CLASSIFICATION AND MULTIPLE TESTING FOR MICROARRAY DATA

by

YAUHENIYA CHERKAS

A Dissertation submitted to the

Graduate School – New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in STATISTICS

written under the direction of

Professor Javier Cabrera

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2010

©2010

YAUHENIYA CHERKAS

ALL RIGHTS RESERVED

ABSTRACT OF THE DISSERTATION

CLASSIFICATION AND MULTIPLE

TESTING FOR MICROARRAY DATA

by YAUHENIYA CHERKAS

Dissertation Director:

Professor Javier Cabrera

This thesis aims to provide a solution to the classification and hypothesis testing problems as well as to create a tool to perform clustering, hypothesis testing or classification tasks automatically via simple menu-driven interface.

Since the first appearance of microarrays in 1995, they became a technique for large gene expression screening worldwide. The quantity of data generated from microarray experiments is enormous, requiring new careful methods of analysis of these high-dimensional data. One of the problems encountered when dealing with this type of data is overfitting. Overfitting happens when information selected is related to the condition of interest only by chance.

This thesis consists of four major parts. The first part contains the overview of microarray methodology and current techniques applied to analyze gene expression data.

The second part uses partial least squares themed idea to develop the algorithm where one can control the FDR (false discovery rate) to extract differentially expressed genes in the analysis of gene expression data. The above procedure can be either used

separately or as a part of the scheme where it provides weights that can be used together with another selection method or as a part of ensemble.

The third part of the thesis deals with the problem of comparing several treatments to the control. In the setting where one wants to find a ‘bump’ in measurements of several groups, the test statistic is considered that is based on maximum and minimum of the group mean differences. Then the derived distribution of a proposed test statistic can be used to make inferences.

The fourth part describes the software developed to provide a menu-driven computing environment for data manipulation and analysis. It includes different methods that can be used to compare expression profiles of genes and methods for gene clustering and various visualization and exploration.

Acknowledgements

I would like to express my sincerest gratitude

- To Professor Javier Cabrera for proposing the subject and for supervising this thesis with such great interest. He has optimally supported me with advice and ideas, and I was always welcome to discuss my work and future directions with him. He has been an excellent mentor, introducing me as a scientist, encouraging me to advance and bringing me into contact with many important researchers in different areas.
- To Professors David Tyler and William Strawderman for serving on my dissertation committee and for the superb courses I had with them.
- To Dr. Dhammika Amaratunga for also being on the committee and his wise advise and helpful insights.
- To Professors Zhang and Kolassa for their guidance and help with graduate program requirements understanding.
- To every member of the Rutgers Statistics department for all the knowledge I received for the interesting courses they taught.
- To Rutgers Department of Statistics and Biostatistics for the opportunity to receive my degree here as well as for the good atmosphere and many enjoyable events.

Dedication

To my mom and Pavel.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANOVA : Analysis of Variance

BIC : Bayesian Information Criterion

BH-FDR : Benjamini and Hochberg procedure for controlling FDR

BY-FDR : Benjamini and Yekutieli procedure for controlling FDR

cDNA : complementary Deoxyribonucleic Acid

DLDA : Diagonal Linear Discriminant Analysis

DQDA : Diagonal Quadratic Discriminant Analysis

ECDF : Empirical Cumulative Distribution Function

FDR : False Discovery Rate

FWER : Family-Wise Error Rate

gFWER : generalized Family-Wise Error Rate

gFDR : generalized False Discovery Rate

IRPLS : Marx's Iteratively Reweighed PLS

KNN : K Nearest Neighbor

LRT : Likelihood Ratio Test

LDA : Linear Discriminant Analysis

mRNA : messenger Ribonucleic Acid

NIPALS: Nonlinear Iterative Partial Least Squares

OLS : Ordinary Least Squares

PCA : Principal Component Analysis

PCR : Principal Component Regression

PLS : Partial Least Squares

PLS1 : Univariate PLS

PLS2 : Multivariate PLS (first)

PLS-FDR : PLS with component coefficient approximation based on FDR-corrected p-values

PLS+LDA : Two-step classification – PLS dimension reduction and LDA

QC : Quality-Control

ROC : Receiver Operating Characteristic

RBI : Resampling-Based Inference

RF : Random Forest

RRR : Reduced Rank Regression

SAM : Significance Analysis of Microarray

SIMPLS: Multivariate PLS (second)

SPCA : Supervised Principal Component Analysis

SVM : Support Vector Machines

PART I.

INTRODUCTION AND

MICROARRAY DATA

Chapter 0

Introduction

0.1 Introduction

Currently technology has reached a point that massive datasets, such as microarray data, have established themselves as a permanent part of the statistical analysis. Problems arising from the high-dimensional data include classification tasks as well as the identification of a relevant subset of genes. A large collection of data is available in the public domains and much progress has been made regarding the development of the technologies and the analyses of the data. However, a number of challenges remain, mostly related to the large-scale nature of the data or to the inherent variability of microarray measurements.

0.2 Dissertation Structure

This dissertation is mainly concentrated on the analysis of the high dimensional microarray data. It starts with the introduction and the description of the datasets used for comparisons. Overview of microarray technology is covered in the Chapter 1. Chapter 2 describes statistical methods for the microarray data analysis and identifies challenges associated with them. Chapter 3 is devoted to the partial least squares methodology and its applications to microarrays. It also describes the simple extension of the PLS algorithm to the case of non-continuous response data. In the Chapter 4 we introduce a modified PLS scheme which uses approximations of the regression coefficients for component weights. We compare this methodology to other classification methods and assess its performance in a variety of scenarios. Chapter 5 talks about finding a ‘bump’ in

measurements when comparing several groups to the control. It contains the description of the method based on the distribution of the proposed statistic and its application to two real-life datasets. Other methods considered are Dunnett's and permutation-based tests. Chapter 6 is devoted to the overview of the R software package that includes methods allowing automatic analysis of datasets. Finally, the dissertation is finished with concluding remarks and perspectives for future research.

0.3 Utilized Datasets

There are several real-life datasets used for the performances comparison of various methods. First, for methodology illustration purpose we will use the *pls* package datasets with the continuous response variable. Then we will turn to publicly available datasets used for classification tasks.

Yarn dataset. A data set with 28 near-infrared spectra (NIR) of PET yarns, measured at 268 wavelengths, as predictors, and density as response (density) (Swierenga, de Weijer, van Wijk, and Buydens 1999).

Gasoline dataset. A data set consisting of octane number (octane) and NIR spectra (NIR) of 60 gasoline samples (Kalivas 1997). Each NIR spectrum consists of 401 diffuse reflectance measurements from 900 to 1700 nm.

Leukemia dataset. This dataset contains gene expression levels of n = 72 patients either suffering from acute lymphoblastic leukemia (ALL, 47 cases) or acute myeloid leukemia (AML, 25 cases) and was obtained from Affymetrix oligonucleotide microarrays.

Breast cancer dataset. This dataset monitors p = 7,129 genes in 49 breast tumor samples. The data were obtained by applying the Affymetrix technology.

Colon cancer datase. In this dataset, expression levels of 40 tumor and 22 normal colon tissues for 6,500 human genes are measured using the Affymetrix technology.

Prostate cancer dataset. The raw data comprise the expression of 52 prostate tumors and 50 non-tumor prostate samples, obtained using the Affymetrix technology.

SRBCT dataset. This dataset contains gene-expression profiles for classifying small round blue-cell tumors of childhood (SRBCT) into four classes (neuroblastoma, rhabdomyosarcoma, non-Hodgkin lymphoma, Ewing family of tumors) and was obtained from cDNA microarrays. There are 88 tissues associated with an expression profile of 2,308 genes, already standardized to zero mean and unit variance across genes.

Lymphoma dataset. This dataset contains gene-expression levels of three most prevalent adult lymphoid malignancies. The total sample size is n = 62, and the expression of p = 4,026 genes is documented.

Brain tumor dataset. This dataset contains n = 42 microarray gene expression profiles from five different tumors of the central nervous system. The raw data were originated using the Affymetrix technology and there are 5,597 genes remained.

National Cancer Institute (NCI) dataset. This comprises gene-expression levels of 5,244 genes for 61 human tumor cell lines which can be divided in 8 classes.

Chapter 1

Microarrays

1.1 Introduction

In recent years, a number of novel biotechnologies have enabled biologists to readily monitor genome-wide expression levels. Microarray technologies, which can measure tens of thousands of gene expression values simultaneously in a single experiment, across different conditions and over time, have been widely used in biomedical research. Since they were introduced in the early nineties, they have found many applications, such as classification of tumors, assigning functions to previously unknown genes, grouping genes into functional pathways, etc. Microarray technology is based on the hybridization of RNA from tissues or cells to either cDNA or oligonucleotides immobilized on a glass chip or rarely on a nylon membrane.

With the wealth of gene expression data from microarrays (such as high density oligonucleotide arrays and cDNA arrays) prediction, classification, and clustering techniques are used for analysis and interpretation of the data. Some applications are for example in molecular classification of acute leukemia (Golub et al., 1999), cluster analysis of tumor and normal colon tissues (Alon et al., 1999), clustering and classification of human cancer cell lines (Ross et al., 2000), diffuse large B-cell lymphoma (DL-BCL; Alizadeh et al., 2000), human mammary epithelial cells and breast cancer (Perou et al., 1999, 2000) and skin cancer melanoma (Bittner et al., 2000). These techniques have also helped to identify previously undetected sub-types of cancer (Golub

et al., 1999; Alizadeh et al., 2000; Bittner et al., 2000; Perou et al., 2000). The problem of 'prediction' may come in various forms of applications as well; the prediction of patient survival duration with germinal center B-like DLBCL compared to those with activated B-like DLBCL using Kaplan–Meier survival curves (Ross et al., 2000).

1.2 Biological Background

There are four forms of life namely Eukaryote, Prokaryote (Bacteria), Archean and Viruses. These are distinguished from each other on the basis of the presence or absence of nuclei and well-structured compartments within their cells. Human life falls under the Eukaryote form, which makes this form of life to be of particular interest. A cell is the structural and functional basic unit of a living organism, and is sometimes called the "building block of life". Each cell is a complex system consisting of many different building blocks enclosed in membrane bag (Figure 1.2.1).

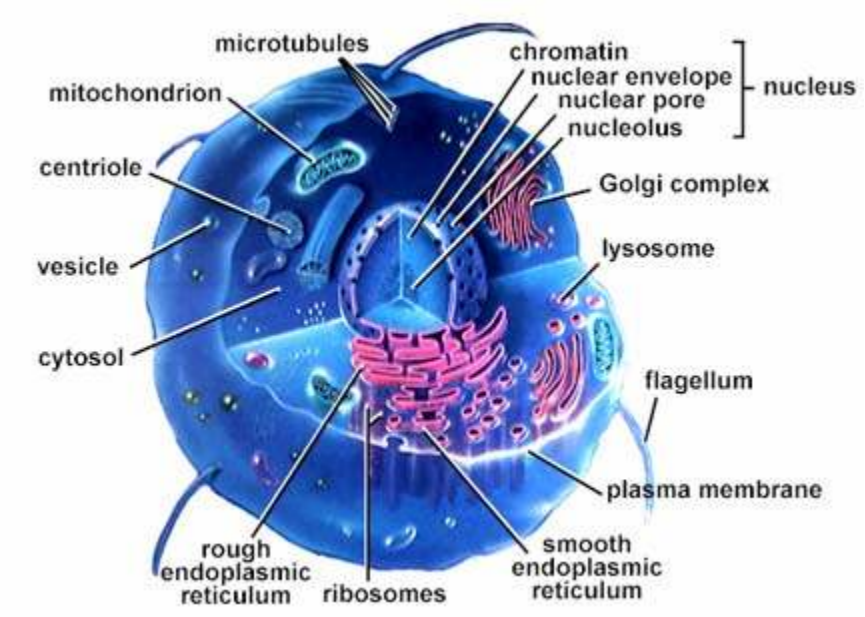


Figure 1.2.1: *A model of a eukaryotic cell (from On-Line Biology Book)*

An essential feature of most (prokaryote and eukaryote) living cells is their ability to grow in an appropriate environment and to undergo cell division. The growth of a single cell and its subsequent division is called the cell cycle.

Cells consist of molecules. There are four basic types of molecules involved in life: (1) small molecules – amino acids, (2) proteins – main building blocks and functional molecules of the cell, (3) DNA – primarily serves as the storage material for genetic information and (4) RNA - can function as a carrier of genetic information, a catalyst of biochemical reactions, an adapter molecule in protein synthesis, and a structural molecule in cellular organelles. Proteins, DNA and RNA are known collectively as biological macromolecules.

DNA is organized as a chain of small molecules, called nucleotides (Figure 1.2.2). There are four different nucleotides Adenosine (A), Guanine (G), Cytosine (C) and Thymidine (T), which are usually referred to as *bases*. DNA may be single or double stranded. DNA forms a double strand by establishing chemical bonds between pairs of complementary bases on the two strands. Adenine binds (only) with Thymine and Guanine binds (only) with Cytosine. This complementarity is a central feature of DNA and it is behind such important processes as replication and gene expression.

Another important molecule is RNA which, like DNA, is constructed from nucleotides, but instead of the Thymine (T), it has a similar molecule, Uracil (U), which is not found in DNA. Because of this difference RNA does not form a double helix, instead they are usually single stranded, but may have complex spatial structure due to complementary links between the parts of the same strand. RNA has different functions in the cell. Mainly, we are interested in its role as an intermediate between DNA and

proteins. It is common to use the term polynucleotide to describe a chain of either DNA or RNA. Some polynucleotide chains are unstable, and, instead of working with them it is common to use their complementary sequence which has to be specifically synthesized. In this case, one talks of cDNA or cRNA.

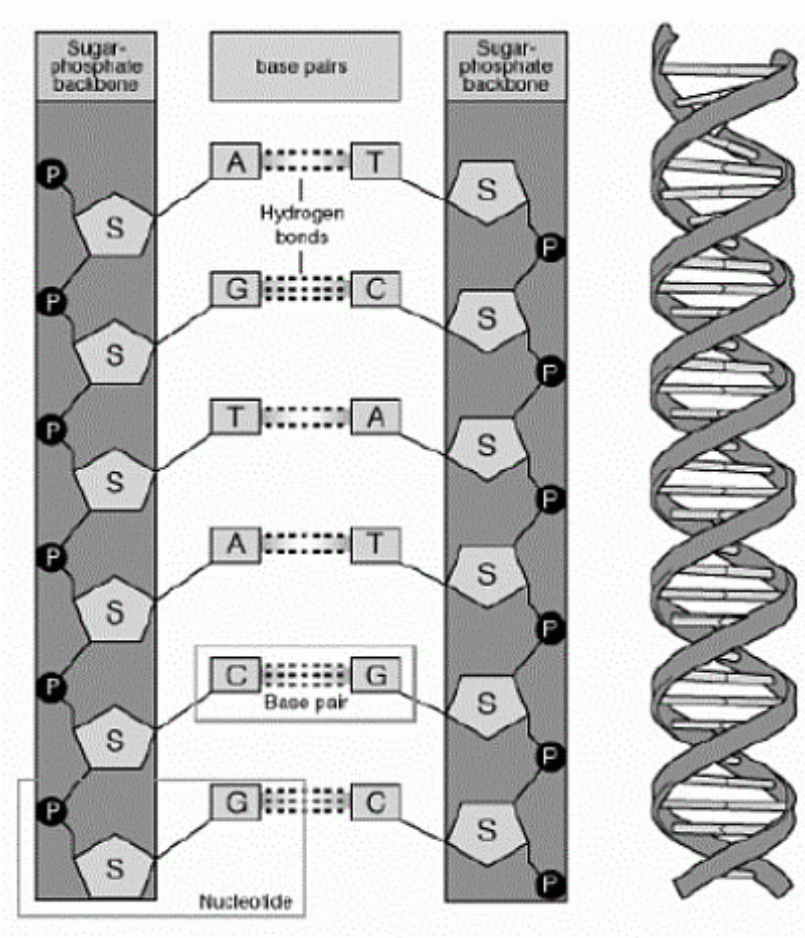


Figure 1.2.2: *The DNA structure*.

A gene is the part of DNA, which contains the genetic code for the chain of amino acids that form a particular protein. The process of deciphering the code, referred to as gene-expression, consists of two steps (Figure 1.2.2).

The first step is called transcription and takes place in the cell core, the nucleus. Messenger ribonucleic acid (mRNA) is created by copying a strand of DNA. The mRNA

then leaves the nucleus and moves into the cytoplasm, where the second part of the process, translation, takes place. Each codon, a triplet of nucleotides of the mRNA sequence, corresponds to an amino acid, which is consecutively attached to a chain forming the protein.

Currently most of the research lies in the field of structural genomics – finding the DNA sequence of various organisms. However, functional genomics which focuses on describing gene functions and gene interactions, and on finding patterns in the expression levels of genes under different conditions is also developing rapidly. Gene-expression that can be quantified by the number of mRNA or proteins produced in the cell is measured by high-throughput technologies, such as microarrays.

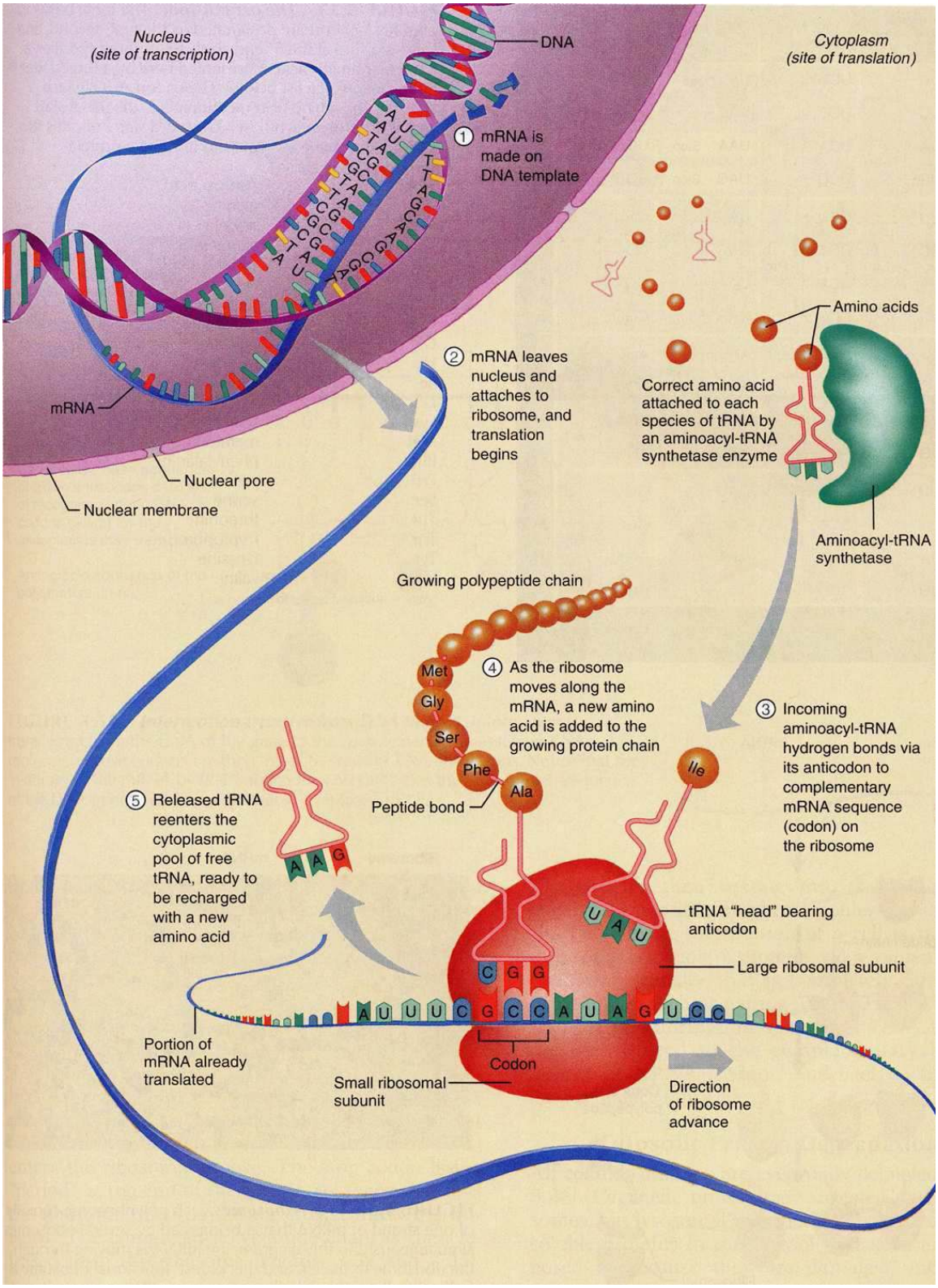


Figure 1.2.2: *Diagram of gene expression (Marieb, 2000)*

1.3 Microarray Technology

To understand the essence of gene expression data, it is necessary to consider the central dogma of molecular biology (Figure 1.3.1) that represents process of reading content of a gene. In order to read the information contained in DNA, first, their functional units, genes are transcribed during transcription into messenger ribonucleic acid (mRNA)), which is based on the complementary DNA strand. mRNA molecules serve as templates for the protein synthesis; they are transported to the cytoplasm and repeatedly read by the ribosomes. Before the mRNA is ready to be translated, it undergoes several processes i.e. splicing, which means that the pre-mRNA is modified to remove certain stretches of non-coding sequences called introns. The stretches that remain include protein-coding sequences and are called exons. Finally, consecutive three nucleotide bases of the mRNA sequence are translated into corresponding amino-acids and linked together to form protein chains.

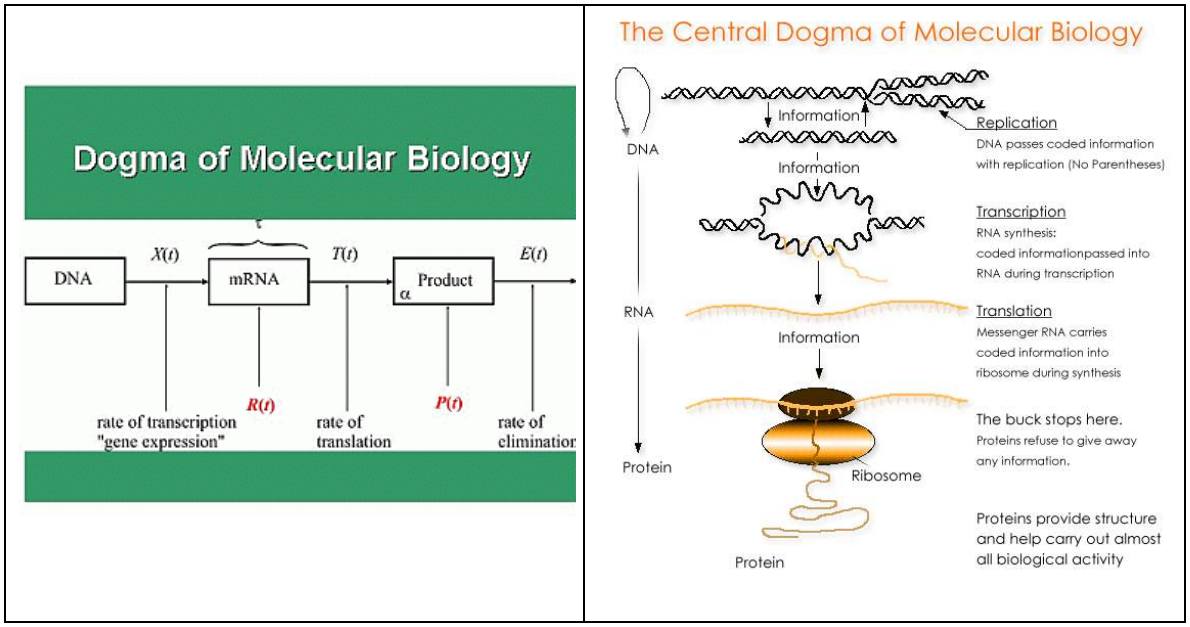


Figure 1.3.1: *The Central Dogma of Molecular Biology.*

In order to understand the role and function of the genes one needs the complete information about their mRNA transcripts and proteins. Unfortunately, exploring the protein functions is very difficult due to their unique 3-dimentional complicated structure and a shortage of efficient technologies. To overcome this difficulty one may concentrate on the mRNA molecules produced by the genes of interest (gene expression) and use this information to investigate the functional roles of the genes. This idea was a motivation for the development of microarrays technique, as a method allowing for studying the interaction between thousands of genes based on their mRNA transcript level.

Although the concept of using microarrays can be traced back 25 years to the introduction of the Southern blot, modern microarray analysis was introduced in 1995 by a Stanford University research team led by Pat Brown and Ron Davis. Their seminal publication was titled "Quantitative monitoring of gene expression patterns with a complementary DNA microarray" and has since been cited over 1,500 times.

