Contributions to Statistical Techniques for the Analysis of Gene and Protein Expression Data

Doctoral Thesis

Submitted to

the Department of Statistics

of the University of Dortmund

in Fulfilment of

the Requirements for the Degree of

'Doktor der Naturwissenschaften'

by

Klaus Jung

from Wiesbaden

Dortmund, June 2006

Supervisor: Prof. Dr. Wolfgang Urfer

2nd Referee: Prof. Dr. Katja Ickstadt

Date of the oral examination: 3 August 2006

Acknowledgements

I would like to thank very much my supervisor Prof. Dr. Wolfgang Urfer for his guidance throughout my research.

My thanks go also to Dr. Karsten Quast and Dr. Guntram Deichsel (Boehringer Ingelheim Pharma GmbH & Co. KG, Biberach, Germany) who provided the microarray gene expression data for this research.

Furthermore, I would like to thank Barbara Sitek, Dr. Kai Stühler and Prof. Dr. Helmut E. Meyer (Medical Proteome-Center, Ruhr-University of Bochum, Germany) for their close collaboration and for providing their protein expression data.

Special thanks go to Prof. Dr. Ali Gannoun (Centre d'Étude et de Recherche en Informatique du CNAM, Paris, France),

as well as to Dr. Nick Fieller and Clare Foyle (Department of Probability and Statistics, University of Sheffield, United Kingdom) for interesting discussions and their help.

I would also like to thank Dr. Jens Decker and Frank-Michael Schleif (Bruker Daltonik GmbH, Bremen, Germany) for their collaboration and for providing their protein expression data.

I am most grateful to my colleagues Marco Grzegorczyk and Nina Kirschbaum as well as to Brigitte Koths and Eva Brune (Department of Statistics, University of Dortmund, Germany) for their help.

In addition, my thanks go to the 'Deutsche Forschungsgemeinschaft e.V.' (Bonn, Germany) for providing a scolarship for my research.

Finally, I would like to thank very much my parents for their encouragement and support.

Dortmund, June 2006 Klaus Jung

Contents

'...our ability to generate data
now outstrips our ability to
analyze it.'
Scott D. Patterson (2003)

1 Introduction

Since the middle of the 1990's two new fields of biochemical research brought along great need of methods for statistical analysis: 'genomics' and 'proteomics'. These two field of research comprise the exploration of the 'genome' and the 'proteome', that is the totality of genes and proteins of an organism, respectively. In particular, a frequent subject of genome and proteome experiments is the exploration of the 'expression' levels of genes or proteins. Simplified, the expression of a biomolecule is its presence in certain cells or biological samples. Scott D. Patterson, one of the pioneers in the field of proteomics, named 'data analysis' to be the 'Achilles heel of proteomics' (cf. Patterson, 2003), and remarked that '... our ability to generate data now outstrips our ability to analyze it.' What he meant by this statement is not only the fact that there is a great need of statistical methods to analyse data from molecular biology experiments but also the fact that the great amount of data produced in proteomics has to be managed and validated. The same was pointed out by Tilstone (2003) for the field of genomics. In fact, so-called 'high throughput' technologies, that allow researchers to measure the expression levels of thousands of genes or proteins at the same time, have meanwhile been established in biochemical research, and statistics are quite important to make the correct inferences from these data. The difficulty of high-throughput experiments is, however, that thousands of variables are measured at only a few objects. Marcus and Meyer (2003) therefore point out the danger of doing proteome experiments with to few replications.

A lot of statistical methods for the analysis of data from genome experiments have

already been developed. However, there has little been done in the statistical analysis of data from proteome experiments. The aim of this work is therefore to transfer some known statistical methods for the analysis of data from genome experiments to similar situations in proteome experiments. In addition, new methods for the analysis of protein expression data are proposed.

To begin with it should be mentioned what the biological and medical interest behind genomics and proteomics is. The expression levels of genes and proteins are different in cells from different types of tissues, for example normal and cancerous. Furthermore, the expression levels of genes and proteins strongly depend on modifications of enzymes (proteins which catalyse chemical reactions) which play an important role in signal pathways (cf. Lodish et al., 2001). Signal pathways can be seen as the flows of information between cells. Inhibition or activation of certain components of a signal pathway influence the expression levels of proteins and thus influences biological processes like proliferation (division and growth of cells), differentiation (specialisation of cells) and cell death. An example is given by the MAPK (mitogen activated protein kinase) signal pathway, a sequence of signals that are initiated by enzymes. First, the enzyme MAPK is activated by another enzyme, the mitogen, then it moves to the cell nucleus where it regulates the expression of genes (cf. Morandell et al., 2005). Thus, it controls certain biological processes like proliferation and differentiation. Another example is the activation of the enzymes tyrosin kinases (Trk) by special molecules, leading itself to the activation of various other pathways like the Ras or MAPK pathways (cf. Sitek et al., 2005). Other influences on the expression of genes and proteins are also certain components of nutrition (cf. Schweigert et al., 2005).
Understanding the relations between the expression levels of genes or proteins in certain types of tissues and the components of signal pathways imbeds the hope to find starting points for new drugs against cancer or other diseases. It is therefore of major interest in genome and proteome studies to compare the expression levels

of genes (or proteins) in cells when certain pathways are activated or not, or when the cells stem from different types of tissues. Due to the fact that gene and protein expression also depends on time (e.g. hours after inhibition of a signal pathway), there is also need for time dependent measurements of expression levels.

Bioanalytical exploration of gene expression can be done by using the DNA microarrays technology, which allows to measure the expression levels of thousands of genes at the same time. One of the most frequent problems which are analysed by using this technology is the comparison of the gene expression in different types of tissues (e.g. normal and cancerous) or differently treated biological samples (e.g. treatment and control group). The subsequent statistical analysis for this problem is done by multiple hypothesis testing. In the context of DNA microarray data, this means testing for each gene whether it is differentially expressed or not. Various statistical approaches to this problem have been made. In this work, focus is set on a nonparametric method for density estimation within a multiple testing procedure for the detection of differentially expressed genes. This procedure was first introduced by Pan et al. (2001). Important improvements to this procedure were contributed by Zhao and Pan (2003) and Gannoun et al. (2004). Here, several further proposals for the improvement of this method are made. In particular, formulas for the calculation of p-values are given and an algorithm for faster implementation of the method is proposed. Additionally, some properties of this method, like its statistical power and its computing time, are evaluated. This evaluation is done in comparison with another method for the detection of differentially expressed genes, namely a permutation method. A new R-implementation of the nonparametric method and a brief introduction of how to use it is presented, too. Not in the focus of this work, but in the context of DNA microarrays important to name, is the big range of other statistical methods being applied to analyse microarray data. Prior to the actual analyses, data has always to be preprocessed by several steps like calibration and normalization. Respective methods have been introduced

and compared by Huber et al. (2002) and Bolstad et al. (2003). In order to see which genes have similar expression profiles over all replicates of a DNA microarray experiment one can group them by hierarchical clustering methods as proposed by Eisen et al. (1998) and Hastie et al. (2000). A very important goal of analysing DNA microarray data is the classification of new samples to known disease classes, e.g. tumor classes. This can be used in making secure diagnoses of the disease states of new patients. Nguyen and Rocke (2003) propose a classification method using partial least squares regression, Tibshirani et al. (2002) and Tibshirani et al. (2003) use a nearest shrunken centroids method to identify subsets of genes that best characterize certain classes. Also of great interest is to find dependencies and interactions between genes, i.e. to find gene networks. Friedman et al. (2000) and Grzegorczyk and Urfer (2004) use Bayesian networks to determine interacting genes. A review about genetic regulatory systems is given in de Jong (2002).

As mentioned above, the major aim of this work is to adapt statistical methods for the analysis of gene expression data to be applicable to protein expression data, too. In general, the structure of both data types is very similar. However, there emerge several specific problems in protein expression data, like outliers in repeated measurements or missing values. Knowing how to handle these problems, allows then the application of the manifold statistical methods for DNA microarray data to protein expression data. Protein expression data can be risen by divers bioanalytical technologies, depending on the biological question and the set of proteins one wants to analyse. Here, protein expression data from two-dimensional gel electrophoresis and from mass spectrometry are analysed.

Two-dimensional gel electrophoresis (2-DE) is one of the most common techniques that biochemists use for measuring the expression levels of proteins. A frequent goal of 2-DE experiments is to find differences in the expression profiles of proteomes in different tissues (cf. Knowles et al., 2003) or in tissues in which certain signal path-

ways were either activated or not (cf. Sitek et al., 2005). Hence, similar to DNA microarray experiments, tests are carried out for each single protein of the observed proteome, testing whether there is a differential expression or not. However, up to now, this is mostly done without adjustment for multiple hypothesis testing (cf. Zhan and Desiderio, 2003). In fact, multiple hypothesis adjustment seems to be a problem in such experiments, because the true number of observed proteins is not known. This is due to the fact that contaminations on the 2-D gels are taken as protein spots by common image analysis softwares. The measurements of these spots are also included in the data and thus are falsely included in the analyses, too. In this work, the problem of multiple hypothesis testing in 2-DE experiments is discussed and a proposition is made how to handle this problem.

In addition, in most 2-DE experiments, the single proteins are analysed with different sample sizes due to missing values in 2-DE data sets. Up to now, the existence of missing values in 2-DE data was neither mentioned in literature nor were there any proposals of how to handle such incomplete data sets. Missing values were also a problem in data from DNA microarray experiments, but the amount of missing values wasn't as big as in 2-DE experiments. Methods for the estimation of missing values in microarray data were for example proposed in Troyanskaya et al. (2001). Here, methods for the estimation of missing values in gene expression data are adapted to protein expression data. In particular, the estimation error for this methods is evaluated when having not only 5% of missing values (like in DNA microarray data), but 20-30%.

As mentioned above, protein expression often depends also on the time after activation of a certain associated signal pathway. Such situations are usually investigated by measuring protein expression at a small couple of subsequent times, yielding longitudinal data. In this work, a corrected analysis of variance model for longitudinal data, originally given in Diggle et al. (1994), is proposed for the analysis of time dependent protein expression data from 2-DE.

Besides 2-DE, another technology for measuring protein expression is mass spectrometry (MS). Often, the samples from each patient are measured repeatedly in a MS experiment, resulting in more than one mass spectrum per sample. However, up to now, these repeated measurements are treated as independent samples. Furthermore, outliers in these measurements can only be detected by visual judgement of MS practitioners involving the drawback of subjective decisions. In this work, a new approach to detect outliers by a standardised statistical method is presented as well as a proposal of how to summarise all mass spectra from the sample of the same patient.
Combined with a magnetic beads technique, which allows the selection of certain proteins (cf. Zhang et al., 2004), MS experiments often result in data sets with less variables than in DNA microarray data, usually only a few hundreds or even less than hundred. Whether this plays a role when applying the above named nonparametric method for the detection of differentially expressed proteins to MS data, will be discussed here, too.

This work is organised as follows. Chapter 2 gives a necessary overview of the biology of gene and protein expression, the bioanalytical instruments and the data structures that results from experiments with DNA microarrays, 2-DE and MS. Especially the biological terms like 'gene expression', 'genome' and 'proteome' are explained more detailed. Subsequent, in chapter 3, the nonparametric analysis of differentially expressed genes is explained and the properties of this method are evaluated. This is followed by chapter 4, where methods for the analysis of protein expression data from 2-DE experiments are detailed, that is the estimation of missing values and the analysis of time dependent protein expression data. Chapter 5 focuses on data obtained from MS experiments, specifically on how to detect outliers in such data and how to use them for the detection of differentially expressed proteins. Finally, the results are summarised and an outlook for further research activities is given in chapter 6.

