several thousand of genes has given the PLS approach a wider application in many microarray studies.

For brief theoretical presentation of PLS procedures, we consider the regression model $Y_i = g(\boldsymbol{X\beta};\, \varepsilon)$ as given in (2.1.1) whose linear form $Y_i = \boldsymbol{X\beta} + \varepsilon$ is as provided in (1.5.1) where $\boldsymbol{X} = (X_1, \dots, X_q)$ is a $n \times q$ matrix of gene expression levels measured on n biological subjects with binary response class $Y_i \in \{0,1\}$ given that $n < q$. With $n < q$ however, it is obvious that the classical least squares regression cannot be used to estimate parameter vector $\boldsymbol{\beta}$ of the above linear regression equation because the $q \times q$ design matrix $\boldsymbol{X}^T\boldsymbol{X}$ on which the estimator of $\boldsymbol{\beta}$ is based is not non-singular. What is being done, according to PLS approach is to represent the linear regression equation $Y_i = \boldsymbol{X\beta} + \varepsilon$ in terms of two separate equations of the form

$$Y = \boldsymbol{TQ}^T + F \tag{2.8.17}$$

$$\boldsymbol{X} = \boldsymbol{TP}^T + E \tag{2.8.18}$$

dropping the subscript i from Y_i for simplicity, where $\boldsymbol{T}$ is a $n \times c$ matrix of the latent components (*factor scores*) for the n obervations, $\boldsymbol{Q}^T$ is a $c \times 1$ vector of regression coefficients (the factor loadings of Y), $\boldsymbol{P}$ is a $q \times c$ matrix of regression coefficients (the factor loadings of $\boldsymbol{X}$), F and E are the residuals of regression models (2.8.17) and (2.8.18) respectively and c is the number of latent components $\boldsymbol{T}$ to be constructed usually fixed by the user. However, the maximum number c of latent components that can be constructed in any given PLS regression is $c = min\ (n, q)$.

The latent component $\boldsymbol{T}$ is usually of the form

$$\boldsymbol{T} = \boldsymbol{XW} \tag{2.8.19}$$

for an appropriate $q \times c$ weight matrix $\boldsymbol{W}$ for $\boldsymbol{X}$.

The estimate of the regression coefficients $\boldsymbol{Q}^T$ in (2.8.17) is usually obtained through the normal least square method as

$$\widehat{\boldsymbol{Q}}^T = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T Y \tag{2.8.20}$$

Once the estimates of vector $\boldsymbol{Q}$ has been determined, the estimates of the original coefficient $\boldsymbol{\beta}$ can then be estimated by

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{WQ}^T = \boldsymbol{W}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T Y \tag{2.8.21}$$

which can be simply expressed in terms of the weight matrix $\boldsymbol{W}$ as

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{X}^T Y \qquad (2.8.22)$$

From the estimator of $\boldsymbol{\beta}$ given by (2.8.22), it is obvious that the only quantity that needs to be determined to get $\widehat{\boldsymbol{\beta}}$ is the weight matrix $\boldsymbol{W}$. Similarly, the estimator of $\boldsymbol{P}^T$ in (2.8.18) can be conceived as

$$\widehat{\boldsymbol{P}}^T = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{X} \qquad (2.8.23)$$

However, several variants of PLS algorithms are available in the literature all of which are targeted at extracting the vector of latent components $\boldsymbol{T}$. The most common among this whose procedure we shall present here is the *non-linear iterative partial least squares* (NIPALS) algorithm due to Wold (1975). The NIPALS algorithm seeks to maximize the objective function

$$\boldsymbol{w}_i = \underset{w}{argmax}\, Cov^2(\boldsymbol{T}, Y)$$

$$= \underset{w}{argmax}(\boldsymbol{T}^T Y Y^T \boldsymbol{T})$$

$$\rightarrow \qquad \boldsymbol{w}_i = \underset{w}{argmax}(\boldsymbol{W}^T\boldsymbol{X}^T Y Y^T \boldsymbol{W}\boldsymbol{X}) \qquad (2.8.24)$$

subject to the constraints that

$$\boldsymbol{w}_i^T\boldsymbol{w}_i = 1 \qquad (2.8.25)$$

and that

$$\boldsymbol{t}_i^T\boldsymbol{t}_j = \boldsymbol{w}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_j = 0 \qquad (2.8.26)$$

for $i \neq j \in \{1, \dots, c\}$. The quantity $\boldsymbol{w}_i$ and $\boldsymbol{t}_i$ are the columns of $q \times c$ and $n \times c$ weight matrix $\boldsymbol{W}$ and latent components $\boldsymbol{T}$ with both $\boldsymbol{w}_i$ and $\boldsymbol{t}_i$ defined as $\boldsymbol{w}_i = (\boldsymbol{w}_{1i}, \boldsymbol{w}_{2i}, \dots, \boldsymbol{w}_{qi})^T$ and $\boldsymbol{t}_i = (\boldsymbol{t}_{1i}, \boldsymbol{t}_{2i}, \dots, \boldsymbol{t}_{ni})^T$ respectively. Thus, the row-vector representations of $\boldsymbol{W}$ and $\boldsymbol{T}$ are given by $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_c)$ and $\boldsymbol{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \dots, \boldsymbol{t}_c)$ respectively. By the objective function given by (2.8.24), each of the weight vector $\boldsymbol{w}_i$ is computed such that the square of the covariance of the response variable Y and latent components $\boldsymbol{T} = \boldsymbol{XW}$ is maximized subject to the conditions that each $\boldsymbol{w}_i$ is of unit norm (by (2.8.25)) and that all the latent vectors $\boldsymbol{t}_i \in \{\boldsymbol{T}\}$ are purely orthogonal (by (2.8.26)).

After the construction of the PLS components, the classification of the response groups would be performed using the c PLS components constructed by adapting any of the standard classification methods such as the *linear discriminant analysis* (LDA), *logistic discriminant* (LD) analysis, *quadratic discriminant analysis* (QDA) and the like. More details about the PLS method for classification can be found in Martens (1985), Wold *et al* (1983), Dai *et al* (2006), Rosipal & Krämer (2006), Boulesteix & Strimmer (2007) and in many other related works. However, in our implementation of the PLS approach for classification, we have adapted the LDA procedure as implemented in the `plsgenomics` library of `R` statistical package. Detail applications of this classification method are provided in the next two chapters.

3 Simulation Studies

3.1 Simulating Microarray data sets

Simulation is the process of emulating the reality using mathematical models. The sole objective is to build models to replicate the actual system. This is often necessary especially when the cost, time and efforts of generating live observations for investigation purposes are rather too unbearable. In such a situation, models that are replica of the condition under study may be simulated to examine the behaviour of the system, proffer solutions to the identified problems and evaluate the practicability of the solutions provided before transferring them to the real world.

For some years back, developing appropriate models to analyse microarray data was such a daunting task due to the sparseness of relevant data sets. This is not unconnected with the huge costs and times involve in generating such data sets. The situations become a lot better in the past few decades due to the advent of several microarray technologies. However, the sensitive nature of microarray studies especially with the involvement of human data has made it more imperative for the investigators to carryout analysis on similar pseudo (simulated) data to ascertain the appropriateness of their methods and results before such could be implemented on live data.

To implement our newly proposed *k*-SS classifier, we intend to simulate typical microarray data set on which the procedure would be tested to ascertain its suitability and results' efficiencies. The performance of our method relative to some of the existing classification methods shall be equally assessed using such simulated data.

The procedure we employed for simulating microarray data set here follows the method adopted, with little modifications, by Bura & Pfeiffer (2003), and Molinaro *et al* (2005) both of which were adaptations of the earlier approaches used by Cook & Lee (1999) and Kepler *et al* (2002). We simulated $n = 100$ observations representing the number of mRNA samples with two distinct biological groups $Y = 0$ (normal patients) and $Y = 1$ (diseased/tumour patients). On each observation, 1000 covariates, $\boldsymbol{X} = (X_1, \dots, X_{1000})'$, representing the observed gene expression profiles were simulated. Each biological group 0 or 1 has 50 observations which we denoted as n_0 for group 0 and n_1 for group 1 with $n_0 + n_1 = n$. The data sets $\boldsymbol{X}|\, Y = 0$ were simulated from multivariate normal distribution with mean μ_0, $\mu_0 \neq 0$ and variance-covariance matrix Σ. That is $[(X_1, \dots, X_{100})'|Y = 0] \sim N(\mu_0, \Sigma)$. Of 1000 genes simulated on group 1 subjects, 5 of them were simulated from the mixture of two multivariate normal densities with the same covariance matrix Σ, and means μ_{11} and μ_{12} respectively, $\mu_{11} \neq \mu_{12}$ and $\mu_{11}, \mu_{12} > \mu_0$. That is, $[(X_1, \dots, X_5)'|Y = 1] \sim [\pi * N(\mu_{11}, \Sigma) + (1 - \pi) * N(\mu_{12}, \Sigma)]$ with the estimate of the mixing parameter π taken to be 0.5. The remaining 995 genes for group 1 were simulated from $N(\mu_0, \Sigma)$ distribution as those in group 0. The 5 genes simulated from multivariate mixture models represent those genes that are differentially expressed. They are the genes whose expression levels are believed to be strongly related to the tumour group. The remaining genes that were simulated from $N(\mu_0, \Sigma)$ densities constitute the genes with relatively low expression levels, but not necessarily zero, only that their expression levels are not as strong as those in the former group. The covariance matrix Σ defined as $\Sigma = \{\sigma_{ij}\}$, has a block structure such that

$$\sigma_{\mathrm{ij}} = \begin{cases} 0.2, & if\ |j-i| \le 5 \\ 0, & otherwise \end{cases} \qquad (3.1.1)$$

The variance-covariance formulation in (3.1.1) is to allow for some level of correlations among the simulated genes, typifying a real gene expression data sets.

The whole data set we simulated is of dimension $n \times q$ (100×1000), $n < q$, as usually the case with microarray data. This is the data we have used to test-run our proposed k-SS method and the data was use for further analysis at various stages in this thesis.

In what follows, we provided the distribution of the test statistics used for the construction of our sequential test procedure.

3.2 Determining the critical values C_α^s of the k-SS tests

As established in Chapter 2, the $\hat{\delta}_{j^s}$, $j = 1, \dots, q-1$, $s = 1{,}2$, are the differences of two minimum average MERs between any successive pairs of selection steps j and $j+1$ in the construction of our sequential test procedures. However, the estimates of the critical values C_α^1 and C_α^2 simply written as C_α^s, for $s = 1{,}2$, as required by our test procedures in (2.4.34), (2.4.35) and (2.4.50), (2.4.51) respectively depend on the theoretical distribution of the test statistic $\hat{\delta}_{j^s}$ or $Z_{\hat{\delta}_{j^s}} = \frac{\hat{\delta}_{j^s} - E\left(\hat{\delta}_{j^s}\right)}{\sqrt{\sigma^2\left(\hat{\delta}_{j^s}\right)}}$ designed for the tests. Based on the methodologies adopted for the construction of our k-SS procedure, we highly suspected that neither of the test statistics $\hat{\delta}_{j^s}$ or $Z_{\hat{\delta}_{j^s}}$ may be fitted by the Gaussian distribution as earlier assumed. Therefore, to determine the true distribution of $\hat{\delta}_{j^s}$, we developed a set of algorithms to simulate the $\hat{\delta}_{j^s} = \pm\left(\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}\right)$ estimates for $s = 1$ or 2 respectively according to our proposed k

sequential selection and prediction procedures. The simulated microarray data matrix of 100 samples by 1,000 genes according to the scheme presented in Section 3.1 is used for simulating the $\hat{\delta}_{j^s}$ values.

The values of the two average MER differences $\hat{\delta}_{j^s} = \pm\left(\hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}\right)$, for $s = 1$ or 2, [$\hat{\delta}_{j^1} = \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ for $s = 1$, $\hat{\delta}_{j^2} = \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\bar{\vartheta}}^{m_1,m_2,\dots,m_j}$ for s =2] were simulated at 50, 100, 200, 500 and 1000 sample sizes according to our k-SS procedures. The `R` code we developed for simulating the $\hat{\delta}_{j^s}$ values is presented in Appendix B.4.

To confirm our suspicion that the Gaussian density might not be suitable to fit the $\hat{\delta}_{j^s}$ values, we compared the empirical distribution (red) of 1000 simulated $\hat{\delta}_{j^1}$'s (for $s = 1$) with the theoretical density function of the normal distribution (blue) (see *Fig 3.1*). The *maximum likelihood estimates* of the two parameters μ and σ^2 of the normal distribution (estimated from the simulated $\hat{\delta}_{j^1}$ data) are computed to be $\hat{\mu} = 0.0069$ and $\hat{\sigma}^2 = 0.0002$. The histogram (green) of the raw $\hat{\delta}_{j^1}$ data is equally presented in *Fig 3.1*. From the results displayed in *Fig 3.1*, it is obvious that the true distribution of the $\hat{\delta}_{j^s}$ is not Gaussian as earlier assumed. This is clearly evident from the deviation of the theoretical Gaussian density function (blue) from the empirical distribution (red) of the $\hat{\delta}_{j^1}$ data in *Fig 3.1*. This lack of Gaussian fit is equally revealed by the quantile-quantile (Q-Q) plot of the simulated $\hat{\delta}_{j^1}$ data as provided again in *Fig 3.1*.

More specifically, the empirical distribution of the $\hat{\delta}_{j^1}$ data obviously suggested a typically skewed distribution for the $\hat{\delta}_{j^s}$'s in contrast to the symmetry property that characterize a typical Gaussian distribution. For clarity purposes however, we presented in *Fig 3.2*, the empirical distributions (*histograms and line graphs*) of the $\hat{\delta}_{j^{s}}$ using the simulated 1,000 $\hat{\delta}_{j^{s}}$ data for $s = 1$ and 2. It can be easily observed from the two plots in *Fig 3.2* that the empirical distribution of the $\hat{\delta}_{j^{1}}$ data (left) is positively skewed while that of $\hat{\delta}_{j^{2}}$ data (right), though similar to that of $\hat{\delta}_{j^{1}}$, is negatively skewed.

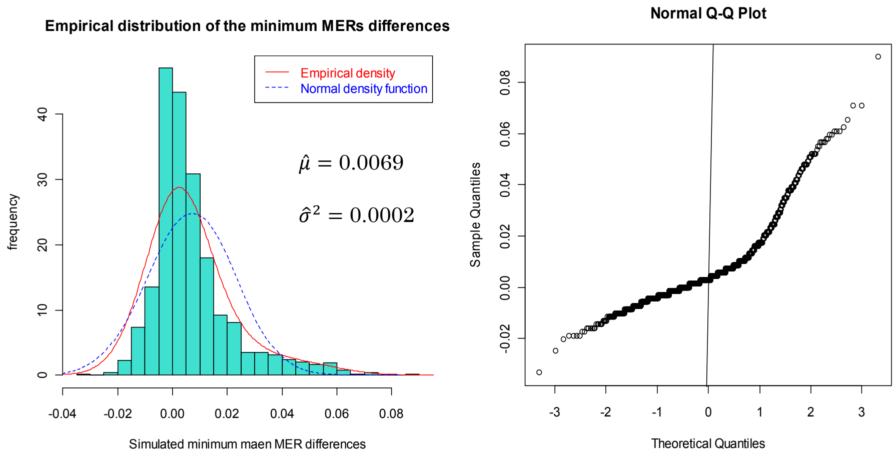


Fig 3.1: The plots in the left present the empirical (red) and the theoretical Normal [N(0.0069,0.0002)] (blue) distributions fitted to the simulated $\hat{\delta}_{j^{1}} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\vartheta}^{m_1,m_2,\ldots,m_{j+1}}$ (the differences of minimum mean MERs) data at 1,000 sample size. The parameters of the Normal distribution are obtained by Maximum Likelihood Estimation (MLE) using the simulated $\hat{\delta}_{j^{1}}$ data. The Q-Q plot (right) clearly indicated lack-of-fit of normal density to the $\hat{\delta}_{j^{1}}$ data.

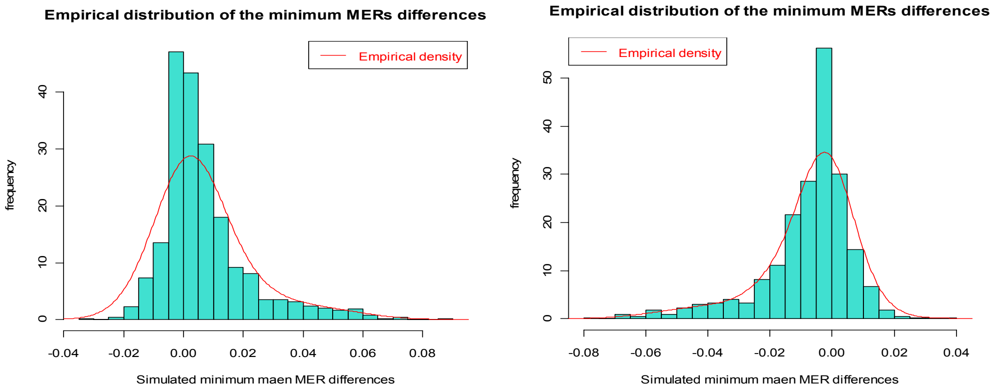


Fig 3.2: The empirical distributions of the simulated 1000 $\hat{\delta}_{j^{1}} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\vartheta}^{m_1,m_2,\ldots,m_{j+1}}$ for s = 1 (left) and $\hat{\delta}_{j^{2}} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}} - \hat{\vartheta}^{m_1,m_2,\ldots,m_j}$ for s = 2 (right)(differences of minimum mean MERs) data.

After several considerations given to some of the common probability distribution functions like Gamma, Exponential, lognormal, Weibull or Beta as well as the Skew-Laplace distribution as used by Fieller & Flenley (1992) for the distribution of particle size to fit $\hat{\delta}_{j^s}$ data, our simulation studies finally revealed that the true distribution of the $\hat{\delta}_{j^s}$ data, $s = 1$ or 2, belong to the *Skew-Normal* parametric class of density functions originally due to Azzalini (1985).

The Skew-Normal (*SN*) densities were developed to capture the continuous variations from normality to non-normality. It is a density function for normal-like data but with lack of symmetry. In what follows, we present the basic theoretical formulations of this distribution and its relevance to our situation under study.

Let $\phi(z)$ be the standard normal density function of random variable Z defined by $\phi(z) = exp(-z^2/2)/\sqrt{2\pi}$ and $\Phi(\lambda z)$ be its distribution function but evaluated at λz. Thus, it is obvious that $\Phi(\lambda z) = \int_{-\infty}^{\lambda z} \phi(t)dt$. If another density function is defined by $\phi(z;\lambda) = 2\phi(z)\Phi(\lambda z)$, then, under this new formulation, random variable Z is said to have a *skew-normal* (*SN*) density with parameter λ, Azzalini (1985,1986). Thus, we have;

$$\phi(z;\lambda) = \frac{2}{\sqrt{2\pi}} exp(-z^2/2) \int_{-\infty}^{\lambda z} \phi(t)dt \qquad (3.2.1)$$

That is, $Z \sim SN(\lambda)$ and in line with the usual $N(0,1)$ notation used to denote the standard normal variable Z, the (standard) skew-normal variate Z with shape parameter λ can be equally written as $Z \sim SN(0,1,\lambda)$ which literally translates to a skew-normal random variable Z with *location parameter* = 0, *scale parameter* = 1 and *shape parameter* = λ. The value of λ determines the shape of the density function $\phi(z;\lambda)$. As the value of λ increases, the skewness of the function also increases and positive values of λ provide positive skewness and vice-versa.

From (3.2.1), the cumulative distribution function (cdf) of $\phi(z;\lambda)$ can be obtained as

$$\Phi(z;\lambda) = 2\int_{-\infty}^{z}\int_{-\infty}^{\lambda z}\phi(v)\phi(u)\,d(v)d(u) \qquad (3.2.2)$$

The histograms and density plots of 10^4 samples drawn from $SN(\lambda)$ family in (3.2.1) at $\lambda = 5$ and -5 are presented in *Fig 3.3*.

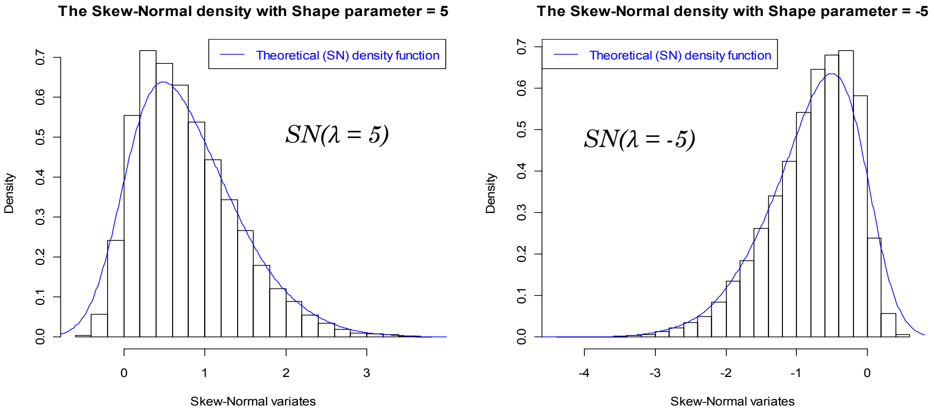


Fig 3.3: The histograms and density plots of 10,000 samples simulated from the Skew-Normal density SN(λ) with shape parameters λ = 5 (left) and λ = -5 (right).

The SN density in (3.2.1) enjoys similar properties of the normal distribution except for symmetry. However, if $\lambda = 0$, it is obvious from (3.2.1) that $\phi(z;0) = \phi(z)$, the standard normal density. For any quantity ξ defined as $\xi = \lambda/\sqrt{1+\lambda^2}$ therefore, both the mean and variance of Z are respectively given as $E_{SN}(Z) = \sqrt{2/\pi}\xi$ and $V_{SN}(Z) = 1 - 2\xi^2/\pi$, Azzalini (1985). Further details on the distributional properties of $\phi(z;\lambda)$ could be found in Azzalini (1985, 1986, 2001, 2005, 2006), Azzalini & Capitanio (1999) and Azzalini *et al* (2003). After the original work of Azzalini and his co-workers on the development of the skew-normal class of distributions, several other variants and modifications of the SN probability functions have been developed, see, for example, Gupta *et al* (2004), Arellano-Valle *et al* (2004), Armando *et al* (2007) among others.

Now, if we consider a transformation on *SN* variate Z of the form

$$\hat{\delta}_{j^s} = \mu_s + \sigma_s Z \qquad (3.2.3)$$

then, random variable $\hat{\delta}_{j^s}$ has a *SN* distribution with location and scale parameters μ_s and σ_s (different from 0 and 1) respectively, and shape parameter still remain λ.

From (3.2.3), it is easy to verify that,

$$E(\hat{\delta}_{j^s}) = \mu_s + \lambda\sigma_s\sqrt{2}\,/\sqrt{\pi(1+\lambda^2)} \qquad (3.2.4)$$

and that

$$V(\hat{\delta}_{j^s}) = \sigma_s^2[1 - 2\lambda^2/\pi(1+\lambda^2)] \qquad (3.2.5)$$

Thus, the distribution of random variable $\hat{\delta}_{j^s}$ can be written as $\phi(\hat{\delta}_{j^s};\mu_s,\sigma_s,\lambda)$, a skew-normal density with location parameter = μ_s, scale parameter = σ_s and shape parameter = λ or as $\hat{\delta}_{j^s}{\sim}SN(\mu_s,\sigma_s^2,\lambda)$. If $\lambda = 0$, it is obvious again from (3.2.4) and (3.2.5) that $E(\hat{\delta}_{j^s}) = \mu_s$ and $V(\hat{\delta}_{j^s}) = \sigma_s^2$ and variable $\hat{\delta}_{j^s}$ would become a (*symmetric*) normal random variable, i.e. $\hat{\delta}_{j^s}{\sim}N(\mu_s,\sigma_s^2)$. On the other hand, when $\mu_s = 0$ and $\sigma_s = 1$, it follows from (3.2.3) to (3.2.5) that $\hat{\delta}_{j^s} = Z{\sim}SN(\lambda)$. Hence, it follows that given any skew-normal variate $\hat{\delta}_{j^s}$ with specified location, scale and shape parameters μ_s, σ_s, λ, respectively, the statistic $Z_{\hat{\delta}_{j^s}} = \frac{\hat{\delta}_{j^s} - E(\hat{\delta}_{j^s})}{\sqrt{V(\hat{\delta}_{j^s})}}$ would have a skew-normal distribution with location, scale and shape parameters 0, 1, and λ respectively simply written as $Z_{\hat{\delta}_{j^s}}{\sim}SN(\lambda)$ as defined in (3.2.1).

Obviously, the statistic $Z_{\hat{\delta}_{j^s}} = \frac{\hat{\delta}_{j^s} - E\left(\hat{\delta}_{j^s}\right)}{\sqrt{V\left(\hat{\delta}_{j^s}\right)}}$ is in the form of the test statistics (2.4.33) to (2.4.35) constructed for our sequential hypothesis tests of (2.4.32) and (2.4.49) with $\hat{\delta}_{j^s} = \pm\left(\hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}\right)$, for $s =$ 1 or 2 respectively (i.e. $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}$ for $s = 1$ and $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}$ for $s = 2$). Hence, to determine the critical values C_α^s, $s =$ 1 or 2, of the test statistics $\hat{\delta}_{j^s}$ as used in (2.4.36) for $s = 1$ and (2.4.50) for $s = 2$, it is sufficient to establish that random variable $\hat{\delta}_{j^s}$ has a skew-normal distribution with location, scale and shape parameters μ_s, σ_s and λ respectively [i.e. $\hat{\delta}_{j^s} \sim SN(\mu_s, \sigma_s^2, \lambda)$] or equivalently that the standardized variate $Z_{\hat{\delta}_{j^s}}$ has a (standard) skew-normal density function with shape parameter λ [i.e. $Z_{\hat{\delta}_{j^s}} \sim SN(\lambda)$] as earlier stated.

Following our simulation procedures, it is quite easy to establish that random variable $\hat{\delta}_{j^s}$ actually follows the skew-normal distribution. Firstly, we fitted the skew-normal density $\phi\left(\hat{\delta}_{j^s}; \mu_s, \sigma_s, \lambda\right)$ to the simulated $\hat{\delta}_{j^s}$, s =1, 2, data at 50, 100, 200, 500 and 1000 sample sizes. The *maximum likelihood estimates* (MLE) of parameters μ_s, σ_s^2, λ, of each of the five fitted skew-normal densities were estimated using *expectation-maximization* (EM) algorithm. Thereafter, random sample of size 10,000 were drawn from each of the fitted *SN* densities. The empirical distributions (*histograms and line graphs*) of the $\hat{\delta}_{j^s}$ data (under all the five samples) are plotted based on the 10,000 samples drawn. These are respectively compared with the theoretical (*skew-normal*) densities using the estimated parameters. Due to space consideration, we only present in *Fig 3.4*, the empirical (red) and theoretical (blue) density plots as well as the respective histograms (green) of the $\hat{\delta}_{j^s}$ data, for $s = 2$, at all the five chosen sample sizes. The quantile-quantile (Q-Q) plot of each of the simulated $\hat{\delta}_{j^s}$ data sets is equally presented in *Fig 3.4*. From the various density plots, the closeness of both the empirical (observed) and theoretical (*skew-normal*) distributions can be easily observed, therefore, confirming the fitness of the skew-normal density to the $\hat{\delta}_{j^s}$ data. This result is corroborated by the respective Q-Q plots as displayed in *Fig 3.4*.

Furthermore, among the popular statistical test procedures that are commonly adopted to establish whether or not a set of data comes from a specified theoretical distribution are the Kolmogorov-Smirnov test (Chakravart *et al*, 1967), Anderson-Darling goodness-of-fit test (Stephens,1974) and the Chi-square goodness-of-fit test (Snedecor & Cochran, 1989) among others. While the approach of Kolmogorov-Smirnov test has been reported to be highly sensitive at rejecting that a data comes from a given theoretical distribution even when it does, (http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm), the method of Anderson-Darling goodness-of-fit test only exists for a very few distribution which does not include the skew-normal density to the best of our knowledge. Therefore, in addition to the probability density function (pdf) and the Q-Q plots presented in *Fig 3.4*, we equally constructed the Chi-square goodness-of-fit test to determine the fitness of the Skew-Normal density to the simulated $\hat{\delta}_{j^s}$ data. The results from the Chi-square test for both $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ data are presented in *Tables 3.1a & 3.1b* respectively. All the results clearly confirmed the appropriateness of the skew-normal distribution to fit the $\hat{\delta}_{j^s}$ data.

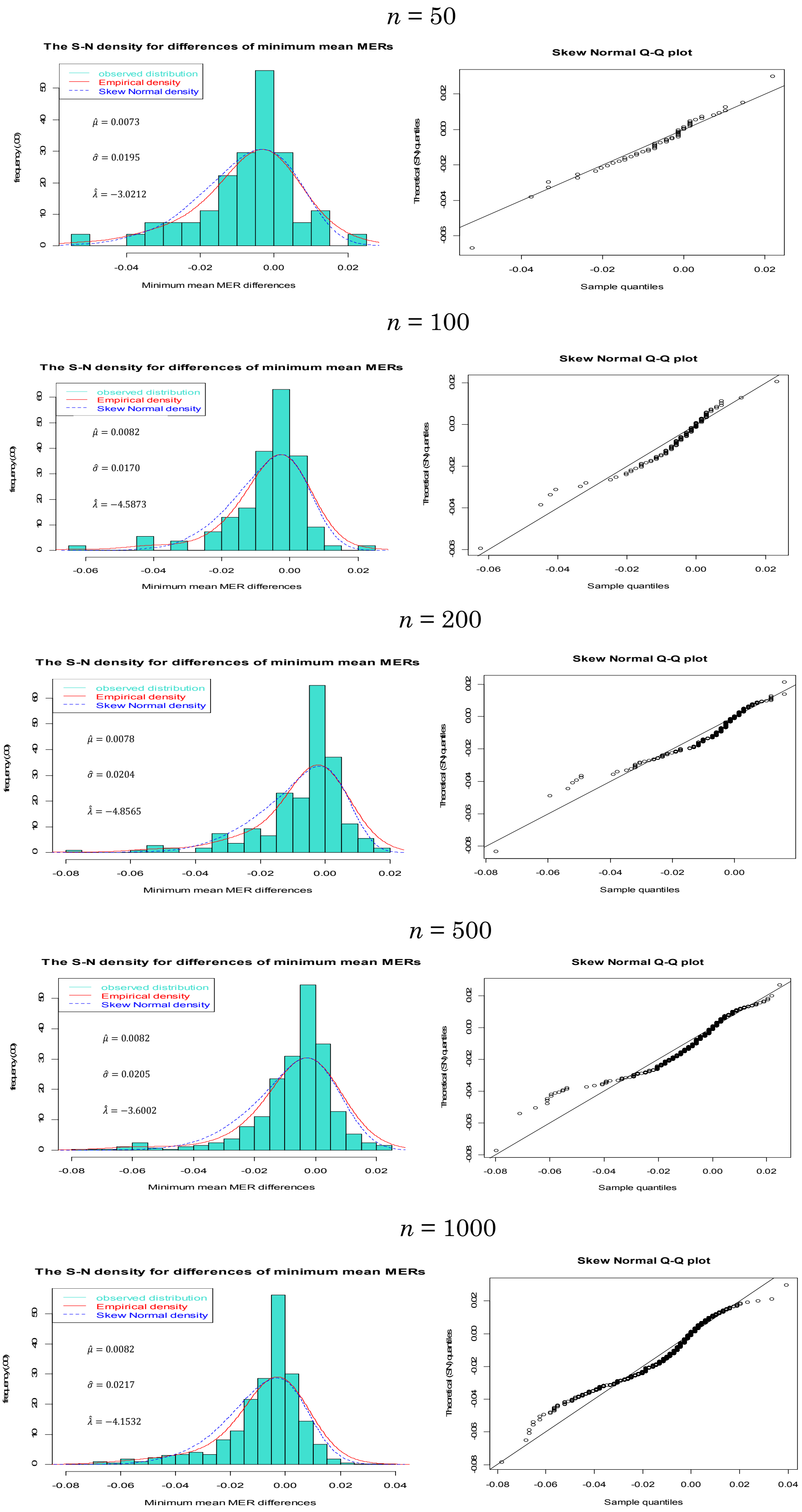


Fig 3.4: The plots in the left showed the empirical (red) and the theoretical (Skew-Normal, blue) distributions of the $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ data at the chosen five sample sizes of 50, 100, 200, 500, and 1000. The estimates of location, scale, and shape parameters μ, σ and λ of the skew-normal densities are indicated for each plot. The corresponding Q-Q plots (right) for each sample are also presented.