A microarray is typically a glass slide onto which DNA molecules are fixed in an orderly manner at specific locations called spots or features (see for example Figure 1.3.2). A microarray may contain thousands of spots and each spot may contain a few million copies of identical DNA molecules that uniquely correspond to a gene. There are several DNA microarray technologies. Currently, two approaches are prevalent: cDNA arrays and oligonucleotide arrays. Both have notable and distinct advantages.

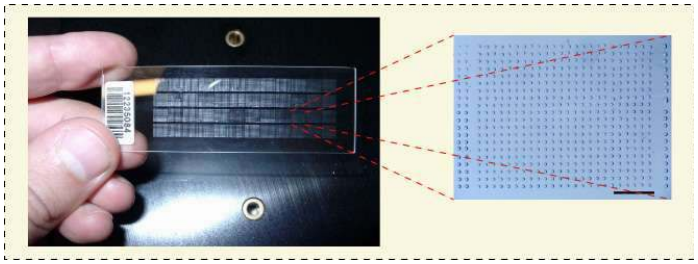


Figure 1.3.2: *Microarray*

1.3.1 cDNA Microarrays

A typical two-channel cDNA microarray (for a graphical display see Figure 1.3.3) is constructed as follows. Messenger RNA (mRNA) from two different biological samples is reverse-transcribed into cDNA, labeled with either green (Cy3) or red (Cy5) dye, and hybridized to DNA sequences which have been spotted onto a glass slide prior to the hybridization. Corresponding to the dyes and different absorption frequencies, the biological signals in the samples are referred to as channels. After hybridization, a laser scanner measures dye fluorescence of each color at a fine grid of pixels. Higher fluorescence indicates higher amounts of hybridized cDNA, which in term indicates a higher gene-expression in the sample. A spot typically consists of a number of pixels. Image analysis algorithms assign pixels to a spot and produce summaries of fluorescence at each spot, as well as summaries of fluorescence in the surrounding unspotted areas (background).

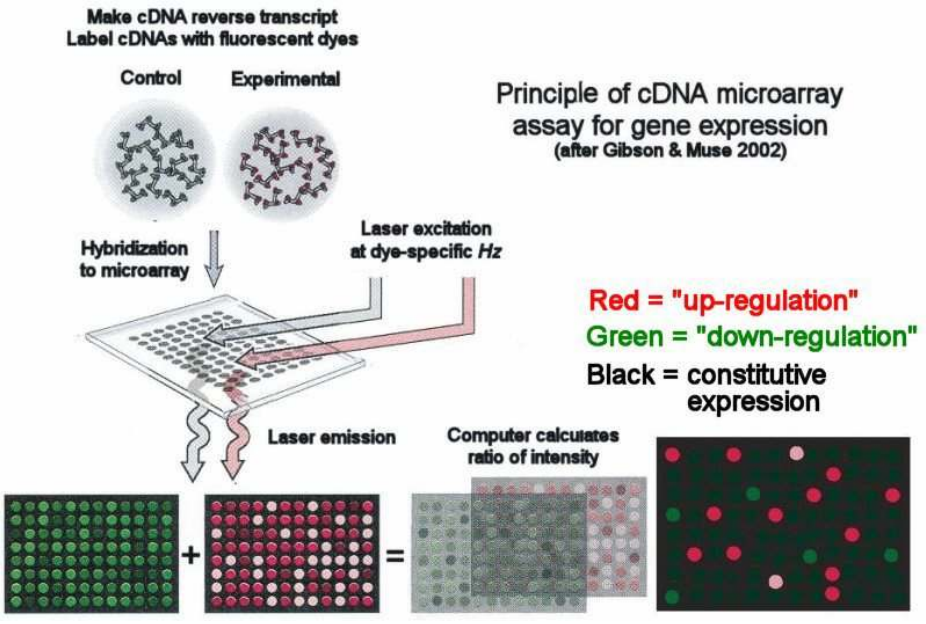


Figure 1.3.3: Two-channel *cDNA processing*

For each spot on the array, a typical output consists of at least four quantities, one of each color (channel) for both the spot and the background. The use of two channels allows for measurement of relative gene-expression across two sources of cDNA, controlling for the amount of spotted DNA. One way of analyzing two channel cDNA arrays is to take the ratios of intensities at each spot. The advantage of the dual channel approach is that it prevents problems in the data that could be caused by variable concentrations of DNA material spotted per DNA sequence. Since both labeled cDNAs compete for the same spot, the relative ratio is still accurate even if the amount of spotted material varies from spot to spot.

The spots of DNA correspond to multiple pixels. An image analysis algorithm first determines the region of the grid containing the spot. One of many types of segmentations techniques is then used to determine the set of pixels belonging to that spot, also referred to as the foreground, and those belonging to the background region. The signal intensity of all pixels belonging to the fore- or background is then summarized per spot by taking the mean or median value. Thus, an output file is created, per label, containing the summarized signal intensity values.

The image files of the two labels are also pseudo-colored and merged by an image analysis algorithm, producing a microarray image. The red or green spots indicate the presence of mRNA from the test or reference population, respectively. If mRNA from both groups is present, the spot has a yellow color. Black spots indicate that no hybridization took place for the particular probe.

1.3.2 Oligonucleotide Microarrays

The second platform to measure gene expression levels is the high-density oligonucleotide array. Here silicon chips contain probes consisting of short oligonucleotide strands, synthesized or deposited on their surface. There exist many types of oligonucleotide arrays. The most popular array type is the Affymetrix GeneChip (Figure 1.3.4). It is composed of 11-20 pairs of oligonucleotides, each of length of 25 base pairs. The two types of probes in each pair are either perfect match (PM) and taken from the gene sequence, or mismatch (MM) and created by changing the middle (13th) base of the PM sequence to reduce the rate of specific binding of mRNA for that gene. The goal of MMs is to control for nonspecific binding of mRNA from other parts of the genome.

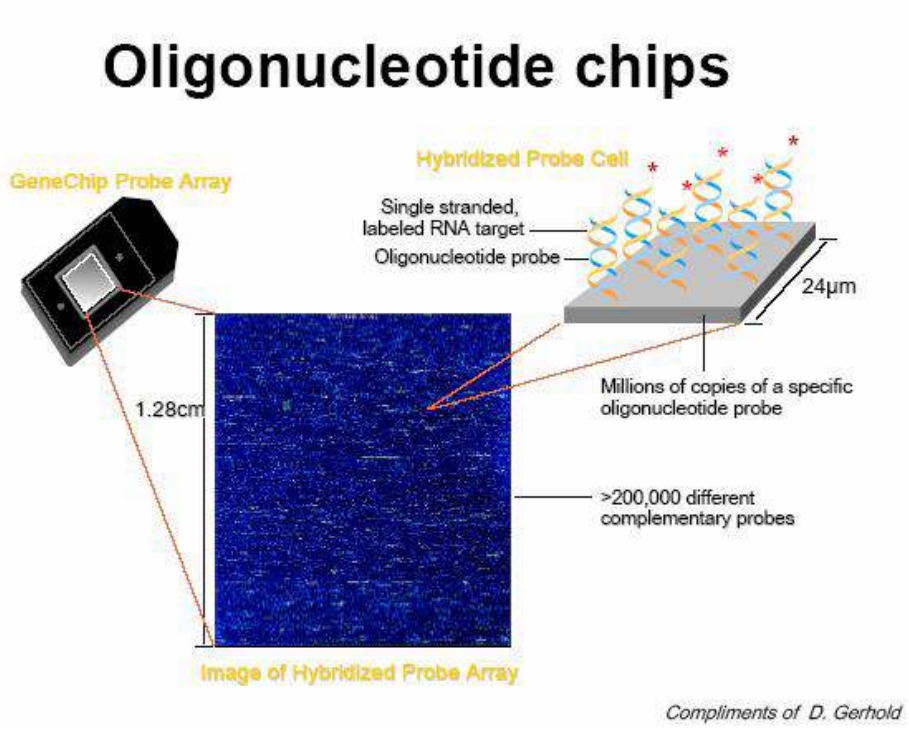


Figure 1.3.4: *Oligonucleotide Chip*

The processing of an oligonucleotide array is, to some extent, similar to that of a cDNA microarray. mRNA, extracted from the tissue under study, is labeled with a fluorescent dye and allowed to hybridize with the probes on the chip. The chip is then scanned to obtain an image. Contrary to cDNA microarrays, GeneChips are one-channel arrays, containing only one biological sample per chip. The different colors on the image indicate the hybridization intensity. The difference in signal intensity, between the perfect and mismatch probes, averaged across all probe pairs of a set, provides an estimate of the gene-expression.

1.3.3 Comparison of cDNA and Oligonucleotide Microarrays

Each of the cDNA and oligonucleotide arrays has their own benefits and disadvantages. cDNA microarrays can be prepared directly from the isolated cDNA clones. Once a set of corresponding PCR products has been generated, microarrays can be created in multiple versions containing the entire set of cDNA sequences, resulting in large-scale arrays for identification of differentially expressed genes of interest or small-scale arrays suitable for specific research applications. They are generally easier to analyze and more flexible. The most important advantage of cDNA microarrays is that they are less expensive to make than a single nucleotide array. However, cDNA microarrays rely on the use of multiple fluorescent dyes. As a result, the comparisons between signal measurements of different colors are subject to dye bias. On the other hand, the approach of using two biological samples per cDNA array leads to a reduction in the between-array variability.

Oligonucleotides can be synthesized either in plates or directly on solid surfaces making it easier to prepare the DNA probes. Also, the probes can be designed to represent unique gene sequences such that cross-hybridization between related gene sequences is minimized to a degree dependent upon the completeness of available sequence information. Oligonucleotide arrays only deal with one biological sample per chip. Thus, twice as many arrays are needed. This makes oligonucleotide microarrays more expensive. However, the fact that they are designed to estimate absolute levels of gene-expression makes them more easily comparable to arrays from different experiments.

Chapter 2

Statistical Data Analysis

2.1 Introduction

In the microarray setting the following are the statistical components of a microarray experiment (Allison et al. 2006):

- Design – The development of an experimental plan to maximize the quality and quantity of information obtained.
- Preprocessing – Processing of the microarray image and normalization of the data to remove systematic variation. Other potential preprocessing steps include transformation of data, data filtering and, in the case of two-color arrays, background subtraction.
- Inference and/or classification – Inference aims at testing statistical hypotheses (these are usually about which genes are differentially expressed). Classification refers to analytical approaches that attempt to divide data into classes with no prior information (unsupervised classification) or into predefined classes (supervised classification).
- Validation of findings – It is the process of confirming the validity of the inferences and various conclusions drawn in the study.

2.2 Design

The importance of design of experiments (DOE) for microarray studies was emphasized by Kerr and Churchill (2001). The problem of designing a microarray

experiment can be decomposed into three distinct layers. First, replication of biological samples is essential in order to draw conclusions that are valid beyond the scope of the particular samples that were assayed. Second, technical replicates increase precision and provide a basis for testing differences within treatment groups. Third, duplication of spotted clones on the microarray slides increases precision and provides quality control and robustness to the experiment.

The basic variation of gene expression data is due to microarray experiments performed with replication. The amount of data gained, quality of data, assessment of the sources of variation, estimation of error variation, and precision of estimates among others are factors contributing to the choice of the design. Usually three types of replication are recognized: (1) spot to spot, (2) array to array, and (3) subject to subject. The replication of spots (i.e., genes) is achieved by depositing probes for the same genes multiple times on the array. Array to array replication refers to multiple hybridizations using the same mix of RNA source. The third type of replication is sampling multiple individuals. The first assesses within array variation (spot-to-spot variation), the second between array variation, and the third biological variation.

Kerr and Churchill emphasized principles of DOE. They described designs commonly used in practice called the reference and loop design (see Figure 2.2.1). The term reference design refers to the original design, where every sample is compared with a common reference. With T treatments and k replicates per treatment, we use kT arrays. If there are technical dye-swaps, these are averaged to form 1 replicate. For more than two varieties, Kerr and Churchill proposed a loop structure design, which collects twice as much data on the varieties of interest than the reference design. A loop is balanced for

dye effects and has two replicates at each node. For T treatments using Tk arrays we have 2k replicates as compared to a reference design for which the same number of arrays yields only k replicates. In these designs, varieties are balanced with respect to dyes.

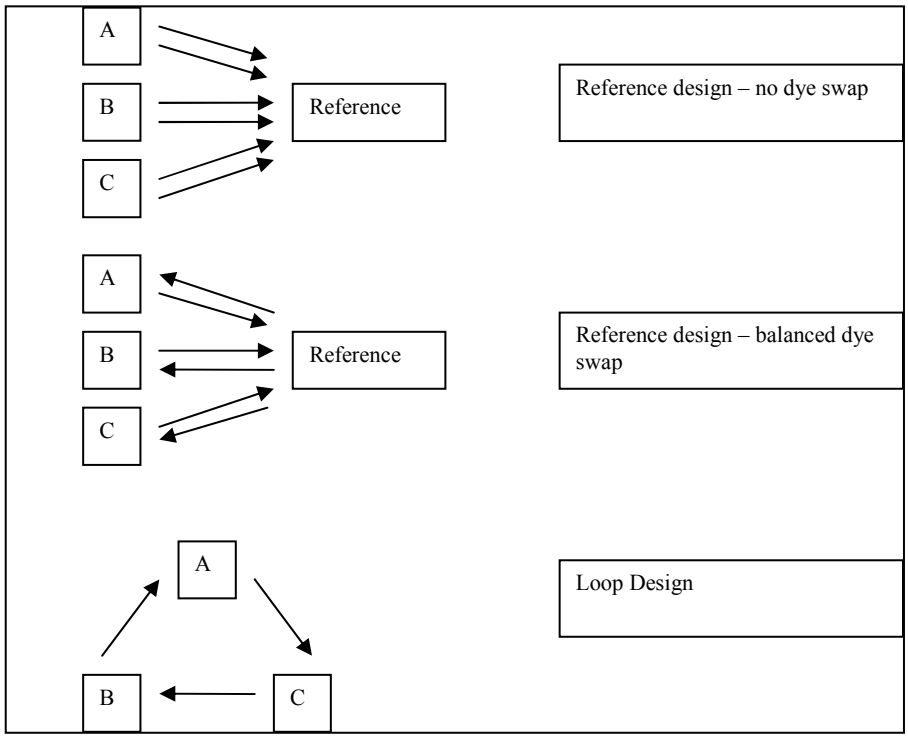


Figure 2.2.1: *Some basic designs for 2-channel microarrays*.

Despite potential gains, carefully designed experiments have not been widely adopted in microarray studies. One reason is that reference design, although inefficient, can be easily extended to more samples by simply adding another array using the same reference. Other reasons are associated cost, physical limitations of the experimental procedure, and innovations in sample preparation, labeling, and detection.

2.3 Preprocessing

Before any kind of microarray data can be analyzed for differential expression several steps must be taken. Raw data must be quality assessed to ensure its integrity. Unprocessed raw data will always be subject to some form of technical variation and thus must be preprocessed to remove as many unwanted sources of variation as is possible, to ensure that results are of the highest attainable level of accuracy. Ideally, the data being assayed should be preprocessed using several different methods, the results of which should be compared to identify which method is of the highest level of suitability. The most appropriate method should then be used to preprocess the raw data before differential expression analysis.

The pre-processing steps include the image analysis, quality control of arrays, background subtraction, summarization of intensities (for oligo microarrays), and normalization (within-and-across) arrays.

Image Analysis

The processing of scanned microarray images can be separated into three major tasks (Yang YH, et al. 2001):

1. Addressing or gridding is the process of identifying the target areas or the combined area of a spot and its background (usually performed by a software).
2. Segmentation allows the classification of pixels either as foreground or as background. According to the geometry of the spots they produce, existing segmentation methods can be categorized into fixed-circle, adaptive circle, adaptive shape and histogram segmentation.

3. Intensity extraction (reduction) involves calculating, for each spot on the array, foreground and background intensities, and possibly, quality measures. For the spot intensity calculation one can use various summary values: mean or median values of the pixel intensities, total sum, ratio, as well as weighted or trimmed mean. Comparisons performed by Yang et al. (2002c) indicated that the differences among the algorithms had very small impact on the spot intensity values. Background intensity calculation methods can be divided into four categories (i) local background intensities, (ii) morphological opening, (iii) constant background, (iv) no adjustment. The choice of background adjustment method can have a large impact on the final output.

Various methods for appropriate quantification of spots on microarrays differ mainly in a way of how spot segmentation and distinguishing foreground from background intensities are carried out.

Quality Control of Arrays

Several methods have been proposed to develop microarray quality-control (QC) measures that quantify the measurement quality for any particular array (for example, using a graphical approach, Chen et al. 2004). A simple quality control procedure can be established at the moment when the spotted image is stored in the database by running a procedure that produces the following items (Amaratunga and Cabrera 2004):

1. An image quality graph could be used to detect specific problems with the array.
2. A side-by-side display of boxplots of gene-expression measures for the sequence of arrays, or a set of summaries based on them, could be used to check whether there are any changes between the arrays.

Normalization

Before multiple microarray measurements can be integrated into a single analysis, the reported measurements need to be normalized, or modified (possibly corrected), to make them comparable. Normalization is useful for a number of situations including: (i) within-slide comparison (ii) multiple-slide comparison, and (iii) paired-slide comparison for dye-exchange experiments (Yang et al., 2001a). It is a matter of adjusting the overall brightness of each scanned microarray image, assuming that the quantity of RNA applied to an array is equal between the arrays.

Regardless of array design, normalization following image acquisition requires two sequential steps: selection and calibration of data derived from genes known not to be affected by the experimental conditions under investigation (called 'invariant' genes).

First, a group of non-differentially expressed or invariant genes has to be identified. Selection criteria include proportion of genes that are expected to change across samples and the availability of control DNA sequences. The following methods have been used:

1. All genes or global normalization (may include trimming of upper/lower extreme values): the assumption underlying this approach is that the total mass of mRNA labeled with either Cy3 or Cy5 is equal. While the intensity for any one spot may be higher in one channel than the other, when averaged over thousands of spots in the array, these fluctuations should average out. Consequently, the total integrated intensity across all the spots in the array should be equal and the ratio of the arithmetic mean equal to one.

2. Housekeeping genes: In the past, the expression levels of housekeeping genes were assumed to be constant and were frequently used to normalize microarray expression data (Camerer et al., 2000). However, it has been found that housekeeping genes are occasionally regulated, too (Foss et al., 1998; Schmittgen and Zakrajsek, 2000; Neuvians et al., 2005). Using housekeeping genes to normalize expression data could, therefore, lead to erroneous conclusions (Yu et al., 2000). Global normalization and normalization to housekeepers may be used when comparing similar samples or when not many changes are assumed. However, if the number of predetermined housekeeping genes is small or their intensities do not cover the full range of signal intensities, this approach may not provide a good fit for non-linear normalization (Tseng et al., 2001).
3. Exogenous control genes: In contrast, exogenous control genes to normalize microarray data is a universally applicable normalization strategy as it does not depend on assumptions like the ones described above. Obviously, external control RNAs should be chosen not to cross-hybridize with RNA from the organism being studied, but should be similar in their general characteristics.
4. Genomic DNA: The rationale behind normalization with genomic DNA is that it represents a constant copy number for a given mass of DNA.
5. Algorithmic selected: Non-differentially expressed genes may be estimated solely by mathematical algorithms instead of biological criteria. This may be achieved by a rank-invariant method that selects signals from spots where the difference of the rank of the Cy3 and Cy5 signals are very close to each other and where the rank of the mean of replicate spots is not within the highest/lowest ranks overall.

For the second step one has to estimate normalization constant or function (linear or non-linear) for either signals or ratios using set of above invariant genes. Irizarry et al. reviews normalization procedures for microarray data. For cDNA arrays the normalization procedure presented in Dudoit et al. (2002) has worked well in practice. For this procedure for each array, a loess curve is fitted to the MVA plot (Mean difference of intensities for two dyes Versus Average intensities for two dyes, Yang et al. 2001, Heldermans et al. 2007) of intensities of the red and green labels and the residuals are considered the normalized log ratios. For normalizing GeneChip arrays various methods have been proposed and reviewed by Bolstad et al. (2002). Quantile normalization was found to perform best. The goal of quantile normalization is to make the distribution of probe intensities the same for arrays i = 1,…, I. The normalization maps probe level data from all arrays, i = 1,…, I, so that an I -dimensional quantile–quantile plot follows the I -dimensional identity line.

2.4 Inference, classification and validation

The objectives of a microarray studies are diverse and can be vaguely separated into two subgroups (Figure 2.4.1): group comparison (inference) and classification. For each objective, there exists a wide range of statistical techniques to analyze the data and to answer the research questions.

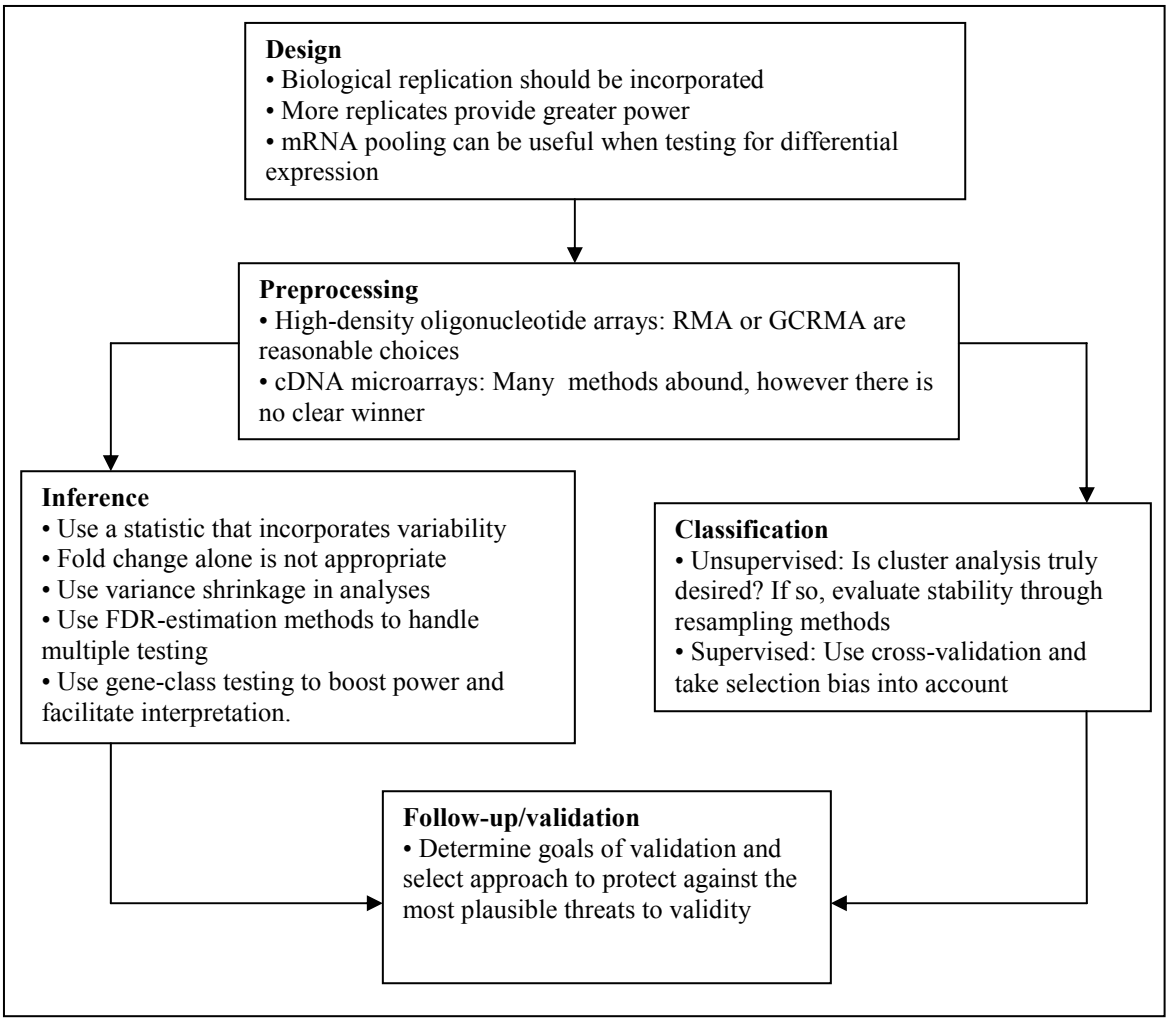


Figure 2.4.1 *Guidelines for the statistical analysis of microarrays (Allison et al. 2005).*

Inference (class comparison) involves making conclusions about the truth of hypotheses that involve unobserved parameters about whole populations, which are based on statistics obtained from samples. The process of **classification** aims at either placing objects (for example, genes) into pre-existing categories (supervised classification), or developing a set of categories into which objects can subsequently be placed (unsupervised classification). To perform a **validation** of a method it is best to have a sufficiently large collection of samples to allow an independent test set and training set. In practice, usually only a limited number of samples are available, and these are needed

for building and training the algorithm. An alternative to using an independent test set is to leave out k when using the cross-validation method.