2 Data from Bioanalytical Instruments and Their Preprocessing

The aim of molecular biology is to understand the molecular processes of organic cells, e.g. signal pathways, and the structures of cellular components like genes or proteins. Cognition from molecular biology research is important for the understanding of diseases and the development of new drugs. Several instruments from the area of analytical chemistry have been adapted to measure the amount of certain biomolecules within cells yielding a lot of data to be analysed by statistical methods. This chapter focuses on the technical procedures of bioanalytical tools which are used to measure gene or protein 'expression'. Gene expression can be measured with DNA microarrays, protein expression can either be measured by mass spectrometry or by two-dimensional gel electrophoresis. Understanding how these instruments work is necessary for the correct interpretation of the data which result from experiments with these technological tools. The specific characteristic of such data is that values for a great number of variables (often a few thousand) are observed upon only a few (five to twenty) numbers of objects. (Data from mass spectrometry experiments can also consist of only 50-500 variables).

In section 2.1, a brief introduction about the role that genes and proteins play within organisms is given. In particular, the structure of the DNA, the process of

protein synthesis and the terms gene and protein expression are explained. Section 2.2 presents the DNA microarray technology and the calculation of respective expression measures. In section 2.3, two-dimensional gel electrophoresis and some preprocessing steps for respective data is detailed. Finally, mass spectrometry and its resulting data structures are discussed in section 2.4.

2.1 Gene and Protein Expression

Functioning, growth and look of organisms are due to specific 'programs' which are coded by genes. Genes can be seen as certain sections on DNA-molecules (DNA = **D**eoxyribo**n**ucleic **A**cid). The specific structure of a gene represents the code for a protein or a part of one. Proteins are the building material of cells and determine therefore the structure of tissues and organisms. Furthermore, they control cellular processes and catalyse chemical reactions. The structure of the DNA and the process of protein synthesis are described in the following two subsections.

2.1.1 DNA, the Construction Plan of Organisms

Eucaryotic organisms contain their complete genetic material in the nucleus of their cells, namely in the form of chromosomes which are made up of DNA. The smallest unit of a DNA-molecule is a nucleotide which consists of a deoxyribose sugar, a phosphate group and a nitrogenous base. Long sequences of nucleotides are called DNA, where the phosphate group of each nucleotide binds to the sugar molecule of the next nucleotide. Small sequences of around ten to thirty nucleotides are called oligonucleotides (they are used for the so called oligonucleotide microarrays). The base molecule, which is bound to the sugar molecule, can either be adenine, guanine, thymine or cytosine. A chromosome is primarily the combination of two complementary DNA strands. Figure 2.1 displays a section of such a DNA double strand. The interface of the two DNA strands are base pairs, where adenine binds with thymine and guanine with cytosine. This complementary structure is used by

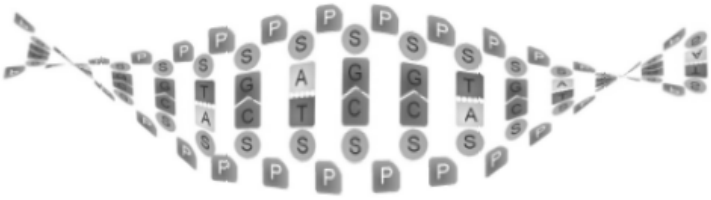


Figure 2.1: Extract from two complementary DNA strands, with P = phosphate, S = sugar, A = adenine, G = guanine, C = cytosine and T = thymine. The anatomy of the molecules causes the DNA to have the shape of a helix.

the DNA microarray technology, as will be seen in section 2.2. A certain section of the base sequence of a single DNA strand can be seen as a gene. This sequence is the code for a certain protein or a part of one. It should be remarked that not each gene codes for only one protein (sometimes, combinations of genes code for a protein). A human cell nucleus contains 46 chromosomes with a total of around 25.000 genes. The totality of all genes of an organism is called its 'genome'. A much greater number is that of the proteins that can be synthesised by the human genome. At present, this number is estimated to lie between 500.000 to 1.000.000. Similar to the genome the totality of these proteins is called the 'proteome'. The respective research fields are called 'genomics' and 'proteomics'.

2.1.2 The Process of Protein Synthesis

A simplified representation of the process of protein synthesis can be given by its two main steps 'DNA transcription' and 'mRNA translation'. At the transcription step, a section of a single DNA strand that represents a certain gene is transcribed into a complementary, single stranded mRNA-molecule (mRNA = messenger Ribonucleic Acid). This molecule is also called the transcript RNA and is the construction plan for a protein or a part of one. mRNA-molecules differ from DNA-molecules only by having the sugar ribose instead of the sugar deoxyribose and the base uracil instead

of the base thymine. This mRNA strand leaves the nucleus through the nucleus pores into the cell plasm. The place where this mRNA molecule is translated into a protein is called ribosome, one of the cell organelles. The mRNA strand adsorbs at the ribosome and the translation step takes place. Any triplet of bases on the mRNA molecule codes for a certain amino acid. A series of amino acids builds a peptide. Peptides with more than 100 amino acids are proteins. The translation and transcription processes are controlled by proteins themselves. Figure 2.2 displays this simplified version of the process of protein synthesis. A more detailed

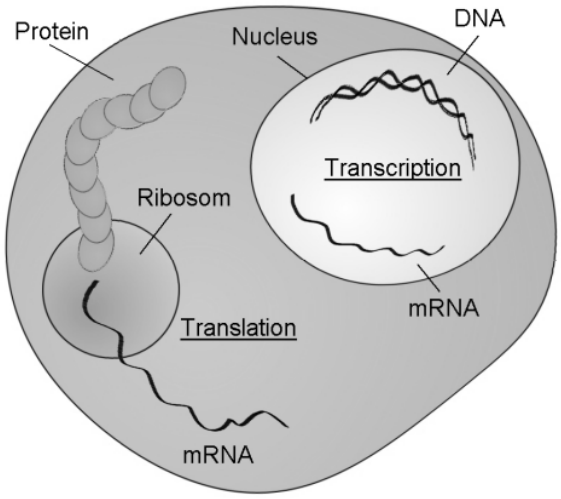


Figure 2.2: Main steps of protein synthesis: transcription of DNA within the nucleus and translation of mRNA at a ribosome.

representation of the process of protein synthesis is given for example in Lorkowski and Cullen (2001).

If the respective mRNA for a certain gene is located within the cell plasm and is thus able to adsorb at a ribosome to start the protein synthesis, this gene is called 'expressed'. The more translatable mRNA exists in the cell plasm the higher is the expression level of this gene. Equally, a protein is called expressed if it has been synthesised and is located within the cell plasm, too. The expression level of a certain gene or protein depends on the cell type and the cell state. If one wants to

know how much of a certain gene or protein is expressed in a certain cell type, the amount of translatable mRNA outside the cell nucleus or the amount of expressed proteins has to be measured. DNA microarrays can be employed to measure the expression levels of thousands of genes at the same time. Expression levels of proteins can simultaneously be measured either by two-dimensional gel electrophoresis or by mass spectrometry.

2.2 Oligonucleotide Microarrays

The most commonly used types of DNA microarrays are 'cDNA microarrays' and 'high density oligonucleotide microarrays'. The latter ones were developed by the Affymetrix Inc. (California) and will be described in the following subsection. For a description of cDNA arrays compare for example Brown and Botstein (1999) and Schulze and Downward (2001).

2.2.1 Measuring Gene Expression with DNA Microarrays

In the first step of a DNA microarray experiment, a glass or silicon chip is sectioned by a grid, where each unit is designated for one gene. The newest generation of oligonucleotide microarrays provides grid sections for each gene of the human genome. Around 10.000 single stranded DNA molecules, or more precisely oligonucleotides, that represent a certain gene are immobilised on the respective grid unit. Next, the expressed genes (single stranded mRNA molecules) are extracted from the cells of interest. These molecules are now tagged by a fluorescent dye and applied to the prepared chip (compare figure 2.3). Because of the complementary structure of DNA and mRNA the single stranded mRNA molecules from the cell sample bind to the single stranded DNA molecules on the chip. This binding process is also called hybridisation. The more molecules of a certain gene are expressed in the sample the more mRNA molecules are hybridised to the DNA on the respective grid unit of the chip. Hence, the grid unit for this gene on the

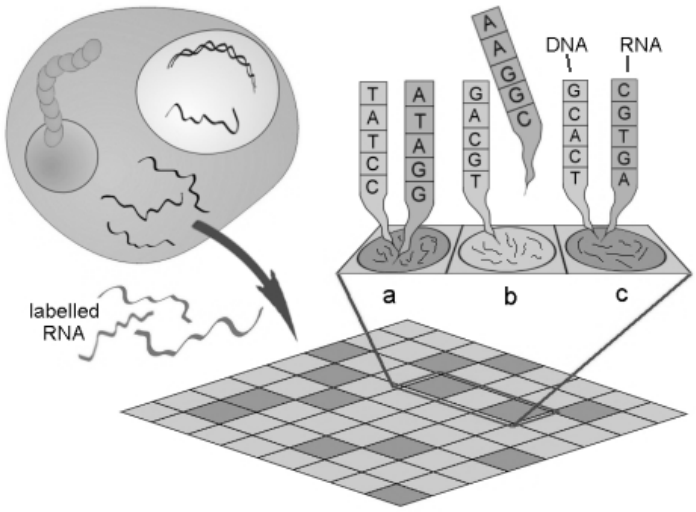


Figure 2.3: Scheme of the workflow of DNA microarray experiments. mRNA binds to complementary DNA at grid units a and c due to complementary bases but not at unit b.

chip gets marked by the dye, too. The strength of the fluorescence at this grid unit can be seen as a measure for the expression level of this particular gene. Using a confocal laser scanner the image of the complete chip is scanned. An example of such an image is given in figure 2.4. More details on oligonucleotide microarrays can also be found in Schulze and Downward (2001) and in Nguyen et al. (2002).

For each grid unit on the array, the scanned image contains an area of 8 times 8 pixels. An image analysis software automatically determines those pixels which belong to the actual hybridisation area and those which belong to the chip background. A difference between the actual hybridisation spot and the background noise is calculated and yields the fluorescence value for this grid unit (cf. Affymetrix, 2001).

The difference when using cDNA microarrays is, that not one but two independent samples (tagged with different dyes) are applied to the same cDNA array.

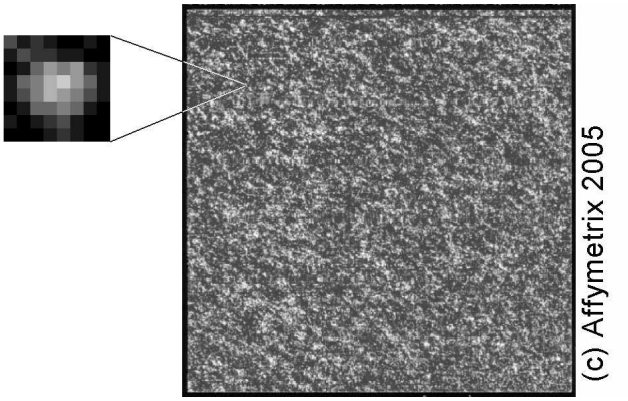


Figure 2.4: Image from a DNA microarray, scanned by a laser scanner. The image contains 8 times 8 pixels per grid unit. The fluorescence intensity within a grid unit is a measure of expression for the respective gene.

2.2.2 Calculation of Expression Values

The most common oligonucleotide microarray is the Affymetrix's GeneChip®. The analyses in this work are based on measurements taken with this chip. In order to get reliable expression values, each gene is represented on 40 grid units on the GeneChip®. From the respective 40 fluorescence values one single expression value has to be derived. To be more precise, each gene is represented as a probe set, containing 20 probe pairs. A probe pair consists of two grid units, where one, called the Perfect Match (PM), contains single stranded DNA molecules from the respective gene and the other one, called the Mismatch (MM), has the same sequence, except for the difference of one base. Under ideal conditions, there should be no hybridisation at the MM sections (because of the one exchanged base) and the MM values should be zero. In reality, however, there are some hybridisations at the MM, because of some modifications of the mRNA molecules. To summarise the values of a probe set to one single expression value for the respective gene, Irizarry et al. (2003) review a couple of measures of expression. One expression value for gene i on array j is for example given by the average difference (called AvDiff) of the the PM and MM values:

$$AvDiff = average\{d_v = (PM_v - MM_v) | v = 1, ..., 20\}. \tag{2.1}$$

However, this linear measure has the disadvantage that differences from high fluorescence values are overestimated. Another measure of expression that is based on a statistical model for the PM and MM values was proposed by Li and Wong (2001).