No. of $\hat{\boldsymbol{\delta}}_{j^1}$'s ($n$) simulated	Estimated parameters of Skew-Normal density fitted to $\hat{\boldsymbol{\delta}}_{j^2}$ data			*Chi-square goodness-of-fit test*	
	Location parameter	Scale parameter	Shape parameter	Critical values	p-values
50	-0.0064	0.0180	3.9363	0.1445	1.0000
100	-0.0072	0.0200	4.9813	1.6946	0.9890
200	-0.0079	0.0197	3.1463	0.1150	0.9998
500	-0.0084	0.0216	4.0917	0.0665	1.0000
1000	-0.0086	0.0212	4.0244	0.1183	1.0000
Average	-0.0077	0.0201	4.0360		

Table 3.1a: The Chi-square goodness-of-fit test to establish the fitness of the simulated $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ data to the Skew-Normal distribution. The Chi-square estimates and the corresponding p-values are respectively shown in the last two columns of the table. The parameter estimates of the fitted SN densities presented are computed using 10,000 random samples drawn from the fitted SN distributions for each respective simulated $\hat{\delta}_{j^1}$ data. All results indicated that the Skew-Normal density fits the $\hat{\delta}_{j^1}$ data.

No. of $\hat{\boldsymbol{\delta}}_{j^2}$'s ($n$) simulated	Estimated parameters of Skew-Normal density fitted to $\hat{\boldsymbol{\delta}}_{j^2}$ data			*Chi-square goodness-of-fit test*	
	Location parameter	Scale parameter	Shape parameter	Critical values	p-values
50	0.0073	0.0195	-3.0212	0.1907	0.9999
100	0.0082	0.0170	-4.5873	0.1278	0.9980
200	0.0078	0.0204	-4.8565	0.2692	1.0000
500	0.0082	0.0205	-3.6002	0.1442	1.0000
1000	0.0082	0.0217	-4.1532	0.1380	1.0000
Average	0.0075	0.0198	-4.0437		

Table 3.1b: The Chi-square goodness-of-fit test to establish the fitness of the simulated $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ data to the Skew-Normal distribution. The Chi-square estimates and the corresponding p-values are respectively shown in the last two columns of the table. The parameter estimates of the fitted SN densities presented are computed using 10,000 random samples drawn from the fitted SN distributions for each respective simulated $\hat{\delta}_{j^2}$ data. All results indicated that the Skew-Normal density fits the $\hat{\delta}_{j^2}$ data.

The family of the skew-normal density functions is implemented in the `sn` library of `R` statistical package. We have employed this to fit the skew-normal distribution to all the simulated $\hat{\delta}_{j^s}$ data sets.

It can be observed from *Tables 3.1a & b* that, except for the sign differences in both location and shape parameters, all the estimated parameters of the skew-normal densities for both $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ variates are essentially similar at each of the selected sample sizes. These are clearly shown by the respective density plots in *Fig 3.2*. While $\hat{\delta}_{j^1}$ has more positive values than negatives and is positively skewed, $\hat{\delta}_{j^2}$ has more negative values than the positives and is negatively skewed. More justifications are provided by the Box-and-Whiskers plots of the simulated $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ values at the five selected sample sizes as shown in *Fig 3.5a* and *Fig 3.5b* respectively. This is more conspicuously presented by the box-plot of the $\hat{\delta}_{j^s}$ data, $s = 1{,}2$, at 1000 sample size as shown in *Fig 3.5c*. Except for their sign differences due to skewness as indicated in all the plots, the two $\hat{\delta}_{j^s}$ data have similar distribution patterns but in the opposite sense.

Fig 3.5a

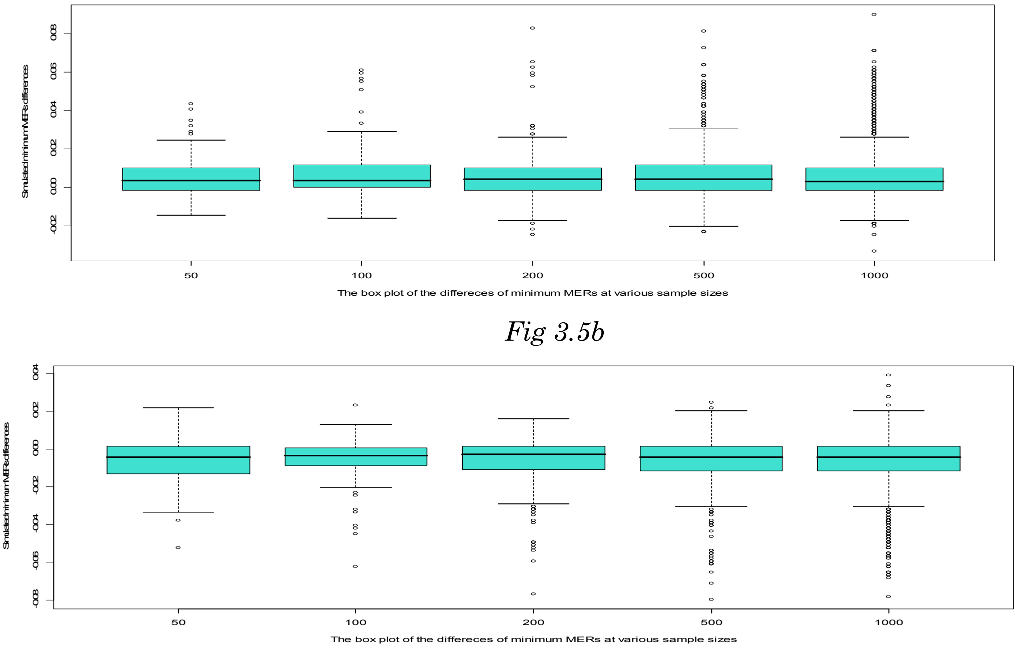


Fig 3.5b

Fig 3.5 a &b: The box-plot of the simulated minimum average MER differences, $\hat{\delta}_{j^1}$ (a) and $\hat{\delta}_{j^2}$ (b) data at the selected five sample sizes.

As previously discussed in Chapter 2 under the two $\hat{\delta}_{j^s}$ k-SS formulations, $s = 1, 2$, the strict inequality $\hat{\vartheta}^{m_1,m_2,\dots,m_{j+1}} < \hat{\vartheta}^{m_1,m_2,\dots,m_j}$ shall be observed as long as the selection of additional gene continues to improve the prediction accuracy of the current models. This will continue to yield positive $\hat{\delta}_{j^1}$ values (or negative $\hat{\delta}_{j^2}$ values) at each successive selection steps until no further improvement is brought into the model despite the inclusion of additional gene. At such selection levels, the condition that $\hat{\delta}_{j^1} \leq 0$ (or $\hat{\delta}_{j^2} \geq 0$) shall hold.

These are the pictures displayed in *Fig 3.2* for the empirical distributions of both $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ simulated data sets.

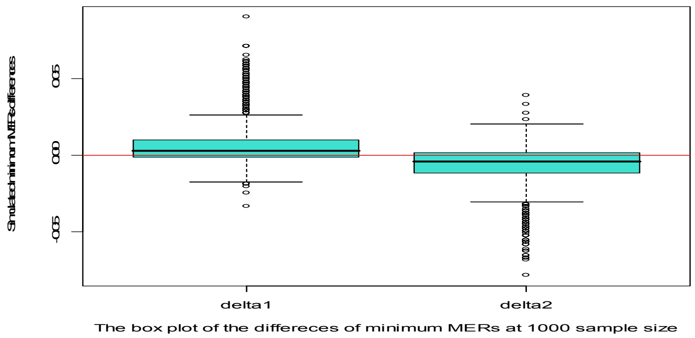


Fig 3.5c: The box-plot of the simulated minimum average MER differences for both $\hat{\delta}_{j^1}$ (delta1) and $\hat{\delta}_{j^2}$ (delta2) at 1000 sample size showing the effects of skewness under the two formulations.

Now that it has been established that the $\hat{\delta}_{j^s}$ data have the skew-normal distribution, it is therefore obvious that the test statistics $Z_{\hat{\delta}_{j^s}} = \frac{\hat{\delta}_{j^s} - E\left(\hat{\delta}_{j^s}\right)}{\sqrt{V\left(\hat{\delta}_{j^s}\right)}}$, $s = 1,\ 2,\ j = 1, \ldots, q-1$, as stated for testing one directional hypotheses sets (2.4.32) and (2.4.49) are also distributed skew-normal. To compute the critical values C_{α}^{s} therefore, we only need to determine the shape parameters of the skew-normal densities $\phi(\hat{\delta}_{j^1}; \lambda_1)$ and $\phi(\hat{\delta}_{j^2}; \lambda_2)$ or simply that of $\phi\left(Z_{\hat{\delta}_{j^1}}; \lambda_1\right)$ and $\phi\left(Z_{\hat{\delta}_{j^2}}; \lambda_2\right)$. We recall that the skewness of the two *SN* densities $\phi\left(Z_{\hat{\delta}_{j^1}}; \lambda_1\right)$ and $\phi\left(Z_{\hat{\delta}_{j^2}}; \lambda_2\right)$ are different only by their signs, such that when $\hat{\delta}_{j^1}$ is positively skewed by λ_1 magnitude $\hat{\delta}_{j^2}$ would be negatively skewed by λ_2 magnitude with $\lambda_2 = -\lambda_1$. Therefore, if the random variable $Z_{\hat{\delta}_{j^1}}$ is distributed skew-normal with shape parameter λ_1 i.e. $Z_{\hat{\delta}_{j^1}} \sim SN(\lambda_1)$, it can be easily shown (Azzalini, 1985, pp172) that random variable $Z_{\hat{\delta}_{j^2}}$ would be distributed skew-normal with shape parameter λ_2 i.e. $[Z_{\hat{\delta}_{j^2}} \sim SN(\lambda_2)]$, $\lambda_2 = -\lambda_1$. From this relationship, another basic property of SN family of distributions as adapted here, using (3.2.2) equally holds that

$$\Phi\left(Z_{\hat{\delta}_{j^1}};\lambda_1\right)=1-\Phi\left(Z_{\hat{\delta}_{j^2}};\lambda_2\right)\leftrightarrow\Phi\left(Z_{\hat{\delta}_{j^1}};\lambda_2\right)=1-\Phi\left(Z_{\hat{\delta}_{j^2}};\lambda_1\right)\quad(3.2.6)$$

Azzalini (1985, pp174), where the absolute value of the 'joint' shape parameter $|\lambda_s|$, $s=1,2$, that satisfies (3.2.6) is to be determined.

To this end therefore, we shall let the joint estimate of the absolute value of the shape parameter for both $\phi(\hat{\delta}_{j^1};\lambda_1)$ and $\phi(\hat{\delta}_{j^2};\lambda_2)$ skew-normal densities be denoted by $\widehat{\lambda^*}$. This can be determined by taken the average of the absolute values of all the estimated shape parameters of the skew-normal densities $\phi(\hat{\delta}_{j^1};\lambda_{1m})$ and $\phi(\hat{\delta}_{j^2};\lambda_{2m})$ fitted for simulated $\hat{\delta}_{j^s}$ data sets, $s=1,2$, at m chosen number of sample sizes, $m=1,\dots,M$. Thus, $\widehat{\lambda^*}$ is obtained by

$$\widehat{\lambda^*}=\frac{1}{2M}\left(\sum_{m=1}^{M}\left|\hat{\lambda}_{1m}\right|+\sum_{m=1}^{M}\left|\hat{\lambda}_{2m}\right|\right)\qquad(3.2.7)$$

Based on the results of our simulations, the estimates of each of the λ_{sm}, $s=1,2$, $m=1,\dots,5$, are provided in *Tables 3.1a & 3.1b* for simulated $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ data sets respectively. From these results, the value of $\widehat{\lambda^*}$ is estimated to be $\widehat{\lambda^*}=\mathbf{4.0398}$ using (3.2.7). Henceforth, this value of $\widehat{\lambda^*}$ shall be used as the true value of parameter λ^* of the skew-normal densities $\phi\left(Z_{\hat{\delta}_{j^1}};\lambda^*\right)$ and $\phi\left(Z_{\hat{\delta}_{j^2}};\lambda^*\right)$ for the critical values $C_\alpha^1=Z_{1-\alpha}(\widehat{\lambda^*})$ and $C_\alpha^2=Z_{1-\alpha}(-\widehat{\lambda^*})$ of our k-SS test procedures (2.4.32) and (2.4.49) respectively at any given value of α.

Therefore, for testing the hypothesis sets (2.3.32) and (2.4.49) the respective test statistics $Z_{\hat{\delta}_{j^1}}=\frac{\hat{\delta}_{j^1}-E\left(\hat{\delta}_{j^1}\right)}{\sqrt{V\left(\hat{\delta}_{j^1}\right)}}$ and $Z_{\hat{\delta}_{j^2}}=\frac{\hat{\delta}_{j^2}-E\left(\hat{\delta}_{j^2}\right)}{\sqrt{V\left(\hat{\delta}_{j^2}\right)}}$ have the skew-normal distributions with shape parameters λ^* and $-\lambda^*$

respectively with λ^* estimated as $\widehat{\lambda^*} = \mathbf{4.0398}$ and $E(\hat{\delta}_{j^s}) = 0$ under H_{0sj}, $s = 1,2$, $j = 1, \dots, q-1$. From the distributions of the two test statistics $Z_{\hat{\delta}_{j^s}}$ given above, one can easily determine the estimates of their critical values C_α^s for our k-SS test procedures in (2.4.32) and (2.4.49) for $s =$ 1 and 2 respectively. These are presented in what follows.

Let us consider one directional hypothesis set given in (2.4.32), i.e. $H_{01j}\colon \delta_{j^1} \leq 0$ vs. $H_{a1j}: \delta_{j^1} > 0$, for $j = 1, \dots q-1$. Since the test statistic $Z_{\hat{\delta}_{j^1}}$ as used in (2.4.33) for this test is distributed skew-normal, $Z_{\hat{\delta}_{j^1}} \sim SN(\lambda^*)$, then, at any significance level α (to be determined by cross-validation), the critical values C_α^1 for this test, as used in (2.4.34) through (2.4.37), shall be estimated by

$$C_\alpha^1 = Z_{1-\alpha}(\widehat{\lambda^*}) \tag{3.2.8}$$

where $Z_{1-\alpha}(\widehat{\lambda^*})$ is the quantile of the skew-normal distribution $\phi\left(Z_{\hat{\delta}_{j^1}}; \lambda^*\right)$ with shape parameter λ^* computed at significance level α.

Similarly, under the one directional hypothesis set in (2.4.49), i.e. $H_{02j}\colon \hat{\delta}_{j^2} \geq 0$ vs. $H_{a2j}: \hat{\delta}_{j^2} < 0$, each of the test statistic $Z_{\hat{\delta}_{j^2}}$ for the test is equally distributed skew-normal, $Z_{\hat{\delta}_{j^2}} \sim SN(-\lambda^*)$, and at any given significance level α, the critical values C_α^2 for this test, as defined in (2.4.50) and (2.4.50), shall be estimated by

$$C_\alpha^2 = Z_{1-\alpha}(-\widehat{\lambda^*}) \tag{3.2.9}$$

where $Z_{1-\alpha}(-\widehat{\lambda^*})$ is the quantile of the skew-normal density $\phi\left(Z_{\hat{\delta}_{j^2}}; \lambda^*\right)$ with shape parameter $-\lambda^*$ at significance level α also to be determined by cross-validation.

Having determined the theoretical distributions of $Z_{\hat{\delta}_{j^s}}$ or $\hat{\delta}_{j^s}$, $s = 1,2$, we then present in what follows, the complete form of our k-SS algorithm. However, it is to be noted that the implementation of either of two k-SS test procedures in (2.4.32) or (2.4.49) on a given microarray data set would essentially yield similar results.

The k-SS algorithm

Input: Training samples n_{TR} and test samples n_{TE} of n biological subjects with binary response group $Y \in \{0,1\}$ and q-dimensional vector $\boldsymbol{X} = (X_1, \dots, X_q)^T$ of genes whose expression levels are measured on all the n samples, $n = n_{TR} + n_{TE}$.
Out-put: The k-SS classifiers and various performance indices.

Step 0-0: #*Search for the first best gene to be selected into the classification model among all the q genes.*

i) Fit *logit* model, $logit(\pi(X_j)) = \alpha + \beta_j X_j$, $j = 1, \dots, q$, on individual gene X_j using the training sample n_{TR}.

ii) Construct the classifiers $\varphi(X_j) = argmax_y\, \hat{p}(y|X_j)$ for each gene X_j, $j = 1, \dots, q$, and predict the two class labels (0,1) of the test sample n_{TE} via the classification scheme;

$$\hat{\varphi}_i(X_j) = \begin{cases} 1, & \text{if } \hat{p}_i(1|X_j) \geq 0.5 \\ 0, & \text{if } \hat{p}_i(0|X_j) < 0.5 \end{cases}$$

iii) Base on ii) above, compute the *misclassification error rates* (MERs), $\hat{\vartheta}_j = \frac{1}{n_{TE}} \sum_{i=1}^{n_{TE}} \left[I_{\{\hat{\varphi}_i(X_j) \neq Y_i\}} \right]$, $0 \leq \hat{\vartheta}_j \leq 1.$ $j = 1, \dots, q$, for each X_j, where $I_{\{.\}} = 1$ if the argument is true and 0 otherwise.

iv) Draw R replicates of training sample n_{TR} randomly, without replacement, from the original n sample and repeat steps i) to iii) on each sub-sample for each gene X_j and compute the average MERs

$$\hat{\hat{\vartheta}}_j = \frac{1}{R \times n_{TE}} \sum_{i=1}^{n_{TE}} \sum_{r=1}^{R} \left[I_{\{\hat{\varphi}_{ir}(X_j) \neq Y_{ir}\}} \right], j = 1, \dots, q.$$

v) Define the minimum average MER from iv) by $\hat{\hat{\vartheta}}^{m_1} = \hat{\hat{\vartheta}}_{(1)} = min\left(\hat{\hat{\vartheta}}_{(1)}, \hat{\hat{\vartheta}}_{(2)}, \dots, \hat{\hat{\vartheta}}_{(q)}\right)$ and select the corresponding gene $X^{m_1} = X_{(1)} \in \{X_{(1)}, X_{(2)}, \dots, X_{(q)}\}$ as the first gene candidate into our classification model.

Step 1-0: #*Search for the next best gene to be included with gene X^{m_1} in the model*

i) For the remaining $q - 1$ genes, construct classification rules as in **Step 0-0** i) to v) above but using each gene pair $X^{m_1}X_{(2)}$, …, $X^{m_1}X_{(q)}$. Obtain the minimum average MERs defined as

$$\hat{\hat{\vartheta}}^{m_1 m_2} = min\left(\hat{\hat{\vartheta}}^{m_{1(2)}}, \hat{\hat{\vartheta}}^{m_{1(3)}}, \dots, \hat{\hat{\vartheta}}^{m_{1(q)}}\right)$$

which is provided by the corresponding gene pair
$X^{m_1}X^{m_2} \in \{X^{m_1}\, X_{(2)}, X^{m_1}\, X_{(3)}, \dots, X^{m_1}\, X_{(q)}\}$.

ii) Select gene X^{m_2} into our classification model which already has gene X^{m_1} to form gene pair $X^{m_1}X^{m_2}$ in the new classification model.

Step 1-1: #*Test for the significance of the gain in prediction accuracy of the current model due to the inclusion of gene X^{m_2}.*

i.) Test one directional hypothesis test of the form:
$H_{011}: \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} \leq 0$ vs. $H_{a11}: \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} > 0$
$\rightarrow H_{011} : \delta_{1^1} \leq 0$ vs. $H_{a11}: \delta_{1^1} > 0$
where $\delta_{1^1} = \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2}$ with its unbiased estimator given by
$\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2}$.

ii.) Use the test statistic, $Z_{\hat{\delta}_{1^1}} = \frac{\widehat{\delta}_{1^1} - E(\widehat{\delta}_{1^1})}{\sqrt{V(\widehat{\delta}_{1^1})}} \sim SN(\lambda^*)$,

$SN(\lambda^*) \rightarrow$ Skew-Normal density with shape parameter λ^*. Under H_{011}, $E(\hat{\delta}_{1^1}) = 0$.

iii.) Construct decision rules (for gene(s) selection(s)):
At some range of significance level α (determined by cross-validation),

a.) accept H_{011} (*reject the selection of gene X^{m_2} into the model*) if

$$\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} \leq Z_{1-\alpha}(4.0398)\sqrt{\sigma^2(\hat{\delta}_{1^1})}$$

b.) reject H_{011} (*accept the selection of gene X^{m_2} into the model*) if

$$\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} > Z_{1-\alpha}(4.0398)\sqrt{\sigma^2(\hat{\delta}_{1^1})}$$

where $Z_{1-\alpha}(4.0398)$ is the quantile of the skew-normal density at the estimated shape parameter $\widehat{\lambda}^* = 4.0398$.

iv.) If the null hypothesis H_{011} is rejected base on decision rule iii.)b.), retain gene X^{m_2} in the model and go back to **Step 1-0** in search of the next best gene to be added to the gene pair $X^{m_1}X^{m_2}$ in the model. If H_{011} is accepted, drop the selected gene X^{m_2} from the model and stop further gene selection.

v.) Execute **Steps 1-0** (i-ii) to **Step 1-1** (i-iv) repeatedly until no more gene satisfies the decision rule iii.)b.) above

vi.) STOP and RETURN the k-sequentially selected (k-SS) informative genes, $k \in \{1, .., q\}$ and various performance indices.

3.3 Applications of *k*-SS method

The new k-SS method proposed here is first applied here on the simulated microarray dataset. The method is later applied on eleven published microarray data sets as presented in Chapters 4 and 5. Details of all the data sets used are provided in the next Chapter.

Since the 100 by 1,000 data matrix we simulated here represents a typical microarray data set, appropriate data normalization and standardization as discussed in Chapter 1 are carried out prior to analysis of the data such that each vector of genes has zero mean and unit standard deviation across the mRNA samples. This is followed by preliminary gene selection using the student-t statistics based on the procedures described in Section 1.4.2. Using the range of the observed p-values from the data as a guide, the cut-point p-value, p^* is taken to be 0.05. The univariate filtering using this student-t method hereby reduced the original $q = 1000$ genes to $q^* = 55$ genes.

We begin the implementation of our k sequential gene selection (k-SS) method by random splitting of the mRNA sample size n using the splitting ratio 19:1 for n_{TR} (training sample) : n_{TE} (test sample) respectively as discussed in Section 2.1. Therefore, with the simulated mRNA sample size $n = 100$, $n_{TR} = 95$ would be used to build our classifier while $n_{TE} = 5$ would be used to evaluate its performance.

Sub-sampling scheme of Monte Carlo Cross-Validation (MCCV) is adopted to ensure stability of results and minimize bias in our estimates. By this, random sample of size $n_{TR} = 95$ is repeatedly drawn from the entire $n = 100$ sample 5000 times without replacement and univariate logit model is fitted on each of the $q^* = 55$ genes using each selected n_{TR} sample. Each of the fitted model is used to predict the response class labels $y \in \{0,1\}$ of the remaining left-out $n_{TE} = 5$ samples from which the misclassification error rates (MERs) are computed. Thereafter, the average MERs $\hat{\bar{\vartheta}}_1, \hat{\bar{\vartheta}}_2, \dots, \hat{\bar{\vartheta}}_{55}$, averaged over the entire 5000 repetitions, are computed. All the 55 genes are then ordered in ascending order of their averaged MER estimates. This resulted into the following genes sequence and their respective average MER estimates (in parenthesis): g5*(0.1737)*, g4*(0.18170)*,…,V879*(0.4850)*, V876*(0.5171).* It should be recalled that the genes labelled g1 to g5 are the 5 simulated genes with up-regulated expression values while genes labelled V6 to V1000 are the 995 simulated genes with moderate gene expressions values according to our simulation procedures as presented in Section 3.1.

Base on the above estimated mean MERs sequence the ordered prediction performance of the genes can be vividly seen. The gene labelled g5 is the gene that provided the best prediction accuracy for having the least mean MER of 0.1737 among the 55 preliminarily selected genes. Hence, gene g5 is the first gene to be selected by our k-SS procedure. This is then followed by searching for the next best gene among the remaining 54 genes to be included in the model with g5. We determined this by fitting the logit model on each of the 54 gene pairs g5g4, … , g5V879, g5V876 and use the fitted model to predict the response category of the test samples. Here again, the mean MER for each prediction is computed and the gene pair that produces the minimum mean MER among the 54 mean MERs is selected for consideration into the model. At this selection step, any of the one directional null hypothesis set of the form $H_{01j}\colon \mu_{\vartheta}^{m_1,m_2,\ldots,m_j} - \mu_{\vartheta}^{m_1,m_2,\ldots,m_{j+1}} \leq 0$ or $H_{02j}\colon \mu_{\vartheta}^{m_1,m_2,\ldots,m_{j+1}} - \mu_{\vartheta}^{m_1,m_2,\ldots,m_j} \geq 0$ as given in (2.4.32) or (2.4.49) respectively with $j = 1$ is to be tested between the two minimum mean MERs obtained at the previous two gene selections.

We shall first consider the use of the hypothesis test (2.4.32) after which the second hypothesis test (2.4.49) shall be considered to illustrate the applications our k-SS method. It shall be finally established thereafter that the k-SS results under the two test formulations are essentially similar.

Using hypothesis test (2.4.32), the test hypothesis required at this gene selection stage is of the form

$$H_{011}\colon \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} \leq 0 \;\; vs.\; H_{a11}\colon \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2} > 0$$

$$\rightarrow \qquad H_{011} : \delta_{1^1} \leq 0 \;\; \text{vs. } H_{a11}\colon \delta_{1^1} > 0 \qquad (3.3.1)$$

where $\delta_{1^1} = \mu_{\vartheta}^{m_1} - \mu_{\vartheta}^{m_1,m_2}$. Based on the decision rules in (2.4.36) and (2.4.37) additional one gene would be selected and added to gene g5 (accepting H_{a11}) if $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} > C_{\alpha}^{1}\sqrt{\sigma^2(\hat{\delta}_{1^1})}$, while the selection of additional one gene would be stopped (accepting H_{011}) if $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} \leq C_{\alpha}^{1}\sqrt{\sigma^2(\hat{\delta}_{1^1})}$ where $C_{\alpha}^{1} = Z_{1-\alpha}(\widehat{\lambda^*})$ is the critical value of the percentage points of the skew-normal distribution as defined in Section 3.2 at some Type I error α to be determined by internal cross-validation.

The value of the shape parameter $\widehat{\lambda^*}$ of the skew-normal density has been estimated to be 4.0398 through simulation studies in the previous section. This shall be used to determine C_{α}^{1} at each selection step. In a nutshell, if the null hypothesis H_{011} is accepted, further variable selection stops, but if the alternative set H_{a11} is accepted, then, additional one gene would be included into the model and the search for the next best gene to be selected begins by repeating the above procedures. The R code we develop to run this test procedure is provided in Appendix B.1.

We would like to reiterate here again that the size α of our k-SS test procedure is not arbitrarily fixed by us but rather, it is being determined through cross-validation. By this, different estimates of the critical values $C_{\alpha}^{1}\sqrt{\sigma^2(\hat{\delta}_{1^1})}$ would be computed over all possible values of α in the interval [0,1] and the value(s) of α at which the decision rule (2.4.36) is satisfied and for which the best prediction results are obtained becomes the size of our test.

Based on these criteria, additional one gene labelled "g3" is selected at step 1 having satisfied the decision rule $\hat{\delta}_{1^1} = \hat{\hat{\vartheta}}^{m_1} - \hat{\hat{\vartheta}}^{m_1,m_2} > C_\alpha^1\sqrt{\sigma^2(\hat{\delta}_{1^1})}$ as given by (2.4.37). Therefore, gene "g3" was added to gene "g5" at step 1 to make gene pair "g5, g3" in the k-SS classification function.

Selection steps *j*	**Min. mean MERs** $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$	**Min. mean MERs** $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$	$\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ $-\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$	**No. of genes selected**	**Decision**
0	0.1831	-	-	1	continue
1	0.1831	0.1132	0.0701	2	✓
2	0.1120	0.0783	0.0337	3	✓
3	0.0787	0.0697	0.0090	4	✓
4	0.0718	0.0602	0.0116	5	✓
5	0.0598	0.0459	0.0139	6	✓
6	**0.0463**	0.0485	-0.0021	×	stop

Table 3.2a: Table of results of k-SS classifier under the $\hat{\delta}_{j^1}$ formulations at each gene selection step for simulated data. Optimal selection (the best prediction result) is achieved at the fifth selection step at which the sixth gene is selected. The size α of the k-SS test, determined by cross-validation, satisfies the range $\alpha \in (0, 0.975]$. The corresponding rage of the critical value $C_\alpha^1\sqrt{\sigma^2(\hat{\delta}_{6^1})}$ of the test statistic $\hat{\delta}_{6^1}$ for this range of α is estimated as $C_\alpha^1\sqrt{\sigma^2(\hat{\delta}_{6^1})} \in (\infty, -1.2081 \times 10^{-4}]$. The six genes selected in order of selection steps 0,1, … ,5 are "g5", "g3", "V192", "V805", "V566", "g2" *respectively.*

At step 2, the decision rule $\hat{\delta}_{2^1} = \hat{\hat{\vartheta}}^{m_1,m_2} - \hat{\hat{\vartheta}}^{m_1,m_2,m_3} > C_\alpha^1\sqrt{\sigma^2(\hat{\delta}_{2^1})}$ was also satisfied with the selection of gene "V192". This was again added to the gene pair "g5, g3" to increase the number of selected informative genes from two ("g5, g3") to three ("g5, g3, V192"). The gene selections and response class predictions processes continue until step 5 at which the gene selection and classification were optimal. At that optimal selection step, step 5, the following six informative genes, "g5", "g3", "V192", "V805", "V566" and "g2", have been selected in that sequence. We present in *Table 3.2a*, the k-SS prediction results which include the minimum mean MERs and their differences as well as the number of gene selected at each gene selection step.

At the 6th selection step however, consideration was being given to the 7th gene to be selected. At this step, the minimum mean MER difference $\hat{\delta}_{6^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_6} - \hat{\vartheta}^{m_1,m_2,\dots,m_7}$ was estimated given the following summary statistics: $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_6} = 0.0463$, $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_7} = 0.0485$, $\hat{\delta}_{6^1} = -0.0021$. Also, with C_α^1 already found to be $C_\alpha^1 = Z_{1-\alpha}(4.0398)$, the estimates of the critical value $C_\alpha^1\sqrt{\sigma^2\left(\hat{\delta}_{6^1}\right)}$ of the test statistic $\hat{\delta}_{6^1}$ as given by the decision rules (2.4.36) and (2.4.37) with $j = 6$ has a range

$$Z_{1-\alpha}(4.0398) \times \sqrt{\sigma^2\left(\hat{\delta}_{6^1}\right)} \in (\infty, -1.2081 \times 10^{-4}] \quad (3.3.2)$$

computed over the corresponding range of significance level α, estimated by cross-validation, given by

$$\alpha \in (0, 0.975] \qquad (3.3.3)$$

It can be easily observed from (3.3.2) that $\hat{\delta}_{6^1} = -0.0021 < Z_{1-\alpha}(4.0398) \times \sqrt{\sigma^2\left(\hat{\delta}_{6^1}\right)}$ over all the range of α as given in (3.3.3). Therefore, by decision rule (2.4.37), further gene selection is stopped and the 7th gene is excluded from k-SS classification model. This simply implies that, our k-SS procedure considers the relative loss in prediction accuracy of -0.0021, the difference between the mean MER $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_6} = 0.0463$ (obtained at 5th selection step from 6 genes) and the mean MER $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_7} = 0.0485$ (obtained at 6th selection step from 7 genes), to be too large enough to warrant the stoppage of further gene selection beyond the 5th selection step. Hence, the reason why the inclusion of the seventh gene at the 6th selection steps is rejected by k-SS criteria.

Based on our simulated microarray data set therefore, the best prediction results are obtained at the 5th selection step at which $k = 6$ informative genes (“g5”, “g3”, “V192”, “V805”, “V566”, “g2”) are selected by our k-SS method. The average prediction accuracy achieved by our k-SS classifier using the six genes is 95.37%. This yielded an average MER of 0.0463.