Class Comparison

The goal of a class comparison is usually to determine a relatively small list of genes, which are under- or over-expressed in one of the classes compared to others. Fold change was the first method used to evaluate whether genes are differentially expressed, and it gives a reasonable measure of effect size. It is now considered to be an inadequate test statistic because it does not incorporate variance and offers no associated level of confidence. A more appealing option is the use of basic parametric or non-parametric test statistics for group comparison, e.g., the t-test or the Wilcoxon rank sum test. In 2001, Tusher et al. proposed a modified version of the t-test, which later was modified by adding a constant, the ”fudge factor”, to the denominator of the t-statistic to adjust for the tendency of selecting low-variance genes. The latter method is known as SAM (Significance Analysis of Microarrays). There are many other methods to perform group comparison based on microarray data, they are discussed in Amaratunga, Cabrera (2004) and Simon et al. (2004).

Supervised classification

Supervised classification (also called ‘class assignment’, ‘class prediction’ or ‘discrimination’) involves finding which features of the known samples (with known class labels) are most useful in separating the known samples. Many supervised classification algorithms are available, but they all are susceptible to overfitting. Methods include logistic regression, linear and quadratic discriminant analysis, nearest neighbor classifiers, decision trees, shrunken centroids, neural networks, random forests, support

vector machines, and many more. The overview of main groups of classification algorithms is included in Section 2.4.1.

Unsupervised classification

Unsupervised methods try to find internal structure or relationships in a data set, instead of trying to determine how to predict a correct answer best. Within unsupervised learning, there are three main classes of techniques:

(1) Feature determination – finding genes with interesting properties without specifically looking for a particular pattern, such as principal-components analysis (PCA)

(2) Cluster determination – assignment of groups to genes or samples with similar patterns of gene-expression, such as nearest neighbor clustering, self-organizing maps, k-means clustering and hierarchical clustering

(3) Network determination – determining graphs representing gene-gene or gene-phenotype interactions using Boolean, Bayesian or relevance networks.

Some of the clustering methods are described in Section 2.4.2.

2.4.1 Overview of Supervised Learning Methods

Many prediction techniques are assessed in the microarray analysis literature. The most prominent classes of techniques are regression, classification trees, nearest neighbor prediction, linear discriminant analysis (LDA), support vector machines (SVM) and neural networks. These methods are general categories of prediction methodologies and include many types of models and options to choose from.

Logistic regression is a technique that uses linear combinations of genes to predict the probability that the samples have a certain characteristic. It can only be built with a small number of genes and therefore will require careful gene filtering. Logistic regression can also be penalized (**lasso, elastic net**).

A **classification and regression tree (CART)** is a decision tree-based method that searches the predictors for cut-point values that best separate the samples into groups. The subsets remaining at the final stage are assigned to a certain class, the one which is most frequently represented in the subset. In a way, the method has its own gene-selection procedure. It determines which genes to use at each splitting node in order to get the best classification. **Random forests** (Breiman 2001) are formed by a combination of tree predictors. Subsets of samples and genes are obtained by independently drawing samples with replacement from the training dataset and by selecting a number of genes at random. A classification tree is estimated for each of the newly formed datasets. A new sample is allocated to the class with the most votes over all the trees in the forest.

Another method is **K-nearest neighbors** (kNN) (Ripley 1996). KNN involves calculating the similarity measure (such as Euclidean distance) between the unknown sample and all of the known samples. The unknown sample is classified by the majority vote in the K nearest neighbors. KNN is technically unsupervised since the correlation coefficients are objectively determined. However, the method could be considered semi-supervised since the choice of the value for K can be determined by the predictability of the known samples, which is dependent on the total sample size.

Linear **discriminant analysis (LDA),** a classical discriminant method, identifies linear combinations of genes that have large ratios of between groups to within group variability. The method is based on the assumption of normally distributed data and equal covariance matrices for the considered classes. Diagonal linear discriminant analysis (DLDA) is a variant of LDA, whereby the covariance matrix is additionally assumed to have a diagonal structure. Diagonal quadratic discriminant analysis (DQDA), also a variant of LDA, assumes diagonal, but not equal covariance matrices for all classes. In a sense, both DLDA and DQDA ignore the correlation structure between expression levels of genes in the microarray data.

Support vector machines, first introduced by Vapnik (2000) in the machine learning theory, are also used to solve classification problems. This method finds the optimal hyperplane in the space of the gene expression values for differentiating the samples based on the characteristic of interest. For this the samples are non-linearly mapped to a very high-dimensional feature space. In this space a hyperplane is designed that provides an optimal separation. SVMs can have linear, polynomial, spline, and other kernels to solve the optimization problem.

Neural Networks are machine learning classification tools that represent the relationship between the expression values and the true classes by a network of connections and nodes. The networks consists of the inputs (gene expression data) connected to hidden layers of nodes which are then connected to the output layer of units, one for each possible outcomes. The connections among inputs, nodes and outputs have weights which are iteratively adjusted to improve the overall prediction. The process is repeated until a vector of weights is generated that best fits the data.

There are also methods to improve the accuracy of classification. One such method is called **bagging** (Breiman 1996). Bootstrap replicates are taken from the training dataset. A tree is constructed for each replicate and the final classification is determined by majority vote. That is, the sample is assumed to belong to the class, to which it is most frequently assigned by the different trees. Bagging is said to be a variance reduction technique, designed to stabilize trees. **Boosting**, proposed by Schapire and Freund (1999), is another form of aggregating trees. A series of classification trees is produced for the training dataset, each time with different weights assigned to the samples. The idea is to give samples misclassified in the previous step more weight in the current one. The final outcome is a weighted majority vote of all created trees. It is believed that bagging is much better than boosting in situations with substantial random noise. Boosting is however expected to reduce both the variance and bias of unstable trees.

2.4.2 Clustering Methods

Cluster analysis is a significant part of unsupervised classification aiming mostly at class discovery (can be used to either classify genes, or samples, or both simultaneously). Different approaches to clustering data can be broadly separated into two classes as it is described with the help of the hierarchy shown in Figure 2.4.2 (Blalock 2003). At the top level, there is a distinction between hierarchical and partitional approaches (hierarchical methods produce a nested series of partitions, while partitional methods produce only one).

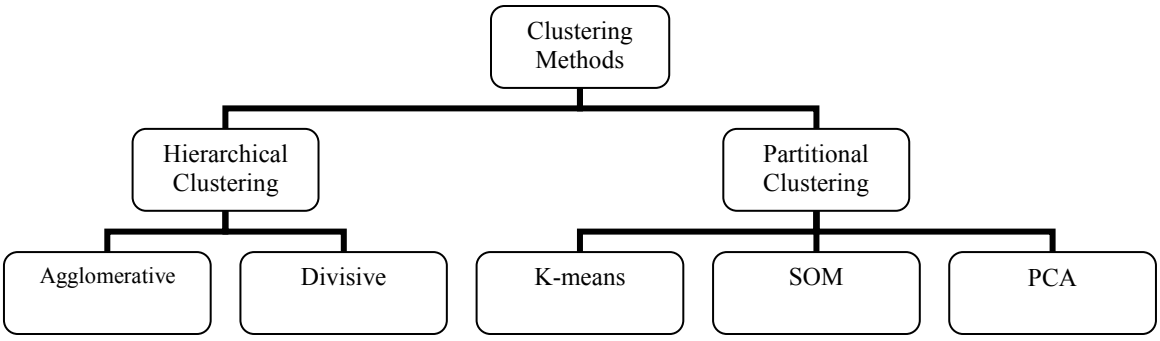


Figure 2.4.2 *Main Unsupervised Methods for Microarray Data Analysis*

Eisen et al. (1998) is one of the first papers to consider clustering analysis for discovering biologically meaningful patterns in microarray data. They used hierarchical clustering (Sokal, Mitchener, 1958). A wide range of algorithms have been proposed since to analyze gene-expression data, such as K-mean clustering, self-organizing maps, model-based clustering, and clustering using ABC dissimilarities.

The K-means (or Lloyd's) algorithm (Lloyd 1957, MacQueen 1967) is used to reposition the cluster centers through the following steps a) observations are assigned to the closest cluster center to form a partition of the data, b) the observations in each cluster are averaged to produce new values for the center vector of that cluster. Steps (a) and (b) are iterated, and the process converges to a local minimum of the total within cluster variance. The self-organizing map (SOM) (Kohonen (1989)) is similar to K-means clustering, with the additional constraint that the cluster centers are restricted to lay in a one or two-dimensional manifold. Model-based clustering assumes that the data have been generated by some, typically probabilistic (Bayesian), model, and tries to find the clustering corresponding to the most probable model. Clustering using ABC dissimilarities (Amaratunga et al., 2008) is based on the idea of aggregating results

obtained from an ensemble of randomly resampled data (where both samples and genes are resampled) and produces a measure of dissimilarity between each pair of samples that can be used in conjunction with another clustering procedure.

Other unsupervised methods, such as principal component analysis (PCA), aim at reducing the dimensionality of the data, making it possible to visualize the latter. Examples of the application of PCA to microarray data can be found in Raychaudhuri and Altman (2000); Yeung and Ruzzo (2001b). Additional techniques for dimension reduction and visualization applied to gene-expression values include correspondence analysis (Fellenberg et al., 2001), biplots (Chapman et al., 2002), and spectral map analysis (Wouters et al., 2003).

2.5 Challenges in Microarray Expression Data

There are many issues arising in the analysis of high-dimensional data such as microarray data. They are mostly related to the data dimensionality and cost. The number of samples is small when compared to the number of variables under consideration, which makes statistical methods prone to overfitting. The size of the datasets is considerable rendering most analytical methods computationally intensive.

2.5.1 Overfitting

A constant concern with supervised learning is the possible overfitting of the data. Overfitting happens when the classifier very precisely distinguishes the training data sets but performs poorly in future observations with new data. The phenomenon of overfitting is shown in the Figure 2.5.1.

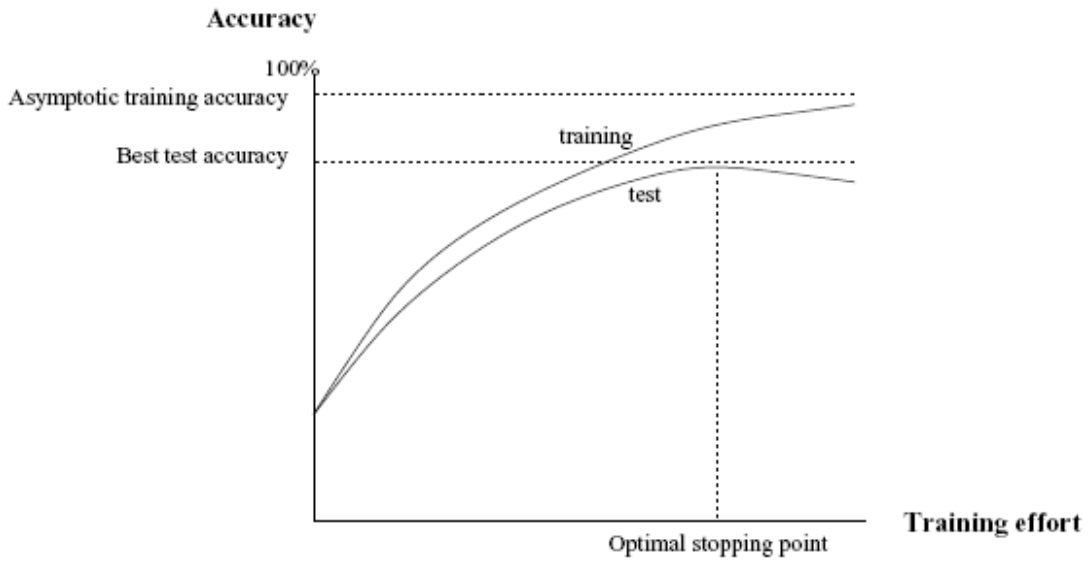


Figure 2.5.1 *Overfitting Phenomenon*

Overfitting can occur when at least one of the following occurs:

- A small training set is more likely to produce overfitting than a large training set. Patterns that show up in a small training set may be spurious, and due to noise. If they carry over to a larger training set, they are likely to reflect actual patterns in the domain.
- Noise in the data is likely to lead to overfitting. It increases the likelihood of spurious patterns that do not reflect actual patterns in the domain.
- Overfitting is more likely with a rich hypothesis space. Overfitting requires the ability to fit the noise in the data, which may not be possible with a restricted hypothesis space.
- A domain with many features is more likely to lead to overfitting. This is particularly an issue with irrelevant features, that are in the domain but have no

impact on the classification. If there are many irrelevant features, it is quite likely that some of them will appear relevant in a particular data set.

2.5.2 Adjustments for multiple comparisons

Due to the size of microarray expression data, multiple hypotheses testing problem is one of main issues in the analysis. Both rejection of true null hypotheses (type I error) and failures to reject false null hypotheses (type II error) can lead to wrong conclusions. For the case of testing multiple hypotheses, the type I error rate can have a variety of generalizations. The two most commonly used error rates in multiple testing are the family wise error rate, abbreviated as FWER, and the false discovery rate, abbreviated as FDR. Multiple testing correction adjusts the individual p-value for each test to keep the overall error rate not exceeding some cutoff value.

The FWER is the probability of rejecting at least one true null hypothesis. The weak, exact, and strong control of FWER correspond to the situation where all the null hypotheses are true, an exact set of null hypotheses is true, and any subset of null hypotheses is true, respectively. The most commonly used method for controlling FWER is the Bonferroni method. The test of each hypothesis H_j is controlled so that the probability of a type 1 error is less that a cutoff value divided by the number of tests performed. Closely related to Bonferroni is Sidak (1967) method, a less conservative one is introduced by Holm (1979). To adjust for correlated structure of gene expression data Westfall and Young in 1993 introduced method that accounts for the dependence structure between the genes (maxT). It requires the estimation of the joint null distribution of the unadjusted unknown p-values. It was later suggested by Dudoit et al.

(2002b) to estimate the null joint distribution of test statistics for all genes by permuting the class labels of the samples.

Control of the FWER can be too stringent in the microarray setting as it may lead to many missed findings. Hence for the purpose of identifying as many genes with significant differences as possible while controlling the portion of false findings, the concept of controlling the FDR becomes popular. The FDR is the expected proportion of false positives among all the rejected null hypotheses. It was first introduced by Benjamini and Hochberg in 1995 (abbreviated as BH procedure) and defined as the expected proportion of false rejection among the rejected hypotheses, FDR=E(Q), where Q =V/R when R > 0, and Q = 0 otherwise (see Table 2.5.1).

The BH-FDR Procedure

Let $P_{(1)} \leq P_{(2)} \leq \ldots \leq P_{(m)}$ be the ordered p-values of and let $H_{(1)}, H_{(2)}, \ldots, H_{(m)}$ be the corresponding null hypotheses. The Benjamini-Hochberg (BH) procedure consists of rejection $H_{(1)}, H_{(2)}, \ldots, H_{(\ell)}$, where ℓ is the largest value of i for which $P_{(i)} \leq \frac{i}{m}\alpha$ .Then the BH-FDR adjusted p-values are given by

$$\tilde{P}_i = \min_{k=i,\ldots,m}\left[\min\left(\frac{m}{i}P_{(i)}, 1\right)\right] \qquad (2.1)$$

The null hypothesis $H_{(i)}$ is rejected if $\tilde{P}_i \leq \alpha$. The BH-FDR procedure controls the FDR for positively dependent test statistics.

Later Yekutieli and Benjamini (2001) proposed a modification of the BY-FDR procedure for controlling the FDR for any joint test statistic distribution. The BY-FDR

procedure consists of a modification factor $\sum_{j=1}^{m}\frac{1}{j}$. Approaches based on the control of the FDR have gained their popularity in the microarray setting, because they lead to a higher power as compared to the methods that control the FWER.

Additionally there are notions of false non-discovery rate (FNR) (see Genovese et al. 2002), positive FDR (see Storey, 2003) and positive FNR. To control the number of false positives, the gFWER was proposed by Hommel and Hoffmann (1988) and defined as the probability of rejecting at least k true null hypotheses, i.e., gFWER= P(V > k). Also, a generalization of the FDR, similar to the way the gFWER generalizes the FWER, was proposed by Sarkar and Guo (2005b). The gFDR is defined the expected proportion of k or more false rejections among all rejections, i.e., gFDR= E (V/R) when V > k, and gFDR= 0 otherwise.

	# not rejected	# rejected	Total
# true null hypotheses	U	V	m_0
# not true null hypotheses	V	S	m_1
Total	W	R	m

Table 2.5.1: *Decisions in multiple testing (Benjamini and Hochberg 1995).*

PART II.

CLASSIFICATION FOR MICROARRAYS

Chapter 3

Partial Least Squares

3.1 Classification for Microarray Data

Several approaches to microarray data classifier construction have been described in the literature (for a brief overview see Section 2.4.1); among the most often used are tree methods, classical discrimination analysis techniques, and machine learning methods. It is evident from literature that a universally best method for classifier creation does not exist. However, there is some effort put into comparison of various classification algorithms (see Table 3.1.1 for summary). For example, Dudoit et al. (2002) compared the performance of nine classification methods for classifying tumors based on gene-expression profiles. They found that simple classifiers such as k nearest neighbor (kNN) and diagonal linear discriminant analysis (DLDA) performed remarkably well as compared to more sophisticated methods like aggregated classification trees. Here one can argue that authors did not pre-select many genes which lead discrimination techniques outperforming other methods. Lee et al. (2005) conducted a more extensive comparison study of the performance considering over twenty methods applied to seven datasets using three gene-selection techniques. Contrary to the findings of Dudoit et al. (2002), Lee et al. (2005) concluded that the more sophisticated classifiers gave better performances than classical methods such as kNN, DLDA, or diagonal quadratic discriminant analysis (DQDA). Additionally, they found that the choice of gene-selection method had much effect on the performance of the classification methods.

Dudoit et al. (2002) 3 data sets MCCV 2:1	• Included: LDA, DLDA, DQDA, Fisher, kNN, trees, tree-based ensembles • Variable selection: F-statistic Conclusion: DLDA and kNN perform best
Romualdi et al. (2003) 2 data sets CV	•Included: DLDA, trees, neural networks SVM, kNN, PAM combined with: •Variable selection/dimension reduction: PLS, PCA, soft thresholding, GA/kNN Conclusion: PLS transformation is recommendable, No classifier uniformly better than the other
Man et al. (2004) 6 data sets LOOCV, bootstrap	• Included: kNN, PCA+LDA, PLS-DA, neural networks, random forests, SVM • Variable selection: F-statistic Conclusion: PLS-DA and SVM perform best
Lee et al. (2005) 7 data sets LOOCV, MCCV 2:1	• Included: 21 methods (e.g. tree ensembles, SVM, LDA, DLDA, QDA, Fisher, PAM) • Variable selection: F-statistic, rank-based score, soft thresholding Conclusion: No classifier uniformly better than the other, rank-based variable selection performs best
Statnikov et al. (2005) 11 data sets LOOCV, 10-fold CV	• Included: SVM, kNN, probabilistic neural networks, back-propagation neural networks • Variable selection: BSS/WSS, Golub et al. (1999), Kruskal-Wallis test Conclusion: SVM performs best
Huang et al. (2005) 2 data sets LOOCV	• Included: PLS, penalized PLS, LASSO, PAM, random forests • Variable selection: F-statistic • Random forests perform slightly better Conclusion: No classifier uniformly better than the other

Table 3.1.1 *Summary of comparison studies of classification methods (Boulesteix 2005)*.

3.2 Introduction to Partial Least Squares

PLS regression is a quite recent technique that generalizes and combines features from principal component analysis and multiple regression. It is particularly useful when one needs to predict a set of dependent variables from a large set of independent predictors. The PLS method was first developed by Herman Wold in the 1960's and 1970's to address problems in econometric path-modeling, and was subsequently adopted

by his son Svante Wold (and many others) in the 1980's to problems in chemometrics and spectrometric modeling. The success of PLS in chemometrics resulted in a lot of applications in other scientific areas including bioinformatics, food research, medicine, pharmacology, social sciences and physiology.

From a data analysis point of view gene expression data are very similar to spectroscopic data. For example, there is often a large amount of systematic variation present. Additionally, a large number of genes across a grid are analogous to the large number of wavelengths in a spectrum. Hence, PLS is very well suited for the analysis of high-dimensional problems arising from the genomic experiments.

On the contrary, other classification methods do not handle case of p>>N very well. To overcome this, methods usually incorporate the extraction of a small subset of interesting variables as a first step using one of the univariate gene selection methods (such as using t-statistic (Hedenfalk et al., 2001), Wilcoxon's rank sum statistic (Dettling and Buhlmann, 2003) or Ben Dor's combinatoric 'TNoM' score (Ben-Dor et al., 2000)). However, aforementioned univariate gene selection methods are all based on the association of individual genes with the response variables. Interactions and correlations between genes are omitted, which excludes biological background from the selection process.

PLS technique presents a wise alternative to above in order to overcome dimensionality and structure issues. Unlike gene selection, this method use all the genes included in the data set, the components then give information or hints about the data's intrinsic structure.

3.3 PLS Method and Algorithms

The multivariate projection methods include partial least squares (PLS) and principal component analysis (PCA) for dimension reduction; correspondence analysis (Fellenberg et al., 2001), biplots (Chapman et al., 2002), and spectral map analysis (Wouters et al., 2003) for dimension reduction and visualization. Multivariate projection methods help to reduce the complexity of high-dimensional data (n genes versus p samples) and provide means to identify gene patterns or subjects in the data. Projected data are typically displayed in a biplot (genes and samples) in a new space. An attractive property of PCA, PLS and their extensions, is that they apply to almost any type of data matrix, e.g., matrices with many variables (columns), many observations (rows), or both.

PLS methods are generally characterized by high computational and statistical efficiency. There are no distributional assumptions associated with PLS, which makes this method flexible and expands the range of problems that may be addressed. However, the literature of PLS is very diverse due to the existence of a large number of algorithmic variants of PLS, which makes it very difficult to understand the principle of PLS.

PLS Method

The underlying idea of PLS regression is to find uncorrelated linear transformations of the original predictor variables which have high covariance with the response variables. These linear transformations can then be used as predictors in classical linear regression models to predict the response variables.

Assume X is a $n \times p$ matrix and Y is a $n \times q$ matrix. The PLS technique works by successively extracting factors from both X and Y such that covariance between the extracted factors is maximized. PLS method can work with multivariate response

variables (i.e., when Y is an $n \times q$ vector with $q>1$). However, we will assume for now that we have a single response variable i.e., Y is $n \times 1$ and X is still $n \times p$, as before.

Formally PLS technique tries to find a linear decomposition of X and Y such that:

$$\begin{gathered} X=TP^{T}+E \text{ and } Y=UQ^{T}+F, \\ \hline \text{where } T_{n\times r}=X-scores \text{ and } U_{n\times r}=Y-scores, \\ P_{p\times r}=X-loadings \text{ and } Q_{l\times r}=Y-loadings, \\ E_{n\times p}=X-residuals \text{ and } F_{n\times l}=Y-residuals \end{gathered} \tag{3.1}$$

Decomposition is finalized to maximize covariance between T and U.

There are multiple algorithms available to solve the PLS problem. They all follow an iterative process to extract the X and Y-*scores*. The three common algorithms for PLS implementation are the kernel algorithm, the classic orthogonal scores algorithm (or NIPALS – nonlinear iterative partial least squares algorithm) (Martens and Naes 1989) and the SIMPLS algorithm (de Jong 1993). The kernel and NIPALS algorithms produce the same results (the kernel algorithm being the fastest of the three). NIPALS (for algorithm see below) is the standard algorithm for computing partial least squares regression components (factors). The PLS Kernel algorithm proposed by Ränner, Geladi, Lindgren, and Wold is based on a simplified version of the EM algorithm for the calculation of covariances matrices when missing data are present. SIMPLS algorithm calculates the PLS factors directly as linear combinations of the original variables. The PLS factors are determined such as to maximize a covariance criterion, while obeying certain orthogonality and normalization restrictions. SIMPLS produces the same fit for single-response models, but slightly different results for multi-response models. It is also

usually faster than the NIPALS algorithm. Other algorithms also mentioned in the literature include various modifications of NIPALS, SIMPLS and kernel selection, weighted algorithms,

NIPALS Algorithm

There are many variants of the NIPALS algorithm which normalize or do not normalize certain vectors. The following algorithm, which assumes that the X and Y variables have been transformed to have means of zero, is considered to be one of most efficient NIPALS algorithms.

Algorithm:

Y - centered and scaled, each X_i has $mean(X_i)=0$, $Var(X_i)=1$ for all i. Initialize $Y_1=Y$, $X^1=X$.