The expression measures used for the analyses in this work are the so called 'signal'-values which are calculated by the Affymetrix's software MAS 5.0 (cf. Affymetrix, 2001). Their calculation algorithm is explained in Affymetrix (2002) and proceeds in five steps. First, the MM and PM intensities are corrected from background noise. Second, an ideal mismatch is calculated and subtracted to adjust the PM intensity. Third, the adjusted PM intensities are log-transformed to stabilise the variance. Fourth, the Tukey's biweight estimator (cf. Huber, 1981) is used to provide a robust mean for the resulting values. Finally, the signal-value is scaled using a trimmed mean.

The resulting signal values of a replicated microarray experiment can be represented in a $r \times n$-matrix with the genes in the rows and the arrays in the columns (cf. table 2.1). For each signal value, MAS 5.0 also derives the so called detection

Gene	Array 1	detection-p-value	...	Array n	detection-p-value
Gene 1	362.92	0.00011	...	911.86	0.00017
Gene 2	2219.46	0.00004	...	4401.18	0.00004
⋮	⋮	⋮	⋱	⋮	⋮
Gene r	2036.66	0.00004	...	4096.52	0.99961

Table 2.1: Extract of gene expression data from a DNA microarray experiment with the genes in the rows, the arrays in the columns and the expression values with their respective detection p-values as entries.

p-value that reflects the reliability of this value (cf. Affymetrix, 2001b). Only genes with small detection p-values should be taken for the actual statistical analyses. For

a given probe set, a two step algorithm determines the respective detection p-value. First, the 'discrimination score' R is calculated for each probe pair of a probe set:

$$R_v = (PM_v - MM_v)/(PM_v + MM_v), \tag{2.2}$$

with $v = 1, ..., 20$. The values of R are then tested against a user-definable threshold θ using the one-sided Wilcoxon's signed rank test (cf. Lehmann, 1986). Increasing θ reduces the number of genes which are falsely retained for further analyses, but may also reduce the number of genes which should rather be retained.

2.3 Two-Dimensional Difference Gel Electrophoresis (2-D DIGE)

2.3.1 Measuring Protein Expression with 2-D DIGE

Two-dimensional gel electrophoresis separates the proteins of a mixture by their charge z and their mass m to distinct spots. Currently, the two-dimensional gel electrophoresis is the separation method with highest resolution power for protein samples. Up to 10.000 proteins can be separated in one gel and therefore are accessible for quantitative analysis (cf. Klose and Kobalz, 1995). The proteins, tagged by a fluorescent dye, cause spots of different size on the two-dimensional gel (see figure 2.5). The size of each spot can be regarded as a measure of expression for its respective protein. An improvement of two-dimensional gel electrophoresis is given by the so-called 'difference gel electrophoresis' (2-D DIGE), developed by Amersham Biosciences (Sweden), which enables the user to put up to three different protein samples on the same gel. These different samples are labelled by different dyes (Cy2, Cy3 and Cy5). After separation the proteins are detected using a confocal fluorescence scanner, where the emission wavelengths are chosen specifically for each of the dyes. An image analysis software automatically determines the boundaries and sizes of the spots. Usually, a 2-D DIGE experiment is designed such that n independent replications of treatment and control samples are put on n

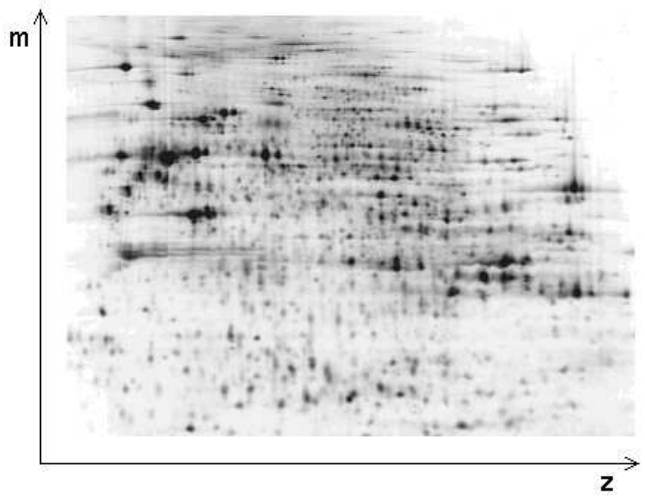


Figure 2.5: Image of a 2-D gel, scanned by a confocal fluorescence scanner. The spots represent the dye-labelled proteins, their sizes are used as measures of expression.

gels, where treatment j and control j are put on the same gel j ($j = 1, ..., n$). The internal standard, a mixture of same amounts of all n treatment and all n control samples, is also put on each gel (see table 2.2).

		Gel 1			...	Gel n		
spot	protein	treat.	contr.	stand.	...	treat.	contr.	stand.
spot 1	protein 1	23.1	24.7	29.3	...	13.4	13.8	14.3
spot 2	protein 2	15.2	11.2	20.6	...	10.3	11.7	12.5
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
spot r	protein r	34.8	42.0	48.1	...	47.9	43.9	44.4

Table 2.2: Protein expression data from a 2-D DIGE experiment. Each spot on a 2-DIGE gel is usually associated with one protein and yields three values: treatment, control and internal standard.

2.3.2 Calibration, Normalisation and Standardisation of 2-D DIGE Data

Before starting the actual statistical analysis of expression values from a 2-D DIGE experiment several preprocessing steps are required. In this section we examine procedures for calibration, normalisation and standardisation of such expression values. In particular, we evaluate the performance of the preprocessing methods that were proposed by Karp et al. (2004). These methods were also discussed in Jung et al. (2005, 2006) and in Sitek et al. (2006). The analyses in this section are based on expression measurements taken within a tumor study. For this study, samples were taken from five independent biological replicates at five different times. Thus, 25 gels were prepared. We call this study the 'TrkA experiment' in this section, details will be discussed in chapter 4.

The first preprocessing step is the calibration of the replicated gels. An impression of the necessity of calibration can be received from figure 2.6, where the raw background subtracted spot volumes (detected from a 2-D DIGE gel by an image analysis software) of the Cy2, Cy3 and Cy5 labelled samples are plotted against each other. The plots show linear dependencies between the different labelled sam-

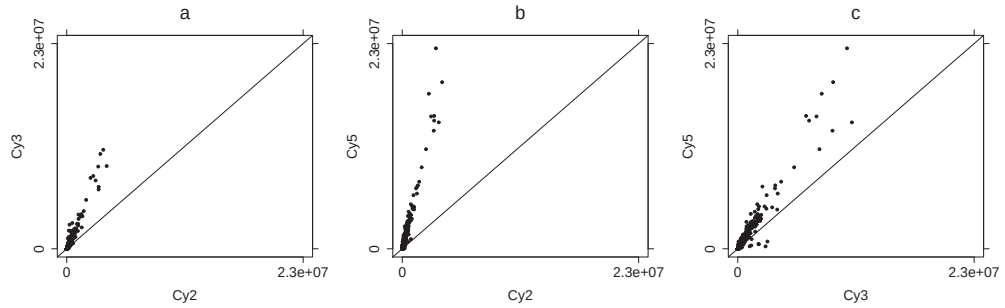


Figure 2.6: Raw background subtracted spot volumes of the Cy2, Cy3 and Cy5 labelled samples plotted against each other.

ples. However, the point clouds appear not on the line of gradient unity and the difference of the different labelled spots increases with increasing spot intensity.

Hence, it can be assumed that the scatter is not only due to biological variation but also to some dye effect. To remove this technical variation given by these dye effects Karp et al. (2004) and Kreil et al. (2004) proposed to use the calibration model

$$y_{ih} = a_h + b_h \tilde{y}_{ih}, \tag{2.3}$$

separately for each gel, with $i = 1, ..., r$ and $h = 1, 2, 3$, where $\tilde{y}_{ih}$ is the measured background subtracted spot volume of the ith spot from the sample that has been labelled with the hth dye. The calibrated value of this spot volume is y_{ih}. The dye effects are adjusted by the scaling factors b_h and the additive offsets a_h compensate for any constant additive bias present after background subtraction. Here, we assume, that the internal standard was labelled with Cy2 ($h = 1$), the treatment with Cy3 ($h = 2$) and the control with Cy5 ($h = 3$). This calibration model was originally developed by Huber et al. (2002) for the calibration of DNA microarrays. A corresponding software package, called 'vsn', for the open source statistic software R (available at http://cran.r-project.org) uses a robust version of maximum likelihood estimation for the estimation of the model parameters. We will call this preprocessing method the 'vsn-method', here. After calibration the spot volumes scatter around the line of gradient unity (see figure 2.7) and the scatter should now represent only the biological variation. This calibration method raises the question

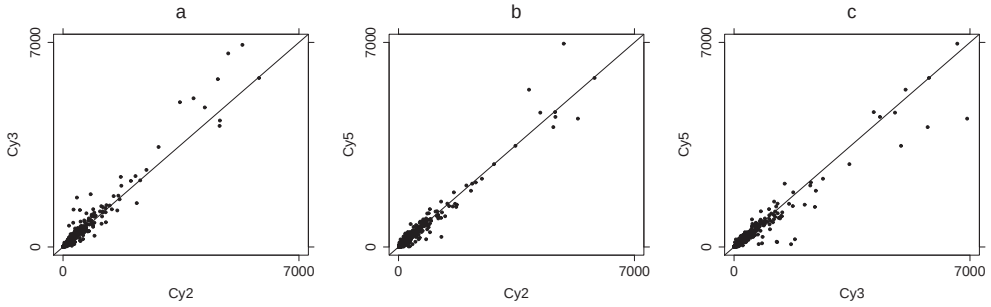


Figure 2.7: Spot volumes, calibrated by the vsn-method, of the Cy2, Cy3 and Cy5 labelled samples plotted against each other.

whether the dye effects are the same for all gels. We analyse this question by com-

paring the estimated parameters when calibrating each gel of the TrkA experiment. Table 2.3 shows the mean and its percentage deviation of the calibration factors and offsets for all gels of this experiment. As we can see, the percentage deviations

h	μ_1 =mean(a_h)	deviation(μ_1)	μ_2 =mean(b_h)	deviation(μ_2)
1	0.0006	128.0%	4.45	166.7%
2	0.0003	134.3%	6.67	155.1%
3	0.0001	125.8%	7.34	154.9%

Table 2.3: The mean and its percentage deviation of the calibration factors and offsets, respectively, when using calibration model 2.3 for each gel of the TrkA experiment.

from the means are higher than 100 % for each parameter, so there are obviously different dye effects for each gel. Hence, the calibration has to be done separately for each gel in the TrkA experiment. However, this observation should be validated in other experiments.

The next preprocessing step includes normalisation and variance stabilisation. In figure 2.7 it can be seen that the deviation of the spot volumes from the different labelled samples calibrated by the vsn-methods is bigger for big values than the deviation for small values. With the calibrated expression values from all five gels that have been prepared with the samples taken at time five of the TrkA experiment, one can calculate the mean and the variance of each spot. There were 1910 spots in these gels. The ranks of the means are plotted against the variances in figure 2.8a. Here, it can be seen that the variance for big values is higher than the variance for small values. Within the standardisation process (see below) the internal standard is subtracted from the treatment and from the control, respectively. That means, that one has to stabilise the variance, because differences obtained from spot volumes with a big variance have another quality than differences obtained from spot volumes with a small variance. One can either apply the logarithm or the arsinh

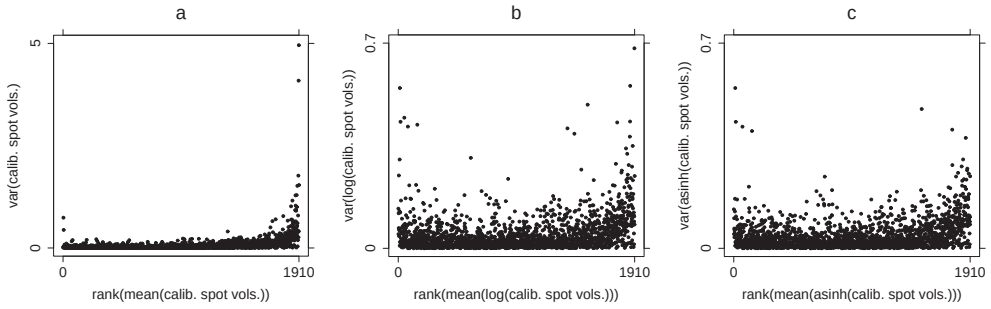


Figure 2.8: Rank of the mean versus the variance of a) the calibrated spot volumes, b) the calibrated and log-transformed spot volumes and c) the calibrated and asinh-transformed spot volumes.