If we adopt the minimum mean MER $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ formulation in the construction of one directional hypothesis set as stated in (2.4.49), the same test procedures above would be followed with the only exception that the test statistic used would now be $\hat{\delta}_{j^2}$ or $Z_{\hat{\delta}_{j^2}}$ with $Z_{\hat{\delta}_{j^2}} \sim SN(-\lambda^*)$ as earlier established in this chapter. The critical value C_α^2 of the test statistic $Z_{\hat{\delta}_{j^2}}$ would be $C_\alpha^2 = Z_{1-\alpha}(-4.0398)$.

According to our k-SS results under the $\hat{\delta}_{j^2}$ formulation, the optimal selection step is also attained at the 5th selection step after the selection of the 6th gene into the model. At the 6th selection step however, consideration is being given to the 7th gene to be selected into the model. The minimum mean MER difference $\hat{\delta}_{6^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_7} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_6}$ is computed at this selection step (step 7) and the following summary statistics are obtained; $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_6} = 0.0473$, $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_7} = 0.0490$ and $\hat{\delta}_{6^2} = 0.0017$. The estimated critical value $C_\alpha^2\sqrt{\sigma^2(\hat{\delta}_{6^2})}$ for the test statistic $\hat{\delta}_{6^2}$ has a range

$$Z_{1-\alpha}(-4.0398) \times \sqrt{\sigma^2(\hat{\delta}_{6^2})} \in [2.1629 \times 10^{-4}, -\infty) \qquad (3.3.4)$$

which is obtained over the corresponding range of α estimated as

$$\alpha \in [0.025, 1) \qquad (3.3.5)$$

Selection steps j	Min. mean MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$	Min. mean MERs $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$	$\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ $-\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$	No. of genes selected	Decision
0	0.1822	-	-	1	continue
1	0.1768	0.1090	-0.0678	2	✓
2	0.1121	0.0772	-0.0349	3	✓
3	0.0817	0.0712	-0.0105	4	✓
4	0.0722	0.0584	-0.0138	5	✓
5	0.0601	0.0461	-0.0140	6	✓
6	**0.0473**	0.0490	0.0017	×	stop

Table 3.2b: Table of results of k-SS classifier under the $\hat{\delta}_{j^2}$ formulations at each gene selection step for simulated data. Optimal selection (the best prediction result) is achieved at the fifth selection step at which the sixth gene is selected. The size α of the k-SS test, determined by cross-validation, satisfies the range $\alpha \in [0.025, 1)$. The corresponding rage of the critical value $C_{\alpha}^{2}\sqrt{\sigma^2(\hat{\delta}_{6^2})}$ of the test statistic $\hat{\delta}_{6^2}$ for this range of α is estimated as $C_{\alpha}^{2}\sqrt{\sigma^2(\hat{\delta}_{6^2})} \in [2.1629 \times 10^{-4}, -\infty)$. The six genes selected in order of selection steps 0,1, … ,5 are "g5", "g3", "V192", "V805", "V566", "g2" respectively.

Based on the above results, it could be observed that $\hat{\delta}_{6^2} = 0.0017 > Z_{1-\alpha}(-4.0398) \times \sqrt{\sigma^2(\hat{\delta}_{6^2})}$, which satisfied the decision rule (2.4.50) over all the range of α as given in (3.3.5). Therefore, the selection of the 7th gene into the model at the 6th selection step is rejected and further gene selection stops. The results' estimates at each selection step as provided by our k-SS procedures are presented in *Table 3.2b*. At the optimal selection step, step 5 after which no additional genes is allowed into the model again, the following sequence of 6 genes, "g5", "g3", "V192", "V805", "V566", "g2" as selected under the $\hat{\delta}_{j^1}$ test formulations have being equally selected. This simply confirms our earlier remark that the use of $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$ or $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$ formulations for the construction of our k-SS procedure would yield similar prediction results.

Results from *Tables 3.2a & b* showed that the average prediction error rate estimated by k-SS method using six genes is about 4.7% under the two test formulations. This shows that, for the simulated microarray data set, the new k-SS method provided prediction accuracy of about 95%. By this result, our k-SS method correctly classified 95% of the subjects in the population from which the data was simulated while it misclassify just about 5% of the subjects. The estimates of other performance measures for the k-SS classifier are provided as follows; sensitivity $\approx$ 96%, specificity $\approx$ 98%, positive predictive value (PPV) $\approx$ 98%, negative predictive value (NPV) $\approx$ 96%, Jaccard Index $\approx$ 91%. All these performance measures as obtained under the two test conditions (2.4.32) and (2.4.49) as considered by our k-SS method is presented in *Table 3.3.* The cross-validated ROC (CVROC) curve and the estimated area under the curve called the *cross-validated* AUC (CVAUC) area, for the optimal k-SS classification model (containing six selected genes) under the $\hat{\delta}_{j^1}$ formulation is presented in *Fig 3.6.*

	Performance Measures on k-SS classifiers						
k-SS formulations	MER	Sensitivity	Specificity	+predictive value	-predictive value	Jaccard Index	No. of selected Genes
$\hat{\delta}_{j^1}$	0.0463	0.9593	0.9789	0.9785	0.9601	0.9131	6
$\hat{\delta}_{j^2}$	0.0473	0.9592	0.9790	0.9786	0.9600	0.9112	6
Average performance	**0.0468**	**0.9593**	**0.9790**	**0.9786**	**0.9601**	**0.9122**	**6**

Table 3.3: Table of estimated performance indices for the k-SS classifier on simulated microarray data set under the two minimum mean MER test formulations $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$.

Due to some argument raised in favour of the use of brier score as an important assessment measure of classification rules (Hand, 1997), we equally obtained the average cross-validated estimates of the brier score, $\hat{\hat{\vartheta}}_{brier}$ to access the performance of the k-SS method. This is estimated to be $\hat{\hat{\vartheta}}_{brier} = 0.0492$ for the simulated microarray data. It can be observed that the estimated brier score of 0.0492 is very close to the estimated MER of 0.0473. To this end, we shall ignore the brier scores estimates in our subsequent analyses.

The plot of the average MERs at each selection steps against the number of genes selected as presented in *Fig 3.7* clearly indicated successive improvements in k-SS prediction results as additional genes are selected into the models.

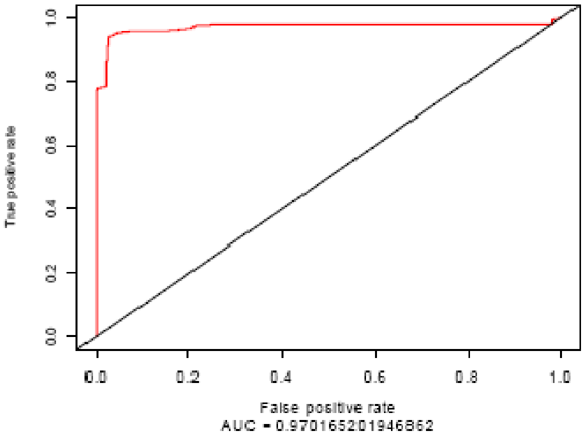


Fig: 3.6: The cross-validated ROC (CVROC) curve for the optimal k-SS prediction results (with six selected genes) under the $\hat{\delta}_{j^1}$ test formulation. The cross-validated AUC area is estimated to be 0.9702.

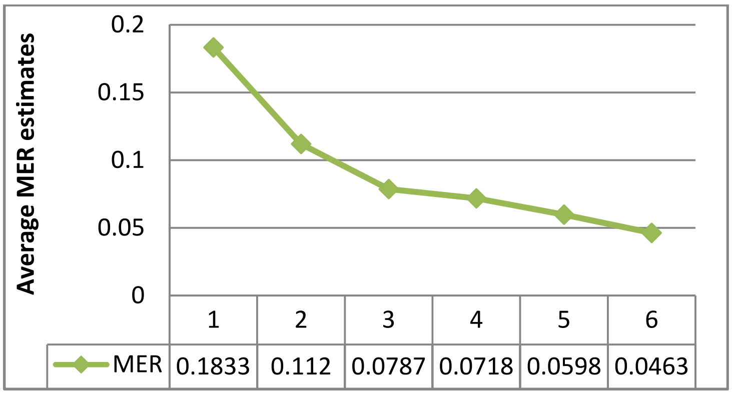


Fig 3.7: The graph of the successive average MER estimates at each selection step against the number of gene selected. The graph shows improvement in prediction accuracy by k-SS method as additional genes are selected into the model until optimal gene selection is reached at the 6th gene selection.

Furthermore, we present in *Fig 3.8* the plots of the estimated minimum mean MER differences for $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ at successive selection steps j against the number of selected genes. It can be easily observed from the plots that both $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$, though having different estimates, provided the same gene selection results and they both reach their optimal selection levels after the selection of the sixth informative genes.

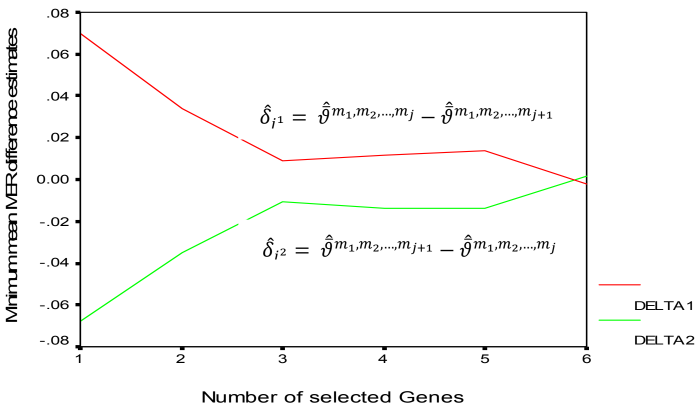


Fig 3.8: The graphs of the successive estimated minimum mean MER differences under the two k-SS test formulations $\hat{\delta}_{j^1}$ (Delta 1) and $\hat{\delta}_{j^2}$ (Delta 2).The optimal gene selection step was reached when k = 6 genes were selected as indicated by the two plots. The optimal selection point is the point at which the $\hat{\delta}_{j^1} \rightarrow -ves$ or $\hat{\delta}_{j^2} \rightarrow +ves$ by some estimated critical values.

Backward checks on the selected genes

As briefly discussed in the last chapter, we intend to examine the importance of each selected genes by our *k*-SS classifier in the presence of other genes in the model. By this, we want to find out if the previously selected genes are still important in the model given that additional new gene is selected into the model. Each of the six selected genes is examined for their relevance in the presence of other selected genes as detailed in Section 2.4.2 under the backward checks procedure. The R code we developed for the implementation of the backward checks on genes selected by *k*-SS method is provided in Appendix B.2.

The results of our backward checks for the six selected genes by our *k*-SS classifier are presented in *Table 3.4.* From the table, it can be easily observed that all the genes selected by *k*-SS method are important in the presence of other selected gene variables in the model. In all cases, the prediction performance of the model without the removed gene are worst than when the removed gene are put back into the model. Based on these results, we can simply suspect that the *k*-SS method only selects the most suitable gene combinations in any given microarray data set. Our suspicion in this regard shall be confirmed when the *k*-SS procedures are applied on real microarray data sets in the next chapter. The box-plot of the results of the backward checks at the 2nd selection step is provided in *Fig 3.9.*

Selection steps *j*	No. of genes selected	MER of full Model	MER of the model when the indicated gene was removed					
0	1	**0.1831**	**g5** -					
1	2	**0.1132**	**g5** 0.2563	**g3** 0.1847				
2	3	**0.0783**	**g5** 0.2192	**g3** 0.1479	**V192** 0.1067			
3	4	**0.0697**	**g5** 0.2189	**g3** 0.1241	**V192** 0.1092	**V805** 0.0779		
4	5	**0.0602**	**g5** 0.1936	**g3** 0.0922	**V192** 0.0893	**V805** 0.07123	**V566** 0.0880	
5	6	**0.0459**	**g5** 0.1677	**g3** 0.0542	**V192** 0.1142	**V805** 0.0616	**V566** 0.0835	**g2** 0.0625

Table 3.4: Results of the backward checks on each of the selected gene by k-SS classifier. The MER indicated against each gene at each selection step is the MER of the model without the indicated gene. The results generally showed that all the selected genes by k-SS method are important in the model.

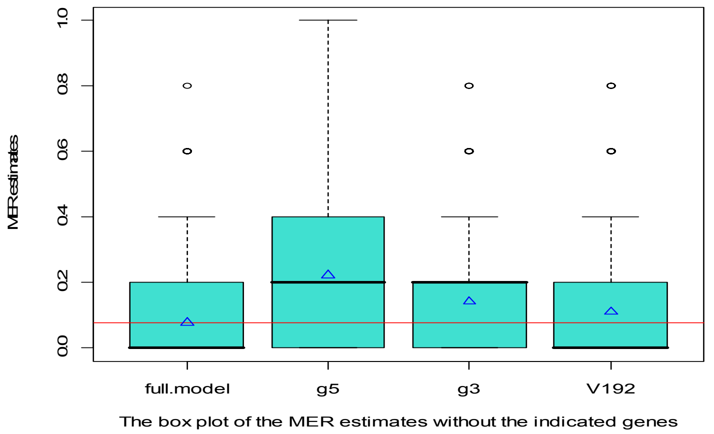


Fig 3.9: The box plot of the backward checks on k-SS selection and prediction results for simulated microarray data. The plot shows the MER of the full model and the models without the indicated gene variables at the third gene selection. The triangular spots are the mean MERs of the models while the red horizontal line indicated the mean MER of the full model. Results from the plot revealed that all the genes selected by k-SS classifier are important in the model.

The sub-sampling technique of Monte-Carlo cross-validation (MCCV) has been adopted in the above implementation of the *k*-SS procedures. It is essential to report that when the cross-validation technique of bootstrap.632+ scheme as proposed by Efron & Tibshirani (1997) was used for the implementation of the k-SS procedures, similar results as in MCCV were obtained. To achieve stable results however, we recommend that sufficient cross-validation runs are used for k-SS implementation. Since the results of the k-SS method under the bootstrap.632+ scheme are essentially similar to those obtained using the MCCV scheme, the results for bootstrap are therefore not reported here to save space. However, the `R` codes we wrote to implement k-SS procedure under the bootstrap.632+ scheme are provided in Appendix B.6.

In the next section, we present the prediction results of three existing classifiers – SVM, k-NN, PLS as implemented in this work on our simulated data and their prediction performances are compared to that of the new k-SS classifier.

3.4. **Applications of some other classifiers**

In this section, we only present the implementation of each of the three selected methods - SVM, k-NN, PLS on simulated microarray data. The results of the remaining five classifiers on published microarray data sets are provided in the relevant section of this thesis.

We begin by using the splitting ratio of 19:1 in favour of training : test samples as used for the construction of our k-SS classifier. For all the analyses performed using the three selected methods, the cross-validation approach of MCCV is adopted with 5000 repetitions.

Support Vector Machines (SVM)

As used for the k-SS implementation, 95% of the sample is used to train the SVM classifier while the remaining 5% is used for its assessment. There are various forms of algorithms that executes SVM for classification. We have adopted the SVM implementation in `R` located under the `e1071` library. Since SVM approach is kernel based whose prediction accuracy is often a function of the type of kernel used for analysis, we shall implement the SVM algorithm using all the four basic kernel functions – i.e. linear, polynomial, radial, and sigmoid kernels as fully discussed in Section 2.8.1. In addition to this, we have discovered that the polynomial kernel implemented in the `e1071` library of `R` is for cubic polynomial by default. We shall, in addition to this, examine the performance of SVM for classification under a polynomial kernel of second degree for possible results' improvements. Thus, all together we have considered five types of kernel for the implementation of SVM and the kernel that provides the best prediction results is finally selected for further inferences.

Performance Measures	**Kernel Types**				
	Linear	*Polynomial3*	*radial*	*sigmoid*	*polynomial2*
MER	0.0340	0.0668	0.0368	0.3812	0.0674
Sensitivity	0.9987	0.9967	0.9975	0.2951	0.9965
Specificity	0.9979	0.9966	0.9988	1.000	0.9968

Table 3.5: Results of support vector machines for classification using simulated microarray data

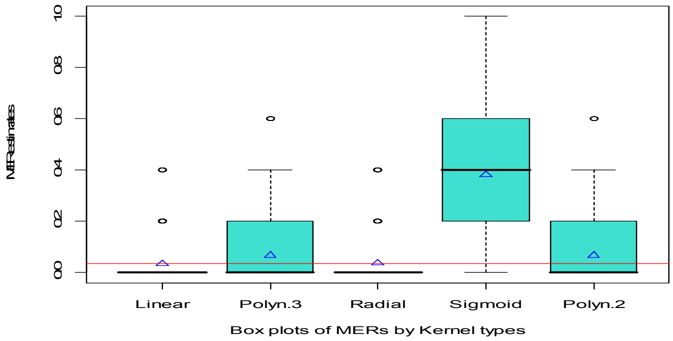


Fig 3.10: The box-plots of average MERs estimates from five support vector machines (SVM) kernels for simulated microarray data. The triangular spots are the mean MERs of the models for each kernel type and the red horizontal line indicated the minimum mean MER.

We present in *Table 3.5*, the classification results from SVM implementation on the simulated microarray data for all the five kernel types. The basic performance measures we reported in the table are the average misclassification error rates MERs, sensitivity and specificity which were all computed over 5000 repetitions using the MCCV sub-sampling scheme.

It can be easily observed from *Table 3.5* that prediction results of SVM using linear or radial kernel seems the best among the five kernel types based on the three performance indices. This superiority performance of the two kernels is clearly shown on the box-plot of the estimated MERs for all the five kernels as presented in *Fig 3.10*. However, the radial basis kernel has been reported in many works to yield more stable results and is generally been preferred in many works (Brown *et al*, 2000; Lee, 2004; etc.). As a result of this, the results of the SVM with radial basis kernel shall be used for further discussions and implementations. Using the radial basis kernel as a standard, the SVM prediction results for the simulated data shows a misclassification error rate (MER) of about 3.7% with 99.75% sensitivity and 99.88% specificity.

k-Nearest Neighbours (k-NN)

As in SVM, the performance of *k*-NN method also depends on the choice of parameter *k*, the number of neighbour to be used for classification. In some studies the value of *k* is fixed *a priori* (Shang & Shen, 2005, Hastie *et al*, 2009) the practice that has been criticized elsewhere for its biasness due to heterogeneity in group samples (Baoli *et al,* 2003). In another studies, the number of neighbours, *k* between 15 and 20 has been suggested (Cover & Hart, 1968; Broder, 1986; etc.) in search for optimal prediction accuracy.

What is however clear is that, prediction accuracy of k-NN classifier largely depends on the number of neighbours adopted for analyses and that the number of neighbour, k, adopted is not unique to all microarray data sets. Therefore, we shall implement the k-NN algorithm for all values of k within the range $1 \leq k \leq 20$ and the best classification results among these as determined through cross-validation shall be chosen as our k-NN result. The k-NN procedure is implemented in the `library(class)` of the `R` statistical package and this we have used for our k-NN implementation.

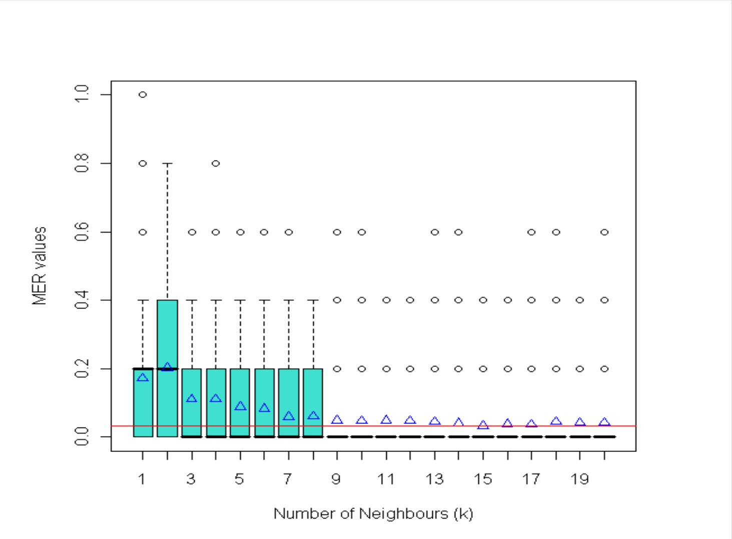


Fig 3.11: The box-plot of the average MERs for k-NN response class prediction at different number of neighbours (k) for simulated microarray data. The best performance occurred at k = 15 neighbours where the least MER is achieved. The triangular spots are the mean MERs of the models at each number of neighbour while the red horizontal line indicated the minimum mean MER.

Using the splitting ration of 19:1 for training : test samples as before, the prediction results under the k-NN method for the simulated microarray data shows the best prediction accuracy at k=15 neighbours. The following performance measures are however estimated: MER = 0.0313, sensitivity = 0.9928 and specificity = 0.9638. The box-plot of the k-NN performance based on MER index at different number of neighbours is presented in *Fig 3.11*.

Partial Least Squares (PLS)

As remarked in the last chapter, the PLS method is, by itself not a classification method but a dimension reduction technique. It is mostly adopted to reduce several thousand of q genes to a very few k gene components, which most often is less than 10 in a high-dimensional microarray data. The number of components, k, constructed from the original q genes are then being used to classify biological subjects into their response groups using any of the standard classification methods. Among the common classification techniques usually adopted for class prediction with PLS components include the *linear discriminant analysis* (Boulesteix & Strimmer, 2005 & 2007), *logistic discriminant analysis* (Nguyen & Rocke, 2002a,b; Fort & Lambert-Lacroix,2005), and *Quadratic discriminant analysis* (Nguyen & Rocke, 2002a,b) among others.

The method that combined dimension reduction of PLS with classification method of the *linear discriminant analysis* (LDA) simply written as PLS-LDA as implemented in the `R` library `plsgenomics` (Boulesteix & Strimmer, 2005 & 2007) is adopted for analyses in this work. The number of the PLS components to be constructed can be fixed *a priori* or determined through cross-validation. Generally, between two to three components have been suggested in some studies (Nguyen & Rocke, 2002a,b,c), while other numbers different from these have are adopted in some others (Ding & Gentleman, 2004). In our implementation of the PLS-LDA, the optimal number of components k desirable for each microarray data set is determined among the first twenty PLS components through cross-validation. By this, the number of component at which the best prediction accuracy is achieved becomes the optimal number of component for each data set.

Based on our simulated microarray data, the classification results of the PLS-LDA revealed a better prediction at just one component. The summary of the estimated performance indices are as follows; MER = 0.0248, sensitivity = 0.9600 and specificity = 0.9994. The box-plot of the MERs at different number of components is presented in *Fig 3.12* where it can be seen that the best prediction is achieved at just one component for the simulated data.

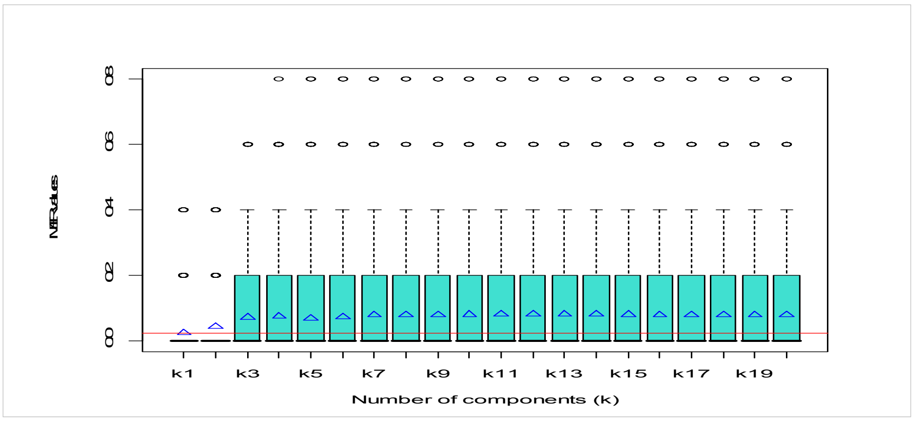


Fig 3.12: The box-plot of the average MERs for PLS-LDA response class prediction at different number of components (k) for simulated microarray data. The best prediction accuracy occurred at the first PLS component (at k = 1) where the least MER is observed. The triangular spots are the mean MERs of the models at different number of components while the red horizontal line indicated the minimum mean MER.

Performance Measures	***Proposed classifier***	***Other classifiers***		
	k-SS	SVM	k-NN	PLS-LDA
MER	0.0463	0.0368	0.0313	0.0248
***CPR**	0.9537	0.9632	0.9687	0.9752
Sensitivity	0.9593	0.9975	0.9928	0.9600
Specificity	0.9790	0.9988	0.9638	0.9994
No. of genes used for prediction	6	1000	1000	1000

*Table 3.6: Summary of the estimated performance indices of the new k-SS classifier and three of the existing classification methods (SVM, k-NN, PLS) on simulated microarray data. The values reported for k-SS are the average estimated prediction performances under the $\hat{\delta}_{j^1}$ and $\hat{\delta}_{j^2}$ k-SS test formulations as reported in Table 3.3. The correct prediction rate (*CPR) is the complement of the estimated MER.*

The summary of the estimated performance measures for our new k-SS classifier and that of other three classifiers (SVM, k-NN, PLS) for simulated microarray data are presented in *Table 3.6*. It can be seen clearly from the table that our k-SS method competes favourably with the three state-of-the art methods in terms of prediction accuracy. The prediction accuracies of all the four classifiers, including the *k*-SS, revolved around 95%. In addition to this, the *k*-SS method has additional advantage of been capable to identify and select those genes that actually contributed to the prediction accuracy estimated. Detail discussions on this and some other benefits of the *k*-SS method shall be provided in the next two chapters.

In the section that follows, we examine the impacts of some random splitting ratios for the training and test samples on the performance of our new *k*-SS classifier as well as other existing classification methods we have so far considered using MCCV sub-sampling scheme.

3.5 **Effects of training-test sample splitting ratios on classifier's performance**

When the sub-sampling techniques of MCCV, bootstrap or any of their variants is to be adopted to improve the performance of any classification rule, the usual practice is to perform a random split of the original sample size *n* into the training and test sample. The idea is to build the classifier using the training sample and evaluates its prediction performance on the test sample. Different splitting ratios between the training and the test samples have been suggested in the literature the most common of which is the ratio 2:1 in favour of training : test sample respectively proposed by Dudoit *et al* (2002). By this, 2/3 of the whole data would be used to train the classifiers and the remaining 1/3 would be used to evaluate their performance via any preferred prediction accuracy indices.

In this section, we seek to examine the effects of some random splitting ratios between the training and test samples on the prediction performance of our k-SS classifier as well as other three classifiers so far considered up to this point. Due to small sample size scenario as common to microarray data sets, we suspected that the common choice of 2:1 splitting ratio might yield unstable and misleading results. Our argument here is that, further reduction of the original sample size n by 1/3rd (used as the training sample) might result into loss of some useful information in the sample that might be needed to construct efficient and stable classification rules. Hence, it is important to keep as much as possible, substantial part of the data in the training set while the remaining few left-out sample shall be used to assess the performance of the classifiers.

To buttress our argument, we shall consider the prediction performances of the k-SS, SVM, k-NN and PLS-LDA classifiers on four different random splitting ratios 1:1, 2:1, 4:1 and 19:1 for training : test samples respectively. This literally translates to using 50%, 66%, 80% and 95% of the whole sample size n as training samples and the remaining 50%, 33%, 20%, and 5% as the test samples respectively.

Splitting ratios	**1:1**	**2:1**	**4:1**	**19:1**
MERs (%)	8.37	6.38	4.92	4.63
No. of genes selected	6	8	9	6

Table 3.7: Table of gene selection and class prediction results by k-SS method at four different splitting ratios of training : test samples. The best prediction results are obtained at 19:1 random splitting ratio, i.e. at 95% training sample(test sample of 5%).

Using our simulated microarray data set, the prediction results of the new k-SS classifier under each of the selected splitting ratios are provided in *Table 3.7*. The corresponding box-plot for these results is provided in *Fig 3.13*.

It can be easily observed from the results of *Table 3.7* and *Fig 3.13* that the performance of the k-SS classifier is sensitive to the choice of splitting ratios between the training and test samples adopted for analysis. The results indicated that the more observations we have in the training samples the better the prediction accuracy of the *k*-SS classifier. The best prediction accuracy (the least mean MER value) however occurred when the *k*-SS classifier is trained with 95% of the whole sample while its prediction performance is only being assessed based on the remaining 5% of the sample.

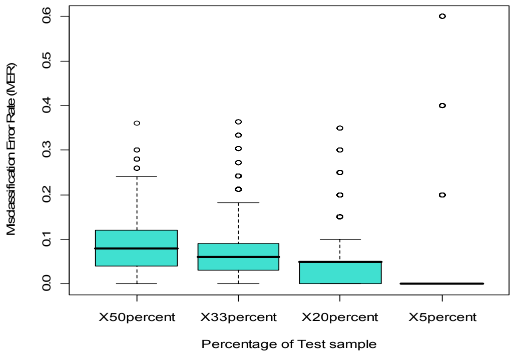
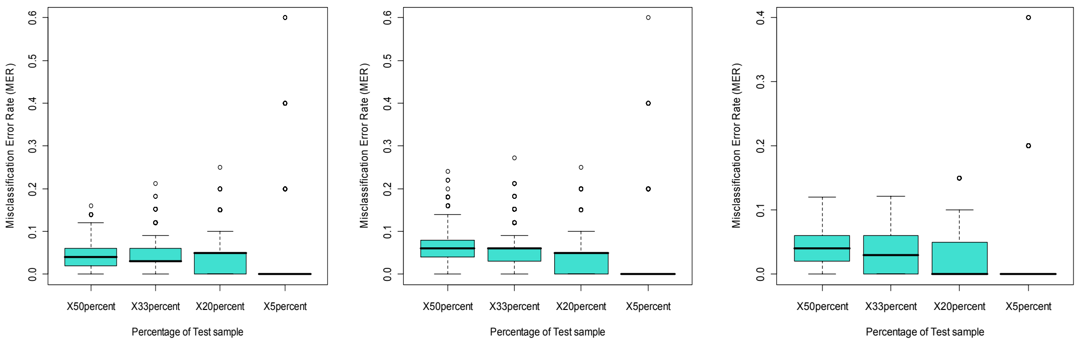


Fig 3.13: The box-plot of the misclassification error rates (MERs) in Table 3.7 for k-SS performances at four different splitting ratios between the training and test samples. The box-plot shows the best prediction accuracy (the least MER value) of the k-SS classifier at 19:1 random splitting ratio i.e. at 95% training sample (test sample of 5%).

Splitting ratios →	**1:1**	**2:1**	**4:1**	**19:1**
Classifiers ↓	**Average MERs (%)**			
SVM	4.63	4.02	3.65	3.53
k-NN	6.22	5.33	4.58	3.13
PLS-LDA	3.69	3.18	2.96	2.48

Table 3.8: Prediction results of SVM, k-NN and PLS-LDA classifier at four different training : test sample splitting ratios. The best prediction results of the three classifiers are obtained at 19:1 random splitting ratio.

We equally present in *Table 3.8* the prediction performances of other three existing classification rules (SVM, *k*-NN, PLS-LDA) at the four splitting ratios 1:1, 2:1, 4:1 and 19:1 for training : test samples respectively. The corresponding box-plots are provided in *Figs 3.14.* All the results also confirmed a better performance of each of the classifiers at the splitting ratio of 19:1 for training : test samples respectively.



SVM

k-NN

PLS-LDA

Fig 3.14: Box-plots of the misclassification error rates (MERs) of SVM, k-NN and PLS-LDA classifiers at four different training : test sample random splitting ratios. The three box-plots showed the best prediction accuracy (the least MER value) of all the three classifiers at 19:1 random splitting ratio i.e. at 95% training sample (test sample of 5%).

In summary, all the above results clearly provided a clear justification of our choice of random splitting ratio of 19:1 in favour of training : test samples respectively while constructing our *k*-SS classifier.

3.6 Applications of AUC preliminary feature selection method

We briefly present here, the discussion of results obtained from the application of AUC preliminary feature selection we proposed in Section 2.6 of this thesis as applied on our simulated microarray data set. Under the student-*t* preliminary feature selection procedure, 55 genes were selected by setting the cut-point of the p-value at 0.05. However, when our proposed AUC criteria as detailed in Section 2.6 were applied, 101 genes were selected at the threshold value of 0.05 for α. When all the 101 genes were ordered in terms of their average AUC values, gene “g5” was found to be the best gene having the highest average AUC value of 0.9075. The worst gene with the least mean AUC value of 0.5950 is gene “V948”. What we can quickly infer from this two results (student-t’s and AUC’s) is that given the same significance level α, the AUC criteria will select more genes than the t-statistic, thereby saving us from the risk of leaving out some of the potentially relevant genes at the primary selection stage for further consideration by standard classification methods. Additional advantage of our AUC preliminary selection procedure is that, it is possible to have idea of the possible predictive power of each gene selected under via their estimated cross-validated AUC values.