1. Calculate the individual regression coefficients of Y_k on each X_i^k
$$w_j^k = < X_i^k , Y_k >$$
2. Form the PLS component as the weighted sum of X_i
$$t_k = \sum w_j^k X_i^k$$
3. Calculate the regression coefficient of Y_k on the component t_k
$$\beta_k = < t_k, Y > / < t_k, t_k >$$
4. Update the X_i^k by orthogonalizing them with respect to t_k.
5. Update by the residuals of the previous linear fit
$$Y_{k+1} = Y_k - \beta_k t_k$$
6. Iterate these 5 steps $k=1,...,g$ (g – number of components desired).

The algorithm produces a sequence of orthogonal vectors $\{t_k\}$ and a sequence of estimators $\{\beta_k\}$.

Table 3.2.1 *The NIPALS Algorithm*

We have to note here that the step where the response Y is updated with the residuals of the previous linear fit can be omitted. This is due to the fact that the set of new predictors $\{X_i^k\}$ is orthogonal to previous components. Hence, the coefficients for any PLS component are the same whether they are calculated regressing on either Y or Y_k due to the following reasoning:

$$\begin{aligned} w_j^k &= < X_j^k, Y_k > = < X_j^k, Y_{k-1} - \beta_{k-1} t_{k-1} > = \\ &= < X_j^k, Y_{k-1} > - \beta_{k-1} < X_j^k, t_{k-1} > = < X_j^k, Y_{k-1} > \overset{ind}{=} \ldots = < X_j^k, Y > \end{aligned} \tag{3.2}$$

Throughout this dissertation we will base our calculations on the NIPALS algorithm. Appendix A contains the overview of R packages that implement various PLS approaches.

Number of PLS Components

There is no widely accepted procedure to determine the right number of PLS components. It is commonly predefined by the user or selected via cross-validation. However, cross-validation is often avoided because of computational limitations and poor performance or strong bias on small sample data sets. Nevertheless, Boulesteix (2005) proposed to use a simple method based on cross-validation measure and subsequently concluded that that for datasets with low error rates, the classes are optimally separated by only one component, whereas subsequent components are useful for data sets with high error rates. Most of the researchers do not use more than two components; only Nguyen and Rocke fixed the number of components at three in their experiments and suggested the classification accuracy is insensitive to this parameter when it is beyond five. Zeng et al. considered linking the threshold with the PLS regression quality via mean classification success (SUC) rate and concluded that the best number of latent

components for sensitive classifiers hardly exceeds three and heavily depends on the dataset. Throughout this dissertation we will consider analysis using only the first component.

3.4 Simple PLS extension to Binary Response Data

For the case of a discrimination between two groups one can consider various modifications to the PLS algorithm. In the simplest case, binary response Y can be treated as a continuous response variable, since PLS regression does not require any distributional assumptions. Other approaches include replacing the binary vector Y with a pseudo-response variable whose expected value has a linear relationship with the covariates (Fort and Lambert-Lacroix, 2005), and also the use of the IRPLS and its improvements for convergence by Nguyen and Rocke, Marx, Ding and Gentleman (with Newton-Raphson algorithm for convergence improvements).

We propose the simple and intuitive modification of the NIPALS algorithm for the binary response. For the modification, we have the component weights to be based on the logistic regression coefficients, and we exclude the step where one updates the response Y with the residuals of the previous fit (as noted in reasoning 3.2 for the continuous case).

Suppose that the response Y is binary and the set of continuous predictors $X=\{x_j\}$ with $j=1,\ldots,p$ is already centered and scaled, i.e. $mean(X_j)=0, Var(X_j)=1$ for all j.

Modified Algorithm:

The PLS modification for binary response could be written as a set of the following steps:

1. Calculate the individual logistic regression coefficients of Y on each X_i^k
$$w_i^k = < X_i^k, Y >$$
2. Form the PLS component as the weighted sum of X_i
$$t_k = \sum w_i^k X_i^k$$
3. Update the X_i^k by orthogonalizing them with respect to t_k.
4. Iterate these 3 steps $k=1,...,g$ (g – number of components desired).

Find the PLS coefficients $\beta_k = \frac{< t_k, Y >}{< t_k, t_k >}$ from logistic regression of Y on components t_k.

Table 3.2.2 *Algorithm Modification for Binary Response Data*

3.5 PLS in Microarray Setting

Within a last few years, many researchers have considered PLS methodology for regression and classification problems, as well as for feature extraction and various modeling of the survival data (Nguyen and Rocke 2002d).

For example, Musumarra et al. based gene selection on the weights vector. They introduced the 'variable influence' measure defined as a function of PLS squared weights and the proportion of SS explained by the corresponding latent component. Boulesteix showed F statistic, which is often used in the gene selection procedure, is a monotonic transformation of the squared PLS weight coefficients.

For classification tasks, two independent comparative studies by Man et al. (2004) and Huang et al. (2005) reported that classification based on PLS regression leads to high

prediction accuracy. Additionally, PLS classification analysis for binary response has been investigated by Huang and Pan (2003) for leukemia and colon cancer data. They suggest determining the best number of latent components by leave-one-out cross-validation. A similar approach is used in a more applied study by Perez-Enciso and Tenenhaus (2003): various binary outcomes such as (i) before versus after chemotherapy treatment in a case-control study, (ii) estrogen receptor positive versus negative tumors and (iii) tumor types are predicted via PLS discriminant analysis.

PLS regression is also employed for multi-class classification in Musumara et al. (2004) for the molecular diagnostic of cancer. Using the software SIMCA, they performed classification on the human cancer NCI data set consisting of the expression levels of 9605 genes in 60 tumor cell lines of eight different types (leukemia, non-small cell lung, colon, melanoma, ovarian, breast, central nervous system and renal).

Other classification studies based on PLS regression include classification of human ovarian tumors (Alaiya et al. 2000), classification of acute leukemia subtypes (Cho et al. 2002), multi-class tumor classification by discriminant partial least squares (Tan et al. 2004), prediction of primary breast cancers (Modlich et al. 2005). A similar approach based on PLS regression to perform classification in the context of meta-analysis is suggested by Huang et al. (2005) for sample classification using weighted partial least squares. Datta (2001) suggests that the partial least squares (PLS) regression may in fact be a powerful tool for exploring relationships between genes' expression profiles, which may translate into biologically meaningful interactions and associations.

There exists another route to classification using partial least squares, first proposed by Nguyen and Rocke (2002a,b) and further studied by Boulesteix (2004) and compared

with other dimension reduction techniques by Dai et al. (2006). This approach first employs PLS as a dimension reduction method and subsequently uses the PLS latent components as predictors in a classical discrimination method (e.g. logistic regression, linear or quadratic discriminant analysis). To apply this method, one has to choose (i) the number of latent components to be extracted in the dimension reduction step and (ii) the classification method to be used for the classification step. In Nguyen and Rocke, three classification methods are studied: logistic regression, linear discriminant analysis and quadratic discriminant analysis. Boulesteix only investigates discriminant analysis. Generally, linear discriminant analysis (LDA) turns out to yield the best classification performance, whereas quadratic discriminant analysis gives worse results. In the comparison study performed by Boulesteix, PLS+LDA turns out to range among the best classification procedures for all the eight studied cancer data sets. According to this study, the most successful other methods are the nearest centroids approach by Tibshirani et al. (2002) and SVM. Additionally, the idea that performance may be improved by modifying PLS dimension reduction in order to adapt it to the specific case of categorical responses was explored by Fort and Lambert-Lacroix (2005). They proposed a two-stage method combining PLS and ridge penalized logistic regression (implemented in R package plsgenomics).

Marx (1996) proposes an extension of the concept of PLS regression into the framework of generalized linear models. This approach, which is denoted as iteratively reweighted partial least squares (IRPLS or IRWPLS), embeds the univariate PLS regression algorithm into the iterative steps of the usual Iteratively Reweighted Least Squares algorithm for generalized linear models, resulting in two nested loops. The loops

are iterated a fixed number of times or until a convergence criterion is reached. The IRPLS method as well as a few adaptations overcoming the convergence problem have been applied both to survival analysis and classification. Binary classification is one of the most common applications of generalized linear models and of Marx's IRPLS algorithm. It has inspired at least two recent papers on the generalization of PLS regression to categorical response variables. The first approach is proposed by Ding and Gentleman (2005) and can be seen as an adaptation of Marx's IRPLS method which solves the problem of separation. The problem of (quasi)separation is avoided by applying bias correction to the likelihood. This method is implemented in the R package gpls. Another classical application of generalized linear models and IRPLS is survival analysis (see Nguyen and Rocke 2002, Park et al. 2002, and Li, Gui 2004).

In the next chapter we will consider the prediction method arising from the partial least squares (PLS) methodology with adjustments made for multiple testing. We will mainly focus on binary classification and study the performance of PLS themed approach in this setting.

Chapter 4

PLS-FDR

4.1 Introduction

As discussed in Chapter 3, PLS methodology proved its usefulness for variety of tasks in microarray experiments. The main step in any PLS algorithm is the calculation of a partial least squares components. Each of the components can be written as a weighted combination of original predictors X_i. As noted by Garthwaite (1994), the PLS components can be obtained as linear combinations of simple linear regression predictors. It was then shown by Nguyen and Rocke (2003) and noted by others (Boulesteix, Fort and Lambert-Lacroix) that the PLS components can be expressed as weighted averages of the original predictor/explanatory variables, with weights depending on the partial correlation coefficients and sample predictor variance. We will initially consider the first partial least squares component and the data that has been previously centered and scaled. In this setting one may think of the PLS component coefficients to be based on the correlation between response vector and predictor variables.

4.2 PLS Weights Approximation

Assume that the data is already centered and scaled. We will base our approximation on the estimation of the tails of the distribution via the p-values. We will initially consider the expression of normal quantile through p-value. Normal distribution is very useful when approximating variety of other distributions including student t and chi-square (see Patel K., Read C.B. 1996), for example:

t-distribution	$t_{n,p} = \sqrt{n\left(\exp\left[z_p^2 c^2 / n\right] - 1\right)} \quad c = \frac{8n+3}{8n+1}$	(Prescott, 1974)	(4.1)
	$t_{n,p} = z_p + \frac{z_p + z_p^3}{4n} + \frac{3z_p + 16z_p^3 + 5z_p^5}{96n^2} + \ldots$	(Cornish-Fisher, 1960)	(4.2)
Chi-squared	$y_p \simeq \frac{\left(z_p + \sqrt{2\nu - 1}\right)^2}{2}$	(Fisher, 1974)	(4.3)
	$y_p \simeq \nu\left(z_p\sqrt{2/(9\nu)} + 1 - 2/(9\nu)\right)^3$	(Wilson-Hilferty,1931)	(4.4)

There are also several approximations of the normal quantile through *p*. For example the following simple expression (see also Patel K., Read C.B. 1996 for others, pp.66-68):

$$z_p = \frac{\ln(\frac{1}{p} - 1)}{b\ln(\frac{1}{p} - 1) + a}; \; a = 1.48 - \frac{0.39}{n}, \quad b = 0.108 \qquad (4.5)$$

J.-T. Lin et al. (1992) considered the use of (4.5) to approximate tails of the t distribution. First, expressing p through z_p:

$$p^* = \left[1 + e^{\frac{a}{(1/z_p - b)}}\right]^{-1} \qquad (4.6)$$

Then, solving normalizing transformation $z = \left[n \cdot \ln(1 + t^2/n)\right]^{1/2}$ for t and substituting (4.6), authors derived the expression for the *t* value that they found to be quite accurate:

$$t^*_{n,p} = \left[n\left(e^{\frac{1}{n\cdot(a/y+b)^2}} - 1 \right) \right]^{1/2} \qquad (4.7)$$

where $y = \ln(1/p - 1) = \ln\left(\frac{1-p}{p}\right) = -\ln\left(\frac{p}{1-p}\right)$.

To derive a simple approximation of $t_{n,p}$ through the p-value, we will use the following series expansion around zero for the *logit* function:

$$-\ln\left(\frac{x}{1-x}\right) \approx -\ln x - x - \frac{x^2}{2} \qquad (4.8)$$

We can first write $1/t^*_{n,p}$ in series of y around infinity:

$$\frac{1}{\sqrt{n\left(e^{\frac{1}{n\cdot(a/y+b)^2}} - 1 \right)}} \approx \frac{1}{\sqrt{ne^{\frac{1}{nb^2}}}} + \frac{ae^{\frac{1}{nb^2}}}{b^3\left(n\left(e^{\frac{1}{nb^2}} - 1 \right) \right)^{3/2}} y - \sqrt{\frac{b^2}{4a^2 n}} y^2 + O\left(\frac{1}{y}\right)^2 \qquad (4.9)$$

Since whenever p is close to zero we have $y \to \infty$ as below:

$$p = 0 \Rightarrow y = -\ln\left(\frac{p}{1-p}\right) = -\ln\left(\frac{0}{1-0}\right) = -\ln(0) = \infty \qquad (4.10)$$

Then the series for $p_0=0$ are derived from (4.9) substituting y and we get the following expression:

$$\frac{1}{\sqrt{n\left(e^{\frac{1}{n\cdot(a/y+b)^2}} - 1 \right)}} \approx \frac{1}{\sqrt{ne^{\frac{1}{nb^2}}}} + \frac{ae^{\frac{1}{nb^2}}}{b^3\left(n\left(e^{\frac{1}{nb^2}} - 1 \right) \right)^{3/2}} \ln(1/p - 1) \qquad (4.11)$$

Which can then be approximated by the Taylor series for p around zero as below:

$$\frac{1}{\sqrt{ne^{\frac{1}{nb^2}}}}+\frac{ae^{\frac{1}{nb^2}}}{b^3\left(n\left(e^{\frac{1}{nb^2}}-1\right)\right)^{3/2}}\ln(1/p-1)\approx\frac{1}{\sqrt{n\left(e^{1/(nb^2)}-1\right)}}-$$
$$-\frac{ae^{1/(nb^2)}}{b^3\left(n\left(e^{1/(nb^2)}-1\right)\right)^{3/2}}\frac{1}{\ln(p)}+\frac{ae^{1/(nb^2)}}{b^3\sqrt{n\left(e^{1/(nb^2)}-1\right)}}\frac{p}{\left(\ln(p)\right)^2}+O(p)^2 \qquad (4.12)$$

Coefficients in (4.12) only depend on *n*. Therefore, we can rewrite the above expression in terms of *p* as follows:

$$1/t^*_{n,p}\approx A(n)-B(n)\frac{1}{\ln(p)}+C(n)\frac{p}{\left(\ln(p)\right)^2} \qquad (4.13)$$

Hence, we can base the approximations on the right-hand terms of (4.13) employing only the *-ln(p)* term of the approximation for further analysis.:

$$b_i\approx-\ln p_i^* \qquad (4.14)$$

Let's consider now the regression coefficients arising from linear or logistic regression. For a simple linear regression, p-values associated with each estimated regression coefficient are based on the test statistic $T_i = b_i/SE$, where b_i is a coefficient of regression Y on each predictor variable X_i. The test statistic T_i follows t distribution; SE only depends on number of observations n if the data has already been centered and scaled. Hence, we can think of T_i as directly depending on the sample regression coefficient b_i and n. Figure 4.2.1 shows the plots of the regression coefficients versus the approximation (4.14) using the original (blue) and BH-FDR corrected (green) p-values.

In the case of a logistic regression, generally a Wald test is used to test the statistical significance of each coefficient b_i in the model. A Wald test calculates a z statistic, which is $z_i = b_i/SE$. Additionally, to assess the importance of a certain predictor variable, one can use the t-test for two-group comparison. Below (Figure 4.2.2) one can see plots of logistic regression coefficients versus the approximation (4.14) with p-values coming from t-test for group means for two simulation scenarios: (i) binary response is generated from a logistic distribution, (ii) binary response is modeled as two groups where mean difference is introduced for a predictor.

(4.13)

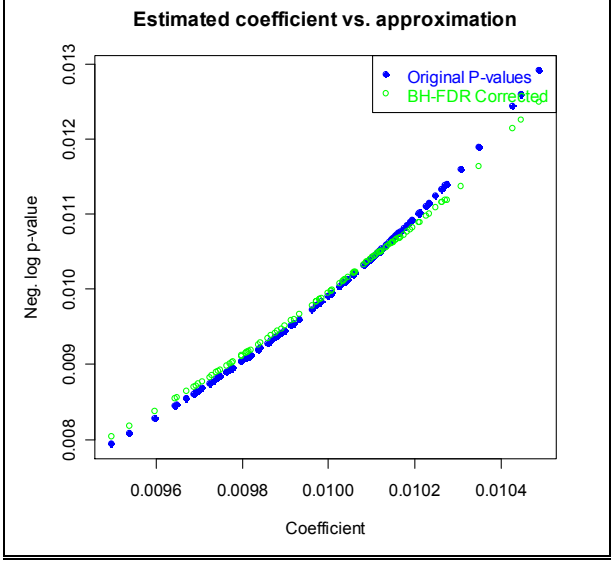


Figure 4.2.1 *Plot of regression coefficients versus the approximation (4.14)*

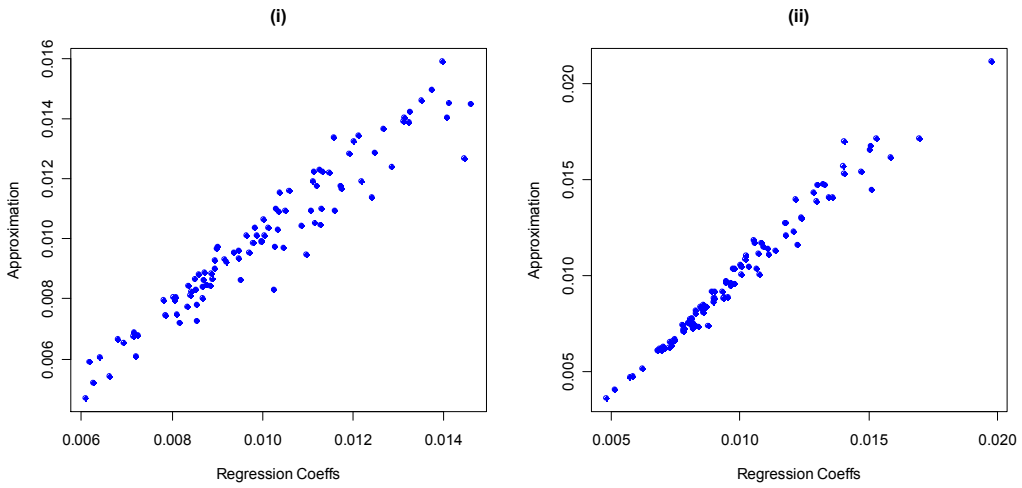


Figure 4.2.2 *Plots of logistic regression coefficients versus the approximation for two simulation scenarios*

In the next section we will consider the modified PLS scheme (PLS-FDR) where component weights are based on the approximation (4.14) and the use of the BH-FDR procedure for adjustment of the p-values to control the false discovery rate.

4.3 FDR-Corrected PLS Scheme (PLS-FDR)

Based on the approximations described in Section 4.2, we propose the modified PLS scheme (call it *PLS-FDR*), where the predictor weights for each component are approximated via the corrected p-values as in (4.14). The method is detailed below.

PLS-FDR Method

Suppose that we have the response Y (centered and scaled if continuous) and the set of already centered and scaled continuous predictors $X=\{x_j\}$ with $j=1,...,p$, i.e. they have $mean(x_j)=0, Var(x_j)=1$ for all j.

PLS-FDR Algorithm (1st component):

The PLS-FDR modification can be written as a set of the following steps:

1. Model the response Y versus each of the predictors X_i, $i=1,\ldots,n$
2. Calculate the p-value for each of the predictors to get the set of p_i, $i=1,\ldots,n$
3. Adjust the set of p-values for the multiple testing using BH-FDR correction to form the set of p_i^*, $i=1,\ldots,n$
4. Calculate weights $\{w_i\}$ for each of the predictors X_i, $i=1,\ldots,n$ using the approximation 4.14 with p_i^*, $i=1,\ldots,n$
5. Form the PLS component as the weighted sum of X_i

$$t = \sum w_i X_i$$

6. Perform the regression of Y on t to get the regression coefficient β.

Table 4.3.1 *PLS-FDR Algorithm for the 1st component*

If we want to calculate more than one component, the algorithm is similar to the one described above. Approximations in this case can for example be derived through the partial correlation coefficients. This is the topic for future investigation.

For the case of the logistic regression we will use the same algorithm with p-values arising from a t-test that compares two groups unless there is an indication that the response is derived from a logistic model as for example in a dose-response studies.

PLS-FDR Weights

When viewing the approximations (4.14) as weights assigned to each of the predictor variables, we can also investigate two more applications associated with the approach. Firstly, approximations can be used as individual weights for algorithms that operate them. Secondly, the classifiers that do not use weights can take advantage of the idea by

embedding them into an ensemble as follows (see Amaratunga et al. (2008) for a scheme with *1/p* weights):

PLS-FDR Ensemble Algorithm:

The PLS-FDR modification can be written as a set of the following steps:

1. Draw a bootstrap sample from the data. Call the observations which are not in the bootstrap sample the "out-of-bag" data.
2. Generate m randomly selected features according to the weights $\{w_i\}$ and use them together with the bootstrap sample to construct a classifier.
3. Use the classifier to predict out-of-bag data to form majority votes.
4. Repeat steps 1-3 N times and collect an ensemble of N rules. Prediction of test data is done by majority votes from predictions from the ensemble of rules.

Table 4.3.2 *Ensemble PLS-FDR Algorithm*

In the following section we will illustrate the performance of PLS-FDR method compared with regular PLS for the continuous response for one component. Then we will concentrate on the class-prediction methods. We will assess the performance of PLS-FDR methodology when compared to other discrimination methods such as SVM, KNN, DL-DQDA, Random Forest, Elastic Net and various extensions of PLS for binary response. Additionally, the ensemble of our weighting scheme with aforementioned classifiers will be compared with single classifiers and the ensemble method as in Amaratunga et al. (2008). Finally, we will look at the weighted classifier approach.

4.4 Simulation Settings and Results

We consider simulated as well as real-life datasets for performance comparison. When using real-life datasets, one can be certain that they adequately represent the complexity of the data structure. However, to assess the performance for the variety of

scenarios, we will also simulate datasets controlling different settings such as sample size, number of predictive genes or correlation structure.

Illustration: Continuous Case

Let's consider the continuous response setting for illustration purposes. We will use two of the real-life datasets available from the *pls* R package (*yarn* and *gasoline* data). Additionally, four scenarios will be simulated for the structure of the normally-distributed *X*-*Y* data:

(i) uncorrelated predictors and noise,

(ii) correlated predictors and noise,

(iii) uncorrelated predictors with addition of predictors mildly related to the response and noise,

(iv) correlated strong and mild predictors and noise.

Table 4.4.1 summarizes the rest of the parameters for the continuous case simulation. Figure 4.4.1 (a-d) show boxplots with the simulation results. Figure 4.4.2 summarizes the performance of *yarn* and *gasoline* datasets.