(=area sine hyperbolicus) on the calibrated values to get a uniformly distributed variance. Figure 2.9 shows the calibrated spot volumes with the logarithm applied on them. However, the logarithm goes very fast to -∞ for small values and can thus

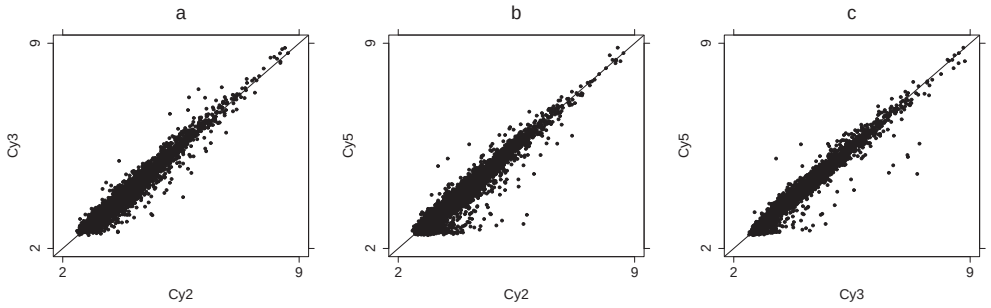


Figure 2.9: Spot volumes, calibrated and log-transformed, of the Cy2, Cy3 and Cy5 labelled samples plotted against each other.

cause a bias for small values. Instead of the logarithm one can also use the arsinh. This is a function that is similar to the logarithm but smoother for small values (see figure 2.10). The relationship between the two functions can be expressed by

$$\lim_{\xi \to \infty} (arsinh\xi - log\xi - log2) = 0. \tag{2.4}$$

The calibrated and arsinh-transformed values are plotted in figure 2.11. The effect of these transformations on the variance-mean-dependencies can be seen in figure 2.8. Figures 2.8 b and c show that after applying the logarithm or the asinh

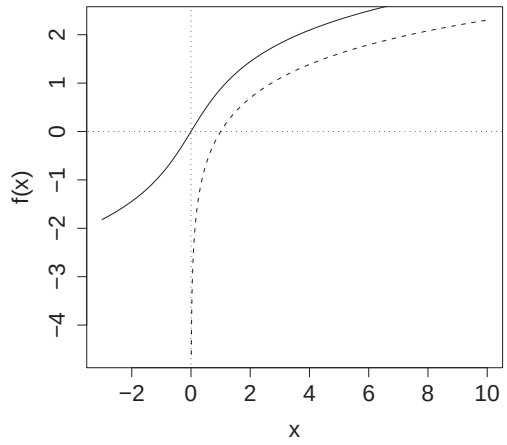


Figure 2.10: Graphs of the asinh(x) (solid line) and the logarithm(x) (dashed line).

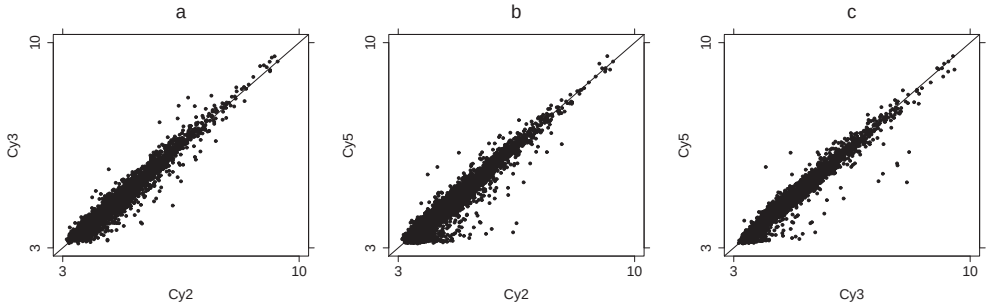


Figure 2.11: Spot volumes, calibrated and arsinh-transformed, of the Cy2, Cy3 and Cy5 labelled samples plotted against each other.

on the calibrated values the variance doesn't depend on the mean any more.

The last preprocessing step is the standardisation. The benefit of the DIGE method is to have an internal standard on each gel. The internal standard is a sample consisting of aliquots from all other samples of the experiment. Subtracting the values of the internal standard from the treatment and control values, respectively, brings all gels on the same level and reduces thus the gel-to-gel variance. The complete preprocessing way for the treatment value for spot i is thus given by either

$$x_i = log(a_2 + b_2\tilde{y}_{i2}) - log(a_1 + b_1\tilde{y}_{i1}), \tag{2.5}$$

or by

$$x_i = asinh(a_2 + b_2\tilde{y}_{i2}) - asinh(a_1 + b_1\tilde{y}_{i1}). \tag{2.6}$$

Similarly, the preprocessed control values are given by

$$x_i = log(a_3 + b_3\tilde{y}_{i3}) - log(a_1 + b_1\tilde{y}_{i1}), \tag{2.7}$$

or by

$$x_i = asinh(a_3 + b_3\tilde{y}_{i3}) - asinh(a_1 + b_1\tilde{y}_{i1}). \tag{2.8}$$

In figure 2.12 the density histogram of the vsn-processed and standardised values for the treatment values is given. This distribution is symmetric and nearly normally

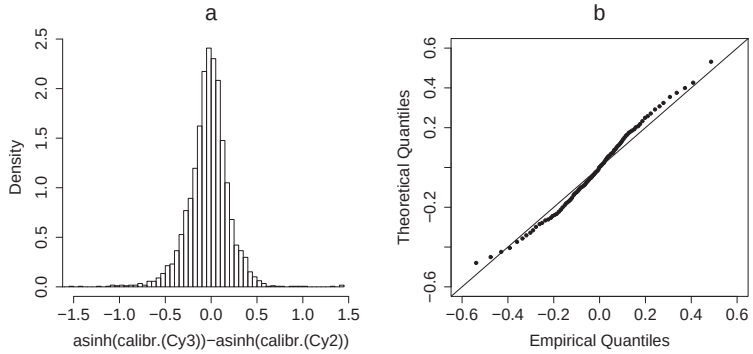


Figure 2.12: a) Density histogram of the preprocessed spot volumes from the treatment sample. b) QQ-plot of these values.

distributed as can also be seen in the QQ-plot.

2.4 Mass Spectrometry

Within proteomics, mass spectrometry (MS) covers a broad range of applications (cf. Aebersold and Goodlett, 2001), including for example protein identification (cf. Nesvizhskii et al., 2003), protein quantification (cf. Gygi et al., 1999) and the analysis of post-translational modifications (cf. Weckwerth et al., 2000). In general, mass spectrometry is used to determine the frequency of the ions of any analyte. This may not only be a protein but also a peptide. An ion is a positively or negatively charged atom or molecule. The most common instruments for

mass spectrometry are electrospray ionization (ESI) or matrix-assisted laser desorption/ionization (MALDI) mass spectrometers. The latter one in combination with a time-of-flight (TOF) mass analyser is described in the following subsection.

2.4.1 Measuring Protein Expression with MS

A simplified illustration of MALDI-TOF MS is given in figure 2.13. In the first

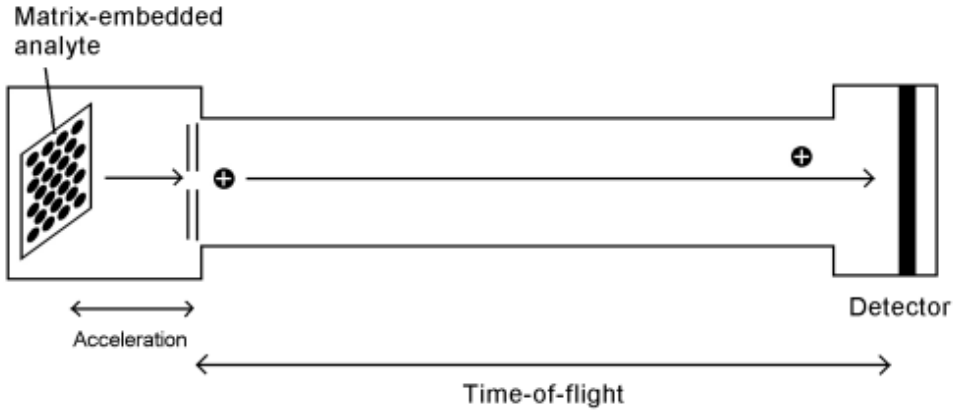


Figure 2.13: Workflow of Matrix-Assisted Laser Desorption/Ionization Time-Of-Flight mass spectrometry (graphic modified from Pusch et al., 2003).

step, the analyte is embedded into a crystalline matrix. Next, ions of the analyte are dissolved from the matrix by laser bombardment and accelerated by an electric field. These ions enter a field-free flight tube. At the end of the tube, the impact of the ions is detected and the time of flight is derived. From the time of flight of an ion the m/z-ratio can be determined, were m is the mass of the ion and z is the number of its charges. Theoretically, this *mass-to-charge ratio* is obtained by

$$\frac{m}{z} = \frac{2 \cdot e \cdot U \cdot t^2}{s^2}, \tag{2.9}$$

were e [1] is the elementary charge, U is the acceleration voltage, t is the time of flight and s is the length of the flight tube. In practice, however, the equation has

[1] The elementary charge $e = 1.6021 \cdot 10^{-19}$C is smallest detectable electric charge.

to be extended by an offset for t and some higher-order terms. The detector, mostly an electron multiplier, also determines the abundance of an ion with the respective m/z ratio. With the m/z ratios and the abundance values the mass spectrum of the analyte can be plotted (cf. figure 2.14). A more detailed explanation of

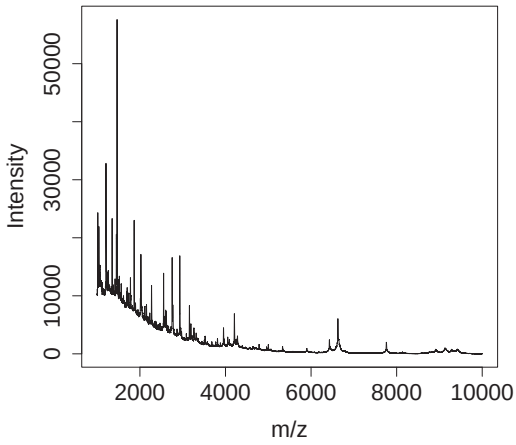


Figure 2.14: Example of a raw mass spectrum with the m/z ratios on the abscissa and the signal intensity on the ordinate.

MALDI-TOF mass spectrometry is given in Pusch et al. (2003).

In the context of protein mixtures, each protein or peptide is represented by a certain m/z value. Hence, the detected abundance of a certain ion with its specific mass-to-charge ratio m/z can be seen as a measure of expression for the respective ionised protein or peptide. Because the raw mass spectrum is nearly a continuous representation of m/z ratios the intensity values for a certain m/z area are summarised as the proteins abundance. Within the data sets analysed in this work, the intensities for the raw spectra were measured with a distance of about 0.1 between the current and the next m/z-value. The calibration of spectra and the determination of the abundance values is discussed in the following subsection.