However, as remarked in Section 2.6, any gene with its AUC value revolving around 0.5 is not expected to *uniquely* provide good prediction of the response class. Due to this fact, we decided to lower the value of the significance level α used by the AUC selection from 0.05 to 0.02. At this level of α, a total of 50 potentially good genes were selected with the best gene, “g5”, having the highest AUC value of 0.9196 while the weakest gene in the group in terms of its AUC contribution has estimated AUC value of 0.6169.

Surprisingly, the use of the AUC preliminary gene selection on our k-SS method yielded the same final gene selection results as those provided by it under the features selection by the t-statistics. For instance, the following six genes, “g5”, “g3”, “V192”, “V805”, “V566”, “g2”, as previously selected by k-SS classifier under the preliminary selection by the t-test are equally selected using AUC preliminary feature selection method. The full results are not presented here due to space consideration.

However, it is necessary to remark that, though, both the AUC and the t preliminary feature selection methods as used with our k-SS method provided similar results based on the simulated microarray data only. It is not unexpected in some instances however to discover some differences in the results provided under the two approaches in terms of the crop and number of genes selected as well as overall prediction performances of the classifiers that might used them. This should be expected because the two methods adopted different criteria for feature selection. If this situation arises, the crop of genes finally selected for class prediction by *k*-SS method under the two approaches might differ and one would expect better classifier's performance under the AUC feature selection criteria. This particular scenario was encountered when the two methods were applied on real microarray data sets. This is discussed in detail in the next chapter.

4 Applications of *k*-SS method to Microarray data sets

4.1 Data descriptions

In this chapter, we present the application of the new k-SS method on real microarray data sets. To start with, the performances of our new classifier are first compared with those provided by three of the existing state-of-the art classification methods to assess its relative worth under the real microarray data situations. Eleven microarray data sets are used to demonstrate the implementation of the k-SS method. Ten of these data sets are published microarray data that are freely available at their respective web links as later provided. The eleventh data set, as analysed in Section 4.2, is base on microarray rectal cancer study carried out in the Department of Surgery, Klinikum rechts der Isar, Technical University of Munich, Munich, Germany. Details about this particular data are provided in the next section. The brief descriptions of other ten data sets are presented in what follows. We want to remark that, only the results of our k-SS method under its $\hat{\delta}_{j^1}$ formulation shall be reported for all the data sets.

Colon cancer data: These data were first analysed by Alon *et al* (1999). They contain 2,000 gene expression profiles of 62 tissue samples with two distinct clinical groups of tumourous (40 tissue samples) and normal (22 tissue samples) subjects. These data are freely available and can be downloaded at http://microarray.princeton.edu/oncology/affydata/index.html.

Leukemia cancer data1: These data set are pre-loaded with any version of R statistical software (http://www.R-project.org) under the package `multtest`. The data contained 3,051 genes whose expression levels were measured on 38 biological samples containing 27 *acute lymphoblastic leukemia* (ALL) and 11 *acute myeloid leukemia* (AML). The data set were also described in Golub *et al* (1999) and is publicly available at http://www-genome.wi.mit.edu/MPR/.

Leukemia cancer data2: The *Leukemia cancer data 2* have 7,129 genes and 72 samples. As in Leukemia cancer data 1, the sample contains 47 ALL and 25 AML biological subjects. More details on these data can be found in Golub *et al* (1999). The data can be freely downloaded at http://www-genome.wi.mit.edu/MPR/.

CNS data: These data described the embryonal tumours of the *central nervous system* (CNS) and were analysed by Pomeroy *et al* 2002. The data contained 7,129 genes and 34 tissue samples. The 34 sample contains 25 *classic* (C) and 9 *desmoplastic* (D) tumour groups.

DLBCL data: These data set were on 7,129 gene expressions of 77 biological samples. The data were analysed in Ship *et al* (2002) to distinguish 58 *Diffuse large B-cell lymphoma* (DLBCL) samples from 19 *follicular lymphoma* (FL) samples. The data are publicly available at www.genome.wi.mit.edu/MPR/lymphoma.

Lung cancer data: These are lung cancer data described in Gordon *et al* (2002). They contained 12,533 genes and 181 samples, 150 of which were those with *malignant pleural mesothelioma* (MPM) and the remaining 31 subjects having *adenocarcinoma* (ADCA) of the lung. The data can be found at http://www.chestsurg.org.

Prostate cancer data1: These are *prostate cancer* data described in Singh *et al* (2002). They contained expression profiles of 12,600 genes that were measured on 102 samples of 52 tumour and 50 normal samples. The data are available at http://www.genome.wi.mit.edu/MPR/prostate.

Prostate cancer data2: These are *prostate cancer* data used by Stuart *et al* (2004). They have expression measures of 12,625 genes on 88 biological subjects with 38 tumour and 50 normal samples. The data are available at www.affymetrix.com.

Prostate cancer data3: These are another *prostate cancer* data described by Welsh *et al* (2001). They contained 12,626 gene expression profiles of 33 samples. The sample has 24 tumour and 9 normal patients. The data are publicly available at http://www.gnf.org/cancer/prostate.

GCM data: These are molecular cancer data described in Ramaswamy *et al* (2001). The data have 16,063 genes with 280 samples 190 of which are tumourous while 90 are normal samples. The data are available at www.genome.wi.mit.edu_MPR_GCM.html.

4.2 Molecular classifications of rectal and colon cancer patients with *k*-SS method

This section presents detail applications of the new *k*-SS method on both rectal and colon cancer microarray data sets.

Rectal cancer data

As briefly highlighted in Section 4.1, the rectal cancer data analysed here are based on microarray study carried out in the Department of Surgery, Klinikum rechts der Isar, Technical University of Munich, Munich, Germany, on preoperative endoscopic biopsy specimen of 43 patients that were diagnosed for locally advanced rectal carcinomas (LARC). In that study, all the 43 patients were subjected to neoadjuvant radiochemotherapy treatments followed by surgical resection. Thereafter, expression profiles of 24,026 probe sets representing 24,026 human genome U133 plus 2.0 gene-chip arrays were measured on each of the 43 patients. At the end of the clinical diagnoses and treatments, it was discovered that 14 out the 43 patients responded very well to neoadjuvant radiochemotherapy treatments while the remaining 29 patients did not respond to these treatments. However, since it was possible to observe the expression profiles of a good number of genes on these patients, the task now is to

i) determine whether it is possible carry out pre-operative prediction of the clinical status (responder or none-responder to neoadjuvant treatment) of any future LARC patients using the gene expression profiles of some of the observed genes.

ii) identify and select those gene sub-set that are really correlated with the two clinical status of the LARC patients in i) for possible determination of appropriate therapeutic measures among other things.

However, the rectal cancer data set analysed here have been recently analysed also by Rimkus *et al* (2008) where some results regarding the prediction of the clinical status of the 43 LARC patients using their gene expression profiles were equally reported. Further details on clinical characteristics of all the 43 patients are provided in that work. We shall discuss some of the results reported in the article later.

By our preferred random splitting ratio of 19:1 in favour of training and test samples, we used $n_{TR} = 41$ sample as training set and $n_{TE} =$

2 sample as the test set. The sub-sampling scheme of MCCV as discussed in Section 2.7 is adopted for analysis. The expression measures for all the genes were normalized so that each gene vector has zero mean and unit variance across the mRNA samples.

Since the crop of genes selected for further analyses at the preliminary selection stage can greatly influence the performance of any classification rule, we shall therefore examine the prediction performance of our k-SS method under the conventional preliminary selection provided by the t-statistics and that of the AUC feature selection criteria as proposed in this work.

i) k-SS applications under the preliminary selection by t-statistic

Here, the preliminary genes selection was performed using the Student-t statistic as discussed in Section 1.4.2. The cut-point we adopted for the p-values of the t-statistic is 0.001 as also used in many studies, (Nguyen & Rocke, 2002 a, b, c; Rimkus *et al*, 2008; etc.). This procedure selected 34 probe sets whose p-values of their estimated t-statistic are less than or equal to the pre-selected implied p-value of 0.001. These are the genes passed into our k-SS algorithm for further analyses.

Results of our analysis on rectal cancer data showed that the k-SS method selected seven genes with gene symbols “SF3A1”, “TOE1”, “RBM18”, “RPL31”, “227353_at”, “ETS2”, “TNFRSF1B” at the end of the 6th selection step to classify/predict the clinical status of the LARC patients in the test sample as shown in *Tables 4.1 & 4.2*. The probe sets numbers, the genes’ symbols and the genes’ names of each selected gene are provided in the *Table 4.1*. Details of the selection and prediction results at each selection steps are provided in *Table 4.2*.

Probe-set Number	Gene Symbol	Gene Name
216457_s_at	SF3A1	Splicing factor 3a, subunit 1, 120kDa
204080_at	TOE1	Target of EGR1, member 1 (nuclear)
238963_at	RBM18	RNA binding motif protein 18
221593_s_at	RPL31	Ribosomal protein L31
227353_at	“227353_at”	“227353_at”
201329_s_at	ETS2	v-ets erythroblastosis virus E26 oncogene homolog 2 (avian)
203508_at	TNFRSF1B	Tumor necrosis factor receptor superfamily, member 1B

Table 4.1: The selected genes from rectal cancer data by k-SS method under the t-test preliminary feature selection. Only the probe-set number is available for the fifth gene selected as shown on the table.

Selection steps	Min. mean MERs $\hat{\vartheta}^{m_1,m_2,\dots,m_j}$	Min. mean MERs $\hat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	$\hat{\delta}_{j^1} = \hat{\vartheta}^{m_1,m_2,\dots,m_j} - \hat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	No. of genes selected	Decision
0	0.1570	-	-	1	continues
1	0.1570	0.0897	0.0673	2	✓
2	0.0940	0.0665	0.0275	3	✓
3	**0.0734**	**0.0609**	0.0125	4	✓
4	0.0539	0.0482	0.0057	5	✓
5	0.0456	0.0018	0.0438	6	✓
6	0.0017	**0.0011**	0.0006	7	✓
7	0.0015	0.0018	-0.0003	×	stops

Table 4.2: Table of results for k-SS classifier at each gene selection step for rectal cancer data under the preliminary selection by the t-test. Optimal selection is attained after the selection of the 7th gene at the 6th selection step. The seven genes selected in order of selection sequence are “SF3A1”, “TOE1”, “RBM18”, “RPL31”, “227353_at”, “ETS2”, “TNFRSF1B”.

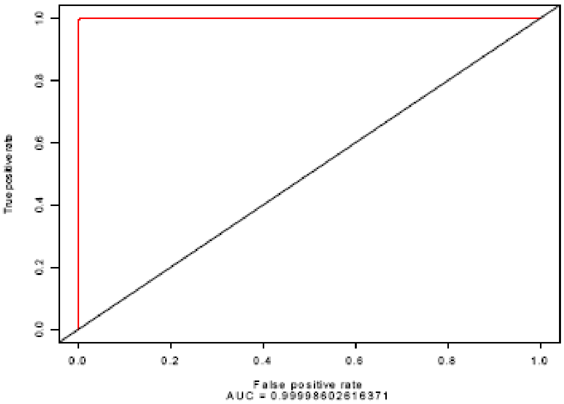


Fig: 4.1: The cross-validated ROC (CVROC) curve estimated by k-SS method from seven selected genes for rectal cancer data. The cross-validated AUC ≈ 1.

It can be easily observed from *Table 4.2* that the *k*-SS method provides correct prediction rate of about 99.89% (average MER of 0.0011) using seven genes. The cross-validated ROC (CVROC) curve and its corresponding AUC area for this result are provided in *Fig 4.1* where it can be seen that the estimated AUC area is almost 1.

To ensure that all the seven selected genes deserve to stay in the model, we perform backward checks on each of the selected genes as discussed in Section 3.3 and the results obtained, as presented in *Table 4.3*, confirmed that all the seven selected genes are important in the model as selected by the *k*-SS classifier and they should all remain in the model.

Selection steps	No. of genes selected	MER of full Model	MER of the model if the indicated gene is removed						
0	1	**0.1570**	**SF3A1** -						
1	2	**0.0890**	**SF3A1** 0.2700	**TOE1** 0.1565					
2	3	**0.0652**	**SF3A1** 0.1334	**TOE1** 0.1587	**RBM18** 0.0961				
3	4	**0.0571**	**SF3A1** 0.1248	**TOE1** 0.1799	**RBM18** 0.0955	**RPL31** 0.0700			
4	5	**0.0474**	**SF3A1** 0.1106	**TOE1** 0.1371	**RBM18** 0.1117	**RPL31** 0.1017	**227353_at** 0.0518		
5	6	**0.0012**	**SF3A1** 0.0020	**TOE1** 0.1200	**RBM18** 0.1201	**RPL31** 0.0025	**227353_at** 0.1164	**ETS2** 0.0459	
6	7	**0.0009**	**SF3A1** 0.0675	**TOE1** 0.0910	**RBM18** 0.1335	**RPL31** 0.0015	**227353_at** 0.1160	**ETS2** 0.0255	**TNFRSF1B** 0.0013

selected under the t-statistic criteria, but might not necessarily be the crop of genes selected under the AUC criteria. Further discussions on this shall be provided later.

The k-SS method, under the AUC preliminary feature selection, selected nine genes in the following sequence with gene symbols “SF3A1”, “TOE1”, “RBM18”, “ZNF24”, “227353_at”, “222303_at”, “CASP1”, “ADPRHL2”, “BLVRA”. The average MER obtained using the 9 genes for prediction is 0.000 translating to 100% correct prediction rate. The k-SS results at each selection steps are presented in *Table 4.4*.

Selection steps j	**Min. mean MERs** $\hat{\vartheta}^{m_1,m_2,\dots,m_j}$	**Min. mean MERs** $\hat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	$\hat{\delta}_{j^1} = \hat{\vartheta}^{m_1,m_2,\dots,m_j} - \hat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	**No. of genes selected**	**Decision**
0	0.1580	-	-	1	continues
1	0.1580	0.0881	0.0699	2	✓
2	0.0904	0.0670	0.0234	3	✓
3	**0.0701**	**0.0470**	0.0231	4	✓
4	0.0471	0.0278	0.0193	5	✓
5	0.0311	0.0020	0.0291	6	✓
6	0.0015	0.0010	0.0005	7	✓
7	0.0009	0.0005	0.0004	8	✓
8	0.0005	0.0000	0.0005	9	✓
9	**0.0000**	0.0000	0.0000	×	stops

Table 4.4: Table of results for k-SS classifier at each gene selection step for Rectal data under the preliminary selection by AUC criteria. Optimal selection is attained when nine genes were selected. The nine informative genes selected in order of selection steps are “SF3A1”, “TOE1”, “RBM18”, “ZNF24”, “227353_at”, “222303_at”, “CASP1”, “ADPRHL2”, “BLVRA”.

It can be easily observed from the results that the first three genes selected here are the same set of genes selected by the k-SS procedure under the preliminary selection by t-statistic. However, at the 3rd selection step, the 4th gene with gene symbol “ZNF24” was selected by k-SS method. The inclusion of this gene with three other previously selected genes (“SF3A1”, “TOE1”, “RBM18”) reduced the average MER from **0.0701** to **0.0470** (*red bold in Table 4.4*), contributing a reduction in prediction error rate by about 33%. The estimates of other performance indices at the end of the genes selection steps provided the following results; *sensitivity* = 100%, *specificity* = 100%, *positive predictive value* (PPV) = 100%, *negative predictive value* (NPV) = 100%, *Jaccard Index* = 100%. The cross-validated ROC curve for the k-SS classifier is presented in *Fig 4.2* where it can be seen that the estimated cross-validated AUC is exactly 1.

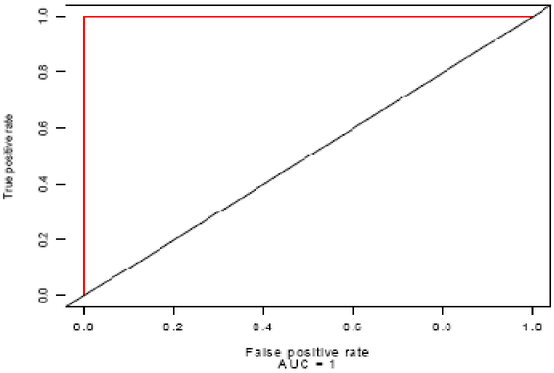


Fig: 4.2: The cross-validated ROC (CVROC) curve estimated by k-SS classifier from nine selected genes for rectal cancer data under the AUC preliminary feature selection criteria. The cross-validated area under the ROC curve (CVAUC) is 1.

In the implementation of the k-SS method using the preliminarily selected genes by the t-test procedure as presented in i) above, it is observed that gene “ZNF24”, which was among the 76 genes preliminarily selected under the AUC criteria, was not among the 34 genes preliminarily selected by the t-statistics criteria *(see Table 4.2)*, hence, it was not available for consideration by the k-SS algorithm during the gene selection and prediction processes. In the sequence of genes selected by the t-statistics, gene “RPL31” was the next best gene available among the remaining genes and this was duly identified and selected by the k-SS classifier at the third selection step. This gene was considered as the fourth best gene due to non-existence of the right gene “ZNF24” *(see Table 4.2)*.

As can be observed from *Table 4.2*, the selection of gene “RPL31” by k-SS classifier at the 3rd selection step reduced the average MER from **0.0734** to just **0.0609** *(red bold in Table 4.2)*, contributing a reduction in prediction error rate by about 17%. This is just about 50% of the gain in prediction accuracy of 33% achieved by *k*-SS method for selecting gene "ZNF24" as presented in ii) using the crop of genes selected under the AUC preliminary selection criteria.

More generally, it can be observed from the above results that the prediction accuracy of the *k*-SS classifiers progressively improves as more suitable genes are selected for prediction at each selection step *(see Tables 2 & 4)*. This improvement shall be more remarkable if all the potentially discriminative genes are selected at the preliminary selection stage for further analyses as obtainable under the AUC selection criteria. It is not surprising however, to observe in *Table 4.*4 *(for k-SS results under the AUC preliminary selection criteria)* that the prediction error rate finally approach zero at the optimal gene selection step, step 10 at which the 9th gene was selected. This result simply underscores the need to adopt a good preliminary selection method that would ensure the selection of all potentially relevant genes at the preliminary selection stage before any standard gene selection and/or classification method like the new *k*-SS technique are implemented on the features selected.

Based on the results obtained under i) and ii) above, we can simply conclude that the best set of genes combination that are capable to discriminate between responder and non-responder LARC patients to neoadjuvant radiochemotherapy treatments are the 9 genes "SF3A1", "TOE1", "RBM18", "ZNF24", "227353_at", "222303_at", "CASP1", "ADPRHL2", "BLVRA" as provided by *k*-SS method under the AUC preliminary selection criteria. Detail information about these nine genes is provided in *Table 4.5*. Further comments on these results are provided in the next chapter under the discussion of results.

Probe-set Number	Gene Symbol	Gene Name
216457_s_at	SF3A1	Splicing factor 3a, subunit 1, 120kDa
204080_at	TOE1	Target of EGR1, member 1 (nuclear)
238963_at	RBM18	RNA binding motif protein 18
203247_s_at	ZNF24	Zinc finger protein 24 (KOX 17)
227353_at	"227353_at"	"227353_at"
222303_at	"222303_at"	"222303_at"
1552703_s_at	CASP1	Caspase 1, apoptosis-related cysteine peptidase
223097_at	ADPRHL2	ADP-ribosylhydrolase like 2
203773_x_at	BLVRA	Biliverdin reductase A

Table 4.5: The selected genes from rectal cancer data by k-SS method using the crop of genes selected at preliminary selection stage by AUC setlecion criteria. Only the probe-set number is available for the fifth and sixth selected genes as shown on the table.

The above results clearly showed that the crop of features selected at the preliminary selection stage has significant influence on the performance of classification rules. The goodness or otherwise of the crop of genes selected at the preliminary selection stage directly depends on the efficiency of selection method adopted. If the selection method adopted at the preliminary selection stage is very efficient like the newly proposed AUC feature criteria, the prediction results of *k*-SS or that of any other classifiers would also be efficient and reliable. But if wrong crop of genes are selected at the preliminary selection stage due to the adoption of inefficient method, then, the prediction performance of any adopted classification rule would be badly affected.

It is important to remark that the rectal cancer data considered here has been earlier investigated by Rimkus *et al* (2008) where the classification procedure of PLS-LDA was adopted using sub-sampling scheme of *leave-one-out cross-validation* (LOOCV). In their results, they reported correct classification rate of responders (specificity) to be 71% while correct classification rate of non-responders (sensitivity) was estimated to be 86%. These results indicated an overall estimated correct prediction rate (CCR) of about 81.4% suggesting a misclassification of about 8 out of the 43 LARC patients. Obviously, this prediction results fell far below the estimated prediction accuracy of 100% provided by our *k*-SS method under the two cases considered above for this same data set.

Colon cancer

These are cDNA microarray colon cancer data that has been previously analysed elsewhere, (Alon *et al*, 1999) using unsupervised technique of two-way hierarchical clustering with single linkage search to separate cancerous from non-cancerous tissues among 62 colon cancer patients. The same data were analysed at different times by Furey *et al* (2000) using *support vector machines* (SVM) and Ding & Gentleman (2004) using *iterative reweighted partial least square* (IRWPLS) methods to classify the biological subjects into two distinct sub-cancer groups of tumour and normal patients.

The data contain the expression profiles of 2,000 genes on 40 tumour and 22 normal colon tissue samples. Our task is to (i) identify and select those genes that are predictive of these two biological groups and (ii) use the selected genes to predict any future (unseen) colon tissue samples as either tumourous or normal using the new *k*-SS method. We shall only present here, the *k*-SS results under the AUC preliminary feature selection.

Results of our *k*-SS method for the colon cancer data revealed the four genes that provided the best discrimination between tumour and normal patients. The probe-set numbers of the four selected genes are "Hsa.8147", "Hsa.5392", "Hsa.1410", "Hsa.490". With these four genes, the *k*-SS prediction accuracy is 93.83% indicating a misclassification of about 4 subjects. The estimates of other performance measures computed by k-SS method are as follows; *sensitivity* = 94.96%, *specificity* = 95.22%, *positive predictive value* (PPV) = 97.32%, *negative predictive value* (NPV) = 91.23%, *Jaccard Index* = 90.94%. The CVROC curve for this data is presented in *Fig 4.3* with the estimated cross-validated AUC area (CVAUC) of 0.9465. The prediction results at each gene selection steps are presented in *Table 4.6*. The results of the backward checks on all the four selected genes are presented in *Table 4.7* where it is clear that all the four selected genes are relevant in the model. The box-plot of one of the results of the backward checks at the third gene selection is provided in *Fig 4.4* where it is revealed that the average MER of the models without the indicated genes are higher than the estimated mean MER of the full model. This evidently underscores the relative importance of each of the selected genes for prediction by k-SS method.

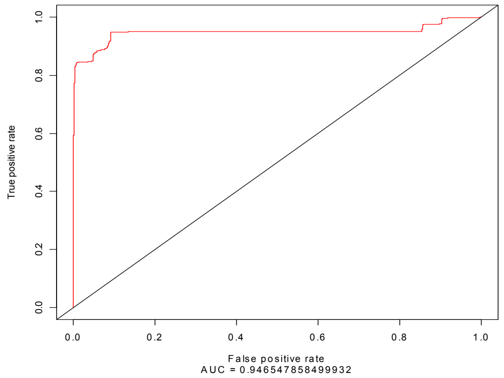


Fig: 4.3: The cross-validated ROC (CVROC) curve estimated by k-SS method using the four selected genes from colon cancer data. The cross-validated area under the ROC curve (CVAUC) is 0.9465.

Selection steps j	**Min. mean MERs** $\widehat{\vartheta}^{m_1,m_2,\dots,m_j}$	**Min. mean MERs** $\widehat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	$\widehat{\delta}_{j^1} = \widehat{\vartheta}^{m_1,m_2,\dots,m_j} - \widehat{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	**No. of genes selected**	**Decision**
0	0.1454	-	-	1	continues
1	0.1454	0.1095	0.0359	2	✓
2	0.1096	0.0679	0.0417	3	✓
3	0.0661	0.0604	0.0057	4	✓
4	0.0617	0.0688	-0.0071	×	stops

Table 4.6: Table of results for k-SS classifier at each gene selection step for colon cancer data. Optimal selection is attained after the selection of the 4th gene at the 3rd selection step. The four genes selected in order of selection sequence are "Hsa.8147", "Hsa.5392", "Hsa.1410", "Hsa.490".

Selection steps *j*	No. of genes selected	MER of full Model	MER of the model when the indicated gene was removed			
0	1	**0.1454**	**Hsa.8147** -			
1	2	**0.1095**	**Hsa.8147** 0.3299	**Hsa.5392** 0.1449		
2	3	**0.0679**	**Hsa.8147** 0.2748	**Hsa.5392** 0.1956	**Hsa.1410** 0.1056	
3	4	**0.0604**	**Hsa.8147** 0.3014	**Hsa.5392** 0.0962	**Hsa.1410** 0.1378	**Hsa.490** 0.0646

Table 4.7: Results of the backward checks on the four selected genes by k-SS classifier from colon cancer data. The MER indicated against each gene is the MER of the model without the indicated gene. The MERs of the full models at each selection step are relatively smaller than that of the models without the indicated genes. This showed that all the selected genes by k-SS method are important in the model.

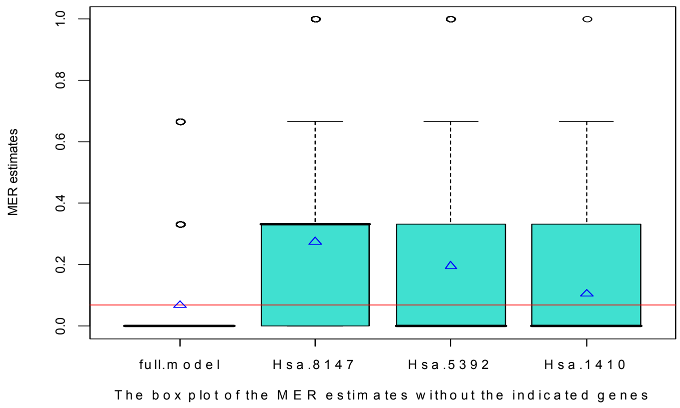


Fig 4.4: The box plot of the backward checks for colon cancer data. It shows the MER of the full model and that of the models without the indicated genes at the third gene selection. The triangular spots are the mean MERs of the models while the red horizontal line indicated the mean of the full model. The estimated average MERs of the model without the indicated genes are relatively higher than that of the full model and indication that all the four genes selected by k-SS classifier are important in the model.

As earlier remarked, this colon cancer data has been previously analysed at different times by Furey *et al* (2000) and Ding & Gentleman (2004) and the two studies reported a misclassification of about 6 of the 62 colon cancer subjects on the average. More specifically, Ding & Gentleman (2004) employs the IRWPLS approach and its variant that incorporated the Firth's procedure, Firth (1992), and selected the first 20 genes with the highest absolute *t*-statistics for classification. The best prediction results reported in their work indicated a misclassification of 7 of the 62 biological subjects. On the other hand, the study of Furey *et al* (2000) misclassified 6 of the 62 subjects using *support vector machines* procedures for classification. In their study, a preliminary selection method that uses the statistic $F_j = \frac{|\bar{x}_j^+ - \bar{x}_j^-|}{\sigma_j^+ + \sigma_j^-}$ was employed, where $\bar{x}_j^+$ and $\bar{x}_j^-$ are the average expression measures of gene j, σ_j^+ and σ_j^- are their respective standard deviations for the two biological groups denoted by + and – signs respectively. The Furey's statistic, though similar to the usual t-statistic, has no theoretical support in statistics for its use. Nonetheless, the k-SS classifier, using just four genes, provided better predictions than any of these earlier methods for this data set.

4.3 *k*-SS results for other microarray data sets

We present the classification results of our k-SS method for other nine publicly available microarray data sets as considered in this work. The remaining data sets whose results are presented under this section are Leukemia data 1 & 2, Prostate data 1, 2 & 3, CNS, DLBCL, Lung and GCM data. The number of genes in each data ranges from 2,000 to 16,000 while the mRNA samples ranges from 33 to 180. The performance measures estimated by the k-SS classifier as shown in *Table 4.8* for each microarray data set are the average MER, correct classification rate (CCR), sensitivity, specificity, PPV, NPV, and Jaccard Index, all of them expressed in percentages. The cross-validated ROC curves as well as their respective cross-validated AUC (CVAUC) area for k-SS classifier for each microarray data set is presented in *Fig 4.5*.

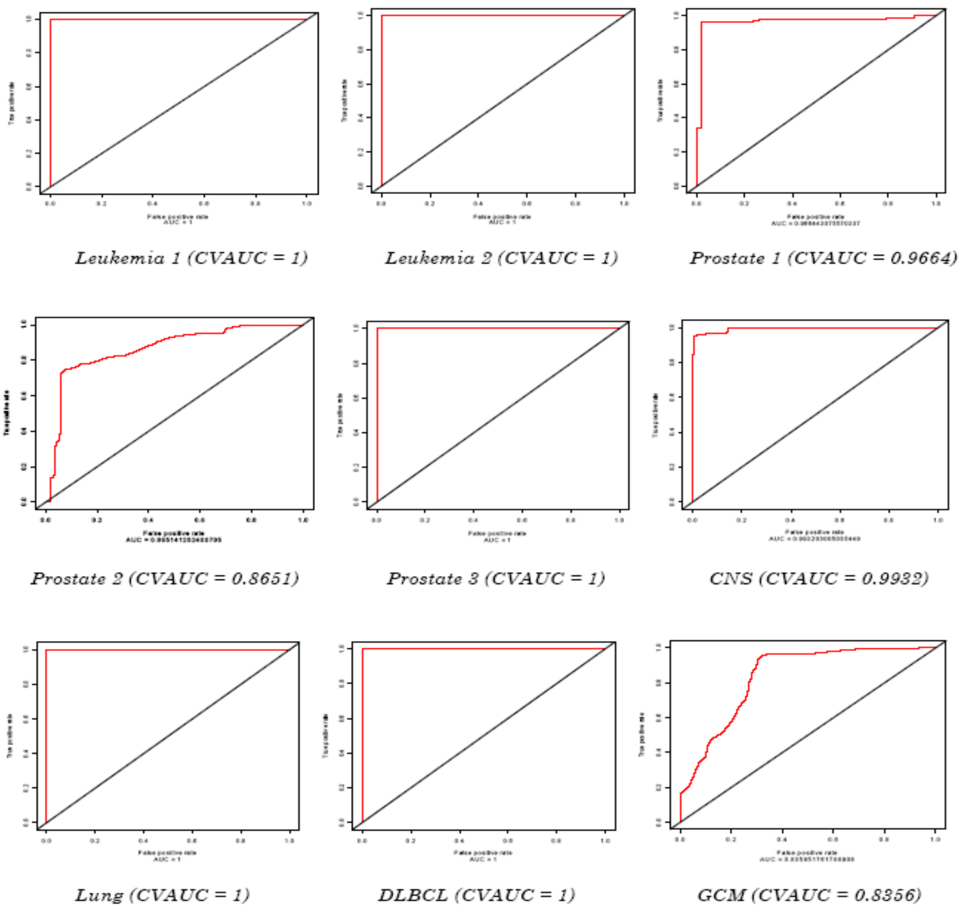
It can be observed from all the results in *Table 4.8* that the new k-SS classifier generally performs very well in all cases of microarray data sets considered. On the overall average, this new method provides about 96% correct classification rate of the tissue samples with an average of 6 selected genes. More discussions on the performance of this new classifier shall be provided in the next chapter.

Microarray data	Estimated performance indices (in %) on k-SS classifier							***No. of selected genes***
	MER	***CCR***	***Sensitivity***	***Specificity***	***PPV***	***NPV***	***Jaccard***	
Leukemia1	0.00	100.00	100.00	100.00	100.00	100.00	100.00	1
Leukemia2	0.00	100.00	100.00	100.00	100.00	100.00	100.00	9
Prostate1	2.85	97.15	97.99	98.00	98.07	97.92	94.45	8
Prostate2	12.01	87.99	84.82	93.89	91.34	89.07	75.10	8
Prostate3	0.00	100.00	100.00	100.00	100.00	100.00	100.00	2
CNS	3.57	96.43	99.54	99.69	99.89	98.85	95.03	4
Lung	0.00	100.00	100.00	100.00	100.00	100.00	100.00	9
DLBCL	0.00	100.00	100.00	100.00	100.00	100.00	100.00	5
GCM	13.17	86.83	96.84	67.63	86.33	91.03	83.28	8
Average performance	**3.51**	**96.49**	**97.69**	**95.47**	**97.29**	**97.43**	**94.21**	**6**

Table 4.8: The various performance indices on the new k-SS classifier for nine published microarray data sets. MER = misclassification error rate, CCR = correct classification rate, PPV = positive predictive value, NPV = negative predictive value.