Parameter Values	Description
$n=40$	Number of observations
$p_0=25$	Number of strong predictors *X*
$p_1=200$	Number of weak predictors *X'*
$p_1=0\text{-}10000$	Number of noise variables
$\{\beta\}\sim U(0,1)$	Regression coefficients
$m=25\%$	Number of test set observations

Table 4.4.1 *Parameters of the simulation (continuous case)*

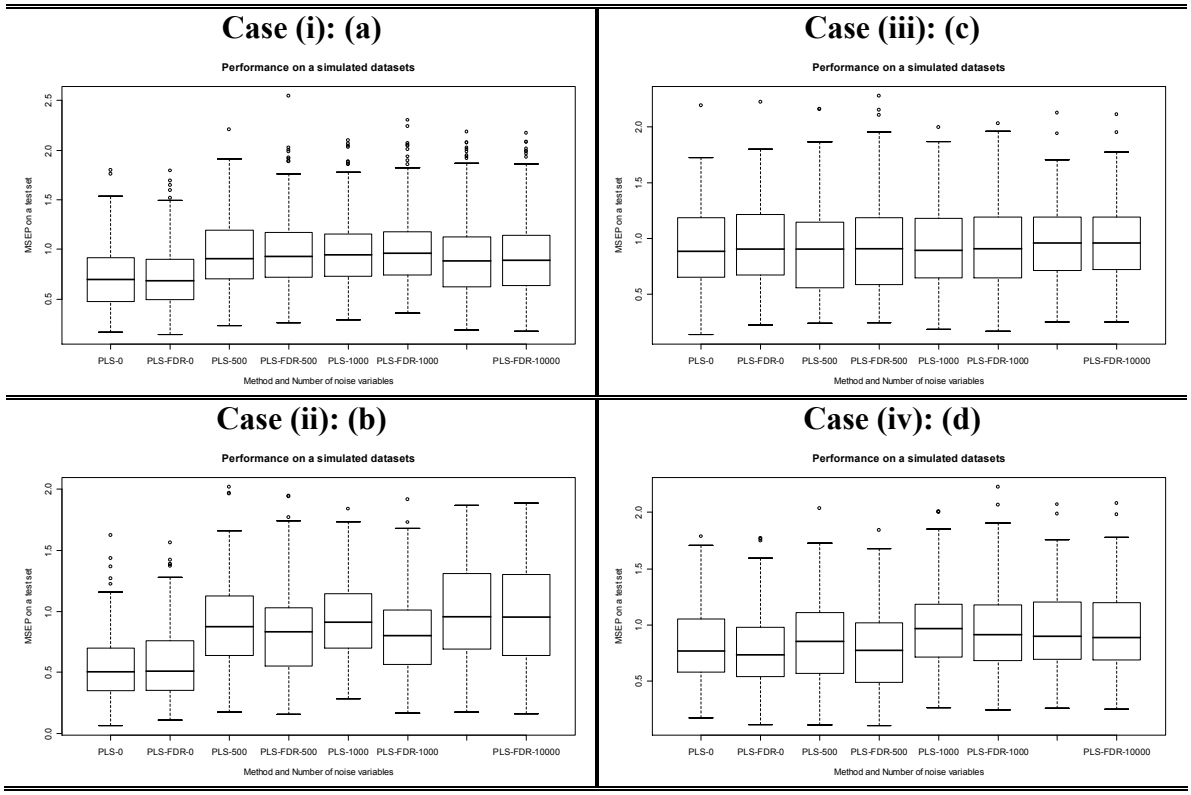


Figure 4.4.1 *Results of the simulation (continuous case)*
(MSEP on a test set containing random 25% of observations)

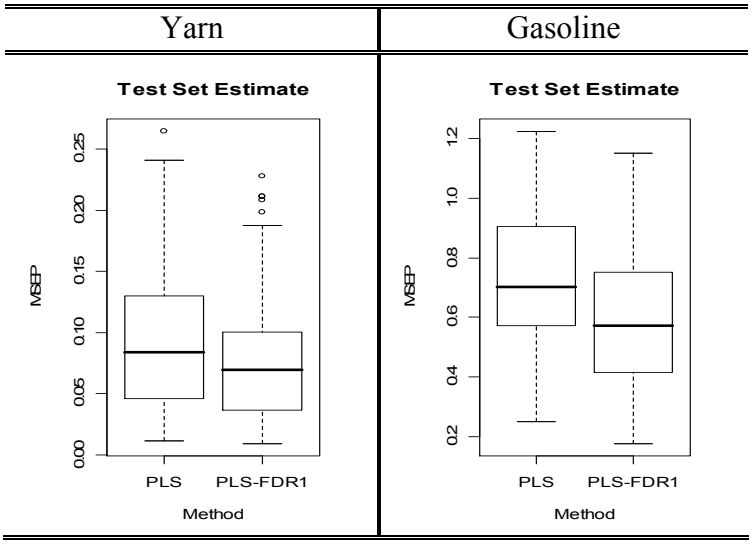


Figure 4.4.2 *Yarn and Gasoline data performance*
(MSEP on a test set containing random 25% of observations)

Performance: Binary Case

We will now turn to the settings of interest, i.e. two-group classification problem. We will compare PLS-FDR with the following nine methods: Linear Discriminant PLS, generalized PLS, DLDA, DQDA, Elastic Net, KNN, Random Forest and SVM.

The comparison schemes are represented in the Table 4.4.2 and we recognize three scenarios: classifier comparisons, combination of classifier and our weighting method, and ensemble of classifier and our weighting scheme. Utilized packages and their options are summarized in Table 4.4.3.

For the simulated data we recognize the following four main settings (as in the continuous case): (i) uncorrelated predictors and noise, (ii) correlated predictors and noise, (iii) uncorrelated predictors with addition of predictors mildly related to the response plus noise, (iv) correlated strong and mild predictors and noise. For every case described above, datasets were simulated according to the parameters as described in Table 4.4.4. Normalized intensities $\{X_{gij}\}$ were modeled as

$$X_{gij} = \mu_g + \tau_{gi} + \varepsilon_{gij} \qquad (4.15)$$

where μ_g $(g = 1, \cdots, G)$ – the effect of the gth gene, τ_{gi} is the effect of the gth gene in the ith class ($i = 1, 2$), and $j = 1, \cdots, n_i$ are sample indexes. The same model was described in Amaratunga and Cabrera (2006). The treatment effect of the gth gene is then $\tau_{gi} = |\tau_{g1} - \tau_{g2}|$. Finally, we assume that $\{\varepsilon_{gij}\}$ are iid observations from a multivariate normal distribution with covariance matrix defined to introduce the correlation between pre-specified genes.

The simulation results for all cases are shown in Figure 4.4.3 (a-d).

The performance of the methods summarized in Table 4.4.2 was also assessed on the real-life datasets (see Section 0.3 for description). In cases where response is not binary (srbct, brain and lymphoma datasets), the subset was used where response takes *{0,1}* values. Datasets are preprocessed by thresholding, filtering, a logarithmic transformation, and standardization as in Dudoit et al. (2002). For fifty simulations, about 25% of observations were retained for the performance assessment. For the ensemble of classifiers the usual square root of total number of genes was sub-sampled prior to the use of a single classifier. For the weighted Elastic Net the weights were constructed as the inverses of the approximation (4.14) to reflect the penalty for each predictor. They were introduced into the net through the parameter *penalty.factor* (number that multiplies lambda to allow differential shrinkage). For the weighted scheme, forty percent of observations were retained for testing.

The results for each dataset are presented in Figures 4.4.4, misclassification means are summarized Table 4.4.5 and Figure 4.4.7. Ensemble methods are compared in Figure 4.4.5, Figure 4.4.6 shows the weighted Elastic Net performance.

Method	Classifier	Ensemble Scheme	Weighted Scheme
PLS-LDA	*Y*	*Y*	
gPLS	*Y*	*Y*	
PLS-FDR	*Y*	–	–
DLDA	*Y*	*Y*	
DQDA	*Y*	*Y*	
Elastic Net	*Y*	*Y*	*Y*
KNN	*Y*	*Y*	
Random Forest	*Y*	*Y*	
SVM	*Y*	*Y*	

Table 4.4.2 *Comparison Schemes*

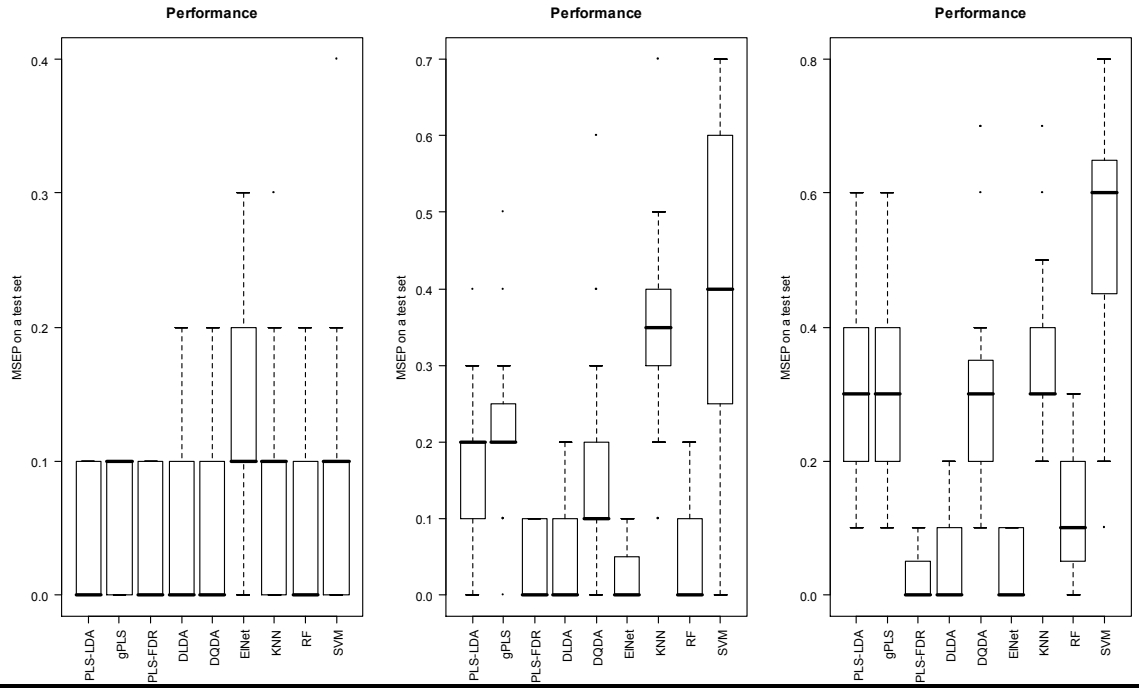
Method	Package/Function	Options
PLS-LDA	*plsgenomics*	*Default (k=1)*
gPLS	*gpls*	*IRWPLS (k=1)*
PLS-FDR	–	*first component*
DLDA	*sma - stat.diag.da*	*Constant cov matrix*
DQDA	*sma - stat.diag.da*	*Varying cov matrix*
Elastic Net	*glmnet*	*α=0.5, λ-auto or [0, 1]*
KNN	*class-knn*	*K=3*
Random Forest	*randomForest*	*300 trees*
SVM	*e1071-svm*	*Linear kernel*

Table 4.4.3 *Employed packages summary*

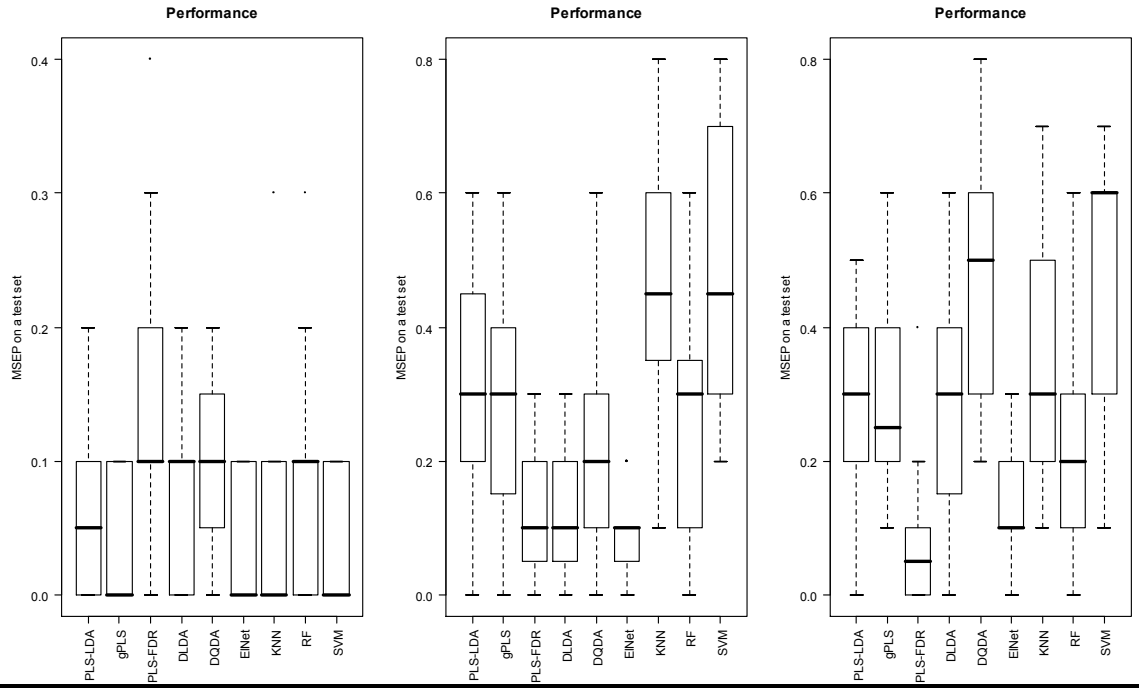
Parameter Values	Description
$n=40$	Number of observations generated
$p_0=25$	Number of strong predictors X
$p_1=200$	Number of weak predictors X'
$p_1=0\text{-}10000$	Number of noise variables
$\mu_g \sim N(0,2)$	Gene Effect
$\tau_{gi} \sim MVN((\mu_{25}, \mu_{200})^T, I)$	Gene-treatment effect (random means)
$\varepsilon_{gij} \sim MVN(0,\Sigma)$	Errors (correlation ranges from 0.2-0.6)
$m=25\%$	Number of test set observations

Table 4.4.4 *Parameters of the simulation (binary case)*

Case (i): (a)



Case (ii): (b)



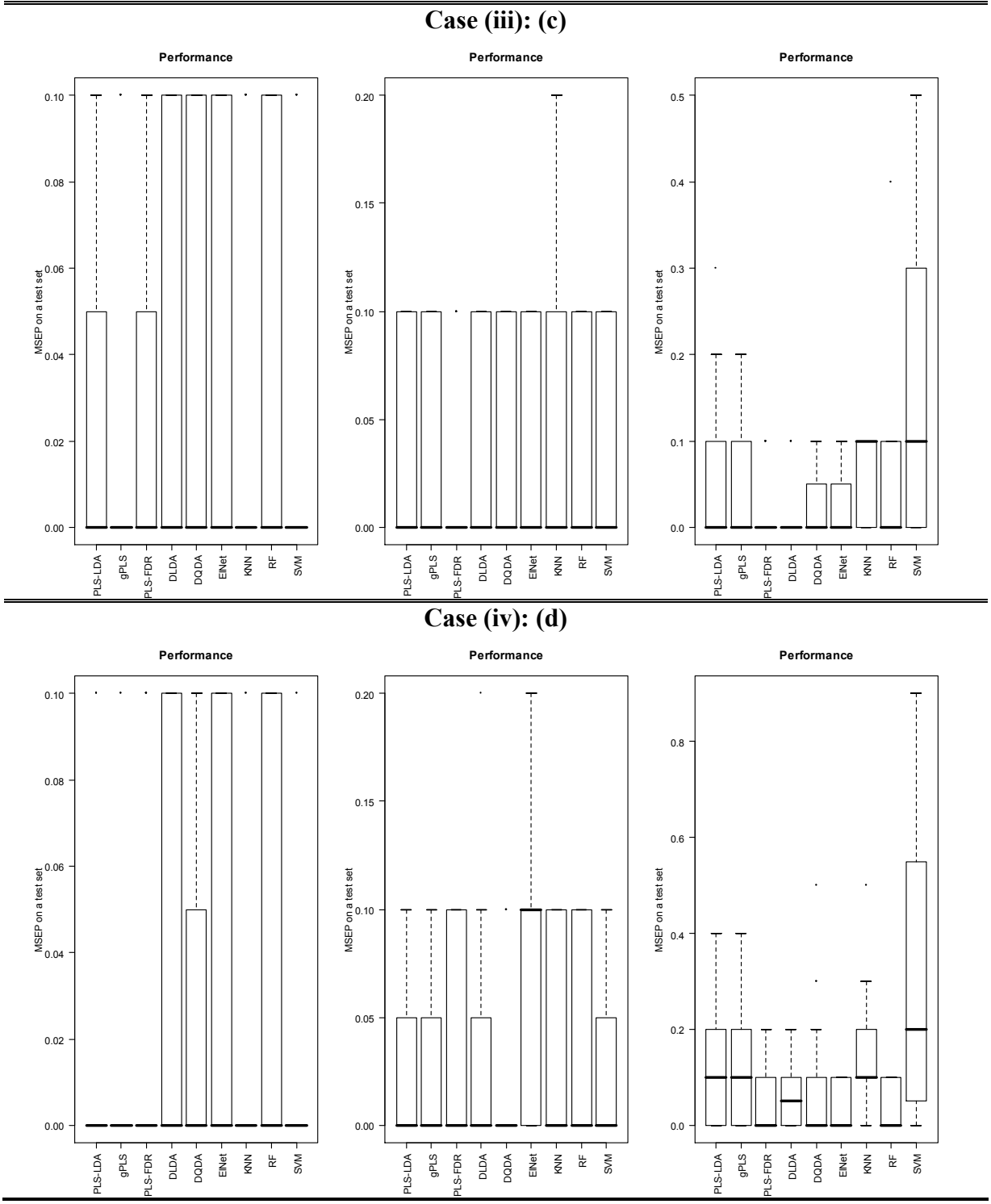


Figure 4.4.3 *Results of the simulation (binary case)*

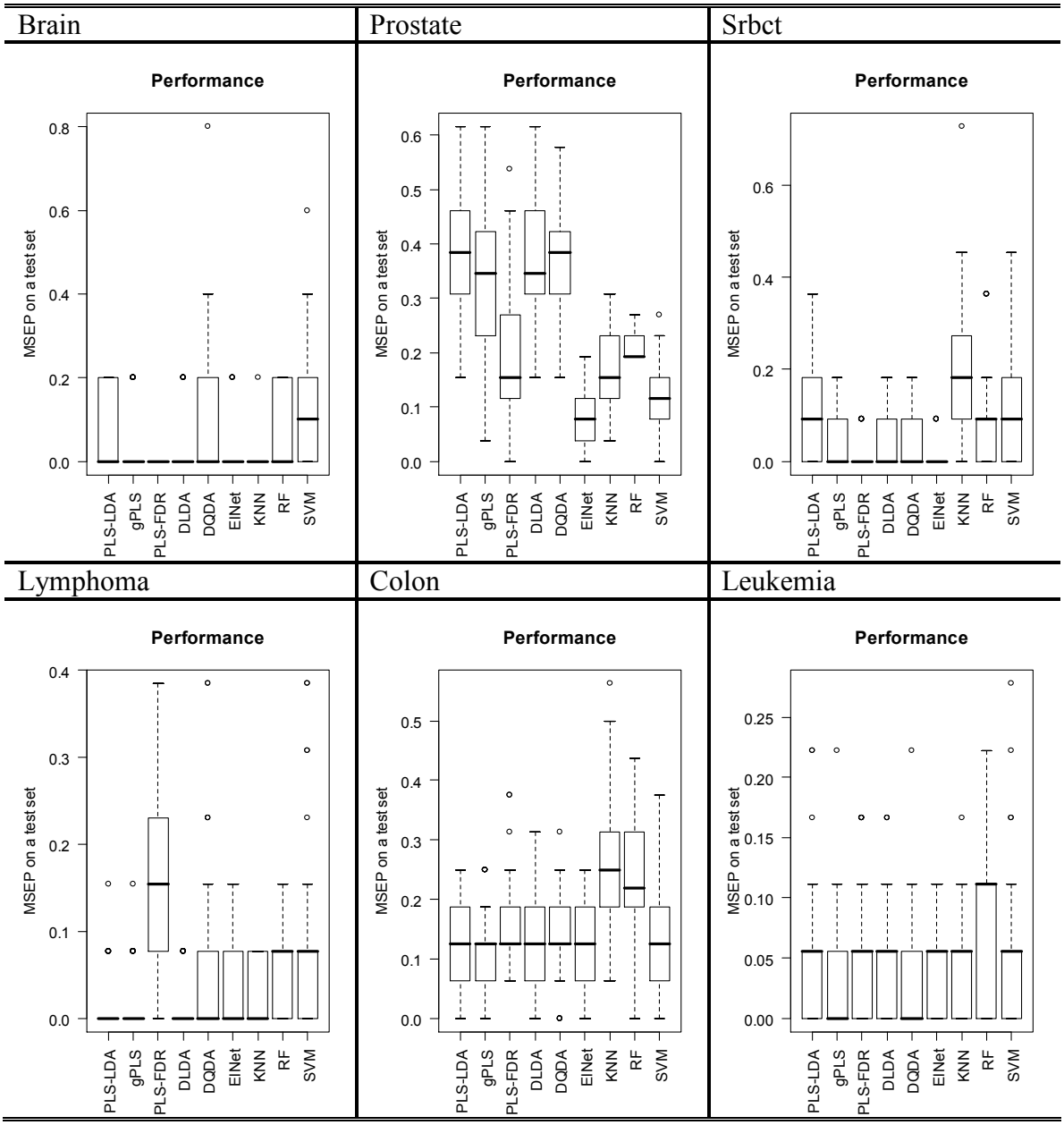


Figure 4.4.4 *Real-life data performance of classifiers*

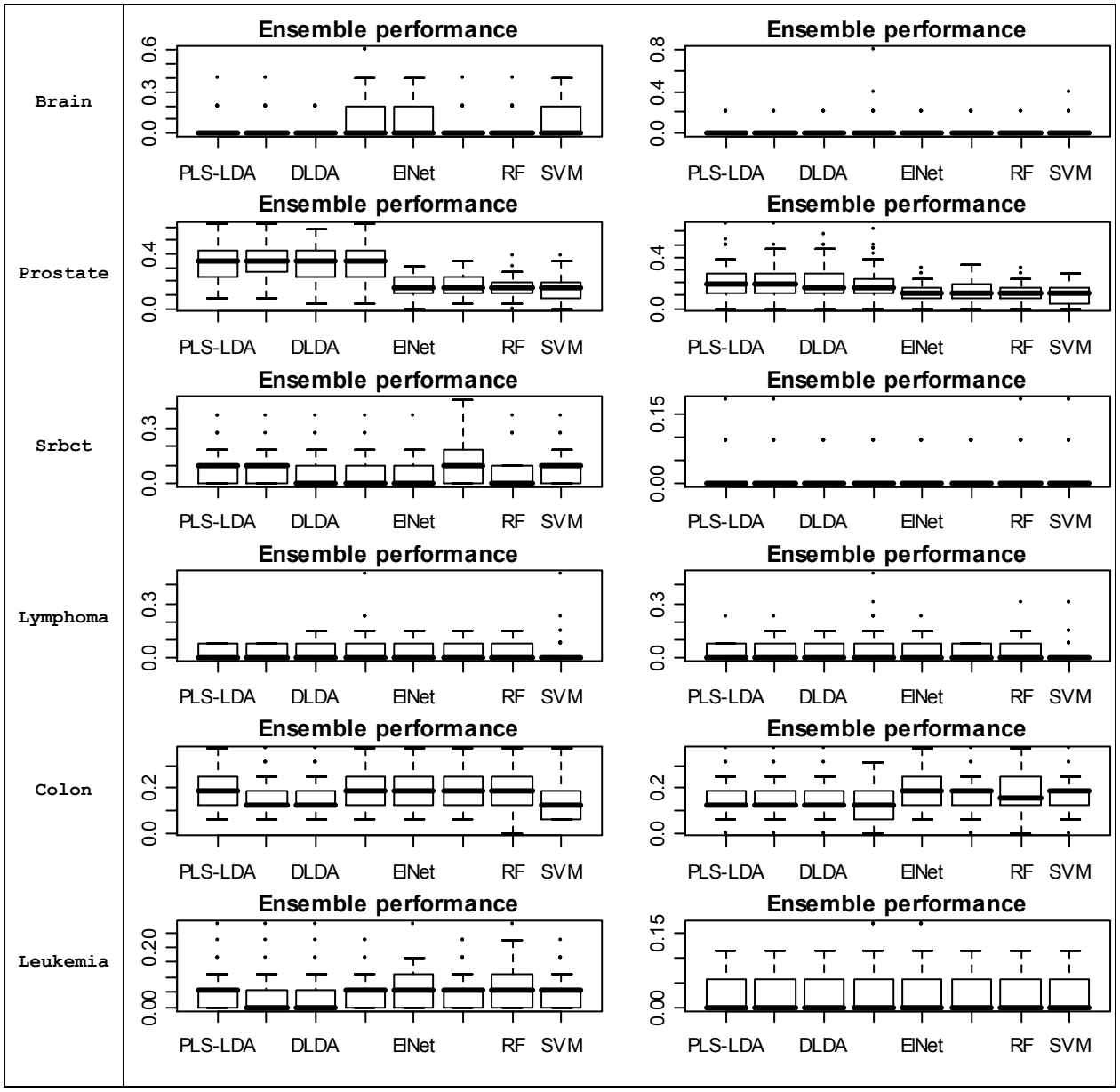


Figure 4.4.5 *Real-life data performance for ensembles with –log(p) and 1/p weights*