2.4.2 Determination of Peak Intensities from a Mass Spectrum

All spectra that result from a MS experiment have to be preprocessed (cf. Villanueva et al., 2005 and Jeffries, 2005). This preprocessing takes place in three main steps: calibration, peak finding and expression calculation. Calibration includes some baseline subtraction as well as an alignment of the m/z-values. The alignment algorithm shifts the m/z-values of all spectra to the right or left respective to a reference spectrum. The peak finding works as follows. A peak can be seen as the intensity values between two minima of the raw mass spectra. Hence, peak finding means to determine the starting and ending m/z-values of all peaks. A problem within this procedure is that sometimes the peaks for certain proteins do overlap. For each peak some centroid m/z-value between the start and end point is reported as reference value. The expression values are determined by either taking the maximum intensity value of a peak, the sum of all intensity values within the peak range (cf. RheaCorporation, 1995) or by deriving the area under the curve within the peak range (e.g. by numerical integration). These summarised peak values can then be represented by a bar plot (see figure 2.15). Also a tabular

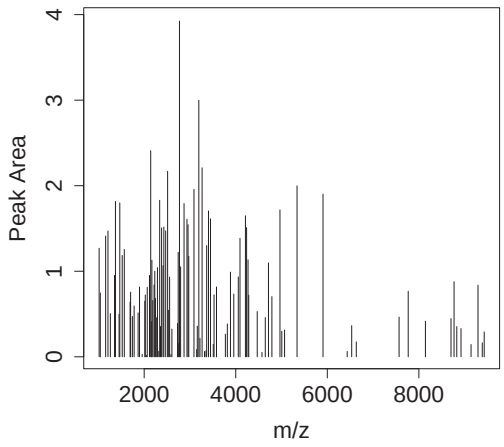


Figure 2.15: Bar plot of preprocessed mass spectrum. Intensities of extracted peaks are plotted against a reference m/z-values.

representation of the measurements can be given like in table 2.4, were the rows

represent proteins (or their respective m/z ratios) and the columns the calibrated spectra.

m/z-value	protein	spectrum 1	...	spectrum n
1016.099	protein 1	33.0215	...	29.0537
1041.409	protein 2	44.7988	...	28.5214
⋮	⋮	⋮	⋱	⋮
9158.490	protein r	40.0333	...	26.8359

Table 2.4: Extract of protein expression data from a mass spectrometry experiment. Each protein has a specific m/z ratio.

3 Analysis of Gene Expression Data

In chapter 2.2, we have seen how DNA microarrays are applied to measure the expression levels of genes and which steps are necessary to preprocess data from DNA microarray experiments. Here, the focus will be turned on the statistical analysis of such DNA microarray data. In section 3.1, a nonparametric method for the detection of differentially expressed genes is given. A software implementation of this method is presented in section 3.2, as well as a brief introduction of how to use the functions of this implementation. The performance of the nonparametric method is evaluated in section 3.3. In section 3.4, finally, the software implementation is used for the analysis of gene expression data that have been collected from normal and cancerous kidney tissues.

3.1 Detection of Differentially Expressed Genes

In many genome studies interest lies on the comparison of the expression profiles of genomes in different types of tissue (e.g. normal and cancerous) or in different treated biological samples (e.g. from treatment group and control group). To be more precise, the goal is to find those genes which are differentially expressed between these different tissues or samples. Since DNA microarrays have been established as the fundamental instrument of genome research, several approaches for the detection of differentially expressed genes were published. The 'significance analysis of microarrays' (SAM), described by Tusher et al. (2001), assigns a score to each gene on the basis of change in gene expression relative to the standard

deviation of repeated measurements. Genes with scores greater than an adjustable threshold are called significant. Efron et al. (2001) introduced a nonparametric 'empirical Bayes' (EB) method. This method also assigns a score to each gene and models the distribution of these scores by a mixture model. This model combines the density of the scores for differentially expressed genes with the density of the scores for non-differentially expressed genes. Based on this idea, the 'mixture model method' (MMM) of Pan et al. (2001) uses these two densities for the construction of a likelihood ratio statistic to identify the significant genes. A fully nonparametric approach is given by Gannoun et al. (2002) and Gannoun et al. (2004), who estimate the densities of the test statistics by nonparametric kernel estimation. This approach allows a very fast implementation, because no bootstrap is necessary like with the above named methods. A completely different approach was proposed by Pepe et al. (2003), who use Receiver Operating Characteristic (ROC) curves to directly compare the distribution of each gene in the treatment and the control group, respectively. Rajagopalan (2003) compares methods which are directly based on the PM and MM values (cf. section 2.2.2).

In Jung et al. (2006b) a renewed approach to Gannoun's fully nonparametric approach was presented. This approach is detailed in this section. A review of other nonparametric methods is given in Troyanskaya et al. (2003).

3.1.1 Multiple Hypotheses Testing

The concrete data situation when searching for differentially expressed genes is the following. Suppose, r genes on n arrays have been observed in a DNA microarray experiment. The expression level of gene i on array j is then denoted by x_{ij}, with $i = 1, ..., r$ and $j = 1, ..., n$. The expression level can for example be the 'signal'-intensity calculated by the Affymetrix's software MAS 5.0 (cf. section 2.2.2). Suppose further, that $n = n_1 + n_2$, where the expression values of the first n_1 arrays were obtained from the treatment samples and the expression values of

the last n_2 arrays were obtained from the control samples. The specified situation is displayed in table 3.1. In the statistical analysis of DNA microarray experiments

	treatment			control		
	array$_1$	...	array$_{n_1}$	array$_{n_1+1}$	...	array$_{n_1+n_2}$
gene 1	x_{11}	...	x_{1n_1}	x_{1n_1+1}	...	$x_{1n_1+n_2}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
gene r	x_{r1}	...	x_{rn_1}	x_{rn_1+1}	...	$x_{rn_1+n_2}$

Table 3.1: Gene expression data from r genes in n_1 treatment arrays and n_2 control arrays.

a test is carried out for each single gene, testing whether this gene is differentially expressed or not, raising the problem of multiple hypotheses testing. For each gene i the null hypothesis that there *is no* differential expression is tested against the alternative hypothesis that there *is* an expression change. Having observed the expression levels of r genes simultaneously this multiple testing procedure can result in a 2×2 contingency table (see table 3.2). This table illustrates also the general

		TEST DECISION		
		gene is not diff. expressed	gene is diff. expressed	
REALITY	gene is not diff. expressed	e_1 (true decisions)	e_2 (type I errors)	$e_1 + e_2$
	gene is diff. expressed	e_3 (type II errors)	e_4 (true decisions)	$e_3 + e_4$
		$e_1 + e_3$	$e_2 + e_4$	r

Table 3.2: Possible result when testing r genes simultaneously for differential expression.

problem of statistical hypothesis testing, that is the test decision can diverge from the real situation, leading to a *type I* or a *type II error*. Of course, when testing

r genes simultaneously, it is desired to have only small numbers e_2 and e_3 of *type I* and *type II errors*, respectively. The probability for a *type I error* is controlled by the fixed level α of the entire testing procedure. (Control of the probability for a *type II error* is discussed in section 3.3). In a single hypothesis test it is often desired to keep $P(\textit{type I error}) \leq \alpha$. In the case of multiple hypotheses testing however, it is desired to control the *family-wise error rate* (FWER) which is defined as $P(\textit{number } e_2 \textit{ of type I errors} \geq 1)$. In order to guarantee that $P(e_2 \geq 1) \leq \alpha$, one can use for example the Bonferroni correction, that is testing each single hypothesis at the *nominal* significance level $\alpha^* = \alpha/r$. The parameter α is called the *global* significance level of the entire multiple testing procedure.

Besides the FWER, the *false discovery rate* (FDR) (cf. Benjamini and Hochberg, 1995) is a widespread error rate which is often desired to be controlled in multiple hypotheses testing. It is defined as the expectation of the ratio $e_2/(e_2 + e_4)$ if $(e_2 + e_4) > 0$ and as 0 if $(e_2 + e_4) = 0$. The advantage of controlling the FDR over controlling the FWER is that it is less 'conservative'. A testing procedure is called conservative, if the real α-level falls below the α-level that has been fixed before the experiment. A conservative testing procedure has also the disadvantage that the probability for a type II error increases. However, having r p-values from a multiple testing procedure which controls the FWER it is possible to derive the so-called q-values to control the FDR (cf. Storey and Tibshirani, 2003, and Storey, 2003). More details on multiple hypotheses testing can be found in Shaffer (1995) and Dudoit et al. (2003).

3.1.2 Nonparametric Approach for Replicated Microarray Experiments

For the multiple testing procedure the expression of gene i on array j can be modelled as

$$x_{ij} = \beta_i + \mu_i \gamma_j + \varepsilon_{ij} \tag{3.1}$$

where $\gamma_j = 1$ for $1 \leq j \leq n_1$ and $\gamma_j = 0$ for $n_1 + 1 \leq j \leq n$, and ε_{ij} are independent random errors with a symmetric distribution about 0. The mean expression levels for gene i under the two conditions are then $\beta_i + \mu_i$ and β_i, respectively. Hence, determining whether gene i is differentially expressed, is equivalent to testing the null hypothesis $H_{0i} : \mu_i = 0$ against the alternative $H_{1i} : \mu_i \neq 0$.

Testing the r null hypotheses by simple t-tests would not be appropriate, since t-tests are restricted by the assumption that the data are normally distributed. Microarray data however, often have a different than the normal distribution. Instead of t-tests, a mixture model can be applied. The idea of this mixture model is to have not only a t-test statistic Z_i (for each gene $i = 1, ..., r$) but also a second null statistic z_i ($i = 1, ..., r$) which has under the global null hypothesis H_0 (none of the genes is differentially expressed) the same distribution as the Z_i's. Although test statistics are usually denoted by capital letters, the null statistics z_i are denoted by lower case letters, here, because this was also done in all other publications of this nonparametric approach.

Denote the density of all Z_i's by $f(Z)$, the density of the Z_i's for only those genes which are differentially expressed by $f_1(Z)$ and the distribution of all z_i's by $f_0(z)$. Further, denote w_1 as the probability that a gene is expressed differentially and $w_0 = 1 - w_1$ as the probability that a gene is not expressed differentially. The density of the Z_i's for all genes can then be expressed by the mixture model

$$f = w_0 f_0 + w_1 f_1. \tag{3.2}$$

Using Bayes' formula (cf. Mood et al., 1974) the probability that gene i is not expressed differentially, given its statistic Z_i, is

$$w_0(Z_i) = \frac{w_0 f_0(Z_i)}{w_0 f_0(Z_i) + w_1 f_1(Z_i)} = \frac{w_0 f_0(Z_i)}{f(Z_i)}, \tag{3.3}$$

and the probability that gene i is expressed differentially, given Z_i, is

$$w_1(Z_i) = 1 - \frac{w_0 f_0(Z_i)}{f(Z_i)}. \tag{3.4}$$

It can be seen that the probability for differential expression increases when the likelihood ratio

$$LR(Z) = f_0(Z)/f(Z) \tag{3.5}$$

decreases. Under the global null-hypothesis, where Z_i and z_i have the same distribution, the likelihood ratio becomes big and the probability for differential expression becomes small.