4.4 ***k*-SS methods versus other classifiers**

In order to determine the goodness of the new k-SS method in comparison to some of the existing classification methods it is necessary to examine its performance relative to some of these classifiers. For this reason, we shall consider the three selected classification methods - SVM, k-NN and PLS-LDA- as presented in Sections 2.8 & 3.4 against which the goodness of our new k-SS classifier would be compared using all the eleven published microarray data sets as presented in Sections 4.1 and 4.2. The comparison of the prediction results of the k-SS method with that of the remaining five classifiers (*Prediction analysis for microarray, Decision trees, Naïve bayes, Top scoring pair, k-Top scoring pair*) is provided in Chapter 5.



Leukemia 1 (CVAUC = 1) *Leukemia 2 (CVAUC = 1)* *Prostate 1 (CVAUC = 0.9664)*

Prostate 2 (CVAUC = 0.8651) *Prostate 3 (CVAUC = 1)* *CNS (CVAUC = 0.9932)*

Lung (CVAUC = 1) *DLBCL (CVAUC = 1)* *GCM (CVAUC = 0.8356)*

Fig 4.5: The cross-validated ROC (CVROC) curves for the prediction results of the k-SS classifier for nine published microarray data sets as shown in Table 4.8. The respective estimates of the cross-validated area under the ROC curve (CVAUC) are equally reported.

The various estimated correct prediction rates (CCR), expressed in percentages, from SVM, *k*-NN, PLS-LDA classifiers as well as that of our new *k*-SS classification method are presented in *Table 4.9*. To ensure that all the classifiers are evaluated on the same platform, we only presented in *Table 4.9* the prediction results of each classifier under preliminary selection by the *t*-statistic.

It can be generally observed from *Table 4.8* that all the four classifiers including the new *k*-SS method provide good predictions of the biological samples in all the eleven microarray data sets considered. However, a closer look at their results revealed that the *k*-SS method has a little edge over other three existing classifiers. Out of the eleven data sets, the prediction rates provided by *k*-SS method is better than that of other classifiers in seven cases, (about 64% of the cases) and performed equally well as others in two instances, (about 18% of the cases) while its prediction performance is slightly lower than others in just two cases, (about 18% of the cases). However, if we consider the average overall performances it can be easily observe that the *k*-SS classifier performs better than all the three existing classifiers considered with respect to their prediction accuracies. In addition, the *k*-SS methods uses a very few sub-sets of genes for classification unlike other earlier methods that used all the available genes for the same purpose.

Microarray data sets	*Number of genes in the data*	*Correct Classification Rate (CCR) (%)*			
		New classifier	**Existing classifiers**		
		k-SS	SVM	*k*-NN	PLS-LDA
Rectal	24,026	**99.89** *(7)*	95.17	93.62	96.73
Colon	2,000	**93.83** *(4)*	81.27	85.65	86.30
Leukemia 1	3,051	**100.00** *(1)*	99.97	**100.00**	**100.00**
Leukemia 2	7,129	**100.00** *(9)*	98.48	93.49	98.63
CNS	7,129	96.43 *(4)*	88.03	**96.75**	91.14
DLBCL	7,129	**100.00** *(5)*	89.22	91.33	91.74
Prostate 1	12,600	**97.15** *(8)*	91.67	90.71	95.36
Prostate 2	12,625	**87.99** *(8)*	78.40	81.38	81.61
Prostate 3	12,626	**100.00** *(2)*	**100.00**	97.45	**100.00**
Lung	12,533	**100.00** *(9)*	98.83	99.74	99.48
GCM	16,063	86.83 *(8)*	87.60	**90.28**	86.23
Average Performance		**96.57**	**91.69**	**92.76**	**93.38**

Table 4.9: The correct classification rates (CCR) of the new k-SS classifier and that of three of the existing methods – SVM, k-NN, PLS-LDA, for eleven published microarray data sets. Out of all the eleven data sets, the k-SS method out-performed other three classifiers in seven instances (about 64% of the cases), it performed equally with others in three cases while it under-performed in just one case. The figures in parenthesis are the number of genes selected for classification by k-SS method from respective microarray data sets. The preliminary feature selection of the t-statistic is used by all the classifiers.

4.5 *k*-SS classifier and cluster analysis

Cluster analysis is one of the earlier unsupervised statistical learning methods commonly adopted for classification and pattern recognition. It is unsupervised because the inherent sub-classes of the subjects are not known a priori and are to be discovered from the data. Therefore, the major aim of clustering is to determine the intrinsic grouping in a set of unlabelled data. When applied to microarray data, it performs the task of revealing some systematic patterns underlying the gene expressions and several sub-classes of the tissue samples. This has been successfully adopted in many microarray studies to identify various sub-classes of cancers in mRNA samples. See Eisen *et al* (1998 & 1999), Alon *et al* (1999), Golub *et al* (1999), Alizadeh *et al* (2000), Gordon *et al* (2002) and Stuart *et al* (2004) among others.

As earlier stated, while applying clustering techniques for classification of mRNA samples, it is assumed that the various subject groups in the data are not previously known and the task is to use the measured genes expression profiles to discover these unknown different biological sub-groups. In other words, it is possible to use the observed gene expression profiles on mRNA samples to discover their various biological sub-groups without an a priori knowledge of those biological groupings through clustering.

In microarray technology, the expression patterns of several thousand of genes are studied simultaneously at the same time. However, if there exist a procedure, like our new *k*-SS method, that can identify and select the few marker genes that are directly related to the existing biological sub-groupings of the mRNA samples, then it would be more appealing and easier while performing clustering, to use only the relevant selected maker genes to identify the different biological groupings of any unidentified future subjects rather than labouring unnecessarily on the entire thousands of genes for the same task. To this end, we shall send the selected *k*-SS classifiers from each microarray data set considered into a suitable clustering algorithm to determining whether they would be capable to identify the inherent biological sub-groups of the unlabelled mRNA samples.

Few of the clustering techniques commonly adopted in the literature are the *k*-Means, fuzzy *c*-Means and hierarchical clustering methods among others. However, the method of *two-way single-linkage hierarchical clustering* (SLHC) has received wider applications in the literature (Alon *et al*, 1999; Alizadeh *et al,* 2000; Gordon *et al*, 2002 etc.) than others and its procedure shall be employed using the *k*-SS selection results.

In the SLHC method as adopted here, the distance matrix between the gene expression data is computed and a linkage or amalgamation rule to determine when two clusters are sufficiently similar to be linked together is defined. By this procedure, a hierarchical tree (dendrogram) is developed which shows the links between all the gene sets and/or between the tissue samples. The clusters are nested together rather than being mutually exclusive as in *k*-means cluster procedure. By this, more and more objects are linked together as larger and larger clusters of increasing dissimilar elements are amalgamated. Therefore, larger clusters created at later stages contained smaller clusters created at earlier stages of agglomeration. In the last step, all objects (genes or tissue samples) are joined together and a horizontal linkage distance is formed. The closer to 1.00 the line that connects two or more genes (or samples) is, the more related the genes (or samples) are to one another. The SLHC becomes a two-way type when both the genes and mRNA samples are clustered simultaneously as performed in Alon *et al* (1999). More details about this clustering method can be found in

Everitt (1980), Alon *et al* (1999), Speed (2003), Lee (2004), Abonyi & Feil (2007) and many other related literatures.

The distance measure we adopted is the *Euclidean distance* metric between any two genes x_j and $x_{j'}$ defined by

$$d_2(x_j, x_{j'}) = \left[\sum_{i=1}^{n}(x_{ij} - x_{ij'})^2\right]^{1/2} \qquad (4.5.1)$$

which is a special case of *Minkowski distance* metric given by

$$d_p(x_j, x_{j'}) = \left[\sum_{i=1}^{n}(x_{ij} - x_{ij'})^p\right]^{1/p}$$

with $p = 2$.

As earlier remarked, to demonstrate the goodness of the genes selected by our k-SS method, only the selected genes from *rectal*, *Leukemia 2*, and *Lung cancer* data sets out of all the eleven microarray data sets are considered for cluster analysis. We have used the clustering software, *cluster 3.0* due to de Hoon *et al* (2004) which is an enhanced version of *cluster* software developed by Eisen *et al,* (1998) for clustering using the SLHC techniques.

Rectal cancer data

In rectal cancer data, the 43 LARC patients consist of 14 responders and 29 non-responders to neoadjuvant radiochemotherapy treatments as obtained from the clinical results. Each subject in the two response group is given a distinct mRNA label. For instance, the 14 responder subjects were given the following labels; p24, p66, p79, p80, p105, p211, p215, p224, p309, p332, p354, p380, p402, and p410. The remaining 29 subjects with mRNA labels different from these fourteen constitute the non-responder patients. Though, we have assumed that the clinical status of the patients is not known, but for the purpose of assessing the cluster output, we have supplied the above mRNA labels for each subject into the cluster algorithm. By this we can easily recognise the subjects' groups clustered together.

The best marker genes selected by *k*-SS method for prediction from rectal cancer data are the following nine genes, as provided in *Table 4.5* "SF3A1", "TOE1", "RBM18", "ZNF24", "227353_at", "222303_at", "CASP1", "ADPRHL2", "BLVRA". These are the nine genes used in our clustering algorithm using two-way SLHC method. The cluster result is provided in *Fig 4.6*.

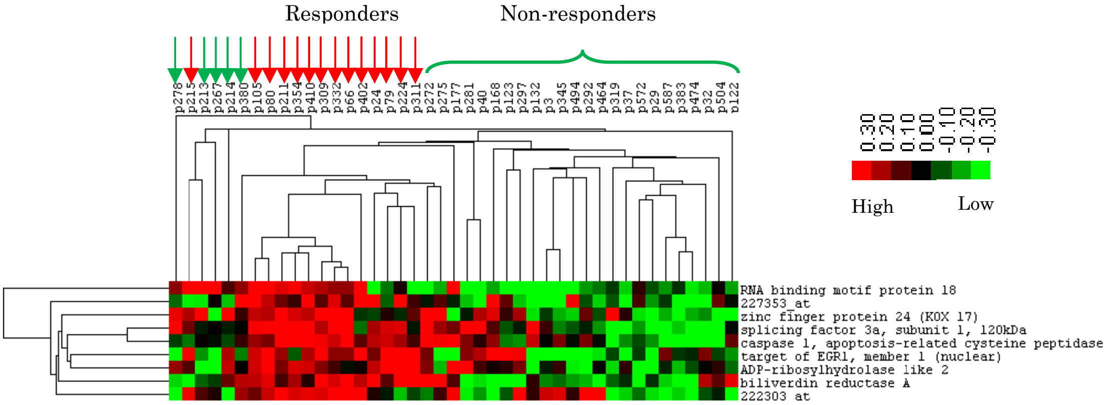


Fig 4.6: The dendrogram of the two-way single linkage hierarchical clustering results using the nine selected genes from rectal cancer data by k-SS method. The cluster shows the two distinct biological groups of LARC patients with responders (indicated with red arrow signs) mostly being those with high expression levels (red fluorescent dyes) of the nine genes and the non-responders (indicated with green arrow or bracket signs) mostly being those with low expression levels (green fluorescent dyes) of the nine genes.

The clustering results as shown in *Fig 4.6* revealed the two distinct groups of LARC patients in the rectal cancer microarray data. The group of responders are mostly the patients with high expression levels of the nine selected genes by *k*-SS classifier, indicated by red fluorescent dyes (Cy5) while the histopathologically non-responders are those patients having low expression levels of the nine genes, indicated by green fluorescent dyes (Cy3). These two groups are clearly identified by the nine genes selected by the *k*-SS method. The fourteen responder subjects are indicated on the dendrogram by red arrow signs while the rest are the non-responders.

Leukemia cancer data 2

The Leukemia cancer data 2 data have 7,129 genes from which 9 informative genes were selected by our *k*-SS method for prediction. There are 72 mRNA samples consisting 47 *acute lymphoblastic leukemia* (ALL) and 25 *acute myeloid leukemia* (AML) patients. The 47 ALL subjects are labelled p01, p02, … , p047 while 25 AML subjects are labelled p148, p149, … , p172.

The nine genes selected by the *k*-SS classifier from these data are Adipsin(M84526), IL-8(M28130), HoxA9(U82759_at), Macmarcks (HG1612), Nucleoside-diphosphate kinase (Y07604), Terminal transferase mRNA (M11722), Cyclin D3(M92287), LTC4 synthase(U50136), and Oncoprotein (Op) 18(M31303). These are the genes used in the two-way SLHC clustering algorithm. The cluster results are displayed by the dendrogram in *Fig 4.7* for these data.

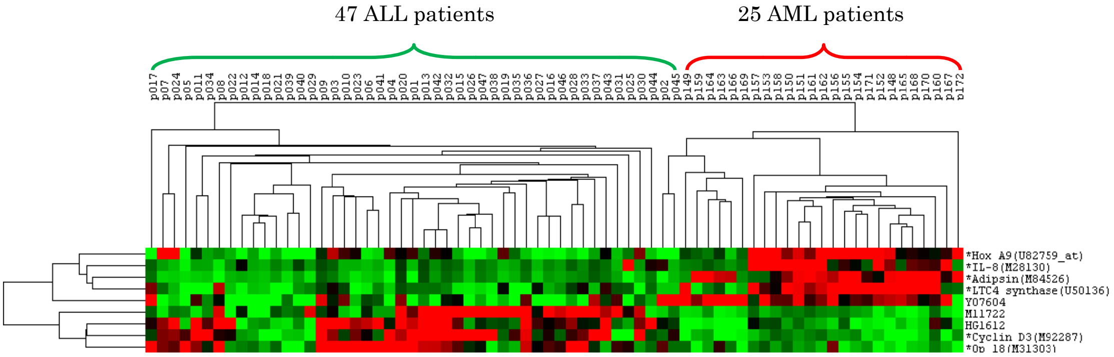


Fig 4.7: The dendrogram of the Single-Linkage Hierarchical clustering (SLHC) result using the nine selected genes by k-SS classifier from leukemia cancer data 2. The two groups of biological subjects with Acute lymphoblastic leukemia (ALL) and Acute myeloid leukemia (AML) are clearly identified by clustering.

It could be observed from the cluster results (*Fig 4.7*) that the AML patients are mostly characterized by having high expression levels of five genes Adipsin(M84526), IL-8(M28130), HoxA9(U82759_at), Nucleoside-diphosphate kinase(Y07604) and LTC4synthase (U50136), for which the ALL subjects group have low expression levels. On the other hand, the ALL patients are mostly those with high expression levels of the remaining four genes Macmarcks(HG1612), Terminal transferase mRNA (M11722), Cyclin D3(M92287) and Op 18 (M31303) for which the AML patients equally have low expressions. The six genes asterisked in *Fig 4.7* were among the fifty differentially expressed genes identified by Golub *et al* (1999). More discussions on this are provided in the next chapter.

Lung cancer data

The lung cancer data contain 12,533 genes and 181 samples, 150 of which are those with *adenocarcinoma* (ADCA) of the lung and the remaining 31 are those with *malignant pleural mesothelioma* (MPM). Except for the rectal, colon and leukemia 1 & 2 data sets where we have information on both gene names and probe-set numbers (rectal, leukemia 2), or probe-set numbers only (colon & leukemia1), we do not have information on both the gene names and probe-set numbers for the remaining seven microarray data sets considered in this thesis. As a result of this, we have labelled the probe-sets in each of the affected data sets including the Lung cancer data 3 as V1, V2, V3, … , and so on, indicating the sequence of available genes in each microarray data set. These are the labels we used in the clustering algorithm for the lung cancer data.

Out of the entire 12,533 genes in the lung cancer data, the following nine genes, "V8005", "V9707", "V2255", "V9607", "V2421", "V8858", "V8537", "V5979", "V6189" were identified and selected by our *k*-SS method for predicting the 181 tissue samples. These nine genes are therefore used for clustering as done earlier and the results of the two-way SLHC are as displayed by the dendrogram in *Fig 4.8.*

As in the previous two microarray data sets considered, it can be observed again here that the nine genes selected by *k*-SS method perfectly revealed the two groups of biological patients (ADCA or MPL) as contained in the lung cancer data.

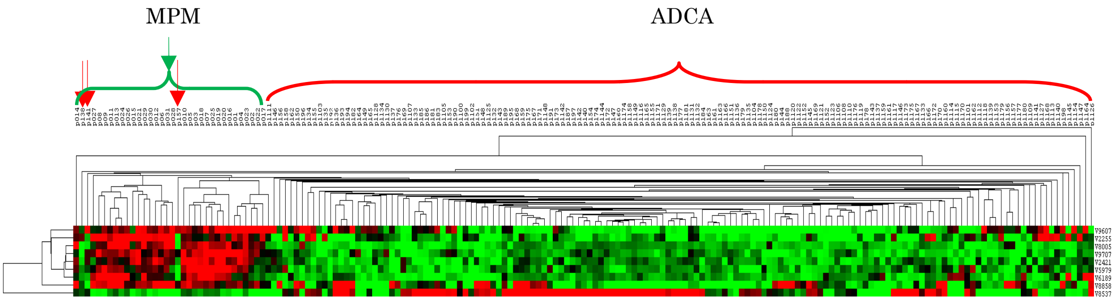


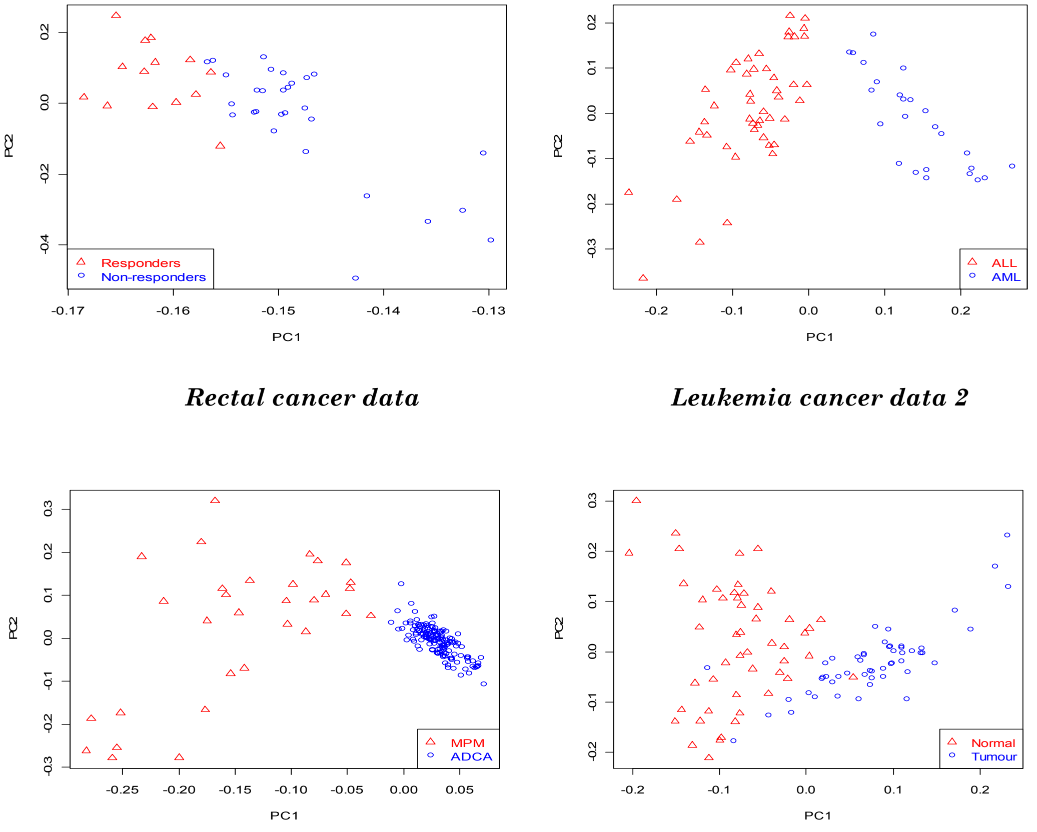
Fig 4.8: The dendrogram of the Single-Linkage Hierarchical clustering (SLHC) results using the nine selected genes by k-SS classifier from lung cancer data. The two biological groups of Malignant Pleural Mesothelioma (MPL) and Adenocarcinoma (ADCA) are clearly identified by clustering. The red bracket and/or arrows indicated the ADCA group while the green bracket denotes the MPM group.

The principal component analysis (PCA)

The discriminatory power of the selected genes by *k*-SS method is equally assessed based on principal component analysis (PCA). The idea is to fit principal component regression model using the selected genes from each microarray data sets and obtain the graphical plots of the first two *principal components* simply called the PCA plots. If the selected genes are good discriminators of the response classes, the number of sub-groups in the response class must be clearly separated on the PCA plots.

We shall again consider the three microarray data sets - Rectal, Leukemia 2 and Lung cancer data sets - as used for clustering in addition to the Prostate 1 data set for the construction of the principal components to assess the efficiency of the *k*-SS classifiers.

As described in Section 4.1, the Prostate 1 cancer data consist of 12,600 genes and 102 samples. The 102 samples consist of 52 tumour (cancerous) and 50 normal (non-cancerous) patients. Our *k*-SS classifier selected 8 informative genes for prediction out of the entire 12,600 genes which eventually yielded correct prediction/classification rate (CCR) of about 97.15% (see *Table 4.8*), indicating a misclassification of about 3 subjects.



Rectal cancer data ***Leukemia cancer data 2***

Lung cancer data ***Prostate cancer data 1***

Fig 4.9: The plots of the first two principal components, PCA plots, constructed using the genes selected by k-SS classifier for four different microarray data sets. All the four PCA plots showed good discriminations of the biological groups of the mRNA samples based on the genes selected by k-SS classifier.

The plot of the first two principal components for each of the data sets is provided in *Fig 4.9.* It can be observed that the different biological groups in each microarray data set are clearly separated on the PCA plots, an indication that the selected genes by *k*-SS classifier are good predictors of the mRNA samples. The two misclassifications (a normal subject misclassified as tumour and a tumour subject misclassified as normal) noticed on the PCA plot for Prostate 1 cancer data is justified by correct prediction rate of 97.15% estimated by *k*-SS classifier using the 8 selected genes as reported for these data in *Table 4.8*.

Based on all the various results as demonstrated in this work, we can generally conclude therefore that the new *k*-SS classifier is capable at selecting the best combination of informative marker genes from several available thousand of genes for good prediction of biological samples in any microarray data sets.

5 Summary of the Study

5.1 Summary of results

This research study is basically designed to address one of the major challenges in microarray studies. The advent of microarray technology which has made it possible to monitor and observe simultaneously the expression levels of several thousand of both relevant and irrelevant genes on a given set of biological subjects has made it more important for us to identify and select the few most relevant genes that are actually related to the tumour conditions being investigated. This task becomes very necessary since the discovery of such relevant genes could tremendously help in the development of appropriate therapeutic measures.

Several methods have being proposed in the literature to carry out this task, but unfortunately a good number of these methods only classify the biological samples into their various cancer sub-groups but not the selection of the relevant informative gene that are easily interpretable with respect to the category of tumour conditions they classified. In addition to this, none of the earlier dimension reduction and/or classification methods like SVM, *k*-NN, PLS, naïve *bayes* (NB), *prediction analysis for microarray* (PAM), *decision tree* (DT), *top scoring pair* (TSP) and the like, has been reputed to be capable at achieving 100 percent prediction accuracy in all cases of tumour classifications in microarray studies.

It is obvious that the cost of misclassify an early stage cancer patient as a normal patient and a normal patient as being cancerous might be too enormous. To avert such negative consequences, it becomes imperative to continuously seeking to develop more efficient classification techniques, like the *k*-SS method proposed here, that could efficiently select the most relevant sub-set of the observed gene chips and provide accurate and stable prediction of biological samples into their various tumour groups in any given high dimensional genomic data.

The new *k*-SS procedure proposed in this thesis is one of the methods targeted at unravel the riddles of dimension reduction, relevant gene selection as well as accurate prediction of various tumour conditions of the mRNA samples as hitherto being desirable in various microarray studies. Given any microarray data set therefore, our new *k*-SS classifier simply adopts unambiguous and easy-to-understand procedures to select only the most informative and biologically relevant marker genes and accurately classify the mRNA samples into their various biological conditions based on the genes selected. This argument is supported by all prediction results provided by our *k*-SS method. For instance, in rectal cancer data, all the 9 selected genes by our *k*-SS procedure are genes encoding proteins. It is clear from the cluster result of *Fig 4.6* for these data that all the selected 9 genes indicated high expressions patterns across all the histopathologically responder patients while they indicated reduced expressions for all the non-responder patients. The two genes “SF3A1” and “TOE1” are genes encoding proteins that perform important function in the nucleus, Rimkus *et al* (2008). Caspases is the family of genes that serve as initiator or executioner of the intrinsic or extrinsic signals that may result into morphological changes that are related to apoptosis, Boatright & Salvesen (2003), Boatright *et al* (2003), Danial & Korsmeyer (2004). Caspase-1 for instance, was the first member of this family whose functions in apoptosis and inflammation have been reported in many studies, Yuan *et al* (1993), Kondo *et al* (1995), Martinon & Tschopp

(2004), Thalappilly *et al* (2006). Among the genes encoding protein that perform transport functions are Biliverdin reductase A (BLVRA) and Zinc finger protein 24 (ZNF24). The BLVRA performs oxidoreductase activities and is capable of initiating several biological processes through energy pathways metabolism. The ZNF24 on the other hand performs transcription regulatory activities and it regulates nucleobase, nucleoside, nucleotide and nucleic acid metabolism (see http://www.biocompare.com/gene/gene_details.asp?geneid=11229#products, HPRD®, for more details on biological functions of these selected genes)

In the leukemia2 cancer data on the other hand, the nine genes selected by *k*-SS method clearly discriminates the *acute myeloid leukemia* (AML) group from *acute lymphoblastic leukemia* (ALL) as shown by cluster result in *Fig 4.7* and PCA plots in *Fig 4.9*. As asterisked on the cluster result of *Fig 4.7*, six of the nine selected genes by *k*-SS classifier have been previously identified as good discriminators between AML and ALL subjects in a microarray study of Golub *et al* (1999). More specifically, the following four genes, *Adipsin, IL-8, HoxA9,* and *LTC4synthase* out of the five genes selected by *k*-SS classifier for which AML subjects have high expression profiles and the two genes, *Cyclin D3* and *Oncoprotein 18 (Op18)* out of the remaining four selected genes by *k*-SS method for which the ALL subjects are up-regulated were among the fifty genes identified by Golub *et al* (1999). More importantly, the two genes *Cyclin D3 and Op 18* have been reported to be genes encoding proteins which are critical to S-phase cell cycle progression, Golub *et al* (1999). It has been further reported (Ross *et al* 1984; Golub *et al* 1999) that some of these identified informative genes encodes topoisomerase II, which is the principal target of the anti-leukemic drug etoposide. All these findings confirm the biological relevance of the genes selected by our new *k*-SS method.

The fact, however remains that all the eleven microarray data sets as used in this thesis have been previously analysed elsewhere at different times to assess the performance of some classification methods. A particular study that interests us among these is the work of Tan *et al* (2005). Except for rectal and leukemia1 cancer data, the remaining nine data sets used in this thesis were also analysed by Tan and his co-workers to assess the performances of their TSP family of classifiers relative to selected five existing classification methods. The two classifiers, PAM and DT that equally perform gene selection as well as classification of biological samples were among the five methods considered in their study.

Like our new *k*-SS method, the TSP family of classifiers which consist of TSP and *k*-TSP, perform gene selection and class prediction and have been adopted for analysis in some studies since they were developed, (Geman *et al* 2004, Xu *et al*, 2005; Price *et al*, 2007, Xu *et al* 2008). We shall therefore, assess the performance of our new *k*-SS classifier relative to that of TSP, *k*-TSP, PAM and DT, all of which perform the same functions like the *k*-SS method as well as one other classifier, Naïve (Idiot) Bayes (NB) which we have not really discussed in this study using the nine microarray data sets as considered in Tan *et at* (2005). For simplicity, we shall only report the various results for the above five classifiers as provided in Tan *et at* (2005), pp 3900 for all the nine microarray data sets and compared these prediction results with the corresponding results provided by our *k*-SS method. The correct classification rates (CCR) estimated by these classifiers are provided in *Table 5.1* while the respective number of genes selected for classification by each of the methods, except NB, is presented in *Table 5.2*.

Method	*Correct classification rates (in %) of classifiers*									
	Colon	Leuk.2	CNS	DLBCL	Prost.1	Prost.2	Prost.3	Lung	GCM	Average
k-SS	**93.83**	**100.00**	96.43	**100.00**	**97.15**	**87.99**	**100.00**	**100.00**	**86.83**	**95.80**
*TSP	91.10	93.80	77.90	98.10	95.10	67.60	97.00	98.30	75.40	88.26
**k*-TSP	90.30	95.83	**97.10**	97.40	91.18	75.00	97.00	98.90	85.40	92.01
*DT	80.65	73.61	67.65	80.52	87.25	64.77	84.85	96.13	77.86	79.25
*PAM	85.48	97.22	82.35	85.71	91.18	79.55	**100.00**	99.45	79.29	88.91
*NB	58.06	**100.00**	82.35	80.52	62.75	73.86	90.91	97.79	84.29	81.17

*Table 5.1: Prediction performances of k-SS method and four other similar gene selection and classification methods (TSP, k-TSP, PAM, DT) as well as NB classifier on nine published microarray data sets. *The reported results are from Tan et al (2005).*

Method	Number of genes used for classification								
	Colon	Leuk.2	CNS	DLBCL	Prost.1	Prost.2	Prost.3	Lung	GCM
k-SS	4	9	4	5	8	8	2	9	8
*TSP	2	2	2	2	2	2	2	2	2
**k*-TSP	2	18	10	2	2	18	2	10	10
*DT	3	2	2	3	4	4	1	3	14
*PAM	15	2,296	4	17	47	13	701	9	47

*Table 5.2: Number of genes selected for classification by each classification method from nine published microarray data sets. *The reported results are from Tan et al (2005).*

It can be observed from *Table 5.1* that the new *k*-SS method performs excellently well than all the five existing classifiers. Although, *k*-SS, TSP, *k*-TSP and PAM classifiers provided average prediction accuracy in the neighbourhood of 90% while DT and NB provided average prediction accuracy in the neighbourhood of 80%, the *k*-SS classifier outperformed all the five classifiers in six of the nine cases (Colon, DLBCL, Prostate 1 & 2, Lung, GCM) while it performed equally in one case each with NB (Leukemia2) and PAM (Prostate 3). The *k*-TSP method slightly performs better than the *k*-SS method in just one instance (CNS) but uses ten genes as against four used by *k*-SS to achieve almost the same result. In the case for which PAM performs equally with *k*-SS (Prostate 3), the *k*-SS method uses only 2 genes to yield 100% correct prediction while PAM uses as large as 701 genes to achieve the same result (see *Table 5.2*). It can be observed generally that PAM uses more genes for classifications than any other classifiers with very little appreciable relative performance over others.

Based on the estimated average prediction accuracies on all the nine binary classification problems presented in *Table 5.1*, it is very clear that the best classifier is the *k*-SS classifier (95.80%) followed by *k*-TSP (92.01%), then PAM (88.91%), TSP (88.26%), NB (81.17%) and lastly DT (79.25%) in that order.

The usual practice in which the random splitting ratio of 2:1 is used to split the original sample size into training sample (2/3) and test sample (1/3) for the construction and assessment of classifiers respectively has been established in this work to be capable of providing unstable and misleading results. Not only in *k*-SS method, other three classifiers considered (SVM, *k*-NN, PLS-LDA) at four different splitting ratios (1:1, 2:1, 4:1, 19:1) all provided their best prediction performances at 19:1 random splitting ratio for which 95% of the sample is used as training and the remaining 5% is used as the test samples. Therefore, due to very small number of biological subjects that characterizes a typical microarray data, and to truly minimize average prediction error variance, we wish to recommend that 95% of the entire *n* mRNA sample should be used to training the classifiers while the remaining 5% should be set aside as independent test sample to assess their performances while adopting any of the sub-sampling schemes (with or without replacement) for cross-validation.