	Classifier	Ensemble - FDR	Ensemble - InvP
Brain	PLS-LDA 0.052 gPLS 0.044 **PLS-FDR 0.000** DLDA 0.028 DQDA 0.096 ElNet 0.016 KNN 0.004 RF 0.096 SVM 0.124	PLS-LDA 0.036 gPLS 0.040 DLDA 0.028 DQDA 0.088 ElNet 0.064 KNN 0.028 RF 0.044 SVM 0.064	PLS-LDA 0.020 gPLS 0.012 DLDA 0.020 DQDA 0.060 ElNet 0.020 KNN 0.016 RF 0.012 SVM 0.048
Prostate	PLS-LDA 0.37846154 gPLS 0.33923077 **PLS-FDR 0.20076923** DLDA 0.36615384 DQDA 0.37307692 ElNet 0.08461539 KNN 0.16846154 RF 0.21538462 SVM 0.12692308	PLS-LDA 0.3392308 gPLS 0.3284615 DLDA 0.3246154 DQDA 0.3192308 ElNet 0.1561539 KNN 0.1723077 RF 0.1515385 SVM 0.1515385	PLS-LDA 0.2069231 gPLS 0.2007692 DLDA 0.1961538 DQDA 0.1938462 ElNet 0.1130769 KNN 0.1369231 RF 0.1176923 SVM 0.1123077
Srbct	PLS-LDA 0.09090910 gPLS 0.05454546 **PLS-FDR 0.01272727** DLDA 0.05090910 DQDA 0.02909091 ElNet 0.01090909 KNN 0.18000002 RF 0.09090910 SVM 0.10545456	PLS-LDA 0.08545455 gPLS 0.07090910 DLDA 0.06363637 DQDA 0.06363637 ElNet 0.05818182 KNN 0.09090910 RF 0.04909091 SVM 0.09272728	PLS-LDA 0.02181818 gPLS 0.02181818 DLDA 0.02000000 DQDA 0.01636364 ElNet 0.01636364 KNN 0.02000000 RF 0.02000000 SVM 0.02545455
Lymphoma	PLS-LDA 0.01692308 gPLS 0.01692308 **PLS-FDR 0.15230769** DLDA 0.01384615 DQDA 0.06769231 ElNet 0.02615385 KNN 0.02000000 RF 0.06153846 SVM 0.08307692	PLS-LDA 0.02615385 gPLS 0.02461539 DLDA 0.02769231 DQDA 0.04461538 ElNet 0.03384616 KNN 0.02461539 RF 0.02461539 SVM 0.03230769	PLS-LDA 0.03076923 gPLS 0.03384615 DLDA 0.02923077 DQDA 0.05846154 ElNet 0.03230769 KNN 0.02923077 RF 0.03538462 SVM 0.02769231
Colon	PLS-LDA 0.12875 gPLS 0.12000 **PLS-FDR 0.15250** DLDA 0.13000 DQDA 0.13875 ElNet 0.11625 KNN 0.24125 RF 0.23125 SVM 0.14125	PLS-LDA 0.17750 gPLS 0.16250 DLDA 0.15875 DQDA 0.17875 ElNet 0.19875 KNN 0.18250 RF 0.18125 SVM 0.15875	PLS-LDA 0.14875 gPLS 0.14625 DLDA 0.14500 DQDA 0.14250 ElNet 0.19750 KNN 0.17250 RF 0.17125 SVM 0.16000
Leukemia	PLS-LDA 0.04555556 gPLS 0.03000000 **PLS-FDR 0.05111111** DLDA 0.04000000 DQDA 0.03000000 ElNet 0.04555556 KNN 0.04222222 RF 0.08888888 SVM 0.05000000	PLS-LDA 0.04777778 gPLS 0.04111111 DLDA 0.04111111 DQDA 0.05666667 ElNet 0.06444446 KNN 0.04888889 RF 0.05777778 SVM 0.04333334	PLS-LDA 0.02444445 gPLS 0.02444445 DLDA 0.02555556 DQDA 0.03444445 ElNet 0.03444445 KNN 0.02555556 RF 0.02666667 SVM 0.02222222

Table 4.4.5 *Comparison of MSEP means for three scenarios on a real-life data*

Brain	Prostate	Srbct	Lymphoma	Colon	Leukemia

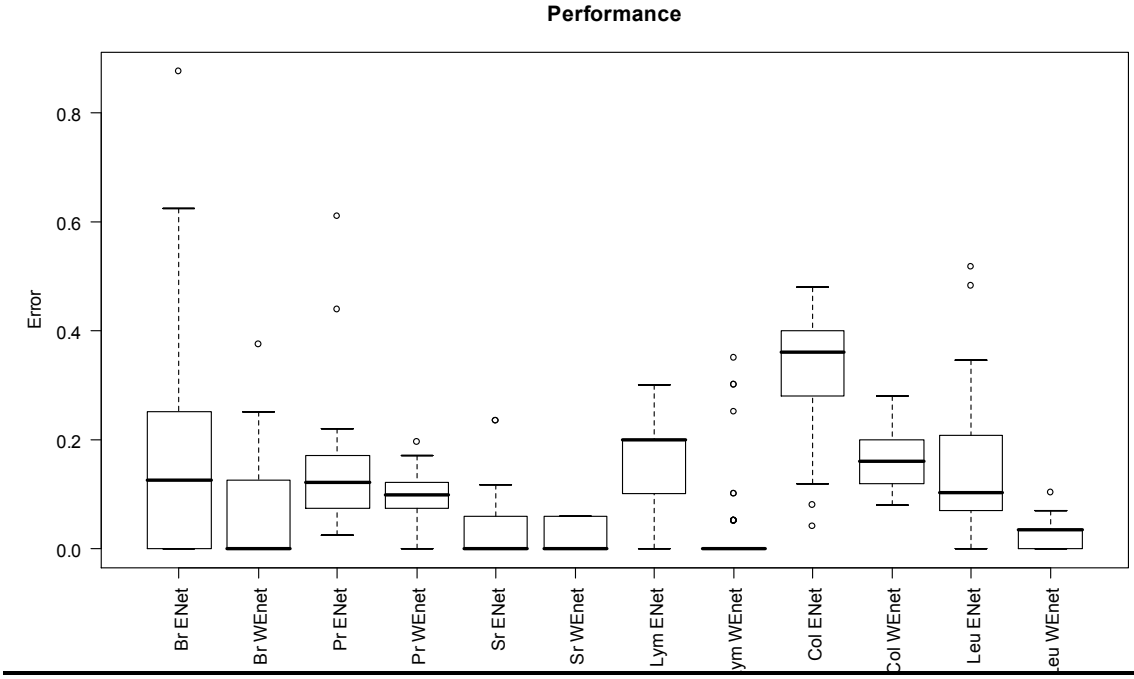


Figure 4.4.6 *Weighted elastic net performance on a real-life data*

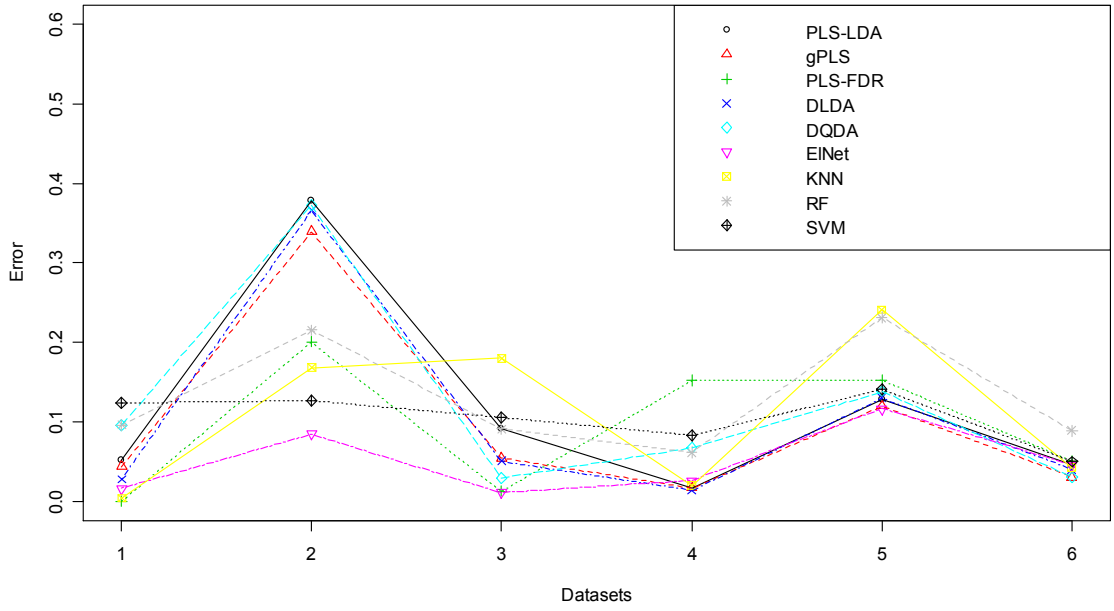


Figure 4.4.7 *Means for all methods (colors) for six datasets*
(1 - brain, 2 - prostate, 3 - srbct, 4 – lymphoma, 5 – colon, 6 – leukemia)

4.5 Discussion

As an illustration of the PLS-FDR methodology, we applied it to continuous response data. In this case, proposed algorithm gives some improvement over regular PLS method especially when correlation exists between predictors. This difference is also present when looking at the real-life datasets *yarn* and *gasoline*, considering only one component for both cases.

We also compared performance of various classification methods to PLS-FDR in terms of misclassification rate for binary response (ratio of number misclassified samples to the total number) for the simulated and six real-life datasets. Due to the computational complexity of the study described above, we limited our comparisons to six datasets and four scenarios for generated datasets with number of noise variables ranging from zero to a thousand.

The results of simulations presented above suggest that as the number of noise variables increases, PLS-FDR performs best on the simulated datasets compared to nine other classifiers including the elastic net. One may also consider other simulated scenarios and non-normality of the generated data for the performance assessment of the proposed classification scheme. This is topic for additional investigation.

When turning to the real-life datasets, PLS-FDR outperforms gPLS, PLS-LDA, DLDA and DQDA most of the time. It is often among the best classifiers for almost all datasets excluding the lymphoma dataset, and is comparable to the Elastic Net. Taking into account the fact that a single classification method is never uniformly better than the

others for all the datasets, we can conclude that the performance of a PLS-FDR is most of the time close to the optimal.

For the ensemble methods, the experimental results show that ensembles generally outperform single classifiers. Additionally, we compared the performance of our ensemble with the one where inverses of FDR-corrected p-values are used as weight (as in Amaratunga et. al (2008)). The performance of an inversed p based ensemble in most cases is better compared to the case with the approximation (4.14). However this does not happen for all combinations of methods and datasets.

Weighted elastic net algorithm with our weights introduced into model as additional penalty factors for each variable showed improvement over the elastic net procedure for all six datasets when using the same set of shrinkage parameters alpha and beta.

We also have to note that classifier comparisons obtained from the simulated and real-life datasets presented in this chapter do not include a gene selection step unlike the studies presented in Table 3.1.1. We can consider the use of our weights for a gene selection procedure, and compare it with common measures, such as t-statistic, the Wilcoxon rank sum or PLS component based ‘variable influence’ (Musumarra et al.). This is a topic for further investigation.

PART III.

COMPARING SEVERAL TREATMENTS

WITH A CONTROL

Chapter 5

Comparison of Several Treatments with a Control

5.1 Introduction

In a microarray experiment, the number of observations is generally very small. Various inference methods used for the analysis of such data mostly either rely on specific assumptions about the distribution of the expression measures, or rely on resampling of the data. Both types are used quite often. Resampling based tests has the advantage of being robust and flexible enough to accommodate almost any new statistic, without the need to derive statistic's distribution. However, they are computationally intensive and p-value distribution derived from a permutation-based approach can be coarse or granular, and it will often be difficult to obtain significant tests.

In this chapter, we discuss the situation of comparisons between several treatments and the control. In a microarray setting, the goal of these comparisons is to find subset of genes whose expression levels differentiate between treatments and the control. To illustrate this setting, we consider the following two experiments of data arising from microarray and also non-microarray settings:

(I) For the first example, suppose we administer a drug and we have 4 groups of observations containing blood measurements at 0- hours; at 0+ hours; at 2 hours and at 24 hours. The dataset consists of m=2375 genes and n_k=7 measurements for each of the 4 ($k=0,...,3$) groups. We are interested in selecting those genes for which the measurement at 2 hours is significantly higher or lower than the other three groups (so called 'bump' in measurements).

(II) For the second example of a non-microarray setting, we consider weight measurements of m=66 patients over three weeks when they are being administered a certain drug. There are 5 to 12 measurements for each patient taken at each week. We are again interested in identifying the patients with the weight measurement at week 2 much higher or lower than those at weeks 1 or 3.

Two of the most common techniques used when dealing with several group comparisons are Dunnett's test (Dunnett 1955, 1964) and permutation-based approach. Dunnett test is a single step procedure that does the many-to-one comparisons simultaneously for every single gene using the multivariate-t distributed test statistic. Both techniques are therefore subjects to the multiple comparisons issue. We will again adjust for the multiple comparisons using the BH-FDR procedure (Section 2.5.2). BH-FDR procedure does not rely on an asymptotic distribution of test statistics which is especially relevant for the examples (I) and (II) above, where the sample sizes for each group of measurements are rather small. Other authors also considered extension of Dunnett's procedure (Cheung and Holland, 1991), testing of medians (Steel, 1959), various step procedures (Nakamura and Imada 2005; Imada and Douke 2007).

The content of this chapter is organized as follows. In Section 5.2 we formulate the problem of comparing several treatments with one control as in experiments (I) and (II) and discuss the statistics arising from the hypotheses. In Section 5.3 we briefly cover Dunnett's procedure, permutation-based approach and detail the distributional approach which is suitable for the settings described above. Section 5.4 presents results of application of these methods for the examples (I) and (II). The chapter ends with a discussion in Section 5.5.

5.2 Hypotheses Formulation

The usual setting in the case of comparing several treatments with a control is to assume normal distribution for the log-transformed gene expression measurements.

Let X_{ijk} be the i^{th} gene-expression of array j in group k, $i=1,...,m$, $j=1,..., n_k$, $k=0,...,3$. Then the following is assumed

$$X_{ijk} \sim N(\mu_{ik}, \sigma_i^2) \qquad (5.1)$$

where μ_{ik} is the mean expression level for the group k for gene i.

In order to assess whether the mean of one of the groups for the gene i dominate the others in a 'bump' fashion, the following hypotheses are to be tested for $i=1,...,n$:

$$H_{0i}: \mu_{i0} = \mu_{i1} = \mu_{i2} = \mu_{i3} \qquad (5.2)$$

$$H_{ai}: \mu_{ik} < \mu_{i0} \text{ for } k=1,...,3 \text{ or } \mu_{ik} > \mu_{i0} \text{ for } k=1,...,3$$

To perform the above hypotheses testing we propose the test statistic

$$T_i = \max(\bar{x}_{i0} - \max(\bar{x}_{i1}, \bar{x}_{i2}, \bar{x}_{i3}), \min(\bar{x}_{i1}, \bar{x}_{i2}, \bar{x}_{i3}) - \bar{x}_{i0}) / SE$$

$$SE = \sqrt{\frac{\sum_{k=0}^{3} \sum_{j=1}^{n_i} (x_{ijk} - \bar{x}_{ik})^2}{\left(\sum_{k=0}^{3} n_k - 4 \right)}} \qquad (5.3)$$

where $\bar{x}_{ik}$ is the mean of gene i for group k, n_k is the sample size.

This test statistics is expected to take small positive or negative values under the null hypothesis. The null hypothesis will be rejected for large positive values of T_i. Note that the above formulation can be extended to any number of groups.

5.3 Approaches

There are several approaches to test (5.2). One alternative is to recycle a related test, such as the F-test or Dunnett's test for multiple comparisons. The power in these cases will be reduced because these related tests are not tailored for these specific hypotheses. Another approach is already mentioned above permutation-based approach. Below we will provide the brief overview of Dunnett's and permutation tests.

One can also try to calculate the distribution of the statistic T_i under the null hypothesis. Despite the fact that different experiments have different number of groups and will require modifications of T_i, this approach will give us a more accurate and powerful result. The framework for the T_i distribution will be detailed in 5.3.3.

5.3.1 Dunnett Approach

Dunnett considered the following set of hypotheses to compare several treatments with a control:

$$\begin{array}{ll} H_{01i}: \mu_{i0} - \mu_{i1} = 0 & H_{11i}: \mu_{i0} - \mu_{i1} \neq 0 \\ H_{02i}: \mu_{i0} - \mu_{i2} = 0 & H_{12i}: \mu_{i0} - \mu_{i2} \neq 0 \\ H_{03i}: \mu_{i0} - \mu_{i3} = 0 & H_{13i}: \mu_{i0} - \mu_{i3} \neq 0 \end{array} \qquad (5.4)$$

The test statistics for the hypotheses formulated above can then be written as

$$\tilde{T}_{ik} = \frac{\bar{x}_{ik} - \bar{x}_{i0}}{s_i \sqrt{\frac{1}{n_k} + \frac{1}{n_0}}} \; i = 1, \ldots, m; \; k = 1, 2, 3 \qquad (5.5)$$

where s_i is the pooled variance for gene i.

Author then proposed the following set of $1-\alpha$ level simultaneous confidence intervals for comparisons for gene i $(i = 1, \ldots, m)$ and the group k $(k = 1, 2, 3)$:

$$(\mu_{ik} - \mu_{i0}) \in (\bar{x}_{ik} - \bar{x}_{i0}) \pm |t|^{\alpha}_{k,\nu,\rho}\, s_i \sqrt{\frac{1}{n_k} + \frac{1}{n_0}} \qquad (5.6)$$

where $|t|^{\alpha}_{k,\nu,\rho}$ is the two-sided upper α quantile of a multinomial t-distribution with the following parameters: $k=3$ variables, equicorrelated with common correlation $\rho=n_k/(n_k+n_0)$, and number of degrees of freedom $\nu = \sum_{k=0}^{3} n_k - 2$. The values of $|t|^{\alpha}_{k,\nu,\rho}$ have been tabulated in Bechhofer and Dunnett (1988). Dunnett (1980) also considered the case of unequal variances and unequal group sample sizes.

5.3.2 Permutation-based Approach

The Dunnett p-values are valid only if the distributional assumption in (5.6) holds. In order to overcome this problem, one can use permutation-based technique, which does not require any assumptions about the distribution of the gene expression values. The set of p-values for this approach are obtained by using a certain number of random permutations of the sample labels and by calculating the test statistics T_i for the newly formed treatment groups. Number of permutations N does not always equal to the total number of possible permutations and could be equal some pre-defined large number in the case when total number of observed values is large. Once the set of test statistics $\{T_i^1, ..., T_i^N\}$ is calculated for all permutations, the p-values are obtained through calculating the percent of values $\{T_i^1, ..., T_i^N\}$ that are larger than the observed value of T_i. If the p-value is less some predetermined significance level then we say that the gene corresponding to the observations is differentially expressed.

5.3.3 Distributional Approach

Permutation-based approach also has limitations. They are mostly due to computational complexity and the fact that sometimes generated p-values are not tight enough to allow one to obtain significant tests. For a particular statistic (5.3) one can then obtain an approximated distribution and draw conclusions from it.

Each of the means of *k* (we have 3 groups for the case (I), and 2 groups for the case (II)) groups follows the normal distribution for gene *i, i=1,...,m*

$$(\overline{x_{i0}},...,\overline{x_{ik}}) \sim N(\underline{\mu^i},\Sigma_i) \qquad (5.7)$$

Let

$$s^2 = \frac{\sum_{k=0}^{K}\sum_{j=1}^{n_i}(x_{ijk}-\overline{x}_{ik})^2}{\left(\sum_{k=0}^{K} n_k - 4\right)} = \frac{\sum_{k=0}^{K}\sum_{j=1}^{n_i}(x_{ijk}-\overline{x}_{ik})^2}{f} \qquad (5.8)$$

then we can rewrite the statistic T_i

$$T_i = \max(\min(\overline{x}_0-\overline{x}_1,...,\overline{x}_0-\overline{x}_k),-\max(\overline{x}_0-\overline{x}_1,...,\overline{x}_0-\overline{x}_k))/s = \\ = \max(y_{(1)},-y_{(k)})/s \qquad (5.9)$$

Then we need to identify the upper percentage points from the probability distribution of T, i.e. number $d_\alpha = d(\alpha,k,f)$ such that

$$\Pr[T_i \le d_\alpha] = 1-\alpha \qquad (5.10)$$

If we set $u = s^2/\sigma^2$, then we can write the probability above as

$$\Pr[T \le d_\alpha] = \\ \int_0^\infty \Pr[\max\frac{(y_{(1)},-y_{(k)})}{\sqrt{\sigma^2}} \le \sqrt{u}\,d_\alpha]\cdot h(u)\cdot du \qquad (5.11) \\ = \int_0^\infty F_{\max(y_{(1)},-y_{(k)})}(\sqrt{u}\,d_\alpha)\cdot h(u)\cdot du$$

where $h(u) = \frac{f^{f/2} e^{-uf/2} u^{f/2-1}}{\Gamma(f/2) \cdot 2^{f/2}}$ is the chi-squared pdf with f degrees of freedom,

$F_{\max(y_{(1)}, -y_{(k)})}(Y)$ is the cdf of $\max(y_{(1)}, -y_{(k)})$.

When we assume equal variances for groups and equicorrelation of mean differences, then under the null hypothesis we can look at the distribution of $(\overline{x}_0 - \overline{x}_1, ..., \overline{x}_0 - \overline{x}_k)$ which is permutation-symmetric multivariate normal. Then we can use the joint distribution of *min* and *-max* of multivariate normal random variables and derive the distribution of their maximum.

Order statistics were studied quite extensively in the literature and results were summarized by several authors (for example Balakrishnan and Rao (1998), David (2003)). The multivariate normal distribution and order statistics arising from it are assessed in detail in Tong (1990).

The joint distribution of order statistics of exchangeable multivariate normal variables is a mixture of the joint distributions of order statistics of i.i.d. normal variables. It equals the following expression:

$$g_{(1,\ldots,k)}(y_1, ..., y_k) = k! \int_{-\infty}^{\infty} \frac{1}{\left(\sigma\sqrt{1-\rho}\right)^k} \times \prod_{t=1}^{k} \phi\left(\frac{\left(y_t - \mu\right)/\sigma + \sqrt{\rho} z}{\sqrt{1-\rho}} \right) \phi(z)\, dz \tag{5.12}$$

Then for the minimum and maximum order statistics we have the following joint distribution:

$$g_{(1,k)}(y_1, y_k) = k! \int_{-\infty}^{\infty} \frac{f_{(1,k)}(v_1, v_k)\, \phi(z)\, dz}{\sigma^2 (1-\rho)} \tag{5.13}$$

where $v_t = \dfrac{(y_t - \mu)/\sigma + \sqrt{\rho}z}{\sqrt{1-\rho}}$, and

$$f_{(i,j)}\left(z_i, z_j\right) = \frac{k!}{(i-1)!(j-i-1)!(k-j)!}\Phi^{i-1}(z_i)\times\left[\Phi\left(z_j\right)-\Phi\left(z_i\right)\right]^{j-i-1}\Phi^{k-j}(-z_j)\phi\left(z_i\right)\phi\left(z_j\right)$$

Substituting (5.13) into (5.11) with zero mean and using the fact that

$$\Pr\left\{\max(y_{(1)}, y_{(k)}) \le Y\right\} = G_{(1,k)}(Y,Y) \qquad (5.14)$$

we get the following expression

$$\begin{aligned}
&\Pr[T \le d_\alpha] = \\
&= \int_0^\infty G_{(1,k)}(\sqrt{u}d_\alpha, -\sqrt{u}d_\alpha)\cdot h(u)\cdot du \qquad (5.15)\\
&= k!\int_0^\infty \int_{-\infty}^{\sqrt{u}d_\alpha}\int_{-\sqrt{u}d_\alpha}^{\infty}\int_{-\infty}^{\infty} \frac{f_{(1,k)}\left(v_1, v_k\right)\phi\left(z\right)}{\sigma^2\left(1-\rho\right)}\cdot dz\cdot dy_1\cdot dy_k\cdot h(u)\cdot du
\end{aligned}$$

For the examples (I) and (II) above we have either *k=2* or *k=3*. The expression (5.15) can then be evaluated for various d_α to obtain set of values corresponding to the set of *α*. Calculations for the percentage point values d_α for expression (5.15) were performed in Wolfram Mathematica for *k=2,3* and *α=0.05, 0.10*. They are reported in Table 5.4.1 below. Additionally, the approximate plots of the T statistic distribution are presented in Appendix C for both cases.

Number of groups k	***Type I error α***	***Percentage points d_α***
2	0.05	1.691265
	0.10	1.274361
3	0.05	1.157234
	0.10	0.866303

Table 5.4.1 *Table of the percentage points*

5.4 Application to the Data

Microarray Data

We will first look at the dataset (I) described in a section 5.1.

First, we will consider the Dunnett single-step testing scheme. It produces p-values by testing all genes simultaneously and the three null hypotheses in (5.4). Table 5.4.2 presents the results using this approach. Among the 2375 genes, the number of genes with at least one significant comparison is 199.

As a next step we applied the permutation-based approach. In our example (I) number of total permutations is around 4.7×10^{14}, which is quite large, so the p-values can be calculated following the above described procedure by drawing N=20000 samples at random, which yielded 20000 values of T_i. The results are also summarized in the Table 5.4.2. Among the 2375 genes, the null hypothesis is rejected for 148 tests identifying 148 genes.

Finally, a distribution-based approach is considered. For three groups and $\alpha=0.05$ there were only 26 genes with a statistic below the cut-off value from Table 5.4.1. Hence, we reject the null hypothesis for these genes. The results of distributional approach are also presented in the Table 5.4.2.