A cut-off point c for $LR(Z)$ can be found by solving the following equation

$$\frac{\alpha}{r} = \int\limits_{LR(z)<c} f_0(z)dz, \tag{3.6}$$

where the left hand side represents the Bonferroni adjusted significance level. The set of differentially expressed genes is then given by all genes i for which $LR(Z_i) < c$. Denoting $\{A = Z : LR(Z_i) < c\}$, equation 3.6 can be rewritten as

$$\frac{\alpha}{r} = \int\limits_{A} f_0(z)dz \approx \int\limits_{-\infty}^{\tilde{Z}_1} f_0(z)dz + \int\limits_{\tilde{Z}_2}^{\infty} f_0(z)dz. \tag{3.7}$$

From this one can specify the set of differentially expressed genes also by the rejection region for the Z_i's:

$$\{Z : Z < \tilde{Z}_1 \ \texttt{or} \ Z > \tilde{Z}_2\}. \tag{3.8}$$

Two statistics Z_i and z_i which satisfy the above requirements have been proposed by Zhao and Pan (2003). They are constructed as follows.

$$Z_i = \frac{\bar{X}_{i(1)} - \bar{X}_{i(2)}}{\sqrt{s^2_{i(1)}/l_1 + s^2_{i(2)}/l_2}}, \tag{3.9}$$

where

$$\bar{X}_{i(1)} = \frac{\sum_{j=1}^{n_1} x_{ij}}{n_1}, \qquad \bar{X}_{i(2)} = \frac{\sum_{j=n_1+1}^{n} x_{ij}}{n_2}, \tag{3.10}$$

are the sample means of the expression levels of gene i in the different groups, and

$$s^2_{i(1)} = \frac{\sum_{l=1}^{l_1} (y_{il} - \bar{Y}_{i(1)})^2}{l_1 - 1}, \qquad s^2_{i(2)} = \frac{\sum_{l=l_1+1}^{l_1+l_2} (y_{il} - \bar{Y}_{i(2)})^2}{l_2 - 1}, \tag{3.11}$$

are modified sample variances, with

$$\bar{Y}_{i(1)} = \frac{\sum_{l=1}^{l_1} y_{il}}{l_1}, \qquad \bar{Y}_{i(2)} = \frac{\sum_{l=l_1+1}^{l_1+l_2} y_{il}}{l_2}, \tag{3.12}$$

and $l_1 = n_1/2$, $l_2 = n_2/2$, and

$$y_{il} = (x_{il} - x_{i,l_1+l})/2 \tag{3.13}$$

for $l = 1, ..., l_1$, and

$$y_{il} = (x_{i,l_1+l} - x_{i,l_1+l_2+l})/2 \tag{3.14}$$

for $l = l_1 + 1, ..., l_1 + l_2$. With (3.8) and (3.9) the null statistic is given by

$$z_i = \frac{\bar{Y}_{i(1)} - \bar{Y}_{i(2)}}{\sqrt{s^2_{i(1)}/l_1 + s^2_{i(2)}/l_2}}. \tag{3.15}$$

Under H_0, both statistics Z_i and z_i have the same distribution. This can be proved by recalling, that under $H_{0i} : \mu_i = 0$ the expression level of gene i is $\beta_i + \varepsilon_{ij}$ for all arrays in both groups ($j = 1, ..., n$), and that the distribution of the random errors ε_{ij} is symmetric about 0. Setting $x_{ij} = \beta_i + \varepsilon_{ij}$ in (3.10), (3.13) and (3.15), it can be seen, that under H_0 the distributions of Z_i and z_i are the same.
A restriction of these two statistics is, however, that both are based on the measurements of an even number of DNA microarrays.

A further requirement for Z_i and z_i to have the same distribution under H_0 is that numerators and denominators of each are independent. This is given for the above stated statistics but was not the case for the statistics used in Gannoun et al. (2004). However, the statistics used here result in some loss of power for the testing procedure.

In order to solve equation (3.6) an estimate $\widehat{LR(Z)}$ for the likelihood ratio is required. Therefor, one first estimates the densities f and f_0 of Z_i and z_i by nonparametric kernel estimation

$$f_r(z) = \frac{1}{rh_r} \sum_{i=1}^{r} K(\frac{z - Z_i}{h_r}) \tag{3.16}$$

and

$$f_{0r}(z) = \frac{1}{rh_{0r}} \sum_{i=1}^{r} K(\frac{z - z_i}{h_{0r}}) \tag{3.17}$$

with kernel function K and the bandwidths:

$$h_r = \hat{\sigma}_r r^{-1/5} \quad \text{and} \quad h_{0r} = \hat{\sigma}_{0r} r^{-1/5}$$

where $\hat{\sigma}_r$ and $\hat{\sigma}_{0r}$ denote the empirical standard deviation of the Z_i's and the z_i's, respectively. As kernel function one can use for example the Gaussian density. In that case one should multiply the bandwidths by the factor 1.144 which minimizes the *integrated mean square error*

$$IMSE(f_r(z), f(z)) = \int_{-\infty}^{\infty} E\{[f_r(z) - f(z)]^2\} dz \tag{3.18}$$

of $f_k(z)$ (cf. Terrell, 1990).

Another useful approach to estimate the null distribution of the test statistic is given in Guo and Pan (2004) who construct weighted permutation scores, using posterior probabilities of having no differential expression. Genes are weighted by these posterior probabilities.

With the kernel density estimates an estimate $\widehat{LR(Z)} = f_{0r}(Z)/f_r(Z)$ for the likelihood ratio can be obtained and one can thus solve equation 3.6. The practical solution of equation 3.6 can be done by using numerical integration (e.g. trapezoidal rule) to calculate the integral (cf. Davis and Rabinowitz, 1984) and by the iteration algorithm given in the next section.

3.1.3 Iteration Algorithm for the Cut-Off Point

The cut-off point c from equation 3.6 can be determined by an iterative algorithm where the true c levels off between a lower bound c_a and an upper bound c_b. The algorithm starts with $c_a = 0$ and $c_b = \max(\widehat{LR(Z)})$. To determine c, the following steps should be repeated until a break-off criterion has been reached:

1. $c = (c_a + c_b)/2$.

2. Calculate the integral
$$I = \int_{LR(z)<c} f_0(z)dz$$
on the right hand side of equation 3.6, e.g. by numerical integration.

3. If $I < \alpha/r$, set $c_b = c$

4. If $I > \alpha/r$, set $c_a = c$

As break-off criterion the difference $|I-\alpha/r|$ should be used. Usually, this difference becomes small enough (around 10^{-7}) after about 20 to 30 runs of the above four steps. Thus, this iteration algorithm leads to a little over- or underplacement of α/r. But, according to experience, a difference of around 10^{-7} doesn't influence the set of differentially expressed genes determined by the nonparametric method.

3.1.4 Calculation of p-Values within Nonparametric Approach

Having determined differentially expressed genes in different types of tissues or in a treatment and control group, it can also be of interest to rank these genes by the strength of their differential expression. Oftentimes, there are some few hundred of significant genes. Ranking them has the benefit that further research can be concentrated on the most significant genes, saving time and money. One possibility to do this is to calculate the p-values for each single test of the multiple testing procedure. Here, the *unadjusted* p-values are considered first. They are defined as the smallest possible nominal significance level α^* for which the null hypothesis of a single test would just be rejected. The smaller the p-value the stronger the evidence against the null hypothesis. Hence, one can order the differentially expressed genes by their respective p-values.

The r unadjusted p-values for the multiple testing procedure of the previous section can be calculated as follows. For gene i the p-value can be obtained by a

modification of equation (3.6):

$$p_i = \int\limits_{\widehat{LZ}(Z)<c^{(i)}} f_{0r}(z)dz, \tag{3.19}$$

where $c^{(i)} =: \widehat{LZ}(Z_i)$. I.e., when the data for gene i results in the value Z_i, the smallest possible significance level for which the null hypothesis would be rejected is given by p_i.

It is also possible to use *adjusted* p-values. The adjusted p-value $\tilde{p}_i$ for gene i is defined as the smallest possible global significance level α of the multiple testing procedure for which H_{0i} would just be rejected. Using the Bonferroni method to control the FWER the adjusted p-value for gene i can be obtained by

$$\tilde{p}_i = \min(rp_i, 1). \tag{3.20}$$

Other p-value adjustments, for example to control the FDR, can be found in Dudoit et al. (2003). Having calculated the p-values the differentially expressed genes can also be determined by calling all genes significant that have adjusted p-values less than or equal α.

3.1.5 Power Calculations for Sample Size Planning

Important for DNA microarray experiments is the question of how many arrays should be used to make the subsequent statistical results reliable. This question concerns not only the high costs of DNA microarrays but also the ethical aspect of taking tissue samples from patients. Pan et al. (2002) proposed to plan sample sizes for the above given nonparametric method by power calculations. Their approach will be detailed in this subsection. The power of a statistical test is the probability of rejecting the null hypothesis. It should be very high when the null hypothesis isn't true. Therefore, one calculates the power of the nonparametric method for different sample sizes and decides for that sample size which results in the largest power.

Because the power also depends very strongly on the specific data one wants to investigate, power calculations are usually based on some pilot data. Let's start the calculations with the expression values of $\kappa = n/2$ arrays from each group. With these data, the densities $f_r(z)$ and $f_{0r}(z)$ as well as the rejection region $\{Z : Z < \tilde{Z}_1 \texttt{ or } Z > \tilde{Z}_2\}$ of the nonparametric method are determined. Now, the power function is given by

$$p(\delta, \alpha) = \int_{-\infty}^{\delta - \tilde{Z}_1} f_{0,r}(z)dz + \int_{\delta + \tilde{Z}_2}^{\infty} f_{0,r}(z)dz, \tag{3.21}$$

where δ is the magnitude of expression change. The power can then be plotted against δ. If the power for a certain δ of interest is too small one can determine the power function for any number $\kappa \cdot k$ ($k \geq 2$) of replicates until the power is big enough. The power function for $\kappa \cdot k$ replicates per group can be determined by the following steps. Estimate the scores $z_{\kappa \cdot k,i}$ (these denote the z_i's based on $\kappa \cdot k$ replicates per group) by

$$z_{\kappa \cdot k,i} = \frac{1}{k} \sum_{j=1}^{k} z_{\kappa,i}^{(j)} \tag{3.22}$$

where $z_{\kappa,i}^{(j)}$ ($j = 1, 2, ..., k$) are k independent realisations of $z_{\kappa,i}$ (z_i's based on κ replicates per group). Now, one estimates the density of the $z_{\kappa \cdot k,i}$'s by kernel estimation, determines the new rejection region and calculates the new power function. When all power functions for possible $\kappa \cdot k$ have been obtained, one can determine an appropriate number of replicates.

3.2 Implementation of the Nonparametric Method

In Jung et al. (2006b) the R-package 'degenes' (=*d*ifferentially *e*xpressed *genes*) with an implementation of the nonparametric method was presented (cf. appendix A). R is an open source statistic software (available at http://www.r-project.org/). It is assumed, that the reader is familiar to the R environment. The implementation provides functions for the data import, the determination of differentially expressed

genes as well as for the sample size determination for replicated DNA microarray experiments.

3.2.1 Data Import

For the problem of determining genes with different expression under two different conditions, *two* data sets are required. In general, two matrices of expression values, with the genes in the rows and the arrays in the columns, have to be read into the working space of the R environment.

In the special case that the data sets are .txt-files, including the names of the genes in the first column, one can use the function `read.values`. There may also be the column names in the first row:

	array1	array2	...	arrayn
gene1	439.60	448.24	...	501.30
gene2	2660.14	2726.31	...	2378.56
⋮	⋮	⋮	⋱	⋮
gener	53.54	63.11	...	71.41

In that case, the data can be imported as follows:

```
R> treatment <- read.values(".../file1.txt", h)
R> control <- read.values(".../file2.txt", h)
```

Here, `file1.txt` contains the values of the treatment group and `file2.txt` the data of the control group. In order to skip the column names in the first row, the parameter `h` has to be set to 1 if there are row names in the .txt-files and 0 if there are no row names.

Another special case is the existence of 'detection p-values' in the data sets, which indicate whether an expression value could be regarded as reliable according to the PM and MM values (cf. Affymetrix, 2001). The data is arranged as in the following matrix, then:

	array1	pvalue	array2	pvalue	...	arrayn	pvalue
gene1	439.60	0.0113	448.24	0.0213	...	501.30	0.2124
gene2	2660.14	0.2347	2726.31	0.3269	...	2378.56	0.3291
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
gener	53.54	0.0014	63.11	0.0083	...	71.41	0.0002

In that case, one has to determine those genes for which reliable expression values exist first. The function `common.genes` filters all genes in that way that only genes for which the median of the p-values in at least one of the two groups is less than 0.05 are retained for the analysis. To apply this function type:

```
R> g <- common.genes(".../file1.txt", ".../file2.txt", h1,
h2)
```

The parameters `h1` and `h2` have to be set to 0 or 1 if there are row names in the first and second .txt-file, respectively, or if there are no row names. The vector `g` represents the indices of only those genes used in the further analysis. After the filtration about 30-40% of the original data will usually be deleted. The data can then be imported by typing:

```
R> treatment <- read.values("C:/.../file1.txt", g, h, pval=TRUE)
R> control <- read.values("C:/.../file2.txt", g, h, pval=TRUE)
```

If the expression values are distributed lognormal one should take logarithms of them (e.g. `R> treatment <- log(treatment)`).