Since the preliminary feature selection is inevitable in the application of virtually all the proposed classification rules including the *k*-SS method due to huge size of a typical microarray data often encountered, it is important therefore to be conscious of the kind of preliminary feature selection methods to be adopted. Most importantly, care must be taken to ensure that the chosen preliminary selection method does not weed out the potentially relevant genes at the preliminary selection stage. However, since none of the existing preliminary selection methods has been reported to be a super-method that is suitable for all cases of microarray data problems, we have also proposed here, a new classifier-like preliminary feature selection method – *the AUC feature selection method-* that is capable at retaining all the potentially relevant features at the close of its preliminary selection exercise. Unlike some of the existing data pruning methods, this new method employs the *v*-fold cross-validation sub-sampling technique to ensure the stability and consistency of the features selected.

5.2 Discussions and conclusion

In this thesis a novel comprehensive but flexible sequential procedure that simultaneously performs dimension reduction, informative gene selection and accurate prediction of tumour conditions of biological samples in any given microarray study has been proposed. The procedure sequentially selects only the most informative *k* genes that are related to the sub-tumour groups in any high dimensional microarray data set, hence, the name *k-sequential selection (k-SS)* given to the method.

It has been demonstrated in this thesis that the new *k*-SS method competes favourably with some of the existing dimension reduction and classification methods. Eleven publicly available microarray data sets have been used to assess the performance of this new classifier relative to eight other existing methods. In virtually all the cases considered, the *k*-SS method exerts its superiority over other methods in terms of prediction accuracies and biological relevance of genes selected. It is hoped that the ability of the *k*-SS method to identify and select only the biologically relevant transcripts shall facilitate pre-operative predictions of several sub-classes of cancers. This shall tremendously help at determining proper therapeutic measures for various kinds of cancers.

In conclusion, the *k*-SS method is a novel dimension reduction and class prediction method that is capable of selecting the most biologically relevant genes in a clearly understood manner, thereby satisfying the yearnings of molecular biologists, physicians and other health workers who are not only interested in the correct classification of different tumour groups but also want to know, in an unambiguous manner, the kind of genes that are related to different tumour conditions of the mRNA samples.

Apart from its simplicity, the *k*-SS method, unlike the 'black-box' approach of some of the earlier methods, is user friendly because the various steps that lead to optimum gene selection and class prediction can easily be understood by any user with very little statistical background.

The new *k*-SS classifier clearly underscores the fact that good variable selection and response class prediction do not necessarily lies in the complexity of the method adopted, as equally remarked by Tan *et al*(2005). The major tasks of informative genes selection and classification of mRNA samples, as often desirable in microarray studies can be accomplished using a very simple, unambiguous but still efficient procedure like the newly developed *k*-SS procedure in this thesis.

Finally, we want to remark that the algorithms that execute the *k*-SS method are developed using `R` statistical software. All necessary `R` codes we developed for its implementation shall be incorporated into the main `R` library within a very short period to facilitate its availability to any interested users.

5.3 **Suggestions for future studies**

The current form of our new *k*-SS method as proposed in this thesis, like any other methods, presents several opportunities for further improvements in order to enhance its general usage. However, whatever modifications intended at this stage shall be addressed in future research works. Few of the areas that come to mind for the benefit of future studies are highlighted in what follows.

Although, binary classification problems are the most common scenario in microarray studies, the dynamic nature of this research area has brought about a few cases that require multiclass prediction problems. An example of this is the three response groups prediction problem of Beer *et al* (2002) using Affymetrix lung cancer microarray data set or the five class predictions using breast cancer data as described in Perou *et al* (2000). However, the suitability of the *k*-SS method to handle multiclass predictions problems has been conjectured in this work. This particular area of application needs to be given thorough practical treatments to enhance its versatility.

More generally, the biological importance of the genes selected by *k*-SS method has been established in this thesis, this particular advantage of the method need to be further demonstrated within the purview of survival analysis where the selected genes could serve as suitable prognostic factors to predict the survival times of cancer patients. This would particularly discourage the use of either the PLS or PCA components, which are often difficult to interpret, to predict the survival times of cancer patients as adopted in some studies, (Nguyen & Rocke, 2002c; Nguyen, 2005). Using the genes selected by *k*-SS method as predictors in survival models would enable us to establish meaningful biological relationship between the gene expression levels and the survival time or status of individual cancer patients. A related study in this regard is the recent study carried out by Yahya & Ulm (2009) in which some histopathological variables were used as predictors of survival times of breast and small-cell lung cancer patients.

Appendix A

List of some symbols and notations

We present some of the symbols and notations used for the construction of k-SS method.

Symbols/Notations	Descriptions/Functions
$\boldsymbol{X} = (X_1, \dots, X_q)$	q-dimensional vector of expression level of q genes measured on n biological samples.
$Y_i \in \{0,1\}$	Binary response variable indicating the two groups (0,1) of biological subjects.
$\varphi_j(X_j)$	The k-SS classifier using gene X_j, $j = 1, \dots, q$
$\varphi^{m_1,m_2,\dots,m_j}(\boldsymbol{x})$	The k-SS classifier using the gene sets $X^{m_1}, X^{m_2}, \dots, X^{m_j}$
$\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$	Minimum average MER estimated using j genes $X^{m_1}, X^{m_2}, \dots, X^{m_j}$
$\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$	Minimum average MER estimated using $(j+1)$ genes $X^{m_1}, X^{m_2}, \dots, X^{m_j}, X^{m_{j+1}}$
$\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$	Estimated difference of the two minimum average MERs using the first formulation $\hat{\delta}_{j^1}$
$\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$	Estimated difference of the two minimum average MERs using the second formulation $\hat{\delta}_{j^2}$
$\hat{\delta}_{j^s}$, $s = 1,2$	The two minimum average MERs
$E\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}\right) = \mu_{\vartheta}^{m_1,m_2,\dots,m_j}$	Expected value of $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_j}$
$\mathrm{E}\left(\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}\right) = \mu_{\vartheta}^{m_1,m_2,\dots,m_{j+1}}$	Expected value of $\hat{\hat{\vartheta}}^{m_1,m_2,\dots,m_{j+1}}$
$E(\hat{\delta}_{j^1}) = \delta_{j^1}$	Expected value of $\hat{\delta}_{j^1}$
$SN(\lambda^*)$	The Skew-normal density with shape parameter λ^*

Appendix B

R functions

B.1 The R function that implements the *k*-SS method using sub-sampling technique of Monte-Carlo cross-validation (MCCV). The following instructions should be noted for using any of the *k*-SS functions provided here:

i) The response variable Y, the vector of the group labels of biological subjects should be in the first column.

ii) The binary group should be coded 0 for normal, and 1 for tumourous or any other outcomes of interest.

# This function returns preliminary genes selected by the t-statistics, and the misclassification error rates (MERs) from logistic discriminant (LD) rules for each of the preliminarily selected genes.

```
##############################################################
          # dat = Microarray data
          # repetitions = Number of cross-validation runs
          # test.sample = Number of test sample to be predicted/classified
          # alpha = t-statistics' p-value cut-point
##############################################################
mer.select <- function(dat, repetitions, test.sample, alpha)
{

t.selection <- function(dat)
{
t.vec <- c()
for (i in 2:ncol(dat))
{
t.statistic <- abs(t.test(dat[, i] ~ dat[, 1], var.equal = F)$p.value)
t.vec <- c(t.vec, t.statistic)
}
names(t.vec) <- names(dat[-1])
return(t.vec)
}

t.result <- t.selection(dat)
t.result <- t.result[t.result <= alpha]
print(sort(t.result, decreasing = F))
dat <- cbind(dat[, 1], dat[, is.element(names(dat), names(t.result))])

dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

mer.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = repetitions,
                dimnames = list(1:repetitions, names(dat)[-1]))

cat("Repetitions done:", "\n"); utils::flush.console()
for (i in 1:repetitions)
{
```

```
repeat
{
samp <- sample(1:nrow(dat), test.sample)
dat2 <- dat[samp, ]
dat3 <- dat[-samp, ]
if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (j in names(dat)[-1])
{
test.data <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, j])
train.data <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, j])

mod <- glm(response ~ x.variable, dat = train.data, family = "binomial")
pred <- predict(mod, newdat = test.data, type = "response")
mer.mat[i, j] <- sum(abs(test.data$response -
                         ifelse(pred < 0.5, 0, 1))) / length(pred)
}
if (i %in% seq(0, repetitions, round(repetitions/10)))
cat(i, "... "); utils::flush.console()
break
}
}
}
return(list("MER" = mer.mat))
}

MER.results <- mer.select(dat, repetitions, test.sample, alpha)

mer <- apply(MER.results$MER, 2, mean)
mer.ordering <- sort(mer, decreasing=F)
mer.ordering
```

# This function returns the k-SS results at each of the gene selection steps

```
################################################################
         #  dat = Microarray data
         #  ordering = mer.ordering (from the previous out-put)
         #  iterations = Number of cross-validation runs
         #  test.sample = Test sample to be predicted/classified
         #  alpha.range = sequence of positive integer from 1 to 1000 (or any
            preferred number) upon which the range of alpha (0,1) is divided
         #  plot.ROC = F (default). If set to T, the plot of ROC curve is
            provided, otherwise, no ROC curve will be plotted.
         #  first = F (default). If set to T, only the first ROC curve at which
            the k-SS criteria is satisfied will be plotted.
         #  cells = c(0,0), specifies the number of cell space to be created for
            ROC curve plot.
################################################################
```

```
library(ROCR)
library(sn)

sequential.selection <- function(dat, ordering, iterations, test.sample,
                                 alpha.range, plot.ROC = F, first = F,
                                 cells = c(0,0))
{
names(dat)[1] <- "response"
dat <- dat[, c("response", names(ordering))]
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

if(plot.ROC == T && first == F) par(mfrow = cells)

final.result <- matrix(NA, ncol = length(alpha.range), nrow = 9)
Mer.mat <- Brier.mat <- Sens.mat <- Spec.mat <- ppv.mat <-
```

```
npv.mat <- match.matrix <- jaccard.matrix <-
matrix(NA, ncol = length(alpha.range), nrow = iterations)

colnames(final.result) <- colnames(match.matrix) <-
colnames(jaccard.matrix) <- colnames(Mer.mat) <-
colnames(Brier.mat) <- colnames(Sens.mat) <- colnames(Spec.mat) <-
colnames(ppv.mat) <- colnames(npv.mat) <- alpha.range
rownames(final.result) <- c("MER", "Jaccard.Index", "Match.Index",
                              "Brier-Score", "Sensitivity",
                              "Specificity", "Positive PV",
                              "Negative PV", "Number of Genes selected")


selection <- (c(names(ordering)[which(ordering == min(ordering))]))[1]
comparison <- rep(FALSE, length(alpha.range))

cat("Gene added:", "\n"); utils::flush.console()
count <- 0

while(length(selection) < length(ordering))
{
count <- count + 1
mer1.vec <- jaccard.vec <- match.vec <- brier.vec <- spec.vec <-
sens.vec <- ppv.vec <- npv.vec <- R.prediction <- R.true.values <- c()

predicted.mer.matrix <- true.mer.matrix <-
matrix(NA, ncol = iterations, nrow = test.sample)

mer2.mat <- matrix(NA, nrow = iterations,
                     ncol = length(names(ordering)[
                     which(!is.element(names(ordering), selection))]))
colnames(mer2.mat) <- names(ordering)[
                        which(!is.element(names(ordering), selection))]

for (j in 1:iterations)
{
samp <- sample(1:nrow(dat), test.sample)
glm1 <- glm(response ~ ., data = dat[-samp, c("response", selection)],
              family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[samp, -1],
                          type = "response") < 0.5, 0, 1)
probab <- predict(glm1, newdat = dat[samp, -1], type = "response")
mer1 <- sum(abs(pred1 - dat[samp, 1])) / test.sample
R.prediction <- c(R.prediction, probab)
R.true.values <- c(R.true.values, dat[samp, 1])

predicted.mer.matrix[, j] <- pred1
true.mer.matrix[, j] <- dat[samp, 1]

brier.score <- sum((dat[samp, 1] - probab)^2) / test.sample
pred1.all <- ifelse(predict(glm1, newdat = dat[ ,-1],
                              type = "response") < 0.5, 0, 1)
mer1.vec <- c(mer1.vec, mer1)

brier.vec <- c(brier.vec, brier.score)


sensitivity <- (sum(c(pred1.all == dat[ ,c("response")])[
                      which(dat[ ,c("response")] == 1)])
                  /length(dat[ ,c("response")][
                          which(dat[ ,c("response")] == 1)]))
specificity <- (sum(c(pred1.all == dat[ ,c("response")])[
                      which(dat[ ,c("response")] == 0)])
                  /length(dat[ ,c("response")][
                          which(dat[ ,c("response")] == 0)]))
spec.vec <- c(spec.vec, specificity)
sens.vec <- c(sens.vec, sensitivity)
ppv <- (sum(c(pred1.all == dat[ ,c("response")])[
              which(dat[ ,c("response")] == 1)])
                / length(pred1.all[which(pred1.all == 1)]))
npv <- (sum(c(pred1.all == dat[ ,c("response")])[
              which(dat[ ,c("response")] == 0)])
               / length(pred1.all[which(pred1.all == 0)]))
ppv.vec <- c(ppv.vec, ppv)
npv.vec <- c(npv.vec, npv)

for (i in names(ordering)[which(!is.element(names(ordering), selection))])
```

```
{
glm2 <- glm(response ~ .,
              data = dat[-samp, c("response", selection, i)],
              family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[samp, -1],
                  type = "response") < 0.5, 0, 1)
mer2 <- sum(abs(pred2 - dat[samp, 1])) / test.sample
mer2.mat[j, i] <- mer2
}
}

jaccard.mat <- predicted.mer.matrix + true.mer.matrix
jaccard.vec <- apply(jaccard.mat, 2, function(x)
                     {sum(x == 2) / sum(x != 0)})
match.vec <- apply(jaccard.mat, 2, function(x)
                     {sum(x == 2 | x == 0) / length(x)})

mean.mer1 <- mean(mer1.vec)
mean.brier <- mean(brier.vec)
mean.mer2 <- colMeans(mer2.mat)
mer.diff <-  mean.mer1 - min(mean.mer2)[1]

cat("selection.step:", count, "\n"); utils::flush.console()
cat("min.average.MER1:", mean.mer1, "\n"); utils::flush.console()
cat("min.average.MER2:", min(mean.mer2)[1], "\n"); utils::flush.console()
cat("difference.delta1:", mer.diff, "\n"); utils::flush.console()

mean.sens <- mean(sens.vec)
mean.spec <- mean(spec.vec)
mean.ppv <- mean(ppv.vec)
mean.npv <- mean(npv.vec)

cat("genes.selected", selection, "\n"); utils::flush.console()

comparison2 <- comparison

var.mer1 <- sum(mer1.vec * (1 - mer1.vec)) / (iterations^2 *
                                                 test.sample)


var.mer2 <- sum(mer2.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
                (1 - mer2.mat[, which(mean.mer2 == min(mean.mer2))[1]])) /
                (iterations^2 * test.sample)

critical.value <-  qsn(1 - alpha.range * 0.001, shape = 4.0398) *
                         ifelse(var.mer1 == 0 || var.mer2 == 0, 0,
                                sqrt(abs(var.mer1 + var.mer2 -
                                2 * cor(mer1.vec, mer2.mat[,
                                which(mean.mer2 == min(mean.mer2))[1]]) *
                                sqrt(var.mer1 * var.mer2))))

comparison <- mer.diff <= critical.value
criteria <- comparison == comparison2

if(sum(criteria) != length(criteria))
{
filled.before <- sum(!is.na(colSums(final.result)))

final.result[, which(criteria == F)[which(criteria == F) %in%
                 which(is.na(colSums(final.result)) == T)]] <-
                 c(mean.mer1, mean(jaccard.vec, na.rm = T),
                 mean(match.vec), mean.brier, mean.sens, mean.spec,
                 mean.ppv, mean.npv, length(selection))
Mer.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Mer.mat)) == T)]] <- mer1.vec
Brier.mat[,        which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Brier.mat)) == T)]] <- brier.vec
Sens.mat[,         which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Sens.mat)) == T)]] <- sens.vec
Spec.mat[,         which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- spec.vec
ppv.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(ppv.mat)) == T)]] <- ppv.vec
npv.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(npv.mat)) == T)]] <- npv.vec
jaccard.matrix[,   which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- jaccard.vec
```

```
match.matrix[,     which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- match.vec

filled.after <- sum(!is.na(colSums(final.result)))

if (plot.ROC == T && filled.before != filled.after)
{
if (first == T && filled.before == 0 && filled.after == 1)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
plot(perf); abline(a=0, b=1)
}
if (first == F)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
plot(perf, main = paste("alpha-factor:", paste(sort(alpha.range,
                          decreasing = T)[
                          (filled.before + 1):filled.after],
                          collapse = ", ")), sub = paste("AUC =",
                          performance(pred, 'auc')@y.values[[1]]),
                          col = "red"); abline(a=0, b=1)
}
}
}
cat("sequential.result.output:", "\n")
utils::flush.console()
print(final.result)

ifelse(sum(comparison) == length(alpha.range),
         break,
         selection <- c(selection, names(mean.mer2[
                        which(mean.mer2 == min(mean.mer2))])[1]))
}
cat("\n")
return(list("RESULT.MATRIX" = final.result,
              "GENE.SELECTED" = selection,
              "MER.MAT" = Mer.mat, "BRIER.MAT" = Brier.mat,
              "SENS.MAT" = Sens.mat, "SPEC.MAT" = Spec.mat,
              "PPV.MAT" = ppv.mat, "NPV.MAT" = npv.mat,
              "JACCARD.MAT" = jaccard.matrix,
              "MATCH.MAT" = match.matrix, "R.PREDICTION" = R.prediction,
              "R.TRUE.VALUES" = R.true.values))
}

KSS.results <- sequential.selection(dat, ordering, iterations, test.sample,
                                                 alpha.range, plot.ROC = T,
                                                 first = F, cells = c(1,1))
```

B.2 The R function that performs backward checks on the genes selected by *k*-SS method under **B.1**.

```
###############################################################
        #   dat = Microarray data
        #   genes = genes selected by k-SS method
        #   iterations = Number of cross-validation runs
        #   test.sample = test sample to predict/classify
        #   bootstrap = F (default) which uses MCCV. If set to T, it uses
            bootstrap cross-validation.
###############################################################

back.check <- function(genes, iterations, test.sample, dat, bootstrap = F)
{
names(dat)[1] <- "response"
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"
mer.mat <- matrix(NA, nrow = iterations, ncol = length(genes)+1)
colnames(mer.mat) <- c("full.model", genes)
names(dat)[1] <- "response"
for (j in 1:iterations)
{
ifelse(bootstrap == F, samp <- sample(1:nrow(dat),
                                 nrow(dat) - test.sample),
                                 samp <- sample(1:nrow(dat), replace = T))
glm1 <- glm(response ~ ., data = dat[samp, c("response", genes)],
              family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)
mer1 <- mean(abs(pred1 - dat[-samp, 1]))
mer.mat[j, 1] <- mer1

for (i in 1:(length(genes)))
{
glm2 <- glm(response ~ ., data = dat[samp, c("response", genes[-i])],
              family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)
mer2 <- mean(abs(pred2 - dat[-samp, 1]))
mer.mat[j, i + 1] <- mer2
}
}
return(mer.mat)
}
KSS.backward.checks <- back.check (genes, iterations, test.sample, dat,
                                   bootstrap = F)
```

B.3 The R function that implements the proposed AUC preliminary feature selection.

# This code returns the number and types of the preliminarily selected genes as well as their cross-validated AUC estimates.

```
##############################################################
          #  dat = Microarray data
          #  alpha = The chosen size alpha for the AUC test
          #  fold = Number of fold chosen for cross-validation
##############################################################
```

```
library(ROCR)
mer.select <- function(dat, alpha, fold)
{

dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

auc.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = fold,
                  dimnames = list(1:fold, names(dat)[-1]))

groups <- sample(rep(1:fold, len = nrow(dat)))
for (k in 1:fold)
{
repeat
{
dat2 <- dat[groups == k, ]
dat3 <- dat[groups != k, ]

if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (m in names(dat)[-1])
{
```

```
test.dat <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, m])
train.dat <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, m])

mod <- glm(response ~ x.variable, data = train.dat, family = "binomial")
pred <- predict(mod, newdata = test.dat, type = "response")
roc <- prediction(pred, test.dat$response)
auc.mat[k, m] <- performance(roc, 'auc')@y.values[[1]]
}
break
}
}
}

mean.auc <- colMeans(auc.mat)
p.1 <- mean.auc / (2 - mean.auc)
p.2 <- 2 * mean.auc^2 / (1 + mean.auc)
sigma <- (mean.auc * (1 - mean.auc) +
     (sum(dat[, 1]) - 1) * (p.1 - mean.auc^2) +
     (length(dat[, 1]) - sum(dat[, 1]) - 1) * (p.2 - mean.auc^2)) /
     (sum(dat[, 1]) * (length(dat[, 1]) - sum(dat[, 1])))

auc.result <- names(mean.auc)[which(mean.auc >= 0.5 + qnorm(1 - alpha) *
                                                      sqrt(sigma))]
auc.select <- sort(mean.auc[auc.result], decreasing = T)
print(length(auc.select))
return(list(auc.select))
}

AUC.selection <- mer.select(dat, alpha, fold)
AUC.selection
```

B.4 The R function that simulates the estimates of the minimum mean MER differences $\hat{\delta}_{j^1} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}}$ and $\hat{\delta}_{j^2} = \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_{j+1}} - \hat{\hat{\vartheta}}^{m_1,m_2,\ldots,m_j}$ as used by the *k*-SS method.

```
#############################################################
        #  dat = data to be used
        #  repetitions = Number of cross-validation runs
        #  test.sample = the number of test sample to predict/classify
        #  alpha = the t-statistics' p-value cut-point
#############################################################

mer.select <- function(dat, repetitions, test.sample, alpha)
{

t.selection <- function(dat)
{
t.vec <- c()
for (i in 2:ncol(dat))
{
t.statistic <- abs(t.test(dat[, i] ~ dat[, 1], var.equal = F)$p.value)
t.vec <- c(t.vec, t.statistic)
}
names(t.vec) <- names(dat[-1])
return(t.vec)
}

t.result <- t.selection(dat)
t.result <- t.result[t.result <= alpha]
print(sort(t.result, decreasing = F))
dat <- cbind(dat[, 1], dat[, is.element(names(dat), names(t.result))])

dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"
mer.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = repetitions,
                dimnames = list(1:repetitions, names(dat)[-1]))

cat("Repetitions done:", "\n"); utils::flush.console()
for (i in 1:repetitions)
{
repeat
{
samp <- sample(1:nrow(dat), test.sample)
dat2 <- dat[samp, ]
dat3 <- dat[-samp, ]
if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (j in names(dat)[-1])
{
test.data <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, j])
train.data <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, j])

mod <- glm(response ~ x.variable, dat = train.data, family = "binomial")
pred <- predict(mod, newdat = test.data, type = "response")
mer.mat[i, j] <- sum(abs(test.data$response -
                        ifelse(pred < 0.5, 0, 1))) / length(pred)
}
if (i %in% seq(0, repetitions, round(repetitions/10)))
cat(i, "... "); utils::flush.console()
break
}
}
}
return(list("MER" = mer.mat))
}

MER.results <- mer.select(dat, repetitions, test.sample, alpha)

mer <- apply(MER.results$MER, 2, mean)
mer.ordering <- sort(mer, decreasing=F)
mer.ordering
```

# This function returns a matrix of $\hat{\delta}_{j1}$ values whose dimension is [iterations by (mer.ordering - 1)]

###

# dat = Microarray data

# ordering = mer.ordering

# iterations = Number of $\hat{\delta}_{j1}$ to be generated from each gene pair

# repetitions = Number of cross-validation run

###

```
sequential.selection <- function(dat, ordering, iterations, repetitions,
                                                         test.sample)
{
names(dat)[1] <- "response"
dat <- dat[, c("response", names(ordering))]
dat <- as.matrix(dat)
dat <- cbind(dat[, 1], scale((dat)[,2:ncol(dat)], center =T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

Mer.mat <- matrix(NA, nrow = iterations, ncol = length(ordering))
colnames(Mer.mat) <- names(ordering)

cat("Iterations:", "\n")

for (j in 1:iterations)
{
selection <- (c(names(ordering)[which(ordering == min(ordering))]))[1]
while(length(selection) != (ncol(dat) - 1))
{
mer1.vec <- c()
mer2.mat <- matrix(NA, nrow = repetitions,
                     ncol = length(names(ordering)[
                     which(!is.element(names(ordering), selection))]))
colnames(mer2.mat) <- names(ordering)[
                        which(!is.element(names(ordering), selection))]

for (k in 1:repetitions)
{

samp <- sample(1:nrow(dat), test.sample)
glm1 <- glm(response ~ ., data = dat[-samp, c("response", selection)],
                          family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[samp, -1],
                          type = "response") < 0.5, 0, 1)
mer1 <- sum(abs(pred1 - dat[samp, 1])) / test.sample
mer1.vec <- c(mer1.vec, mer1)

for (i in names(ordering)[which(!is.element(names(ordering), selection))])
{
glm2 <- glm(response ~ ., data = dat[-samp, c("response", selection, i)],
                          family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[samp, -1],
                  type = "response") < 0.5, 0, 1)
mer2 <- sum(abs(pred2 - dat[samp, 1])) / test.sample
mer2.mat[k, i] <- mer2
}
}
mer1 <- mean(mer1.vec)
mer2.vec <- colMeans(mer2.mat)
mer.diff <-   mer1 - min(mer2.vec)[1]
Mer.mat[j, names(mer2.vec[
                   which(mer2.vec == min(mer2.vec))])] <- mer.diff
selection <- c(selection, names(mer2.vec[
                                 which(mer2.vec == min(mer2.vec))])[1])
}
if (j %in% seq(0, iterations, round(iterations / 1)))
cat(j, "... "); utils::flush.console()
}
cat("\n")
return(Mer.mat)
}

mini.mean.mer.diffiference <-  sequential.selection(dat, ordering, iterations,
                                                     repetitions, test.sample)
```

B.5 The R function that implements the *k*-SS method using the new AUC preliminary feature selection under the sub-sampling technique of Monte-Carlo cross-validation (MCCV).

# This function returns preliminary genes selected by newly proposed AUC criteria, and the Misclassification error rates (MERs) from logistic discriminant (LD) rules for each preliminarily selected genes.

```
##############################################################
          #  dat = Microarray data
          #  repetitions = Number of cross-validation runs
          #  test.sample = Number of test sample to be predicted/classified
          #  alpha = The chosen size alpha for the AUC test
          #  fold = the number of fold used for cross-validation
##############################################################
```

```
library(ROCR)
mer.select <- function(dat, repetitions, test.sample, alpha, fold)
{

dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"
auc.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = fold,
                  dimnames = list(1:fold, names(dat)[-1]))

groups <- sample(rep(1:fold, len = nrow(dat)))
for (k in 1:fold)
{
repeat
{
dat2 <- dat[groups == k, ]
dat3 <- dat[groups != k, ]

if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (m in names(dat)[-1])
{
test.dat <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, m])
train.dat <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, m])

mod <- glm(response ~ x.variable, data = train.dat, family = "binomial")
pred <- predict(mod, newdata = test.dat, type = "response")
roc <- prediction(pred, test.dat$response)
auc.mat[k, m] <- performance(roc, 'auc')@y.values[[1]]
}
break
}
}
}

mean.auc <- colMeans(auc.mat)
p.1 <- mean.auc / (2 - mean.auc)
p.2 <- 2 * mean.auc^2 / (1 + mean.auc)
sigma <- (mean.auc * (1 - mean.auc) +
          (sum(dat[, 1]) - 1) * (p.1 - mean.auc^2) +
          (length(dat[, 1]) - sum(dat[, 1]) - 1) * (p.2 - mean.auc^2)) /
          (sum(dat[, 1]) * (length(dat[, 1]) - sum(dat[, 1])))

auc.result <- names(mean.auc)[which(mean.auc >= 0.5 +
                              qnorm(1 - alpha) * sqrt(sigma))]

cat("preliminary.features.selected:", "\n")
print(sort(mean.auc[auc.result], decreasing = T))
utils::flush.console()

dat <- cbind(dat[, 1], dat[, is.element(names(dat), auc.result)])

mer.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = repetitions,
                dimnames = list(1:repetitions, names(dat)[-1]))

cat("Repetitions done:", "\n"); utils::flush.console()

for (i in 1:repetitions)
{
repeat
{
samp <- sample(1:nrow(dat), test.sample)
dat2 <- dat[samp, ]
dat3 <- dat[-samp, ]
if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (j in names(dat)[-1])
{
test.data <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, j])
train.data <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, j])

mod <- glm(response ~ x.variable, dat = train.data, family = "binomial")
pred <- predict(mod, newdat = test.data, type = "response")
mer.mat[i, j] <- sum(abs(test.data$response -
                     ifelse(pred < 0.5, 0, 1)))/length(pred)
}
if (i %in% seq(0, repetitions, round(repetitions/10))) cat(i, "... ")
break
}
}
}
return(list("MER" = mer.mat))
}
MER.results <- mer.select(dat, repetitions, test.sample, alpha, fold)
```

```
mer <- apply(MER.results$MER, 2, mean)
mer.ordering <- sort(mer, decreasing=F)
mer.ordering
```