Additionally, Appendix B contains identifiers of genes selected as differentially expressed by each approach, 25 of the genes overlap.

Weight Data

Now turning to the non-microarray experiment (II) with weight data, we will apply the same three approaches as above. There are $k=2$ groups with various sample sizes for $m=66$ patients. Despite unequal number of observations in each group, we will

still use the distributional approximation derived in Section 5.3. This is due to complication in the joint distribution of order statistics, since multivariate normal distribution in the case of unequal group sizes is no longer permutation-symmetric.

Results of three approaches are summarized in Table 5.4.3. Additionally, Table 5.4.4 contains identifiers of patients, for whom the null hypothesis was rejected. Permutation-based approach rejects null hypothesis for 50 patients, while Dunnett test selects only 42. Distribution-based approach selected the smallest number of patients again which equals 38. The sets of selected patients overlap for 22 members.

Approach	**(1) Dunnett**	**(2) Permutation**	**(3) Approximated Distribution**
Number of genes declared significant	199	148	26

Table 5.4.2 *Number of significant genes for dataset (I) identified using (1) Dunnett, (2) permutation, (3) distribution-based approaches.*

Approach	**(1) Dunnett**	**(2) Permutation**	**(3) Approximated Distribution**
Number of patients declared significant	42	50	38

Table 5.4.3 *Number of patients declared significant for dataset (II)*

Approach	**(1) Dunnett**	**(2) Permutation**	**(3) Approximated Distribution**
Patients declared significant	1 2 3 4 5 6 7 8 9 11 14 16 17 18 19 21 23 24 26 27 30 31 33 34 36 38 40 41 44 45 46 47 48 50 51 53 54 55 56 63 65 66	1 2 3 4 5 6 7 8 9 10 11 16 17 18 19 20 21 23 24 26 27 29 30 31 32 33 34 36 37 38 39 41 43 44 45 46 47 48 50 51 52 53 54 55 56 58 59 61 64 65	2 3 4 7 9 10 11 14 16 17 18 21 23 24 25 28 30 31 32 35 37 38 39 42 44 45 46 49 51 52 53 56 58 59 60 63 65 66

Table 5.4.4 *IDs of Patients declared significant for dataset (II)*

5.5 Discussion

The aim of the experiments considered in this chapter was to find 'bump' in measurements for examples (I) and (II) above. In this chapter we considered three approaches – Dunnett's test, permutation-based and distribution-based techniques. The results presented in the Section 5.4 reveal substantial differences between different methods employed. The distribution-based procedure led to the smallest number of significant findings than the other two testing methods. With a small sample size (as in the weight dataset case), the permutation approach tends to pick the larger number of patients than Dunnett's and distribution-based techniques. When the sample size is larger (for the microarray dataset), the number selected by a permutation-based method is in the middle of other two approaches.

The drawback of Dunnett's method is that the correction made to the test statistic weakens the significance of truly significant results. For the permutation-based approach, the issue related to the small sample size is that p-values generated are not tight enough to allow one to obtain significant tests and leads to distorted conclusions. Therefore, distribution-based approach gives a way to correct for the above-mentioned issues and obtain relevant results.

PART IV.

MENU-DRIVEN PACKAGE

FOR ANALYSIS WITH R

Chapter 6

PfarMineR

6.1 Introduction

Statistics is a fast growing field and novel methods for analysis and exploration of data emerge constantly. The R software (Ihaka and Gentleman (1996); R Core Development Team (2004)) is a tool widely used for implementation of the newly developed methods as R packages, which then are quickly disseminated via CRAN, Bioconductor or other web resources. Once these new methods prove their usefulness, they are incorporated into commercial software and become standard tool for the data exploration. However, the commercial software (primarily SAS or SPSS) while generally providing an excellent production environment, lack the timely implementation of state of the art methodology.

A large group of statisticians who work in regulated production environments use SAS or SPSS software and may not have the training in R necessary to apply new methods to their data. Hence there is a need for the bridge software (in this case an R package). For this reason we introduce PfarMineR to make the transition between R and SAS/SPSS easy for individuals not familiar with the R command-line environment.

PfarMineR is a menu-driven interface to a subset of methods in R that is easily expandable and can be tailored to users' needs. Compared to other menu-driven packages (see Fox (2005)), PfarMineR does not require any add-ons, has simple intuitive menus and dialogs, can work automatically or with a user-specified options. This approach is suited the best for statisticians that are not experts in R and students of a specific course that uses PfarMineR as a teaching tool.

The purpose of this chapter is to outline the package capabilities and functions as well as underline options to customize and further expand the software. In the next section we will summarize the package design tree and give an overview of currently included features. Section 6.3 will describe the way to implement modifications. Then we will end this chapter with a discussion.

6.2 Summary of the Design

Once the PfarMineR package has been installed and loaded into R, it automatically creates the menu called "PfarMineR" consisting of the four main submenus and additional four command buttons. The summary for submenus is provided in the Table 6.2.1. The functions of the command buttons are described in the Table 6.2.2. Extensive additional details for both submenus and command buttons that include options and names of functions can be found in the Appendix 1.

Submenu	Data	Exploratory Data Analysis	Classifications	Clustering
Main Function	Contains methods of data extraction and variable manipulation	Contains methods of data extraction and variable manipulation	Contains various classification methods	Contains various clustering methods
Description	1.Load SAS, Excel and TXT files. 2.Save or load data frame can be saved or loaded 3.Provide data summary 4.Edit Variables	1.Data Visualization 2.Response Visualization 3. Basic EDA Stats 4.Data Transformations 5.Robust Regression 6.Lasso 7.Simulation Methods and Bootstrap	1.ARF 2.CART 3.Neural networks 4.SVM 5.Naive Bayes 6.Random Forest 7.Boosting	1.Hierarchical 2.PAM 3.Silhouette Plots 4.K-means 5.Model-based
Output	1. 'Edit Variables' menu contains variable names and variable manipulation options; 2. Summary of the dataset 3. Number of missing values.	Depends on a method 1-7 (see Appendix 1)	Depends on a method 1-7 (see Appendix 1)	Depends on a method 1-5 (see Appendix 1)

Table 6.2.1 *Main sub-menus overview*

Report All	Commands History	Add Method	MANUAL (AUTOMATIC) OPTIONS
Creates the pdf file (REPORTALL.pdf) in the current directory that contains the output of all methods and functions included in the package.	Brings up the window that shows the history of commands used in the current session.	Creates the new menu button with the function assigned to it that was specified by the user.	Switch between automatic default and manual user-specified options.

Table 6.2.2 *Command buttons overview*

6.3 Implementation of Modifications

The present package is structured in a way that allows for user-specified updates and modifications. In this section we will describe those capabilities and provide a series of steps for guiding the user towards implementing the following three types of updates: (i) modification of the output; (ii) selection of the options; (iii) implementation of a new method.

Modification of the Output

To allow flexibility in output destination and contents, the following are the options implemented in the PfarMineR. First of all, each method includes “Output to Window” or “Output to PDF” items for the destionation selection. Secondly, selection of the output contents is done via the pop-up menu. For most methods the pop-up menu includes options such as display of the object, object summary, graphical output or all of the above. Examples of the output are presented in the Appendix 2.

Selection of the Options

Another important feature of the software is the ability to customize any options of the selected method. This feature has to be first turned on via the “MANUAL OPTIONS” command button in the main menu. Each method is provided with the set of reasonable defaults that are applied automatically. With the “MANUAL OPTIONS” switch turned on, the user is communicated via the dialog screen where he can either accept default settings or to modify them according to his preferences. Once parameters are specified, they are automatically substituted for the method of interest.

Implementation of a new method

Finally, the user is given the ability to expand range of methods provided by the PfarMineR with a new method. This can be done through the "Add Method" command button. Then the user has to perform the following steps:

(i) provide the name for the new item in the menu;

(ii) provide the name of the function associated to this menu item;

(iii) write the function corresponding to the template (see Table 6.3.1 for the template and example)

Function Arguments	`Dataset (xdatset) and output type (pdf) and others that needed be a specific method`
Capture responses commands	`response<-xdatset[,attributes(xdatset)$tab[,2]==TRUE]` `attributes(xdatset)$tab[attributes(xdatset)$tab[,2]==TRUE,1]<-FALSE`
Capture predictors commands	`x<-xdatset[,attributes(xdatset)$tab[,1]==TRUE]`
Output to PDF commands	`pdf(file=pdfname,width=10,height=7.5)` `dev.off()`

Choose output type commands	`outtp<-'All'`
	`if (type==0) {`
	`outtp<-select.list(c('All','Object','Summary','Graphics'), title = "Choose Output Type") }`

Change parameters commands	`if (manual==TRUE) {`
	`defp<-' weights= NULL, subset= NULL,na.action= na.omit'`
	`s<-winDialogString("Change options below", default=defp)`
	`eval(parse(text=paste("tmp <- list(",s,")")))}`

Example	`myknn=function(xdatset){ library(class);`
	`response<-xdatset[,attributes(xdatset)$tab[,2]==TRUE]`
	`attributes(xdatset)$tab[attributes(xdatset)$tab[,2]==TRUE,1]<-FALSE`
	`x<-xdatset[,attributes(xdatset)$tab[,1]==TRUE]`
	`knn(x,x,cl=response,k=1,l=0,prob=FALSE,use.all=TRUE)}`

Table 6.3.1 *New method template*

6.4 Discussion

The PfarMineR package is a simple and elegant tool that serves variety of purposes. First of all, it provides intuitive interface appealing to those who only begin to use the R software. This group of users will be able to slowly develop R programming skills while already using R features. Then PfarMineR can also be used by more advanced users to perform initial crude analysis of the dataset and then go into detailing the analysis options. Finally, the package provides updatable flexible environment, with a modular to expand architecture, which allows the implementation of new methods at user's convenience. The download link is available in the Appendix 3.

PART V.

CONCLUDING REMARKS

Chapter 7.

Concluding Remarks and Future Research

In this dissertation we considered several questions arising from the analysis of microarray data. We were primarily interested in classification and hypothesis testing problems for gene expression data.

The first part of the dissertation was an introduction to the microarrays and statistical analysis of microarray data. Then in Chapters 3 and 4, we assessed the problem of classification tasks in microarray experiments and proposed the PLS-FDR scheme. We noted that PLS-FDR scheme can also be used in two more settings such as ensemble and weighting. Comparing the performance of proposed scheme with other classifiers, we observed that for simulated data PLS-FDR outperforms other classifiers when number of noise variables gets large. For real-life datasets we found that the proposed methodology is also among the best classifiers for most datasets. PLS-FDR generated weights also improve the performance of each single classifier if used as a weighting scheme or part of the ensemble.

One of the interesting applications of the PLS-FDR approximations would be the use of weights for gene-selection purpose. It was previously noted that the gene-selection method that leads to the largest percentage of truly differentially expressed genes does not necessarily lead to the lowest misclassification rate. Hence, we can compare our weights with other methods such as Wilcoxon, prediction analysis of microarray, or

extreme-value distribution based gene selection. Additionally, we can explore relations between various univariate and multivariate gene-selection methods.

Another application we would like to consider is solving clustering problems. We previously looked at the application of weights to the problem of gene clustering and observed some promising results. This topic can be explored further and the results can be compared with those of supervised clustering (Dettling 2005) and other methods.

We can also look at the Sliced Inverse Regression (SIR) idea proposed by Duan and Li (1991). The SIR is a tool for reducing the dimension that does not require parametric or non-parametric model fitting. This idea deserves further research.

In the next part of the dissertation we focused on the hypothesis testing for microarray data. We concentrated on the experiments where one wants to find a 'bump' in group measurements. We applied three approaches – Dunnett's test, permutation-based and distribution-based – to the analysis of gene-expression and weight data. We concluded that the use of derived distribution of the test statistic has the advantage over Dunnett's and permutation-based tests. It does not alter the test statistic like Dunnett's method, and has no limitations for the small sample size as in the permutation-based approach. We may also consider the performance of the three tests in the carefully simulated datasets to see if the selected genes are truly those that were simulated as differentially expressed. Additionally, we can look at the distribution of the test statistic in more general settings of unequal variances and group sample sizes. These are topics for additional investigation.

The last part of the dissertation was devoted to the PfarMineR package. The package can also be expanded to include novel methods as they emerge in the statistical society. As flexible as this tool is, there is also room for simplifications and improvements of the interface.

We would like to conclude this dissertation with the goal of functional genomics and microarray technology which is to understand the relationship between an organism's genome and its phenotype. The analysis of gene expression data is a step toward the fulfillment of this goal. The methods described in this dissertation contribute to the set of methodologies for the analysis of gene expression data.

APPENDIX A

Overview of R Packages That Implement Various PLS Approaches

Package	Description
plsgenomics	This package implements PLS regression (using the function simpls from the pls.pcr package) with user-friendly features such as the choice of the number of components. It also implements the classification method PLS+LDA (discussed by Nguyen and Rocke, and Boulesteix) as well as the ridge PLS method.
pls.pcr	This package implements the two main variants of multivariate PLS regression SIMPLS and PLS2 as well as PCR.
pls	This package is an extension of the earlier package pls.pcr including, e.g. various plot functions and a formula interface.
gpls	This package implements the classification method using generalized PLS (see Ding B. and Gentleman R. Classification using generalized partial least squares. 2005.).
plss	These programs implement PLS regression based on splines transformations of the predictors.

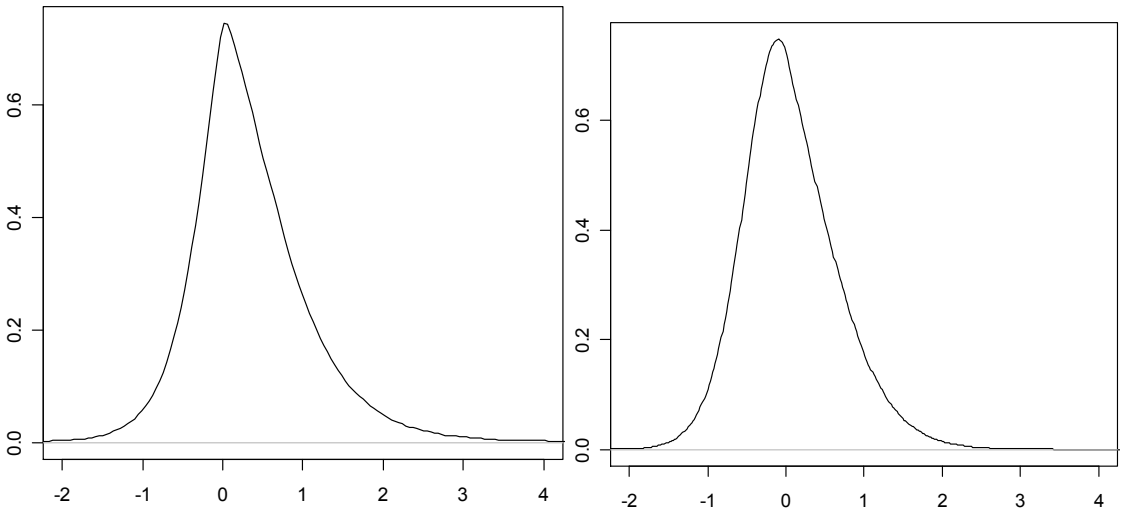
APPENDIX B

Differentially expressed genes identified by each procedure

Approach	Dunnet	Permutation	Approximated Distribution
Genes	A20 A26 A48 A64 A69 A101 A118 A138 A152 A167 A168 A171 A178 A193 A209 A218 A222 A228 A240 A275 A282 A317 A322 A364 A371 A376 A378 A393 A394 A407 A415 A421 A426 A430 A444 A448 A449 A453 A461 A462 A463 A479 A490 A512 A514 A516 A549 A554 A558 A567 A591 A594 A606 A609 A626 A631 A633 A638 A639 A644 A657 A660 A662 A664 A675 A690 A702 A711 A716 A746 A757 A825 A834 A851 A855 A862 A875 A892 A898 A910 A923 A951 A970 A975 A985 A1001 A1011 A1036 A1040 A1071 A1093 A1115 A1133 A1142 A1154 A1159 A1164 A1170 A1187 A1189 A1193 A1210 A1223 A1241 A1254 A1257 A1259 A1279 A1297 A1307 A1323 A1325 A1327 A1345 A1370 A1374 A1383 A1389 A1412 A1413 A1458 A1464 A1474 A1477 A1479 A1488 A1504 A1506 A1507 A1512 A1513 A1536 A1537 A1542 A1560 A1566 A1576 A1584 A1595 A1596 A1598 A1601 A1627 A1644 A1659 A1666 A1683 A1700 A1711 A1720 A1729 A1747 A1750 A1755 A1776 A1792 A1802 A1822 A1830 A1834 A1857 A1874 A1886 A1900 A1905 A1928 A1945 A1952 A1978 A1987 A2004 A2017 A2023 A2033 A2051 A2059 A2061 A2062 A2067 A2085 A2113 A2122 A2139 A2144 A2160 A2178 A2185 A2190 A2200 A2220 A2248 A2251 A2254 A2267 A2303 A2316 A2334 A2352 A2353	A20 A26 A48 A58 A64 A79 A83 A138 A149 A192 A205 A206 A218 A222 A228 A230 A235 A273 A322 A363 A371 A376 A387 A407 A415 A421 A442 A443 A448 A449 A453 A461 A477 A490 A498 A510 A512 A516 A521 A522 A538 A558 A594 A601 A618 A632 A643 A654 A664 A665 A690 A716 A769 A773 A782 A825 A834 A850 A875 A886 A887 A910 A937 A939 A951 A971 A1011 A1017 A1019 A1033 A1064 A1071 A1077 A1133 A1142 A1154 A1159 A1164 A1170 A1180 A1187 A1197 A1213 A1241 A1242 A1254 A1307 A1323 A1327 A1345 A1382 A1383 A1422 A1425 A1458 A1463 A1477 A1504 A1506 A1507 A1512 A1513 A1536 A1539 A1542 A1568 A1569 A1595 A1596 A1598 A1601 A1602 A1626 A1627 A1644 A1724 A1729 A1747 A1750 A1755 A1776 A1792 A1822 A1830 A1834 A1882 A1928 A1937 A1940 A1966 A2002 A2004 A2017 A2061 A2067 A2104 A2110 A2113 A2142 A2190 A2192 A2199 A2303 A2316 A2319 A2322 A2352 A2353	A20 A26 A376 A558 A632 A664 A690 A875 A951 A1011 A1254 A1323 A1327 A1383 A1477 A1507 A1513 A1536 A1596 A1598 A1627 A1747 A1750 A1830 A2061 A211
Intersection	A20 A26 A376 A558 A664 A690 A875 A951 A1011 A1254 A1323 A1327 A1383 A1477 A1507 A1513 A1536 A1596 A1598 A1627 A1747 A1750 A1830 A2061 A2113		

APPENDIX C

Approximate plots of the T statistic distribution



Plots of a T-statistic density for k=2 (left) and k=3 (right)

APPENDIX 1

Package Tree and Layout

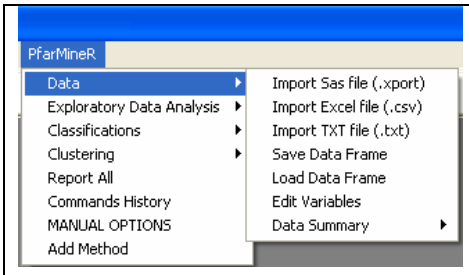
DATA MENU		
Menu Name	*Function*	*Input/Options*
Import Sas file (.xport)	browse	File name/location Dataset name
Import Excel file (.csv)	browse2	
Import TXT file (.txt)	browse3	
Output Description and Notes: Reads the dataset, identifies variables automatically as numeric or categorical, response (those including "RSP" or "RESP" in their name) or independent, and whether they are included in the analysis. Also creates additional menu with variable names and characteristics, so that they can be changed. Dataset name provided is for user's use for later. Initially all variables are included in the analysis, and only those are considered categorical that are text variables. Output is dataset summary and number of missing values.		
Save Data Frame	frames	File Name
Output Description and Notes: Saves the current dataset and its attributes (variable types, etc.) in the current folder as '.txt' file with the name provided by user. File gets created in the current directory. Dataset can later be loaded via the 'Load' menu button.		
Load Data Frame	framei	File Name/Location
Output Description and Notes: Loads dataset from the file specified, prints dataset summary and number of missing values.		
Edit Variables	edtvars	Current Dataset
Output Description and Notes: Creates or updates "Edit Variables" menu based on the current dataset – menu where you can choose if the variable is numeric or categorical, response or independent, included or not in the analysis.		
Data Summary	prdatasumm	Current Dataset Output type to PDF or Window
Output Description and Notes: Prints dataset summary - variable name, type, whether included in analysis, whether response or independent; and number of missing values.		
EXPLORATORY DATA ANALYSIS MENU		
Menu Name	*Function*	*Input/Options*
Data Visualization	visual	Current Dataset Output type to PDF or Window
Response Visualization	rvisual	
Output Description and Notes: Visualization of variables against each other and/or response(s) – done as scatterplot for two continuous variables, boxplot for continuous and categorical, and barplot for two categorical ones. To look at the distribution of a single variable, there is either histogram or barplot for cases of continuous or categorical variables respectively. The output is broken into pages with the largest dimension of a page being 5x5 and pages are arranged from left to right.		
Basic EDA Stats	sumstat sumstatby	Current Dataset Categorical Variable Name (for summary by this variable) Automatic or By Variable Output type to PDF or Window
Output Description and Notes: Basic summary for categorical variable – levels, number of times each level happens, percentage of total number. For continuous variables – mean, standard deviation, minimum, first quartile, median, 3rd quartile. There is also an option to calculate those statistics for each level of chosen categorical variable.		
Data Transformations	transf	Current Dataset Variable Name Type of Transformation Required constant(s) Output type to PDF or Window
Output Description and Notes: Transformations available for continuous variables: add a constant (x+c),		

raise to a power (x^c), take a root ($x^{1/c}$), inverse (1/x), take a logarithm (log(x)), exponentiate (e^x), automatic ($(x-a)^b$ where a and b are constants determined automatically to make the QQ plot straight line), standardization ((x-mean)/sd), center at zero (x-mean), restore original. Outputted then are the QQ plots - original and after transformation – and the histogram with curve over. For automatic transformation constants a and b are also outputted. For categorical variables user can recode levels and restore original. Two levels can be merged if recoded to the same level. Then new levels of a variable are outputted.		
Robust Regression	robregr (via lm, rlm, and glm)	Current Dataset Output type to PDF or Window
Output Description and Notes: For continuous response there is summary of the linear regression, ANOVA, summary of the robust regression, residual boxplot, QQ plots for least-squares and robust residuals, least-squares and robust residuals vs. fitted that are shown. For categorical - summary of the logistic regression, analysis of deviance table, response barplot, model success, ROC for prediction.		
LASSO	pfarlas (via lars, gl1ce)	Current Dataset Output type to PDF or Window
Output Description and Notes: LASSO runs with all the variables and then the summary is printed along with the optimum parameter value and non-zero coefficients. Plotted are cross-validated error curve and plot of lasso fit. For the categorical response there is Generalized Regression With L1-constraint on the Parameters performed. Output consists of lasso object, plot for residuals vs. fitted values, and link vs. response predicted values.		
Simulation Methods and Bootstrap	simbootstr (via boot)	Current Dataset Output type to PDF or Window
Output Description and Notes: Performs a nonparametric bootstrap for continuous response predictors. Output is least-squares estimates, bias, standard error, percentile confidence interval (bootstrap percentile method), BCA confidence interval (adjusted bootstrap percentile method)		
CLASSIFICATIONS MENU		
Menu Name	*Function*	*Input/Options*
ARF	myarf (via arf)	Current Dataset Output type to PDF or Window
Output Description and Notes: Output consists of the report files (for PDF "arfreport.pdf") for each response.		
CART	cart (via rpart)	Current Dataset Manual (then require minimum bucket size, minimum split, complexity parameter) or automatic Output type to PDF or Window
Output Description and Notes: Rpart object is plotted and its summary is displayed.		
Neural networks	neunet (via nnet)	Current Dataset Output type to PDF or Window
Output Description and Notes: Neural Network summary is showed for each response.		
SVMs	svms (via svm)	Current Dataset Output type to PDF or Window
Output Description and Notes: Response can be either a factor (for classification tasks) or a numeric vector (for regression). For the continuous response output is the model summary. For the categorical one - model summary, SVM object plots for each combination of two predictors on a separate page (crosses indicate support vectors, the colors represent the classes of the data points).		
Naive Bayes	naibay (via naiveBayes)	Current Dataset Output type to PDF or Window
Output Description and Notes: After performing the Naive Bayes Classifier for Discrete Predictors shows the model information and the table of predicted versus original values		
Random Forests	ranfor (via randomForest)	Current Dataset Output type to PDF or Window
Output Description and Notes: Performs Breiman's random forest algorithm for classification and regression and output consists of the model information, the plot of the error rates or MSE of a random		