3.2.2 Determination of the Differentially Expressed Genes

After reading the data into the working space, the differentially expressed genes can be calculated by using the function `deg`:

```
R> genes.table <- deg(treatment, control, ref, alpha)
```

Here, the parameter `ref` is the percentage of artificial observations to be created (e.g. set `ref=0.005` for 0.5%). Artificial observations can be used to improve

the kernel estimation in the case that there are only a few number of genes to be tested. It should be set to zero for data sets with more than 500 genes. Details on the calculation of artificial observations are given in section 5.2. The parameter `alpha` specifies the global significance level α. The results of `deg` can be printed by typing `genes.table`.

```
R> genes.table
$values
[1] 330 489 495 ...
```

If the expression values were filtered before using the function `deg()` (in the case of detection p-values in the data set), the user has to return the indices from the vector `g` and retrieves the indices of the differentially expressed genes in the original data set by typing:

```
R> g[genes.table$values]
```

To calculate the p-values the function `pvalue` can be used. Type for example

```
R> pv <- pvalue(treatment, control, ref, alpha)
```

The parameters are the same as for the function `deg`, here. The result of the function `pvalue` is a list (`$unadjusted, $adjusted`) containing the unadjusted and the adjusted p-values.
A ranking list of the significant genes ordered by their unadjusted p-values can be obtained by

```
R> pv.values <- pv$unadjusted[genes.table$values]
R> genes.list <- cbind(genes.table$values[order(pv.values)],
pv.values[order(pv.values)])
```

3.2.3 Determination of the Necessary Sample Size

If some pilot data, taken from earlier studies, are available, it is possible to determine the necessary sample size for future studies (cf. subsection 3.1.5). The

algorithm used here starts with $\kappa = n/2$ arrays under each condition (or group) and calculates the power function for the above given multiple testing procedure. The user should make sure that κ is an even number. The data have to be imported as described in section 3.2.1. Now, the power function for the multiple testing procedure with κ arrays for each condition can be calculated as follows:

```
R> power.plot(treatment, control, ref, alpha)
```

Parameters are the same as in the functions `deg` and `pvalue`. This will produce a graph sheet with the expression change δ on the x-axis and the power on the y-axis. Next, for any $k > 2$ one can calculate the power function for $\kappa \cdot k$ arrays by typing:

```
R> zmk(k, treatment, control, ref, alpha)
```

When all power functions, for possible $\kappa \cdot k$ replicates have been obtained, one can determine an appropriate number of replicates by considering the desired power, the global significance level and the targeted expression changes.

Investigations and recommendations about the necessary sample size for microarray experiments are also given in Pavlidis et al. (2003). An approach to determine the optimal sample size with respect to the FDR was proposed by Müller et al. (2004).

3.3 Performance of the Nonparametric Method

3.3.1 Average Power

In section 3.1.1, the error types of multiple testing procedures were discussed. While the probability α for the FWER or the FDR is given by the global significance level of the procedure, the probability β for the type II error depends on several characteristics of the data and the statistical method as well as on user defined requirements. Specifically, these are a) the α-level, b) the variance σ^2 of the data,

b) the sample size n, d) the type of testing procedure and e) the magnitude δ of differential expression to be detected. Of course, in the case of multiple hypothesis testing, it is not of interest to have a small probability β for the type II error but to have a small type II error rate (similar as the type I error rates FWER and FDR). One common type II error rate is the average probability of a type II error within the multiple testing procedure. In order to keep this error rate small one seeks for testing procedures which have a high *average power*. The average power is the average probability of rejecting a false null hypothesis (e.g. calling a significant gene significant). In this section the average power of the nonparametric method is compared to a permutation test (see appendix B). Both methods control the FWER. The average power for the two methods was determined by a simulation study with the following steps (cf. Dudoit et al. 2003).

1. Expression values from $r = 10000$ genes in a treatment and control group were generated. For 9900 genes the expression values were generated randomly from a $N(0, 0.2)$-distribution. For the remaining 100 genes the expression values were generated randomly from a $N(0, 0.2)$-distribution within the control group and from a $N(\delta, 0.2)$-distribution within the treatment group, were δ is the magnitude of differential expression. These settings reflect the situation of many real data sets from microarray experiments.

2. The two multiple testing procedures were applied to the data and the number e_3 of type II errors was recorded as well as the numer e_4 of correctly rejected null hypothesis. As global significance level $\alpha = 0.05$ was chosen.

For each expression change δ of interest steps 1 and 2 were repeated $\tilde{B} = 10$ times and the average power was derived by

$$Average\ power = 1 - \frac{\sum_{b=1}^{\tilde{B}} e_3^b/(e_3^b + e_4^b)}{\tilde{B}}, \tag{3.23}$$

where e_3^b =*(number of type II errors whithin the bth run)* and e_4^b =*(number of correctly rejected null hypothesis whithin the bth run).*

In a first flow, the simulated data consisted of 6 arrays for each group, in a second flow, expression values were simulated for 12 arrays for each group. The average power with respect to the expression change δ is plotted in figure 3.1 for the non-parametric method and in figure 3.2 for the permutation test. From these plots it

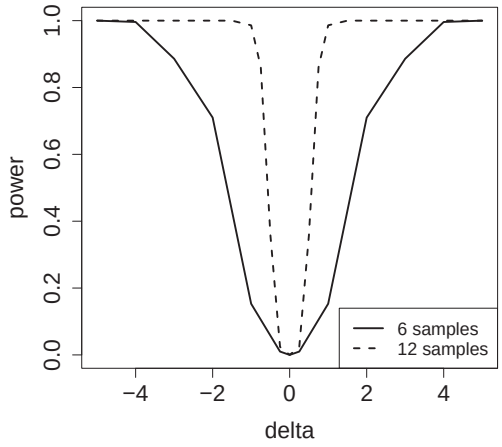


Figure 3.1: Average power for the nonparametric method with different sample sizes.

can be seen, that the power increases when the expression change δ increases, too. This is a plausible effect, because a big expression change is more easy to detect than a small one. It should be noted that for $\delta = 0$ the average power is equal to the nominal testing level $\alpha^* = \alpha/r$. The graphs also show that the average power increases with the number of observations. Altogether, the permutation test has a higher power than the nonparametric method.

3.3.2 Computing Time

Data that have been collected by bioanalytical high-throughput instruments are usually multi-dimensional with a high number of variables. Not until the development of a new and very fast generation of computer processors at the beginning of this decade rendered the analysis of gene expression data possible with comfortable

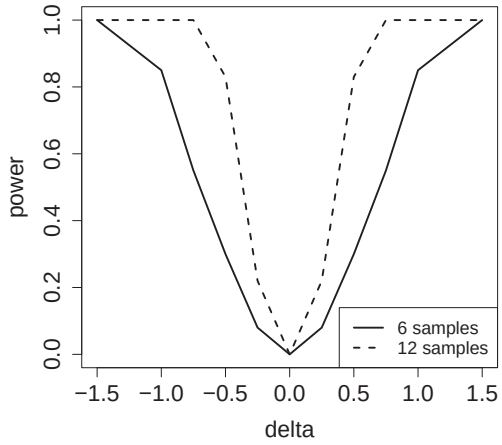


Figure 3.2: Average power for the permutation procedure with different sample sizes.

computing times. However, it is still worth to compare statistical methods by their computing time. In this section the computing time of the R-implementation of the nonparametric method (see section 3.2) is compared with an R-implementation of the permutation method (see appendix B). Both implementations were applied to a real data set of expression values from 10521 genes with 14 arrays of a treatment and control group, respectively. The computations were carried out on a computer with an AMD Athlon 2000+ processor with 1.67 GHz. The computing time for the nonparametric method was 1 minute and 9 seconds. The run of the permutation procedure took much longer, namely 34 minutes and 41 seconds. There are certainly faster implementations of the permutation procedure. But repetitive execution of tests for a large number of permutations will take a lot of time, independent of the implementation.

3.3.3 Breakdown of the Nonparametric Method

In the case that there are no differentially expressed genes in the data set the theoretical distribution of the Z_i and the z_i is the same. Hence, the likelihood ratio $LR(Z) = f_0 Z / f(Z) = 1$ for all Z. That means, that equation 3.6 can't be solved,

then, and the nonparametric testing procedure breaks. However, to be able to report a result, the cut-off point c should be set zero in that situation. If $c = 0$, none of the $LR(Z_i)$ is smaller than c and none of the genes is called differentially expressed.

It should be remarked, that in the case of this breakdown of the nonparametric method, it is also not possible to derive p-values as explained in section 3.1.3. The parameter $c^{(i)}$ should be set arbitrary bigger than one and the integral in equation 3.19 will be calculated from -∞ to ∞. The p-value for each gene will then be equal one.

3.3.4 Conclusions

We have seen in section 3.3.1 that the nonparametric testing procedure has overall a lower power than the permutation algorithm. On the other hand, the nonparametric method is very fast, as was discussed in section 3.3.2. Based on these insights, it can be recommended to use the nonparametric method to get first results and impressions from the data. For exact results one should also take the time to run the permutation algorithm. In addition, it is also possible to improve the power of both methods by converting the p-values into so called q-values and to control thus the FDR (cf. Storey and Tibshirani, 2003). In the case that there are no differentially expressed genes, one should incorporate the recommendations of section 3.3.3.

3.4 Example: Comparison of Normal and Cancerous Kidney Tissues

In Jung et al. (2006b), the software implementation of the nonparametric method (see section 3.2 and appendix A) was applied to gene expression data that were obtained from an examination of kidney-mRNA using the Affymetrix U-133A GeneChip®.

This DNA microarray allows to measure the expression levels of about 22.000 human genes simultaneously. The expression levels for one of the two data sets to be compared were obtained from tissues of kidney tumors, the expression levels for the other one were obtained from normal kidney tissues. Each of the two data sets consisted of the expression levels from 14 arrays. At first, the function `common.genes()` was used to discard those genes, for which the median of the detection p-values in at least one of the two groups was greater than or equal 0.05. Thereby, r =10521 genes remained for the actual analysis. Hence, the nominal significance level for each single test was $\alpha^* = 0.05/10521 = 4.7524 * 10^{-6}$. After reading the expression values into the working space of the R-environment (by using the function `read.values()`), the function `deg()` was applied to find the differentially expressed genes between the two tissue types. This function calculates first the values of the two statistics Z_i and z_i (see equations 3.9 and 3.15, respectively). Next, the densities $f_r(z)$ and $f_{0r}(z)$ of the distributions of these statistics are determined by kernel estimation (see equations 3.16 and 3.17). The graphs of these densities is plotted in figure 3.3. The likelihood ratio $LR(z)$ (see equation 3.5), i.e. is the quotient of the

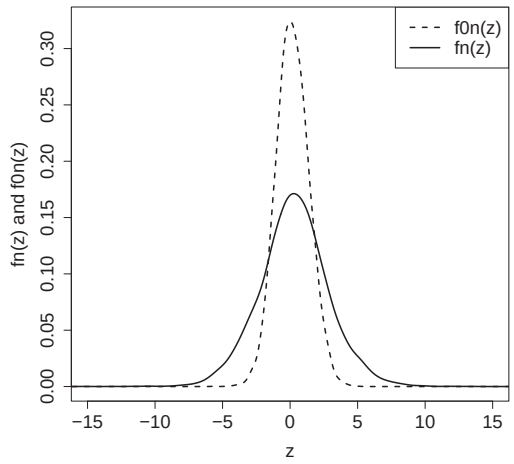


Figure 3.3: Density estimates f_{0n} and f_n of the distributions of the statistics Z_i and z_i, respectively.

both densities, is plotted in figure 3.4. Using the iteration algorithm from section

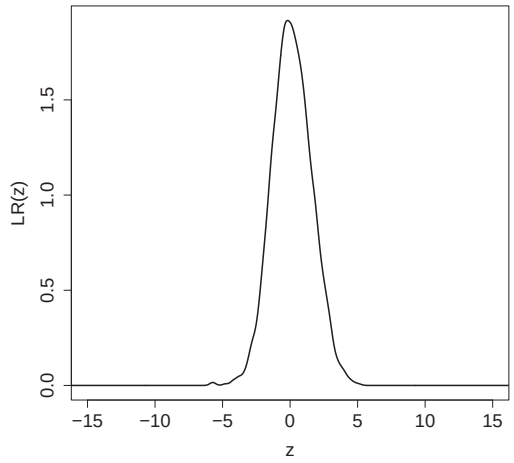


Figure 3.4: Likelihood ratio $LR(z)$.