# This function returns the k-SS results at each of the gene selection steps

```
##############################################################
        #   dat = Microarray data
        #   ordering = mer.ordering (from the previous out-put)
        #   iterations = Number of cross-validation runs
        #   test.sample = Test sample to be predicted/classified
        #   alpha.range = sequence of positive integer from 1 to 1000 (or any
            preferred number) upon which the range of alpha (0,1) is divided
        #   plot.ROC = F (default). If set to T, the plot of ROC curve is
            provided, otherwise, no ROC curve will be plotted.
        #   first = F (default). If set to T, only the first ROC curve at which
            the k-SS criteria satisfied will be plotted.
        #   cells = c(0,0), specifies the number of cell space to be created for
            ROC curve plot.
##############################################################
```

```
library(ROCR)
library(sn)

sequential.selection <- function(dat, ordering, iterations, test.sample,
                                 alpha.range, plot.ROC = F, first = F,
                                 cells = c(0,0))
{
names(dat)[1] <- "response"
dat <- dat[, c("response", names(ordering))]
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

if(plot.ROC == T && first == F) par(mfrow = cells)

final.result <- matrix(NA, ncol = length(alpha.range), nrow = 9)
Mer.mat <- Brier.mat <- Sens.mat <- Spec.mat <- ppv.mat <-
npv.mat <- match.matrix <- jaccard.matrix <-
matrix(NA, ncol = length(alpha.range), nrow = iterations)

colnames(final.result) <- colnames(match.matrix) <-
colnames(jaccard.matrix) <- colnames(Mer.mat) <-
colnames(Brier.mat) <- colnames(Sens.mat) <- colnames(Spec.mat) <-
colnames(ppv.mat) <- colnames(npv.mat) <- alpha.range
rownames(final.result) <- c("MER", "Jaccard.Index", "Match.Index",
                              "Brier-Score", "Sensitivity",
                              "Specificity", "Positive PV",
                              "Negative PV", "Number of Genes selected")


selection <- (c(names(ordering)[which(ordering == min(ordering))]))[1]
comparison <- rep(FALSE, length(alpha.range))

cat("Gene added:", "\n"); utils::flush.console()
count <- 0

while(length(selection) < length(ordering))
{
count <- count + 1
mer1.vec <- jaccard.vec <- match.vec <- brier.vec <- spec.vec <-
sens.vec <- ppv.vec <- npv.vec <- R.prediction <- R.true.values <- c()

predicted.mer.matrix <- true.mer.matrix <-
matrix(NA, ncol = iterations, nrow = test.sample)
```

```
mer2.mat <- matrix(NA, nrow = iterations,
                     ncol = length(names(ordering)[
                     which(!is.element(names(ordering), selection))]))
colnames(mer2.mat) <- names(ordering)[
                        which(!is.element(names(ordering), selection))]

for (j in 1:iterations)
{
samp <- sample(1:nrow(dat), test.sample)
glm1 <- glm(response ~ ., data = dat[-samp, c("response", selection)],
              family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[samp, -1],
                          type = "response") < 0.5, 0, 1)
probab <- predict(glm1, newdat = dat[samp, -1], type = "response")
mer1 <- sum(abs(pred1 - dat[samp, 1])) / test.sample
R.prediction <- c(R.prediction, probab)
R.true.values <- c(R.true.values, dat[samp, 1])

predicted.mer.matrix[, j] <- pred1
true.mer.matrix[, j] <- dat[samp, 1]

brier.score <- sum((dat[samp, 1] - probab)^2) / test.sample
pred1.all <- ifelse(predict(glm1, newdat = dat[ ,-1],
                              type = "response") < 0.5, 0, 1)
mer1.vec <- c(mer1.vec, mer1)

brier.vec <- c(brier.vec, brier.score)


sensitivity <- (sum(c(pred1.all == dat[ ,c("response")])[
                      which(dat[ ,c("response")] == 1)])
                  /length(dat[ ,c("response")][
                          which(dat[ ,c("response")] == 1)]))
specificity <- (sum(c(pred1.all == dat[ ,c("response")])[
                      which(dat[ ,c("response")] == 0)])
                  /length(dat[ ,c("response")][
                          which(dat[ ,c("response")] == 0)]))
spec.vec <- c(spec.vec, specificity)
sens.vec <- c(sens.vec, sensitivity)
ppv <- (sum(c(pred1.all == dat[ ,c("response")])[
              which(dat[ ,c("response")] == 1)])
                / length(pred1.all[which(pred1.all == 1)]))
npv <- (sum(c(pred1.all == dat[ ,c("response")])[
              which(dat[ ,c("response")] == 0)])
               / length(pred1.all[which(pred1.all == 0)]))
ppv.vec <- c(ppv.vec, ppv)
npv.vec <- c(npv.vec, npv)

for (i in names(ordering)[which(!is.element(names(ordering), selection))])
{
glm2 <- glm(response ~ .,
              data = dat[-samp, c("response", selection, i)],
              family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[samp, -1],
                  type = "response") < 0.5, 0, 1)
mer2 <- sum(abs(pred2 - dat[samp, 1])) / test.sample
mer2.mat[j, i] <- mer2
}
}

jaccard.mat <- predicted.mer.matrix + true.mer.matrix
jaccard.vec <- apply(jaccard.mat, 2, function(x)
                     {sum(x == 2) / sum(x != 0)})
match.vec <- apply(jaccard.mat, 2, function(x)
                     {sum(x == 2 | x == 0) / length(x)})

mean.mer1 <- mean(mer1.vec)
mean.brier <- mean(brier.vec)
mean.mer2 <- colMeans(mer2.mat)
mer.diff <-  mean.mer1 - min(mean.mer2)[1]

cat("selection.step:", count, "\n"); utils::flush.console()
cat("min.average.MER1:", mean.mer1, "\n"); utils::flush.console()
cat("min.average.MER2:", min(mean.mer2)[1], "\n"); utils::flush.console()
cat("difference.delta1:", mer.diff, "\n"); utils::flush.console()

mean.sens <- mean(sens.vec)
mean.spec <- mean(spec.vec)
mean.ppv <- mean(ppv.vec)
mean.npv <- mean(npv.vec)

cat("genes.selected", selection, "\n"); utils::flush.console()

comparison2 <- comparison

var.mer1 <- sum(mer1.vec * (1 - mer1.vec)) / (iterations^2 *
                                                 test.sample)


var.mer2 <- sum(mer2.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
                (1 - mer2.mat[, which(mean.mer2 == min(mean.mer2))[1]])) /
                (iterations^2 * test.sample)

critical.value <-  qsn(1 - alpha.range * 0.001, shape = 4.0398) *
                         ifelse(var.mer1 == 0 || var.mer2 == 0, 0,
                                sqrt(abs(var.mer1 + var.mer2 -
                                2 * cor(mer1.vec, mer2.mat[,
                                which(mean.mer2 == min(mean.mer2))[1]]) *
                                sqrt(var.mer1 * var.mer2))))

comparison <- mer.diff <= critical.value
criteria <- comparison == comparison2

if(sum(criteria) != length(criteria))
{
filled.before <- sum(!is.na(colSums(final.result)))

final.result[, which(criteria == F)[which(criteria == F) %in%
                 which(is.na(colSums(final.result)) == T)]] <-
                 c(mean.mer1, mean(jaccard.vec, na.rm = T),
                 mean(match.vec), mean.brier, mean.sens, mean.spec,
                 mean.ppv, mean.npv, length(selection))
Mer.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Mer.mat)) == T)]] <- mer1.vec
Brier.mat[,        which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Brier.mat)) == T)]] <- brier.vec
Sens.mat[,         which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Sens.mat)) == T)]] <- sens.vec
Spec.mat[,         which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- spec.vec
ppv.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(ppv.mat)) == T)]] <- ppv.vec
npv.mat[,          which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(npv.mat)) == T)]] <- npv.vec
jaccard.matrix[,   which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- jaccard.vec
match.matrix[,     which(criteria == F)[which(criteria == F) %in%
                   which(is.na(colSums(Spec.mat)) == T)]] <- match.vec

filled.after <- sum(!is.na(colSums(final.result)))

if (plot.ROC == T && filled.before != filled.after)
{
if (first == T && filled.before == 0 && filled.after == 1)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
plot(perf); abline(a=0, b=1)
}
if (first == F)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
plot(perf, main = paste("alpha-factor:", paste(sort(alpha.range,
                          decreasing = T)[
                          (filled.before + 1):filled.after],
                          collapse = ", ")), sub = paste("AUC =",
                          performance(pred, 'auc')@y.values[[1]]),
                          col = "red"); abline(a=0, b=1)
}
}
}
cat("sequential.result.output:", "\n")
utils::flush.console()
```

```
print(final.result)

ifelse(sum(comparison) == length(alpha.range),
         break,
         selection <- c(selection, names(mean.mer2[
                        which(mean.mer2 == min(mean.mer2))])[1]))
}
cat("\n")
return(list("RESULT.MATRIX" = final.result,
              "GENE.SELECTED" = selection,
              "MER.MAT" = Mer.mat, "BRIER.MAT" = Brier.mat,
              "SENS.MAT" = Sens.mat, "SPEC.MAT" = Spec.mat,
              "PPV.MAT" = ppv.mat, "NPV.MAT" = npv.mat,
              "JACCARD.MAT" = jaccard.matrix,
              "MATCH.MAT" = match.matrix, "R.PREDICTION" = R.prediction,
              "R.TRUE.VALUES" = R.true.values))
}

KSS.results <- sequential.selection(dat, ordering, iterations,
                                          test.sample, alpha.range,
                                          plot.ROC = T, first = F,
                                          cells = c(1,1))
```

B.6 The R function that implements the *k*-SS method using bootstrap .632+ sub-sampling scheme under the preliminary feature selection by the *t*-statistics.

# This function returns preliminary gene selection by the t-statistic, and the Misclassification error rates (MERs) from logistic discriminant (LD) rules for each preliminarily selected genes.

```
###############################################################
         #  dat = Microarray data
         #  repetitions = Number of cross-validation runs
         #  alpha = the t-statistics' p-value cut-point
###############################################################
```

```
 mer.select <- function(dat, repetitions, alpha)
{
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

t.selection <- function(dat)
{
t.vec <- c()
for (i in 2:ncol(dat))
{
t.statistic <- abs(t.test(dat[, i] ~ dat[, 1], var.equal = F)$p.value)
t.vec <- c(t.vec, t.statistic)
}
names(t.vec) <- names(dat[-1])
return(t.vec)
}

t.result <- t.selection(dat)
t.result <- t.result[t.result <= alpha]
cat("preliminary.features.selected", "\n")
print(sort(t.result, decreasing = F))
utils::flush.console()
dat <- cbind(dat[, 1], dat[, is.element(names(dat), names(t.result))])

mer.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = repetitions,
                    dimnames = list(1:repetitions, names(dat)[-1]))
cat("Repetitions done:", "\n"); utils::flush.console()
for (i in 1:repetitions)
{
repeat
{
samp <- sample(1:nrow(dat), replace = T)
dat2 <- dat[-samp, ]
dat3 <- dat[samp, ]
if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (j in names(dat)[-1])
{
test.data <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, j])
train.data <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, j])

mod <- glm(response ~ x.variable, dat = train.data, family = "binomial")
pred <- predict(mod, newdat = test.data, type = "response")
mer.mat[i, j] <- 0.632 * sum(abs(test.data$response -
                              ifelse(pred < 0.5, 0, 1))) / nrow(dat) +
                   0.368 * sum(abs(train.data$response -
                              ifelse(mod$fitted.values < 0.5, 0, 1))) /
                              nrow(dat2)
}
if (i %in% seq(0, repetitions, round(repetitions/10)))
cat(i, "... "); utils::flush.console()
break
}
}
}
return(list("MER" = mer.mat))
}

MER.results <- mer.select (dat, repetitions, alpha)
mer <- apply(MER.results$MER, 2, mean)
mer.ordering <- sort(mer, decreasing=F)
mer.ordering
```

# This function returns the k-SS results at each of the gene selection steps

```
##########################################################
          #  dat = Microarray data
          #  ordering = mer.ordering (from the previous out-put)
          #  iterations = Number of cross-validation runs
          #  alpha.range = sequence of positive integer from 1 to 1000 (or any
             preferred number) upon which the range of alpha (0,1) is
             divided
          #  plot.ROC = F (default). If set to T, the plot of ROC curve is
             provided, otherwise, no ROC curve will be plotted.
          #  cells = c(0,0), specifies the number of cell space to be created for
             ROC curve plot.
##########################################################
```

```
library(ROCR)
library(sn)
sequential.selection <- function(dat, ordering, iterations, alpha.range,
                           plot.ROC = F, cells = c(0,0))
{
names(dat)[1] <- "response"
dat <- dat[, c("response", names(ordering))]
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center =T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

if(plot.ROC == T) par(mfrow = cells)

final.result <- matrix(NA, ncol = length(alpha.range), nrow = 2)
Mer.mat <- match.matrix  <- matrix(NA, ncol = length(alpha.range),
                                     nrow = iterations)
colnames(final.result) <- colnames(Mer.mat) <- alpha.range

rownames(final.result) <- c("MER", "Number of Genes selected")

selection <- (c(names(ordering)[which(ordering == min(ordering))]))[1]
comparison <- rep(FALSE, length(alpha.range))

cat("Gene added:", "\n"); utils::flush.console()
count <- 0

while(length(selection) < length(ordering))
{
count <- count + 1

mer1.vec <-  R.prediction <- R.true.values <-
mer1.test.vec <- mer1.train.vec <- c()

mer2.mat <- mer2.test.mat <- mer2.train.mat <-
              matrix(NA, nrow = iterations,
                     ncol = length(names(ordering)[
                            which(!is.element(names(ordering),
                                              selection))]))
colnames(mer2.mat) <- colnames(mer2.test.mat) <-
colnames(mer2.train.mat) <-
names(ordering)[which(!is.element(names(ordering), selection))]

for (j in 1:iterations)
{
samp <- sample(1:nrow(dat), replace = T)
glm1 <- glm(response ~ ., data = dat[samp, c("response", selection)],
                          family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)
probab <- predict(glm1, newdat = dat[-samp, -1], type = "response")

mer1.test <- mean(abs(pred1 - dat[-samp, 1]))
mer1.train <- mean(abs(ifelse(glm1$fitted.values < 0.5, 0, 1) -
                                                       dat[samp, 1]))
mer1 <- 0.632 * mer1.test + 0.368 * mer1.train

R.prediction <- c(R.prediction, probab)
R.true.values <- c(R.true.values, dat[-samp, 1])

mer1.vec <- c(mer1.vec, mer1)
mer1.test.vec <- c(mer1.test.vec, mer1.test)
mer1.train.vec <- c(mer1.train.vec, mer1.train)

for (i in names(ordering)[which(!is.element(names(ordering), selection))])
{
glm2 <- glm(response ~ ., data = dat[samp, c("response", selection, i)],
                          family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)

mer2 <- 0.632 * mean(abs(pred2 - dat[-samp, 1])) +
        0.368 * mean(abs(ifelse(glm2$fitted.values < 0.5, 0, 1) -
                                                          dat[samp, 1]))

mer2.mat[j, i] <- mer2
mer2.test.mat[j, i] <- mean(abs(pred2 - dat[-samp, 1]))
mer2.train.mat[j, i] <- mean(abs(
                               ifelse(glm2$fitted.values < 0.5, 0, 1) -
                                                          dat[samp, 1]))
}
}

mean.mer1 <- mean(mer1.vec)
mean.mer2 <- colMeans(mer2.mat)
mer.diff <-  mean.mer1 - min(mean.mer2)[1]

cat("selection.step:", count, "\n"); utils::flush.console()
cat("min.average.MER1:", mean.mer1, "\n"); utils::flush.console()
cat("min.average.MER2:", min(mean.mer2)[1], "\n"); utils::flush.console()
cat("difference.delta1:", mer.diff, "\n"); utils::flush.console()

cat("genes.selected", selection, "\n"); utils::flush.console()
```

```
comparison2 <- comparison

var.mer1 <- .632^2 * (iterations^2 * nrow(dat[-samp, ]))^(-1) *
                       sum(mer1.test.vec * (1 - mer1.test.vec)) +
            .368^2 * (iterations^2 * nrow(dat))^(-1) *
                       sum(mer1.train.vec * (1 - mer1.train.vec))

var.mer2 <- .632^2 * (iterations^2 * nrow(dat[-samp, ]))^(-1) *
             sum(mer2.test.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
              (1 - mer2.test.mat[, which(mean.mer2 ==
                                   min(mean.mer2))[1]])) +
            .368^2 * (iterations^2 * nrow(dat))^(-1) *
            sum(mer2.train.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
            (1 - mer2.train.mat[, which(mean.mer2 == min(mean.mer2))[1]]))

critical.value <-  qsn(1 - alpha.range * 0.001, shape = 4.0398) *
                        ifelse(var.mer1 == 0 || var.mer2 == 0, 0,
                           sqrt(abs(var.mer1 + var.mer2 -
                                2 * cor(mer1.vec, mer2.mat[,
                                which(mean.mer2 == min(mean.mer2))[1]]) *
                                sqrt(var.mer1 * var.mer2))))
comparison <- mer.diff <= critical.value
criteria <- comparison == comparison2

if(sum(criteria) != length(criteria))
{
filled.before <- sum(!is.na(colSums(final.result)))

final.result[, which(criteria == F)[which(criteria == F) %in%
               which(is.na(colSums(final.result)) == T)]] <-
               c(mean.mer1, length(selection))
Mer.mat[, which(criteria == F)[which(criteria == F) %in%
          which(is.na(colSums(Mer.mat)) == T)]] <- mer1.vec

filled.after <- sum(!is.na(colSums(final.result)))

if (plot.ROC == T && filled.before != filled.after)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
        plot(perf, main = paste("alpha-factor:",
            paste(sort(alpha.range, decreasing = T)[
                  (filled.before + 1):filled.after],
            collapse = ", ")),
            sub = paste("AUC =", performance(pred, 'auc')@y.values[[1]]),
            col = "red"); abline(a=0, b=1)
}
}
cat("sequential.result.output:", "\n")
utils::flush.console()
print(final.result)

ifelse(sum(comparison) == length(alpha.range),
        break, selection <- c(selection, names(mean.mer2[
                              which(mean.mer2 == min(mean.mer2))])[1]))
}
cat("\n")
return(list("RESULT.MATRIX" = final.result, "GENE.SELECTED" = selection,
             "MER.MAT" = Mer.mat, "R.PREDICTION" = R.prediction,
              "R.TRUE.VALUES" = R.true.values))
}
KSS.results <- sequential.selection (dat, ordering, iterations, alpha.range,
                                               plot.ROC = T, cells = c(0,0))
```

B.7 The R function that implements the *k*-SS method using bootstrap .632+ sub-sampling scheme under the new AUC preliminary feature selection.

# This function returns preliminary genes selected by the new AUC feature selection method, and the Misclassification error rates (MERs) from logistic discriminant (LD) rules for each preliminarily selected genes.

```
#################################################################
             #  dat = Microarray data
             #  repetitions = Number of cross-validation runs
             #  alpha = The chosen size alpha for the AUC test
             #  fold = Number of fold chosen for cross-validation
#################################################################
```

```
library(ROCR)
mer.select <- function(dat, repetitions, alpha, fold)
{

dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center = T, scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

auc.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = fold,
                  dimnames = list(1:fold, names(dat)[-1]))

groups <- sample(rep(1:fold, len = nrow(dat)))
for (k in 1:fold)
{
repeat
{
dat2 <- dat[groups == k, ]
dat3 <- dat[groups != k, ]

if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (m in names(dat)[-1])
{
test.dat <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, m])
train.dat <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, m])

mod <- glm(response ~ x.variable, data = train.dat, family = "binomial")
pred <- predict(mod, newdata = test.dat, type = "response")
roc <- prediction(pred, test.dat$response)
auc.mat[k, m] <- performance(roc, 'auc')@y.values[[1]]
}
break
}
}
}

mean.auc <- colMeans(auc.mat)
p.1 <- mean.auc / (2 - mean.auc)
p.2 <- 2 * mean.auc^2 / (1 + mean.auc)
sigma <- (mean.auc * (1 - mean.auc) +
          (sum(dat[, 1]) - 1) * (p.1 - mean.auc^2) +
          (length(dat[, 1]) - sum(dat[, 1]) - 1) * (p.2 - mean.auc^2)) /
          (sum(dat[, 1]) * (length(dat[, 1]) - sum(dat[, 1])))

auc.result <- names(mean.auc)[which(mean.auc >= 0.5 +
                              qnorm(1 - alpha) * sqrt(sigma))]

cat("preliminary.features.selected:", "\n")
print(sort(mean.auc[auc.result], decreasing = T))
utils::flush.console()

dat <- cbind(dat[, 1], dat[, is.element(names(dat), auc.result)])
```

```
  mer.mat <- matrix(NA, ncol = ncol(dat) - 1, nrow = repetitions,
                  dimnames = list(1:repetitions, names(dat)[-1]))

cat("Repetitions done:", "\n"); utils::flush.console()
for (i in 1:repetitions)
{
repeat
{
samp <- sample(1:nrow(dat), replace = T)
dat2 <- dat[-samp, ]
dat3 <- dat[samp, ]
if(length(unique(dat2[, 1])) != 1 && length(unique(dat3[, 1])) != 1)
{
for (j in names(dat)[-1])
{
test.data <- data.frame("response" = dat2[, 1], "x.variable" = dat2[, j])
train.data <- data.frame("response" = dat3[, 1], "x.variable" = dat3[, j])

mod <- glm(response ~ x.variable, dat = train.data, family = "binomial")
pred <- predict(mod, newdat = test.data, type = "response")
mer.mat[i, j] <- 0.632 * sum(abs(test.data$response -
                             ifelse(pred < 0.5, 0, 1))) / nrow(dat) +
                 0.368 * sum(abs(train.data$response -
                             ifelse(mod$fitted.values < 0.5, 0, 1))) /
                             nrow(dat2)
}
if (i %in% seq(0, repetitions, round(repetitions/10)))
cat(i, "... "); utils::flush.console()
break
}
}
}
return(list("MER" = mer.mat))
}
MER.results <- mer.select(dat, repetitions, alpha, fold)
mer <- apply(MER.results$MER, 2, mean)
mer.ordering <- sort(mer, decreasing=F)
mer.ordering
```

# This function returns the k-SS results at each of the gene selection steps

```
##################################################################
          # data = Microarray data
          # ordering = mer.ordering (from the previous out-put)
          # iterations = Number of cross-validation runs
          # alpha.range = sequence of positive integer from 1 to 1000 (or any
            preferred number) upon which the range of alpha (0,1) is
           divided plot.ROC = F (default). If set to T, the plot of ROC curve
           is provided, otherwise, no ROC curve will be plotted.
          # cells = c(0,0), specifies the number of cell space to be created for
            ROC curve plot.
##################################################################
```

```
library(ROCR)
library(sn)
sequential.selection <- function(dat, ordering, iterations, alpha.range,
                           plot.ROC = F, cells = c(0,0))
{
names(dat)[1] <- "response"
dat <- dat[, c("response", names(ordering))]
dat <- as.matrix(dat)
dat <- cbind(dat[,1], scale((dat)[,2:ncol(dat)], center =T , scale = T))
dat <- as.data.frame(dat)
names(dat)[1] <- "response"

if(plot.ROC == T) par(mfrow = cells)

final.result <- matrix(NA, ncol = length(alpha.range), nrow = 2)
Mer.mat <- match.matrix  <- matrix(NA, ncol = length(alpha.range),
                                     nrow = iterations)
colnames(final.result) <- colnames(Mer.mat) <- alpha.range
```

```
rownames(final.result) <- c("MER", "Number of Genes selected")

selection <- (c(names(ordering)[which(ordering == min(ordering))]))[1]
comparison <- rep(FALSE, length(alpha.range))

cat("Gene added:", "\n"); utils::flush.console()
count <- 0

while(length(selection) < length(ordering))
{
count <- count + 1

mer1.vec <-  R.prediction <- R.true.values <-
mer1.test.vec <- mer1.train.vec <- c()

mer2.mat <- mer2.test.mat <- mer2.train.mat <-
              matrix(NA, nrow = iterations,
                     ncol = length(names(ordering)[
                            which(!is.element(names(ordering),
                                              selection))]))
colnames(mer2.mat) <- colnames(mer2.test.mat) <-
colnames(mer2.train.mat) <-
names(ordering)[which(!is.element(names(ordering), selection))]

for (j in 1:iterations)
{
samp <- sample(1:nrow(dat), replace = T)
glm1 <- glm(response ~ ., data = dat[samp, c("response", selection)],
                          family = "binomial")
pred1 <- ifelse(predict(glm1, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)
probab <- predict(glm1, newdat = dat[-samp, -1], type = "response")

mer1.test <- mean(abs(pred1 - dat[-samp, 1]))
mer1.train <- mean(abs(ifelse(glm1$fitted.values < 0.5, 0, 1) -
                                                       dat[samp, 1]))
mer1 <- 0.632 * mer1.test + 0.368 * mer1.train

R.prediction <- c(R.prediction, probab)
R.true.values <- c(R.true.values, dat[-samp, 1])

mer1.vec <- c(mer1.vec, mer1)
mer1.test.vec <- c(mer1.test.vec, mer1.test)
mer1.train.vec <- c(mer1.train.vec, mer1.train)

for (i in names(ordering)[which(!is.element(names(ordering), selection))])
{
glm2 <- glm(response ~ ., data = dat[samp, c("response", selection, i)],
                          family = "binomial")
pred2 <- ifelse(predict(glm2, newdat = dat[-samp, -1],
                          type = "response") < 0.5, 0, 1)

mer2 <- 0.632 * mean(abs(pred2 - dat[-samp, 1])) +
        0.368 * mean(abs(ifelse(glm2$fitted.values < 0.5, 0, 1) -
                                                          dat[samp, 1]))

mer2.mat[j, i] <- mer2
mer2.test.mat[j, i] <- mean(abs(pred2 - dat[-samp, 1]))
mer2.train.mat[j, i] <- mean(abs(
                               ifelse(glm2$fitted.values < 0.5, 0, 1) -
                                                          dat[samp, 1]))
}
}

mean.mer1 <- mean(mer1.vec)
mean.mer2 <- colMeans(mer2.mat)
mer.diff <-  mean.mer1 - min(mean.mer2)[1]

cat("selection.step:", count, "\n"); utils::flush.console()
cat("min.average.MER1:", mean.mer1, "\n"); utils::flush.console()
cat("min.average.MER2:", min(mean.mer2)[1], "\n"); utils::flush.console()
cat("difference.delta1:", mer.diff, "\n"); utils::flush.console()

cat("genes.selected", selection, "\n"); utils::flush.console()

comparison2 <- comparison
var.mer1 <- .632^2 * (iterations^2 * nrow(dat[-samp, ]))^(-1) *
                       sum(mer1.test.vec * (1 - mer1.test.vec)) +
            .368^2 * (iterations^2 * nrow(dat))^(-1) *
                       sum(mer1.train.vec * (1 - mer1.train.vec))

var.mer2 <- .632^2 * (iterations^2 * nrow(dat[-samp, ]))^(-1) *
             sum(mer2.test.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
              (1 - mer2.test.mat[, which(mean.mer2 ==
                                   min(mean.mer2))[1]])) +
            .368^2 * (iterations^2 * nrow(dat))^(-1) *
            sum(mer2.train.mat[, which(mean.mer2 == min(mean.mer2))[1]] *
            (1 - mer2.train.mat[, which(mean.mer2 == min(mean.mer2))[1]]))

critical.value <-  qsn(1 - alpha.range * 0.001, shape = 4.0398) *
                        ifelse(var.mer1 == 0 || var.mer2 == 0, 0,
                           sqrt(abs(var.mer1 + var.mer2 -
                                2 * cor(mer1.vec, mer2.mat[,
                                which(mean.mer2 == min(mean.mer2))[1]]) *
                                sqrt(var.mer1 * var.mer2))))
comparison <- mer.diff <= critical.value
criteria <- comparison == comparison2

if(sum(criteria) != length(criteria))
{
filled.before <- sum(!is.na(colSums(final.result)))

final.result[, which(criteria == F)[which(criteria == F) %in%
               which(is.na(colSums(final.result)) == T)]] <-
               c(mean.mer1, length(selection))
Mer.mat[, which(criteria == F)[which(criteria == F) %in%
          which(is.na(colSums(Mer.mat)) == T)]] <- mer1.vec

filled.after <- sum(!is.na(colSums(final.result)))

if (plot.ROC == T && filled.before != filled.after)
{
pred <- prediction(R.prediction, R.true.values)
perf <- performance(pred, "tpr", "fpr" )
        plot(perf, main = paste("alpha-factor:",
            paste(sort(alpha.range, decreasing = T)[
                  (filled.before + 1):filled.after],
            collapse = ", ")),
            sub = paste("AUC =", performance(pred, 'auc')@y.values[[1]]),
            col = "red"); abline(a=0, b=1)
}
}
cat("sequential.result.output:", "\n")
utils::flush.console()
print(final.result)

ifelse(sum(comparison) == length(alpha.range),
        break, selection <- c(selection, names(mean.mer2[
                              which(mean.mer2 == min(mean.mer2))])[1]))
}
cat("\n")
return(list("RESULT.MATRIX" = final.result, "GENE.SELECTED" = selection,
             "MER.MAT" = Mer.mat, "R.PREDICTION" = R.prediction,
              "R.TRUE.VALUES" = R.true.values))
}
KSS.results <- sequential.selection (dat, ordering, iterations, alpha.range,
                                               plot.ROC = T, cells = c(0,0))
#############################################################################
```

References

[1] Abonyi J & Feil B, *Cluster analysis for data mining and system identification* (2007), Birkhäuser verlang AG, Berlin.

[2] Affymetrix Inc. GeneChip expression analysis technical manual, technical report (2001a), Santa Clara, California.

[3] Affymetrix Inc. New statistical algorithm for monitoring gene expression on GeneChip probe arrays. Technical note, (2001b). http://www.affymetrix.com /pdf/algorithms.pd

[4] Akaike H, *A new look at the statistical model identification*. IEEE Transactions on Automatic Control **19**.6 (1974): 716–723.

[5] Akaike H, *Information measures and model selection*. Bulletin of the International Statistical Institute, **50**(1983): 277-290.

[6] Alizadeh AA, Eisen MB, Davis RE, MAC *et al*, *Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling*. Nature, **403**.6769 (2000): 503-511.

[7] Alon U, Barkai N, Notterman DA *et al, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*. PNAS, **96** (1999):6745-6750.

[8] Ambroise C & McLachlan G, *Selection bias in gene extraction on the basis of microarray gene-expression data*. PNAS, **99**.10 (2002): 6562-6566.

[9] Anderson PK, Borgan Ø, Gill RD & Keiding N, *Statistical Models Based on Counting Processes* (1993), Springer-Verlag, New York.

[10] Arellano-Valle RB, Gomez HW & Quintana FA, *A new class of skew-normal distributions*. Communications in Statistics: Theory and Methods **33** (2004): 1465–1480.

[11] Armando J, Domínguez-Molina, González-Farías G, Ramos-Quiroga R & Gupta AK, *A matrix variate closed skew-normal distribution with applications to stochastic frontier analysis*. Commun. Statist. – Theory & Methods **36** (2007).

[12] Azzalini A, *A class of distributions which includes the normal ones*. Scand. Jour. Stat., **12** (1985): 171-178.

[13] Azzalini A & Capitanio A, *Statistical applications of the multivariate skew normal distributions*. Jour. Royal Stat. Soc., B **61** (1999): 579–602.

[14] Azzalini A, *A note on regions of given probability of the skew-normal distribution. Metron,* **LIX** (2001): 27–34.

[15] Azzalini A, *Further results on a class of distributions which includes the normal ones*. Statistica, **46** (1986): 199-208.

[16] Azzalini A, *Skew-normal family of distributions*. In Kotz, S., Balakrishnan, N., Read, C. B., & Vidakovic, B, editors, Encyclopaedia of Statistical Sciences, **12** (2006): 7780–7785, John Wiley & Sons, New York, second edition.

[17] Azzalini A, *The skew-normal distribution and related multivariate families (with discussion). Scand. J. Statist.* **32** (2005): 159–188 (C/R 189–200).

[18] Azzalini A, Dal-Cappello T & Kotz S, *Log-skew-normal and log-skew-t distributions as model for family income data*. Journal of Income Distribution, **11** (2003): 12–20.

[19] Bamber D, *The area above the ordinal dominance graph and the area below the receiver operating graph*. Jour. Math. Psych. **12** (1975): 387-415.

[20] Baoli L, Shiwen Y & Qin L, *An improved k-nearest neighbor algorithm for text categorization*. Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shenyang, China, (2003).

[21] Beer DG *et al, Gene-expression profiles predict survival of patients with lung adenocacinoma*. Nat. Med., **8** (2002): 816-824.

[22] Bendel RB & Afifi AA, *Comparison of stopping rules in forward regression.* Jour. Amer. Stat. Assoc., 72 (1977): 46-53.

[23] Bennett KP & Campbell C, *Support vector machines: Hype of Hallelujah?* SIGKDD Explorations, **2**.2(2000): 1-13.

[24] Bennett KP & Mangasarian OL, *Robust linear programming discrimination of two linearly inseparable sets*. Optimization Methods and Software, **1** (1992):23–34.

[25] Berkson J, *Application of logistic function to bio-assay*. Jour. Amer. Stat. Assoc., **39** (1944): 357-365.

[26] Berry DA, *Seymour Geisser, 1929–2004*. Jour. Royal Stat. Soc., A **168** (2005): 245-6.

[27] Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S *et al, Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses*. PNAS, **98**.24 (2001):13790-13795.

[28] Bicciato S *et al, Pattern identification and classification in gene expression data using an auto-associative neural network model.* Biotechnology and Bioengineering, **81**.5 (2003): 594-606.

[29] Binder H & Schumacher M, *Adapting prediction error estimates for biased complexity selection in high-dimensional bootstrap sample*. Statistical Applications in genetics and Molecular Biology, 7.1.12 (2008): 1-26.

[30] Bittner M, Meltzer P, Chen Y, Jiang Y *et al, Molecular classification of cutaneous malignant melanoma by gene expression profiling*. Nature, **406**.6795 (2000): 536-540.

[31] Bliss CI, *The calculation of the dosage-mortality curve*. Annals of Applied Biology **22** (1935): 134-167.

[32] Boatright KM & Salvesen GS, *Mechanisms of caspase activation*. Curr. Opin. Cell Biol., **15** (2003): 725–731.

[33] Boatright KM, Renatus M, Scott FL, Sperandio S, *et al, A unified model for apical caspase activation*. Mol. Cell, **11**(2003): 529–541.

[34] Botstein D & Risch N, *Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease*. Nature Genetics Supplement, 33 (2003): 228-237.

[35] Boulesteix A-L & Strimmer K, *Partial least squares: a versatile tool for the analysis of high-dimensional genomic data*. Briefings in Bioinformatics, 8 (2007): 32-44.

[36] Boulesteix A-L& Strimmer K, *Predicting transcription factor activities from combined analysis of microarray and ChIP data: a partial least squares approach*. Theor. Biol. Med. Model, (2005): 2-23.

[37] Boulesteix A-L, Strobl C, Augustin T & Daumer M, *Evaluating microarray-based classifiers: an overview*. Cancer Informatics, **4** (2008): 77-97.

[38] Breiman L, *Better subset selection using the non-negative garrotte*. Technical Report, University of California, Berkeley (1993).

[39] Brier, *Verification of forecasts expressed in terms of probabilities*. Monthly Weather Review, **78** (1950): 1-3.

[40] Broder AJ, *Strategies for efficient incremental nearest neighbor search*. Pattern Recognition, **23**.1-2 (1986): 171–178.

[41] Brown MPS, Grundy WN, Lin D *et al, Knowledge-based analysis of microarray gene expression data by using support vector machines*. Proceedings of the National Academy of Science, USA, **97** (2000): 262-267.