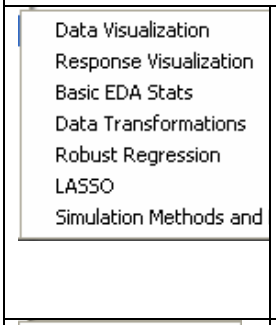
forest object, and the dot-chart of variable importance as measured by a random forest.			
Boosting	mybst (via gbm)	Current Dataset Output type to PDF or Window	
Output Description and Notes: Performs Generalized Boosted Regression Modeling. Output consists of the optimal number of boosting iterations for an object and the summary based on the estimated best number of trees. Also there are two plots - the first one shows performance measures. The second one is the relative influence barplot.			
CLUSTERING MENU			
Menu Name	*Function*	*Input/Options*	
Hierarchical	clusth (via hclust)	Current Dataset Output type to PDF or Window	
Output Description and Notes: Cluster Dendrogram			
PAM	clustpam (via pam)	Current Dataset Output type to PDF or Window	
Output Description and Notes: Outputs the optimal number of clusters based on average silhouette width, the plot of number of clusters versus average silhouette width, the PAM object for the best number of iterations, and the silhouette plot for the above object.			
Silhouette Plots	clustsilh (via silhouette)	Current Dataset Number of clusters Output type to PDF or Window	
Output Description and Notes: Showed are summary of the silhouette clustering object and the silhouette plot for this object.			
K-means	clustkmean (via kmeans)	Current Dataset Number of clusters Output type to PDF or Window	
Output Description and Notes: The output consists of the k-means object summary and plots of variables against each other showing clusters in different colors.			
Model-based	modbase (via Mclust)	Current Dataset Output type to PDF or Window	
Output Description and Notes: The optimal model according to BIC for EM initialized by hierarchical clustering for parameterized Gaussian mixture models produces the following model-based clustering plots: BIC values used for choosing the number of clusters, pairs plot showing the classification, projections of the data showing location of the mixture components, classification, and uncertainty			
Menu Name	*Function*	*Input*	*Output type*
Report All	reportall	Current Dataset	PDF file 'REPORTALL .pdf' in the current directory containing the output of all methods
Commands History	history		Window pops up containing the commands used in the current session
Add Method	addmenubutn	Menu Name Function Name Source Code File	New menu button with the function assigned as specified by the user

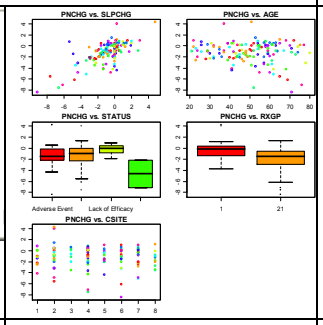
APPENDIX 2

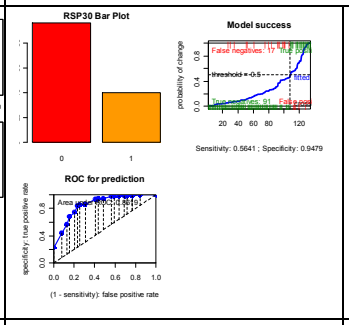
Examples of Output

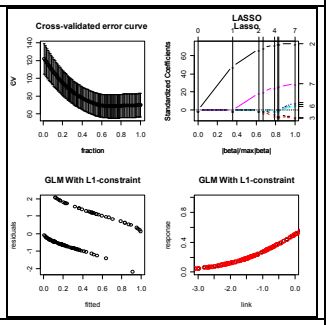


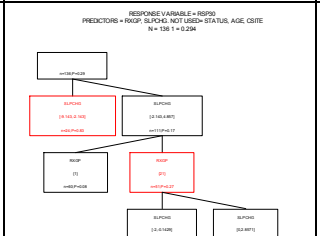

```
          Variable Type Included Response
PAIN0     Numeric        No       Independent
SLEEP0    Numeric        No       Independent
PAIN      Numeric        No       Independent
SLEEP     Numeric        No       Independent
PNCHG     Numeric        Yes      Response
SLPCHG    Numeric        Yes      Independent
PNPCH     Numeric        No       Independent
SLPPCH    Numeric        No       Independent
RSP30     Categorical    Yes      Response
RSP50     Numeric        No       Independent
AGE       Numeric        Yes      Independent
STATUS    Categorical    Yes      Independent
RXGP      Categorical    Yes      Independent
SITE      Numeric        No       Independent
PTID      Numeric        No       Independent
LASTDAY   Numeric        No       Independent
DAY1      Numeric        No       Independent
LASTDYDB  Numeric        No       Independent
CSITE     Numeric        Yes      Independent

WARNING!
 14  MISSING VALUES
```

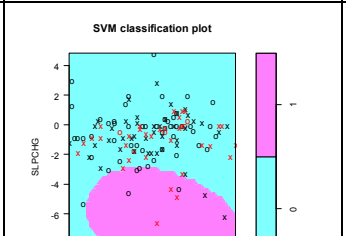


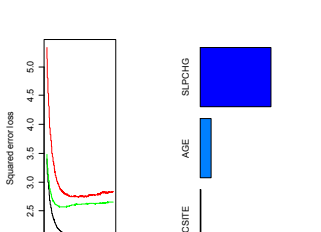


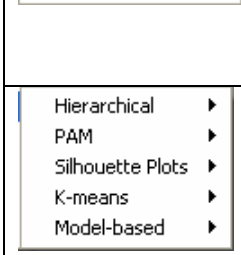


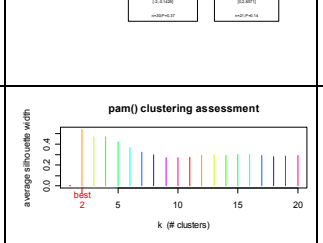


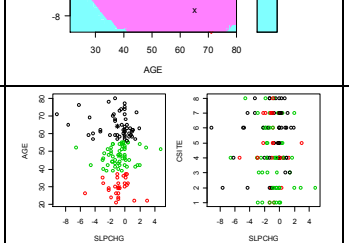


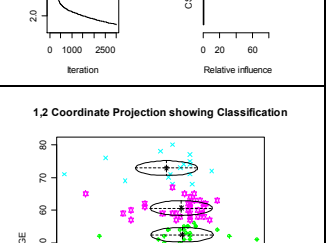












APPENDIX 3

Package PfarMineR

The package can be downloaded at:

http://stat.rutgers.edu/~ycherkas/Codes/PfarMineR_1.0.zip

REFERENCES

[1] Affymetrix (1999) Affymetrix Microarray Suite User Guide, Version 4 edn. Affymetrix Santa Clara, CA.

[2] Affymetrix (2004) GeneChip Expression Analysis Technical Manual, Rev.4. Affymetrix Santa Clara, CA.

[3] Allison, D. B., Cui, X., Page, G. P., Sabripour, M. (2006) Microarray data analysis: from disarray to consolidation and consensus. Nat Rev Genet. Jan;7 (1):55-65.

[4] Allison, D. B., Cui, X., Page, G. P., Sabripour, M. (2006) Microarray data analysis: from disarray to consolidation and consensus. Nat Rev Genet., 7, 55–65.

[5] Amaratunga, D., Cabrera, J. (2001) Statistical analysis of viral microchip data. Journal of the American Statistical Association, 96 1161–1170.

[6] Amaratunga, D., Cabrera, J. (2004) Exploration and Analysis of DNA Microarray and Protein Array Data. Wiley, New York.

[7] Amaratunga, D., Cabrera, J. (2006) Differential expression in DNA microarray and protein array experiment. Technical Report 06-001, Department of Statistics, Rutgers University.

[9] Amaratunga, D., Cabrera, J., Kovtun, V. (2008) Microarray learning with abc. Biostatistics, 9, 128–136.

[10] Arnold, B. C., Balakrishnan, N. (1989) Relations, bounds and approximations for order statistics. Springer-Verlag, New York.

[11] Bechhofer, R. E., Dunnett, C. W. (1988) Percentage points of multivariate Student t distributions. In: Selected Tables in Mathematical Statistics, vol. 11. American Mathematical Society, Providence, RI.

[12] Benjamini, Y., Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of Royal Statistical Society, Biostatistics, 57, 289–300.

[13] Benjamini, Y., Yekutieli, D. (2001) The control of the false discovery rate in multiple testing under dependency. Annal of Statistics, 29 (4), 1165–1188.

[14] Benjamini, Y., Yekutieli, D. (2005a) False Discovery Rate-Adjusted Multiple Confidence Intervals for Selected Parameters. Journal of the American Statistical Association, 100, 71–81.

[15] Boes, T., Neuhäuser, M. (2005) Normalization for Affymetrix GeneChips. Methods Inf Med. ;44 (3):414-7.

[16] Bolstad, B. M., Irizarry, R. A., Astrand, M., Speed, T. P. (2002) A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. Bioinformatics, 19, 185–193.

[17] Boulesteix, A. L., Strimmer, K. (2007) Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. BMC Bioinformatics, 8 (1), 32–44.

[18] Boulesteix, A. L., Tutz, G., Strimmer, K. (2003) A CART-based approach to discover emerging patterns in microarray data. Bioinformatics, 19, 2465–2472.

[19] Breiman, L. (1996) Bagging predictors. Machine Learning, 24, 123–140.

[20] Breiman, L. (2001) Random forests. Machine Learning, 45, 5–32.

[21] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984) Classification and Regression Trees, Chapman & Hall, New York.

[22] Broët, P., Richardson, S., Radvanyi, F. (2002) Bayesian Hierarchical Model for Identifying Changes in Gene Expression from Microarray Experiments. J Comput Biol.;9 (4):671-83.

[23] Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Furey, T.S., Jr., M.A., Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl Acad. Sci., 97, 262–267.

[24] Causton, H. C., Quackenbush, J., Brazma, A. (2003) Microarray gene expressions data analysis: a beginner's guide. Wiley-Blackwell.

[25] Chen, D. T. (2004) A graphical approach for quality control of oligonucleotide array data. J. Biopharm. Stat., 14, 591–606.

[26] Chen, Y., Dougherty, E. R., Bittner, M. L. (1997) Ratio-based decisions and the quantitative analysis of cDNA microarray images. J. Biomed. Opt., 2, 364–374.

[27] Cheung, S. H., Holland, B. (1992) Extension of Dunnett's multiple comparison procedure with differing sample sizes to the case of several groups. Computational Statistics & Data Analysis, Vol. 14, Issue 2, August, 165-182.

[28] Cortes, C., Vapnik, V. (1995) Support-Vector Networks. Machine Learning, 20, 273–297.

[29] Crick, F. (1970) Central Dogma of Molecular Biology. Nature 227, 561-563.

[30] David, H. A., Nagaraja, H. N. (2003) Order statistics. 3rd ed. Wiley.

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

[32] Dettling, M., Buhlmann, P. B. (2003). Boosting for tumor classification with gene expression data. Bioinformatics 19 (9), 1061–1069.

[33] Diaz-Uriarte, R., Alvarez de Andres, S. (2006) Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7:3.

[34] Ding, B., Robert Gentleman, R. (2005) Classification Using Generalized Partial Least Squares. Journal of Computational and Graphical Statistics. Jun, Vol. 14, No. 2: 280-298

[35] Dudoit, S., Fridlyand, J., Speed, T. P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association, 98, 77–87.

[36] Dudoit, S., Yang, Y. (2002) Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. Garret, R. Irizarry and S. Zeger, editors, The Analysis of Gene Expression Data : Methods and Software. Springer, New York.

[37] Dudoit, S., Yang, Y. H., Callow, M. J, Speed, T. P. (2000) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical Report, #578.

[38] Dudoit, S., Yang, Y., Bolstad, B. (2002) Using R for the analysis of DNA microarray data. R News 2 (1), 24–32.

[39] Dudoit, S., Yang, Y., Speed, T., Callow, M. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Statistica Sinica 12 ,111-140.

[40] Dunnett, C. W. (1955) A multiple comparison procedure for comparing several treatments with a control. JASA, 50, 1096–1121.

[41] Dunnett, C. W. (1964) New tables for multiple comparisons with a control. Biometrics, 20, 482–491.

[42] Dunnett, C. W. (1980) Pairwise Multiple Comparisons in the Unequal Variance Case. Journal of the American Statistical Association. 75 (372): 796-800.

[43] Dunnett, C. W., Sobel, M. (1955) Approximations to the probability integral and certain percentage points of a multivariate analogue of Student's t-distribution. Biometrika 42, 258-260.

[44] Dunnett, C. W., Sobel, M. (1954) A Bivariate Generalization of Student's t-Distribution, with Tables for Certain Special Cases. Biometrika, Vol. 41, No. 1/2 (Jun.), 153-169.

[45] Efron, B., Hastie, T., Johnstone, I., Tibshirani,R. (2004) Least angle regression. Ann. Stat., 32, 407–499.

[46] Efron, B., Tibshirani, R. (2002) Empirical Bayes Methods and False Discovery Rates for Microarrays, Genetic Epidemiology, 23, 70–86.

[47] Efron, B., Tibshirani, R., Storey, J. D., Tusher, V. (2001) Empirical Bayes Analysis of a Microarray Experiment,Journal of the American Statistical Association, 96, 1151–1160.

[48] Eickhoff, B., Korn, B., Schick, M., Poustka, A., van der Bosch, J. (1999) Normalization of array hybridization experiments in differential gene expression analysis. Nucleic Acids Res., 27, 33.

[49] Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. USA, 95, 14863–14868.

[50] Frank, I. E., Friedman, J. H. (1993) A statistical view of chemometrics regression tools. Technometrics, 35 109–148.

[51] Freund, Y., Schapire, R. E. (1999) A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, 14, 771–780.

[52] Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16, 906–914.

[53] Gautier, L., Cope, L., Bolstad, B. M., Irizarry, R. A. (2004) affy--analysis of Affymetrix GeneChip data at the probe level. Bioinformatics. Feb 12;20 (3):307-15.

[54] Ge, Y., Dudoit, S., Speed, P. T. (2003) Resampling based multiple testing for microarray data analysis, technical report, 633, University of Berkeley.

[55] Genovese, C., Wasserman, L. (2001) Operating characteristics of the FDR procedure. Technical Report. New York, Carnegie Mellon University 2001.

[56] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., Lander, E. S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 286, 531–537.

[57] Guo, Y., Hastie, T., Tibshirani, R. (2007) Regularized linear discriminant analysis and its application in microarrays. Biostatistics, 8 (1), 86–100.

[58] Haldermans, P., Shkedy, Z., Van Sanden, S., Burzykowski, T., Aerts, M. (2007) Using linear mixed models for normalization of cDNA microarrays. Statistical Application in Genetics and Molecular Biology, 6 (1), 19.

[59] Hastie, T., Tibshirani, R., Friedman, J. (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York.

[60] Helland, I. S. (1990) Pls regression and statistical models. Scandivian Journal of Statistics, 17, 97–114.

[61] Hochberg, Y. (1995) A sharper Bonferroni procedure for multiple tests of significance, Biometrika, 75, 800–802.

[62] Hochberg, Y., Tamhane, Y.C. (1987) Multiple Comparison Procedures. New York: Wiley.

[63] Holm, S. (1979) A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics, 6, 65–70.

[64] Hoskuldson, A. (1988) Pls regression methods. Journal of Chemometrics, 2, 211-228.

[65] Ideker, T., Thorsson, V., Siegel, S., Hood, L. (2000) Testing for differentially expressed genes by maximum-likelihood analysis of microarray data. Journal of Computational Biology 7 (6) 805–817.

[66] Imada, T., Douke, H. (2007) Step down procedure for comparing several treatments with a control based on multivariate normal response. Biom J. Feb; 49 (1): 18-29.

[67] Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., Speed, T. P. (2003) Summaries of Affymetrix GeneChip probe level data. Nucleic Acids Res., 31, e15–e15 (1).

[68] Irizarry, R. A., Gautier, L., Cope, L. (2003) The Analysis of Gene Expression Data. Springer.

[69] Irizarry, R. B., Hobbs, F. C., Beaxer-Barclay, Y., Antonellis, K., Scherf, U., Speed, T. (2003a) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4, 249–264.

[70] Kerr, K. M., Martin, M., Churchill, G. A. (2000) Analysis of variance for gene expression microarray data. Journal of Computational Biology, 7, 819–838.

[71] Kerr, M. K., Churchill, G. A. (2001) Experimental design for gene expression microarrays. Biostatistics, 2, 183-201.

[72] Kerr, M. K., Churchill, G. A. (2001) Statistical analysis of a gene expression microarray experiment with replication. Biostatistics, 2, 183–201.

[73] Kerr, M., Churchill, G. (2001) Experimental design for gene expression microarrays. Biostatistics, 2 183–201.

[74] Kimball, A. W. (1951) On dependent tests of significance in the analysis of variance. Ann. Math. Stat. 22, 600-602.

[75] Lee, J. W., Lee, J. B., Park, M., Song, S. H. (2005) Extensive comparison of recent classification tools applied to microarray data. Computational Statistics and Data Analysis, 48, 869–885.

[76] Lee, N. H., Saeed, A. I. (2007) Microarrays: an overview. Methods Mol Biol., 353, 265-300.

[77] Lehmann, E. L., Romano, J. P. (2005) Generalizations of the familywise error rate. Ann. Statist., 33 (3), 1138–1154.

[78] Leung, Y. F., Cavalieri, D. (2003) Fundamentals of cDNA microarray data analysis. Trends Genet. Nov 19 (11), 649-59.

[79] Lin, J. (1992) Approximating the Student's t-Tail Probability and Its Inverse. Probability in the Engineering and Informational Sciences, 6, 133-137.

[80] Lonnstedt, I., Speed, T. P. (2002) Replicated microarray data. Statistica Sinica,12 31–46.

[81] Ma, S., Song, X., Huang, X. (2007) Supervised group Lasso with applications to microarray data analysis. BMC Bioinformaics, 6, 60.

[82] Mah, N., Thelin, A., Lu, T., Nikolaus, S., Kühbacher, T., Gurbuz, Y., Eickhoff, H., Klöppel, G., Lehrach, H., Mellgård, B., Costello, C. M., Schreiber, S. (2004) A comparison of oligonucleotide and cDNA-based microarray systems. Physiol Genomics. Feb 13;16 (3):361-70.

[83] Mann, H. B., Whitney, D. R. (1947) On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18, 50-60.

[84] Nadarajah, S., Kotz, S. (2008) Exact Distribution of the Max/Min of Two Gaussian Random Variables. IEEE Transactions On VLSI Systems, Vol. 16, No. 2, Feb.

[85] Nguyen, D. V., Rocke, D. M. (2004) On partial least squares dimension reduction for microarray-based classification: a simulation study. Computational Statistics & Data Analysis, 46, 407–425.

[86] Nguyen, D. V., Rocke, D. M. (2002) Tumor classification by partial least squares using microarray gene expression data. Bioinformatics. Jan; 18 (1): 39-50.

[87] Nguyen, D. V., Rocke, D. M., David, M. (2004) On partial least squares dimension reduction for microarray-based classification: a simulation study. Computational Statistics & Data Analysis, 46, issue 3, p. 407-425.

[88] Pan, W. (2003) On the use of permutation in and the performance of a class of nonparametric methods to detect differential gene expression. Bioinformatics, 19, 1333–1340.

[89] Patel, K., Read, C. B. (1996) Handbook of the normal distribution. CRC Press.

[90] Pillai, K. C. S., Ramachandran, K. V. (1954) On the distribution of the ratio of the i-oth observation in an ordered sample from a normal population to an independent estimate of the standard deviation. Ann. Math. Stat. 25, 565-572.

[91] Quackenbush, J. (2001) Computational analysis of microarray analysis. Nature Reviews Genetics, 2, 418–427.

[92] Rproject CRAN. http://http://www.r-project.org

[93] Ränner, S., Geladi, P., Lindgren, F., Wold, S. (1995) A PLS Kernel Algorithm for data sets with many variables and few objects. Part II : cross-validation, missing data and examples. Journal of Chemometrics, Vol 9, 459-470.

[94] Rao, C. R., Balakrishnan, N. (1998) Order statistics: applications. Elsevier.

[95] Reiner, A., Yekutieli, D., Benjamini, Y. (2003) Identifying differentially expressed genes using false discovery rate controlling procedures. Bioinformatics, 19 (3), 368–375.

[96] Ripley, B. D. (1996) Pattern Recognition and Neural Networks. Cambridge.

[97] Ritchie, M. E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., Smyth, G. K. (2007) A comparison of background correction methods for two-colour microarrays. Bioinformatics, Oct.,23 (20): 2700–2707.

[98] Rosipal, R., Krämer, N. (2006) Overview and recent advances in partial least squares. Subspace, Latent Structure and Feature Selection, 34-51.

[99] Sanchez, G. (2009) Understanding Partial Least Squares Path Modeling. Academic Paper, March.

[100] Schena, M. (1999) DNA Microarrays: A Practical Approach, Oxford University Press.

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

[102] Schulze, A., Downward, J. (2001) Navigating gene expression using microarrays - a technology review . Nat Cell Biol. Aug;3 (8):E190-5. Review.

[103] Schwender, H., Belousov, A. (2006) Comparison of preprocessing methods for affymetrix microarrays. A Magazine of the American Statistical Association, 19 (3):16, Summer..

[104] Sengupta, J. M., Bhattacharya, N. (1958) Tables of random normal deviates. Sankhya 20, 249-286.

[105] Simon, R. M., Dobbin, K. (2003) Experimental design of DNA microarray experiments.Simon RM et al. Biotechniques.

[106] Smith, L., Greenfield, A. (2003) DNA microarrays and development. Hum Mol Genet. Apr 1;12 Spec No 1:R1-8.

[107] Smyth, G. K., Speed, T. P. (2003) Normalization of cDNA microarray data. Methods 31, 265-273. [PubMed ID 14597310]

[108] Speed, T., Yang, Y. (2002) Direct versus indirect design for cDNA microarray experiments. Technical report 616, Department of Statistics, University of California, Berkeley.

[109] Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., Levy, S. (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics, 21, 631–643.

[110] Steel, R. G. D. (1959) A multiple comparison rank sum test: treatments versus control. Biometrics 15, 560-572.

[111] Storey, J. D. (2001) A direct approach to false discovery rates. Technical Report. Stanford, CA: Stanford University.

[112] Storey, J. D., Tibshirani, R. (2003) Statistical significance for genome-wide studies, Proceedings of the National Academy of Sciences, 100, 9440–9445.

[113] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. Journal of the Royal statistical society, series B, 58.

[114] Tibshirani, R. J., Efron, B. (2002) Pre-validation and inference in microarrays. Statistical Applications in Genetics and Molecular Biology: Vol. 1 : Iss. 1, Article 1.

[115] Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D., Brown, P. (1999) Clustering methods for the analysis of DNA microarray data. Oct.

[116] Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS, 99, 6567–6572.

[117] Tong, Y. L. (1990) The Multivariate Normal Distribution. Springer, New York.

[118] Tukey, J. W. (1953) The problem of multiple comparisons. Unpublished manuscript. In The Collected Works of John W. Tukey VIII. Multiple Comparisons, 1948- 1983, 1-300. Chapman and Hall, New York.

[119] Tusher, V. G., Tibshirani, R., Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response, Proceedings of the National Academy of Sciences, 98 , 5116–5121.

[120] Vapnik, N. V. (2000) The nature of Statistical Learning Theory, (2nd ed.), Springer, New York.

[121] Venables, W. N., Ripley, B. D. (2003) Modern Applied Statistics with S. Springer, New York.

[122] Westfall, P. H., Young, S. S. (1993) Resampling based multiple testing, Willy.

[123] Yang, T., Buckley, M., Dudoit, S., Speed, T. (2002) Comparison of methods for image analysis on cDNA microarrays. Journal of Computational and Graphical Statistics 11, 108–136.

[124] Yang, Y. H., Buckley, M. J., Speed, T. P. (2001) Analysis of cDNA microarray images. Brief Bioinform., 2, 341–349.

[125] Yang, Y. H., Dudoit, S., Luu, P., Speed, T. P. (2001) Normalization for cDNA microarray data. In Microarrays: Optical Technologies and Informatics, M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), Proceedings of SPIE, 4266, 141–152.

[126] Yekutieli, D., Benjamini, Y. (1999) Resampling-Based False Discovery Rate Controlling Multiple Test Procedures for Correlated Test Statistics, Journal of Statistical Planning and Inference, 82, 171–196.

[127] Zou, H., Hastie, T. (2005) Regularization and variable selection via the elastic net. J. R. Statist. Soc. B 67, Part 2, 301–320.

Vita

Yauheniya Cherkas

2001 BS in Applied Mathematics and Computer Science,

State University of Belarus, Belarus

2003 BS in Mathematics, Temple University

2005 MA in Mathematics, Temple University

2010 PhD in Statistics, Rutgers University, the State University of New Jersey