[42] Bruce A, Johnson A, Lewis J, Raff M, Roberts K & Walters P, *Molecular Biology of the Cell* (2002), Fourth Edition. Garland Science, New York & London.

[43] Bura E & Pfeiffer RM, *Graphical methods for class prediction using dimension reduction techniques on DNA microarray data*. Bioinformatics. **19**.10 (2003): 1252 – 1258.

[44] Burges CJC, *A Tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, **2** (1998): 121-167.

[45] Burnside J, Ouyang M, Anderson A, Bernberg E, Lu C, Meyers BC, Green PL, Markis M, Isaacs G, Huang E, & Morgan RW, *Deep Sequencing of Chicken microRNAs*. BMC Genomics **9**.1 (2008): 185. doi:10.1186/1471-2164-9-185. PMID 18430245.

[46] Chakravarti L & Roy, *Handbook of Methods of Applied Statistics,* **1**(1967), John Wiley and Sons, 392-394.

[47] Chu F & Wang L, *Applications of support vector machines to cancer classification with microarray data.* International Journal of Neural Systems, **15**.6 (2005): 475-484.

[48] Cleveland WS, *LOWESS: A program for smoothing scatter plots by robust locally weighted regression*. The American Statistician, **35** (1981): 54.

[49] Cleveland WS, *Robust locally weighted regression and smoothing scatter plots*. Jour. Amer. Stat. Ass., **74** (1979): 829-836.

[50] Cook RD & Lee H, *Dimension reduction in binary response regression*. Jour. Amer. Stat. Soc., 94, (1999): 1187-1200.

[51] Cooper GC, Hausman RE, *The Cell: A Molecular Approach* (2004), 3rd edition, 261–276, 297, 339–344, Sinauer.

[52] Cornfield J, *Joint dependence of the risk of coronary heart disease on serum cholesterol and systolic blood pressure: A discriminant function analysis*. Federation Proceedings, **21**(1962): 58-61.

[53] Cortes C & Mohri M, *AUC optimization vs. error rate minimization*. In Sebastian Thrun, Lawrence Saul, and Bernhard Scholkopf, editors, Advances in Neural Information Processing Systems **16** (2004). MIT Press.

[54] Cortes C & Vapnik VN, Support vector networks. *Machine Learning*, **20** (1995):273–297.

[55] Cover T & Hart P, *Nearest neighbour pattern classification*. Proc. IEEE Trans. Inform. Theory, **11** (1967): 21-27.

[56] Cover TM & Hart PE, *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, **13**.1 (1967): 21–27.

[57] Cristianini N & Shawe-Taylor J, *An introduction to support vector machines* (2000), Cambridge University Press, UK.

[58] Dai JJ, Lieu L & Rocke D, *Dimension reduction for classification with gene expression microarray data*. Statistical Applications in Genetics and Molecular Biology, 5.1.6 (2006): 1-19.

[59] Danial NN & Korsmeyer SJ, *Cell death: critical control points*. Cell, **116** (2004): 205–219.

[60] de Hoon MJL, Imoto S, Nolan J & Miyano S, *Open Source Clustering Software*. Bioinformatics. **20**.9 (2004): 1453-1454.

[61] Dettling M & Buhlmann P, *Boosting for tumour classification with gene expression data*. Bioinformatics, **19.**9 (2003): 1061-1069.

[62] Dice LR, *Measures of the amount of ecological association between species*. Ecology, **26** (1945): 297-302.

[63] Dillon WR & Goldstein M, *Multivariate analysis* (1984), Wiley, New York.

[64] Ding B & Gentleman R, *Classification using generalized partial least squares*. Bioconductor project 5(2004): 1-29.

[65] Dorfman DD, & AIf E, *Maximum likelihood estimation of parameters of signal detection theory and determination of confidence intervals-rating-method data*. J Math Psych **6** (1969): 487-496.

[66] Draper N & Smith H, *Applied regression analysis* (1981), 2nd ed., John Wiley & Sons, New York.

[67] Du YP, Kasemsumran S, Maruo K *et al*, *Ascertainment of the number of samples in the validation set in Monte Carlo cross validation and the selection of model dimension with Monte Carlo cross validation.* Chemometrics and Intelligent Laboratory Systems, **82**(1-2)(2006): 83-89.

[68] Dudoit S & Fridlyand J, *Bagging to improve the accuracy of a clustering procedure.* Bioinformatics, **19.**9 (2003): 1090-1099.

[69] Dudoit S, Fridlyand J & Speed TP, *Comparison of discriminant methods for the classification of tumors using gene expression data.* Jour. Amer. Stat. Assoc., **97** (2002): 77-87.

[70] Dudoit S, Yang YH, Callow MJ & Speed TP, *Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments*. Statistica Sinica, **12** (2002): 111-139.

[71] Efron B & Gong G, *A leisurely look at the bootstrap, the Jackknife and cross-validation*. The American Statistician, **37** (1983): 36-48.

[72] Efron B & Tibshirani R, *An introduction to the Bootstrap* (1993), Chapman & Hall, London. Efron B *Estimating the error rate of a prediction rule: improvements and cross-validation.* Jour. Amer. Stat. Assoc., **78** (1983): 316-331.

[73] Efron B & Tibshirani R, *Improvements on cross-validation: The .632+ bootstrap method*. Jour. Amer. Stat. Assoc., **92** (1997): 548-560.

[74] Efron B, *The efficiency of logistic regression compared to normal discriminant function analysis.* Jour. Amer. Stat. Assoc., **70** (1975): 892-898.

[75] Eisen MB, Spellman PT, Brown PO & Bostein D, *Cluster analysis and display of genome-wide expression patterns*. Proc. Natl. Acad. Sci. USA, **95** (1998): 14863-14868.

[76] Ernst LA, Gupta RK, Mujumdar RB & Waggoner AS, *Cyanine dye labelling reagents for sulfhydryl groups*. Cytometry**10**.1(1989): 3-10. PMID: 2917472.

[77] Everitt B, *Cluster analysis* (1980), 2nd Ed., John Wiley & Sons, Inc. New York.

[78] Fawcett T, *An introduction to ROC analysis*. Pattern Recognition Letters, **27** (2006): 861-874.

[79] Fieller NRJ & Flenley EC, *Statistics of particle size data*. Applied Statistics, 41.1 (1992): 127-146.

[80] Firth D, *Bias reduction, the jeffrey's prior and glim*. In Fahrmeir L, Francis B, Gilchrist R & Tutz G, editors, Advances in GLIM and statistical modelling, (1992): 91-100.

[81] Fix E & Hoges J, *Discriminatory analysis, nonparametric discrimination: consistency properties* (1951), Technical report, Randloph Field, Texas, USAF School of Aviation Medicine.

[82] Furey TS, Cristianini N, Duffy N et al., *Support vector machine classification and validation of cancer tissue samples using microarray expression data*. Bioinformatics,. **16**.10 (2000): 906-914.

[83] Geisser S *Predictive Inference: An Introduction* (1993), CRC Press, USA.

[84] Geman *et al, Classifying gene expression profiles from pairwise mRNA comparisons*. Stat. Applic. Geneti. Molic. Biol., **3**.1.19 (2004): 1-19

[85] Gerds TA & Schumacher M, *Efron-type measures of prediction error for survival analysis*. Biometrics, **63**.4 (2007): 1283-1287.

[86] Giordano TJ, et al, *Distinct transcriptional profiles of Adrenocortical tumours uncovered by DNA microarray analysis*. Am J Pathol, **162**.2 (2003): 521-531.

[87] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, BloomÞeld CD & Lander ES, *Molecular Classification of Cancer: Class Discovery andClass Prediction by Gene Expression Monitoring*. Science, **286**.5439(1999): 531-537.

[88] Gordon GJ, Jensen, RV et al, *Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma.* Cancer Research **62** (2002): 4963–4967.

[89] Green D & Swets JA, *Signal detection theory and psychophysics* (1966), John Wiley and Sons, New York.

[90] Gupta AK, Gonzalez-Farias G & Dominguez-Molina JA, *A multivariate skew normal distribution.* Jour. Multivariate Analysis, 89 (2004): 181-190.

[91] Hand DJ, *Construction and assessment of classification rules* (1997), John Wiley & Sons, New York.

[92] Hanley JA & McNeil BJ, *The meaning and use of the area under a reaceiver operating characteristic (ROC) curve.* Radiology **143** (1982): 29-36.

[93] Hastie T & Tibshirani R, *Classification by pair-wise coupling*, Annals of statistics, **26**(2) (1998): 451-471.

[94] Hastie T, Tibshirani R & Friedman J, *The elements of statistical leaning* (2009), 2nd Ed., Springer, New York.

[95] Hazel JE, Binary *coefficients and clustering in stratigraphy*. Geological Society of America Bulletin, **81**.11 (1970): 3237-3252.

[96] Hedenfalk I, Duggan D, Chen Y, Radmacher M, et al, *Gene-Expression Profiles in Hereditary Breast Cancer.* The New England Journal Medicine,. 344.8 (2001): 539-548.

[97] Hermann T, Patel DJ, *RNA bulges as architectural and cognition motifs.* Structure **8**.3 (2000): R47. doi:10.1016/S0969-2126(00)00110-6. PMID 10745015.

[98] Hertz J, Krogh A, Palmer RG, *Introduction to the theory of neural computation* (1991), Addison-Weasley, Redwood City, CA.

[99] Holland JK, Kemsley EK & Wilson RH, *Use of Fourier Transform Infrared Spectroscopy and Partial Least Squares Regression for the Detection of Adulteration of Strawberry purees*. J Sci Food Agric, 76 (1998): 263-269.

[100] Hosmer DW & Lemeshow S, *Applied logistic regression* (1989), John Wiley & Sons, New York.

[101] Hsieh FS & Turnbull BW, *Nonparametric and semi-parametric estimation of the receiver operating characteristic curve*. The Annals of Statistics **24** (1996): 25-40.

[102] Huber et al, *Variance stabilization applied to microarray data calibration and to the quantification of differential expression*. Bioinformatics,**18** (2002): 896-8104.

[103] Huber W, Heydebreck Av & Vingron M, *Analysis of microarray gene expression data*. (2003) Chichester: John Wiley & Sons.

[104] Huber W, Heydebreck Av & Vingron M, *Low-level analysis of microarray experiments* (2005), Wiley-VCH.

[105] Human Protein Reference Database® (HPRD®), Johns Hopkins University. http://www.biocompare.com/gene/gene_details.asp?Geneid =11229# products

[106] Hwang D, Schmitt WA, Stephanopoulos G & Stephanopoulos G, *Determination of minimum sample size and discriminatory expression patterns in microarray data*. Bioinformatics, **18** (2002): 1184-1193.

[107] Ioannidis JP, *Microarray and molecular research: noise discovery?* The Lancet, **365** (2005): 454-455.

[108] Jaccard P, *Etude comparative de la distribution florale dans une portion des Alpes et des Jura*. Bulletin de la Societe Vaudoise des Sciences Naturelles, **37** (1901): 547-579.

[109] Johnson RA & Wichern DW, *Applied multivariate statistical analysis* (1992), Prentice Hall, New Jersey.

[110] Karush W, *Minima of functions of several variables with inequalities as side constraints* (1939), Master's thesis, Dept. of Mathematics, University of Chicago.

[111] Kepler TB, Crosby L & Morgan KT, *Normalization and analysis for DNA microarray data by self-consistency and local regression*. Genome Biol., 3(2002): RESEARCH0037.

[112] Kerr MK, Martin M & Churchill GA, *Analysis of variance for gene expression microarray data*. Jour. Computational Biology, **7**(2000): 819-837.

[113] Khan J, Wei JS, Ringner M *et al.*, *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks.* Nature Medicine, **7**.6 (2001): 673.

[114] Kleinbaum DG & Kupper LL, *Applied regression analysis and other multivariate methods* (1978), Wadsworth Publishing Company, Inc., Belmont California.

[115] Kondo S, Barna BP, Morimura T, Takeuchi J *et al*, *Interleukin-1 betaconverting enzyme mediates cisplatin-induced apoptosis in malignant glioma cells*. Cancer Research, **55** (1995): 6166–6171.

[116] Kuhn HW & Tucker AW Nonlinear programming. *Proc.* 2*nd Berkeley Symposium on Mathematical Statistics and Probabilities*, (1951): 481–492.

[117] Kuramochi M & Karypis G *Gene classification using expression profiles: a feasibility study*. Int'l Jour. On Artificial Intellegence Tools, 14.4 (2005): 641-660.

[118] Lachenbruch PA, *Discriminant analysis* (1975), Hafner, New York.

[119] Lee J, Park M & Songs S, *An extensive comparison of recent classification tools applied to microarray data.* Computational Statistics and Data Analysis, **48** (2005): 867-885.

[120] Lee M-LT, *Analysis of microarray gene expression data* (2004), Springer, New York.

[121] Lee M-LT *et al*, *The importance of replication in microarray gene expression studies. Statistical methods and evidence from repetitive cDNA hybridizations*. Proc. Natl. Acad. Sc., USA **97** (2000): 9834-9839.

[122] Liu B *et al*, *A combinational feature selection and ensemble neural network method for classification of gene expression data.* BMC Bioinformatics, **5** (2004.): 136.

[123] Lonning PE, Sorlie T & Borresen-Dale A-L, *Genomics in breast cancer - therapeutic implications.* Nature Clinical Practice Oncology, **2.**1(2005): 26-33.

[124] Man MZ, Dyson G, Johnson K, & Liao B, *Evaluating methods for classifying expression data.* Jour. Biopharmaceutical Statistics, **14** (2004): 1065-1084.

[125] Martens H, *Multivariate Calibration.* (1985), Dr. technical thesis, Technical University of Norway, Trondheim Ž.

[126] Martinon F & Tschopp J, *Inflammatory caspases: linking an intracellular innate immune system to autoinflammatory diseases.* Cell, **117** (2004): 561–574.

[127] McCormick GP, *Nonlinear Programming: Theory, Algorithms, and Applications* (1983), John Wiley and Sons, New York.

[128] McCullagh P & Nelder JA, *Generalized Linear Models* (1989), 2nd ed. London: Chapman & Hall.

[129] McLachlan GJ, *Discriminant analysis and statistical pattern recognition* (1992), John Wiley & Sons, Inc., New York.

[130] Mikkola S, Nurmi K, Yousefi-Salakdeh E, Strömberg R, Lönnberg H, *The mechanism of the metal ion promoted cleavage of RNA phosphodiester bonds involves a general acid catalysis by the metal aquo ion on the departure of the leaving group.* Perkin transactions 2 (1999): 1619–26. doi*:*10.1039/a903691a.

[131] Molinaro AM, Simon R & Pfeiffer RM, *Prediction error estimation: a comparison of re-sampling methods*. Bioinformatics, **21**.15 (2005):3301-3307.

[132] Morrison, AM, *Receiver operating characteristic (ROC) curve preparation - A Tutorial*. Boston: Massachusetts Water Resources Authority. Report, 2005: 1-5.

[133] Muller HJ, *The development of the gene theory*. In Leslie C. Dunn (ed.), Genetics in the 20th Century. Essays on the Progress of Genetics During its First 50 Years. (1951), 77-99, MacMillan, New York.

[134] Naik P & Tsai C-L, *Partial least squares estimator for single-index models.* Jour. Royal. Stat. Soc., B **62**.4 (2000): 763-771.

[135] Nelder JA & Wedderburn R, ***G**eneralized Linear Models* Jour. Royal Stat. Soc., A*,* **135**.3 (1972): 370-384.

[136] Nguyen DV & Rocke DM, *Tumour classification by partial least squares using gene expression data*. Bioinformatics,**18** (2002a): 39–50.

[137] Nguyen DV & Rocke DM, *Classification of acute leukemia based on DNA microarray gene expressions using partial least squares*. In Lin, S.M and Johnson,K.F. (eds), *Methods of Microarray Data Analysis*. Kluwer, Dordrecht, (2002b): 109–124.

[138] Nguyen DV & Rocke DM, *Partial least squares proportional hazard regression for application to DNA microarray survival data.* Bioinformatics. 18.12 (2002c): 1625–1632.

[139] Nguyen DV & Rocke DM, *Multi-class cancer classification via partial least squares with gene expression profiles. Bioinformatics*, **18**(2002d): 1216–1226.

[140] Nguyen DV, *Partial least squares dimension reduction for microarray gene expression data with a censored response*. Mathematical Biosciences, **193** (2005): 119–137.

[141] Ochiai A, *Zoogeographical studies on the soleoid fishes found in Japan and its neighbouring regions.* Bulletin of the Japanese Society of Scientific Fisheries, **22** (1957): 526-530.

[142] O'Gorman TW & Woolson RF, *Variable Selection to Discriminate Between Two Groups: Stepwise Logistic Regression or Stepwise Discriminant Analysis? The American Statistician*, **45**.3 (1991): 187-193.

[143] Olby RC, *Mendel no Mendelian?*. History of Science, **17**(1979): 53-72.

[144] Pease AC, Solas D, Sullivan EJ, Cronin MT, Holmes CP, Fodor SP, *Light-generated oligonucleotide arrays for rapid DNA sequence analysis. PNAS* **91**(1994).: 5022-5026. doi:10.1073/pnas.91.11.5022. PMID 8197176.

[145] Peng S, *et al, Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines.* FEBS Letters, **555** 2 (2003): 358-362.

[146] Perou CM, Sùrlie T, Eisen MB *et al*, *Molecular portraits of human breast tumours*. Nature, **406**.17(2000): 747-752.

[147] Pohar M, Blas M & Turk S, *Comparison of Logistic Regression and Linear Discriminant Analysis: A Simulation Study*. Metodološki zvezki, **1**.1 (2004): 143-161.

[148] Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM *et al*, *Prediction of central nervous system embryonal tumour outcome based on gene expression* Nature, **415**.6870 (2002):436-42.

[149] Price ND, Trent J, El-Naggar AK et al *Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leiomyosarcomas.* PNAS, **104**.9 (2007): 3414–3419.

[150] Ramaswamy S, Tamayo P, Rifkin R *et al*, *Multiclass cancer diagnosis using tumor gene expression signatures.* PNAS, **98**.26 (2001): 15149–15154.

[151] Rimkus C, Friederichs J *et al, Microarray-based prediction of tumour response to neoadjuvant radiochemotherapy of patients with locally*

advanced rectal cancer. Clinical Gastroenterology and Hepatology, **6** (2008): 53-61.

[152] Ripley RB, *Pattern recognition and neural networks* (1996), Cambridge University Press, U.K.

[153] Rosipal R & Krämer N, *Overview and recent advances in partial least squares*. Saunders et al (Eds.): SLSFS 2005, LNCS 3940 (2006): 34-51.

[154] Ross W, Rowe T, Glisson B, Yalowich J & Liu L, *Cancer Res.* **44** (1984): 5857.

[155] R Development Core Team, *R: A language and environment for statistical computing*. R foundation for Statistical computing, Vienna, Austria (2007), ISBN 3-900051-07-0, URL http://www.R-project.org.

[156] Salazar M, Fedoroff OY, Miller JM, Ribeiro NS & Reid BR, *The DNA strand in DNAoRNA hybrid duplexes is neither B-form nor A-form in solution.* Biochemistry 32 (1993): 4207–15. PMID 7682844.

[157] SAS Institute Inc., *Logistic regression examples using SAS system* (1995), Version 6, Cary, NC: SAS Institute Inc.

[158] Schwarz G, Estimating the dimension of a model. *Annals of Statistics* **6**.2 (1978):461-464.

[159] Seligman DA & Pullinger AG, *A multiple stepwise logistic regression analysis of trauma history and 16 other history and dental cofactors in females with temporomandibular disorders*. Jour. Orofac Pain, 10.4 (1996): 351-61.

[160] Shalon D, Smith SJ & Brown PO, *A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization.* Genome Res **6** (1996): 639-645. doi:10.1101/gr.6.7.639. PMID 8796352.

[161] Shang C & Shen Q, *Aiding classification of gene expression data with feature selection: A comparative study*. Int'l. Jour. Comput'l. Intelligence Research, **1**.1(2005): 68-76.

[162] Shapiro J & Brutlag DL, *Fold Miner: Structural Motif Discovery Using an Improved Superposition Algorithm*. *Protein Science*, **13** (2004): 278-294.

[163] Shipp MA *et al*, *Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning*. Nat. Med., **8**.1 (2002): 68-74.

[164] Shtatland ES, Cain E & Barton MB, *The perils of stepwise logistic regression and how to escape them using information criteria and the output delivery system*. NESUG paper (2000): 222-226.

[165] Simpson GG, *Notes on the measurement of faunal resemblance*. American Journal of Science, A, **258** (1960): 300-311.

[166] Singh D, Febbo PG, Ross K *et al, Gene expression correlates of clinical prostate cancer behaviour*. Cancer Cell, **1** (2002): 203-209.

[167] Smola AJ & Schölkopf B, *A tutorial on support vector regression*. Statistics and Computing, **14** (2004): 199-222.

[168] Smola AJ, *Learning with Kernels* (1998), PhD thesis, Technische Universität Berlin, GMD Research Series No. 25.

[169] Smyth GK & Speed TP, *Normalization of cDNA microarray data.* Methods, **31** (2003): 265-273.

[170] Smyth GK, Yang YH & Speed TP, *Statistical issues in cDNA microarray data analysis*. (2002), Totowa: Humana Press.

[171] Snedecor GW & Cochran WG, *Statistical Methods* (1989), Eighth Edition, Iowa State University Press.

[172] Sokal RR & Michener CD *A statistical method for evaluating systematic relationships*. University of Kansas Scientific Bulletin, **28** (1958): 1409-1438).

[173] Sokal RR & Sneath PHA, *Numerical taxonomy: The principles and practice of numerical classification.* WH Freeman, San Francisco, (1973).

[174] Speed T, *Statistical analysis of gene expression microarray data* (2003), Chapman & Hall, London.

[175] SPSS Inc., Prentice Hall Inc. (2003), Chicago.

[176] Stata Corporation, Texas 77845 USA. http://www.stata.com/

[177] Statnikov A, Aliferis CF, Tsamardinos L, Hardin D & Levy S, *A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis*. Bioinformatics, **21** (2005): 631-643.

[178] Steinhoff C & Vingron M, *Normalization and quantification of differential expression in gene expression microarrays*. Briefings in Bioinformatics, **7**.2 (2006): 166-177.

[179] Stephens MA, *EDF Statistics for Goodness of Fit and Some Comparisons*. Journal of the American Statistical Association, **69** (1974): 730-737.

[180] Stevens MW, Leong AS-Y, Fazzalari NL, Dowling KD & Henderson DW, *Cytopathology of Malignant Mesothelioma: A Stepwise Logistic Regression Analysis*. Diagnostic Cytopathology, **8**.4 (1992): 333-341.

[181] Stuart RO, Wachsman W, Berry CC *et al, In silico dissection of cell-type-associated patterns of gene expression in prostate cancer*. PNAS, **101**.2 (2004): 615-620.

[182] Su AI, Welsh JB *et al, Molecular classification of human carcinomas by use of gene expression signatures*. Cancer Research, 61 (2001): 7388-7393.

[183] Swets J, *Measuring the accuracy of diagnostic system*, Science, **240** (1988): 1285-1293.

[184] Swets JA, *ROC analysis applied to the evaluation of medical imaging techniques*. Invest. Radiol. **14** (1979): 109-121.

[185] Swets JA, Dawes RM & Monahan J, *Better decisions through science.* Scientific American **283** (2000): 82–87.

[186] Tan AC, Naiman DQ, Xu L, Winslow RL & Geman D, *Simple decision rules for classifying human cancers from gene expression profiles.* Bioinformatics, **21**.20 (2005): 3896-3904.

[187] Tang T, François N, Glatigny A, Agier N, Mucchielli MH, Aggerbeck L & Delacroix H, *Expression ratio evaluation in two-colour microarray experiments is significantly improved by correcting image misalignment.* Bioinformatics **23** (2007): 2686-2691. doi:10.1093/bioinformatics/btm399. PMID 17698492.

[188] Thalappilly S, Sadasivam S, Radha V & Swarup G, *Involvement of caspase 1 and its activator Ipaf upstream of mitochondrial events in apoptosis.* FEBS Journal **273** (2006): 2766–2778

[189] Thomas JG *et al, An efficient and robust statistical modelling approach to discover differentially expressed genes using genomic expression profiles.* Genome Research, **11** (2001): 1227-1236.

[190] Tibshirani R, *Regression shrinkage and selection via the LASSO.* Jour. Royal Stat. Soc., B.**58**.1(1996): 276- 288.

[191] Tseng GC, Oh M-K *et al, Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variation and assessment of gene effects.* Nucleid Acid Research, **29** (2001): 2549-2557.

[192] Valenzuela TD, Roe DJ, Cretin S *et al, Estimating Effectiveness of Cardiac Arrest Interventions: A Logistic Regression Survival Model.* Circulation, 96 (1997): 3308-3313.

[193] Vapnik V *Statistical learning theory* (1998), John Wiley & Sons, New York.

[194] Vapnik VN & Chervonenkis A, *A note on one class of perceptrons.* A*utomation and Remote Control*, 25, 1964.

[195] Vapnik VN & Chervonenkis A, *Theory of Pattern Recognition (in Russian)*(1974). Nauka, Moscow (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung.* 1979*,* Akademie-Verlag, Berlin).

[196] Vapnik VN & Lerner, A *Pattern recognition using generalized portrait method. Automation and Remote Control*, **24** (1963): 774–780.

[197] Vapnik VN, *Estimation of Dependences Based on Empirical Data* (1982). Springer, Berlin.

[198] Vapnik VN, *Statistical Learning Theory* (1998). John Wiley and Sons, New York.

[199] Vapnik VN, *The Nature of Statistical Learning Theory*(1995). Springer, New York.

[200] Volmer M, Bolck A, Woithers BG, *et al, Partial Least-Squares Regression for Routine Analysis of Urinary Calculus Composition with Fourier Transform Infrared Analysis*. Clinical Chemistry, 39/6(1993): 948-954.

[201] Welch BL, *The generalization of "student's" problem when several different population variances are involved*. Biometrika **34** (1947), 28-35

[202] Welsh JB, Sapinoso LM, Su AI *et al, Analysis of Gene Expression Identifies Candidate Markers and Pharmacological Targets in Prostate Cancer*. Cancer Research, **61** (2001): 5974-5978.

[203] Witten IH & Frank E, *Data mining practical machine learning tools and techniques with JAVA implementations* (2000), Morgan Kaufmann Publishers, London.

[204] Wold H, *Estimation of Principal components and related models by iterative least squares*. In: Krishnaiah PR (ed). Multivariate Analysis, New York: Academic press, (1966): 391-420.

[205] Wold H, *Nonlinear Iterative Partial least Squares (NIPALS) modeling: some current developments*. In: Krishnaiah PR (ed). Multivariate Analysis, New York: Academic press, (1973): *383-407.*

[206] Wold H, *Path models with latent variables: The NIPALS approach*. In H.M. Blalock (edition), Quantitative Sociology: International perspectives on mathematical and statistical model building, (1975): 307–357.

[207] Wold S, Wold, Martens H, Wold H, *The multivariate calibration problem in chemistry solved by the PLS method*. In: A. Ruhe, B. Kagstrom (Eds). Proc.

[208] Wolfinger RD, Gibson G *et al, Assessing gene significance from cDNA microarray expression data via mixed models*. Jour. Comput. Biol., **8** (2001): 625-637.

[209] Xu L, Tan AC, Geman D & Winslow RL, *Merging microarray data from separate breast cancer studies provides a robust prognostic test. BMC Bioinformatics* **9**(2008): 125.

[210] Xu L, Tan AC, Naiman DQ, Geman D & Winslow RL, *Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data*. Bioinformatics, **21**(2005): 3905-3911.

[211] Xu Q-S & Liang Y-Z & Du YP, *Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration*. Journal of Chemometrics, **18**.2 (2004): 112-120.

[212] Xu Q-S & Liang Y-Z, *Monte Carlo cross-validation*. Chemometrics and Intelligent Laboratory System, **56**.1 (2001): 1-11.

[213] Yahya WB & Ulm K, *Survival analysis of breast and small-cell lung cancer patients using conditional logistic regression models*. International Journal of Ecological Economics & Statistics, **14**.S09 (2009): 15-35.

[214] Yang YH *et al, Normalization for cDNA microarray data, in microarrays*: Optical technologies and informatics, 4266, Proc. SPIE, Bittner, ML et al, Eds., 141-152.

[215] Yang YH, Dudoit S *et al, Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.* Nucleid Acids Research, **30**.4(2002): e15

[216] Ye J, Li T & Janardan R, *Using uncorrelated discriminant analysis for tissue classification with gene expression data.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, **01**.4 (2004): 181-190.

[217] Yuan J, Shaham S, Ledoux S, Ellis HM & Horvitz HR, *The C. elegans cell death gene ced-3 encodes a protein similar to mammalian interleukin-1b-converting enzyme*. Cell, **75** (1993): 641–652.

[218] Zhang B & Srihari SN *A, Fast Algorithm for Finding k-Nearest Neighbors with Non-metric Dissimilarity*. Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), IEEE, 2002: 1-6.

[219] Zhang H, Yu C-Y & Singer B, *Cell and tumour classification using gene expression data: Construction of forests*. PNAS, **100**.7 (2003): 4168-4172.

[220] Zhang H, Yu C-Y, Singer B & Xiong M, *Recursive partitioning for tumour classification with gene expression microarray data*. PNAS, **98**.12 (2001): 6730-6735.

[221] Zou KH, *Receiver operating characteristics (ROC) literature research* (2002), http://splweb.bwh.harvard.edu:8000/pages/ppl/zou/roc.html

[222] Zucknick M, Richardson S & Stronach EA, *Comparing the characteristics of gene expression profiles derived by univariate and multivariate methods*. Statistical Application in Genetics and Molecular Biology, **7**.1 (2008): 1-31.

Curriculum Vitae

Name:	Waheed Babatunde YAHYA
Sex:	Male
Place of Birth:	Osogbo
Nationality:	Nigerian
Marital Status:	Married *(with children)*

Secondary Education:

- 1980 – 1985 Nawair - ud - deen Grammar School, Osogbo, Nigeria

Pre-University Education & Qualification obtained:

- 1989 - 1992 Osun State College of Education, Ila – Orangun, Nigeria, *National Certificate in Education (N.C.E), Mathematics / Geography* (*Distinctions*)
- 1995 - 1997 Kwara State Polytechnic, Ilorin, Nigeria, *Higher Diploma in Public Accounting & Auditing*

University Education & Qualification obtained:

- Sept. 1997 - May 2001 University of Ilorin, Ilorin, Nigeria, *Bachelor of Science, B.Sc. (Hons.) in Statistics.*
- July 2001 - Dec. 2003 University of Ilorin, Ilorin, Nigeria, *Master of Science (M.Sc.) in Statistics.*
- Dec. 2003 – Sept. 2004 University of Ado-Ekiti, Nigeria, *Postgraduate Diploma in Financial Management (PGDFM).*
- Sept. 2004 - Aug. 2006 University of Ado-Ekiti, Nigeria, *Master of Business Administration (MBA).*
- Dec. 2006 - March 2007 Goethe-Institute, Mannheim (*affiliated to University of Mannheim*), Germany, *Deutsch-Sprachkurs (Intensive 8)*
- April 2007 - June 2009 Ludwig-Maximillians-University of Munich, Munich, Germany, *Ph.D. Statistics (Dr. rer. Nat).*

Work Experience:

- Jan. 1993 - Sept. 1994 ***Sales Representative,*** Doyin Pharmaceuticals Ltd., Nigeria.
- Oct. 1994 - Aug. 1997 ***Sales / Operations Manager,*** Hasdel Oil (Nig.) Ltd., Nigeria.
- Sep. 2001 - Dec. 2003 ***Graduate Assistant,*** Department of Statistics, University of Ilorin, Ilorin, Nigeria.
- Dec. 2003 - Sept. 2007 ***Assistant Lecturer,*** Department of Statistics, University of Ilorin, Ilorin, Nigeria.
- Oct. 2007 - date ***Lecturer II,*** Department of Statistics, University of Ilorin, Ilorin, Nigeria.
- April 2007 - Aug. 2008 Technical University of Munich, Munich, Germany, *Research Assistant to Prof. Dr. Kurt Ulm.*

Merits & Awards:

- The School of Science prize of Osun State College of Education Ila-Orangun, Nigeria for being the *best student in academic performance in the final N.E.C examinations for 1991/1992 session.*
- *Prize for being the best student in academic performance* in the Department of Statistics, University of Ilorin, Nigeria, for 1997/1998 session.
- Deutscher Akademischer Austausch Dienst (DAAD) Scholar.
- STIBET Studentship awards, Technical University of Munich, Munich, Germany for 2007/2008 session.