3.1.3, the cut-off point c is determined as a solution of equation equation 3.6. This cut-off point c specifies the region on the z-axis were the integral in equation 3.6 has to be calculated, that is those z with $LR(z) < c$. Here, this region is given by $(-\infty, -6.17)$ and $(5.43, \infty)$, as can be seen in figure 3.5. The integral from equation

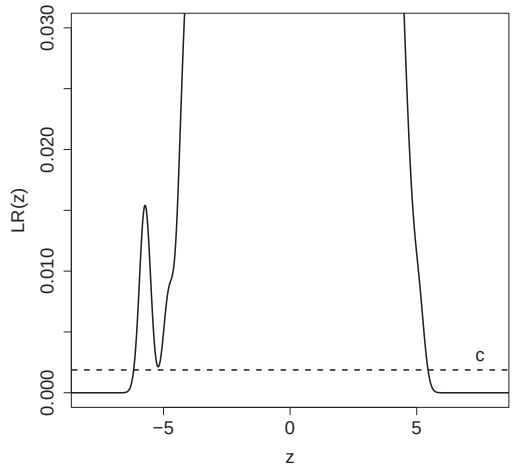


Figure 3.5: Likelihood ratio $LR(z)$ with cut-off point c.

3.6 over this region is illustrated in figure 3.6. Now, the function `deg` determined 339 genes for which $LR(Z_i) < c$, that is 339 gene were determined to be expressed

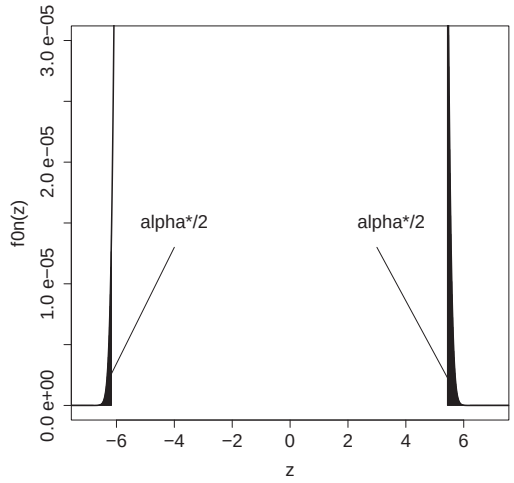


Figure 3.6: Illustration of the integral from equation 3.6.

differentially. Table 3.3 displays the top 10 differentially expressed genes, ordered by their adjusted p-values. These p-values were calculated by the function `pvalue`. Here, the row numbers refer to the original data matrix with about 22.000 genes, the name is Affymetrix's gene name. The complete set of differentially expressed genes is listed in appendix C.

Now, the kidney data are used to determine an appropriate number of replicates for future experiments. Therefore, power functions based on different numbers of replicates are calculated as described in 3.1.5 and 3.2.3. The result is illustrated in figure 3.7. As one can see, the power increases strongly by doubling the number of arrays from 10 to 20. But the gain in power is only less by adding again 10 more replicates.

rank	row-no.	name	p-value
1	17678	217773_s_at	> 0
2	18389	218484_at	> 0
3	19262	219358_s_at	$5.32*10^{-260}$
4	3108	203039_s_at	$8.10*10^{-257}$
5	17888	217983_s_at	$3.64*10^{-211}$
6	17889	217984_at	$2.12*10^{-204}$
7	1247	201178_at	$1.83*10^{-171}$
8	1047	200978_at	$6.09*10^{-160}$
9	973	200904_at	$3.41*10^{-131}$
10	11613	211671_s_at	$2.31*10^{-126}$

Table 3.3: Top 10 differentially expressed genes in kidney data.

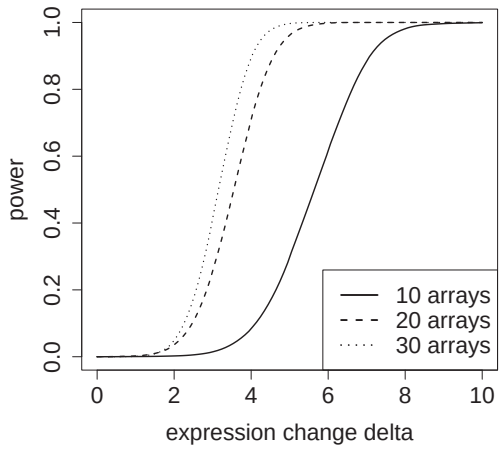


Figure 3.7: Power of the nonparametric method in reference to the expression change δ and for different sample sizes.

4 Analysis of Protein Expression Data from 2-D DIGE Experiments

As mentioned in the introduction, most of the questions that were posed to gene expression data from DNA mciroarray experiments have also to be answered in analyses of protein expression data. Furthermore, the statistical methods that can be used to answer these questions are similar for gene and protein expression data, respectively. However, in several cases some adaptions to the methods or the data have to be made. In section 2.3, the application of two-dimensional difference gel electrophoresis (2-D DIGE) for measuring protein expression was explained, as well as the preprocessing of 2-D DIGE data. Here, the focus will be turned on missing values in such data and on the analysis of time-dependent protein expression data.

The first subject of this chapter will be the estimation of missing values. Data sets obtained from experiments with 2-D gel electrophoresis often contain a lot of missing values, because these experiments are usually carried out with replications of gels and not each protein spot appears on each gel, due to technical nuisances. This drawback makes the transfer of known methods for gene expression data, or the general application of multivariate methods, to 2-D DIGE data more complicated. The estimation of missing values will be discussed in section 4.1. Furthermore, protein expression is often measured repeatedly over several times. The question is then to find significant differences in the temporal course of differently treated samples. For this purpose, a model for the analysis of time dependent protein

expression data will be introduced in section 4.2. Additionally, the problem of multiple hypothesis testing in 2-D DIGE experiments will be discussed in section 4.3. The methods in this chapter are explained for the special case of experiments with 2-D DIGE experiments, but they are also applicable to other forms of 2-D gel electrophoresis experiments.

4.1 Missing Values in 2-D DIGE Data

Like DNA microarrays experiments, 2-D DIGE experiments should be done with replications. That is each sample is applied to more than one gel in order to be able to asses the technical variations. A problem of replicating 2-D gels is, however, that not each protein spot appears on each gel. Table 4.1 displays the amounts of detected protein spots on the gels from a 2-D DIGE experiment with five replications. In this experiment, there have 1057 spots been detected on gel no. 1 and 1267 on

Gel no.	1	2	3	4	5
# spots	1057	1267	1226	1792	1138
# joint spots	1057	650	470	417	330

Table 4.1: Numbers of detected protein spots on five 2-D gels that were prepared with the same biological sample. The last row contains the portion of jointly detected spots from gel 1 up to the gel in the respective column.

gel no. 2, but there is only an intersection of 650 spots which appear jointly on both gels. After five replications there remain only 330 spots which have in fact five values available for the statistical analysis. The number of jointly existent spots decreases still more when measuring protein expression over several points in time. Many statistical methods, however, need complete data sets, especially those for multivariate data. These methods could also be applied to protein expression data if the data sets were complete. One possible method to overcome this problem is to estimate the missing values by using the available ones. In Jung et al. (2005,

2006a), methods for the estimation of missing values in 2-D DIGE data were evaluated. These methods have already been applied to incomplete DNA microarray data, but not with such great amounts of missing values. Unlike with data from DNA microarrays, where the number of missing values is mostly around 5-15%, 2-D DIGE data contains mostly around 20-30% of missing values. The application of these methods to 2-D DIGE data is thus a complete new challenge.

To begin with, some notations are given. Let X be the $r \times n$ matrix of protein expression values from a 2-D DIGE experiment, where the rows are referred to protein spots and the columns are referred to replications (gels). Hence, x_{ij} is the expression value of protein i on gel j, with $i = 1, ..., r$ and $j = 1, ..., n$, as given below.

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{i1} & x_{ij} & x_{in} \\ \vdots & & \vdots \\ x_{r1} & \dots & x_{rn} \end{pmatrix} \tag{4.1}$$

In the following subsections, let this matrix represent the preprocessed data, i.e. after calibration, normalisation and standardisation. Furthermore, X could either represent the matrix of the preprocessed values from the treatment or from the control. Although treatment and control matrices have missing values at the same positions in 2-D DIGE data – because both samples are applied to the same gel – the estimation of the missing values is done separately for both matrices.

4.1.1 Row Mean Method

Obviously, the most simple technique to estimate a missing value in the context of 2-D DIGE data is given by the row mean method. Let $X_i = (x_{i1}, ..., x_{in})^T$ be the ith row of X where one or more values are missing. Within X_i, let Q_i be the set of non missing values. These values are denoted by x'_{iu}, $u = 1, ..., v$, and

$X_i' = (x_{i1}', ..., x_{iv}')^T$. If x_{ij} belongs to the set of missing values in X_i the row mean estimator for this value is then given by the average of X_i', that is

$$\hat{x}_{ij} = \frac{1}{v} \sum_{r=1}^{v} x_{iu}' \tag{4.2}$$

This estimator only includes the non missing values of the same protein were the value x_{ij} is missing.

4.1.2 k Nearest Neighbor Method

A more elaborate technique is given by using the k nearest neighbor method. This method was proposed by Troyanskaya et al. (2001) for the estimation of missing values in DNA microarray data. This technique makes use of the fact that some proteins have similar expression profiles due to similar biological functions. Therefore, available values from other proteins than the protein with the missing value are applied for the estimation. To specify what is meant by similar expression profile in the statistical or mathematical sense one can define distances between each pair of proteins. These distances can be derived by using the available values from the proteins of each pair. In the following, three distances between each pair $(X_i = (x_{i1}, ..., x_{in})^T, X_{i'} = (x_{i'1}, ..., x_{i'n})^T)$ of rows of X are defined. The Euclidean distance is given by

$$d_1(X_i, X_{i'}) = \tag{4.3}$$
$$= \sqrt{(x_{i1} - x_{i'1})^2 + (x_{i2} - x_{i'2})^2 + \ldots (x_{in} - x_{i'n})^2},$$

the Chebyshev distance is given by

$$d_2(X_i, X_{i'}) = sup|x_{ij} - x_{i'j}|, \tag{4.4}$$

with $j = 1, \ldots, n$, and the Mahalanobis distance is given by

$$d_3(X_i, X_{i'}) = \sqrt{(X_i - X_{i'})^T A^{-1} (X_i - X_{i'})}, \tag{4.5}$$

where A is the empirical covariance matrix of the n gels.

The principle of the k nearest neighbor method is now the following. For row X_i the k nearest neighbors are those rows of X with the k smallest distances to X_i. More details on the k nearest neighbor method can be found in Ripley (1996). This method was also used in nonparametric estimation of the density (see for example Rosenblatt, 1979) and regression (see for example Devroye, 1978) as well as in classification problems (see for example Ketskemety, 2004). With the above given notations missing values can be estimated as follows. Let X_i be the row where the value x_{ij} is missing. Let Q_i be the set of non missing values of X_i. We denote these values again by x'_{iu}, $u = 1, ..., v$, and $X'_i = (x'_{i1}, ..., x'_{iv})^T$. Let X_s, $s \neq i$, be the row s of the Matrix X. Suppose that x_{sj} is available and at least v other x_{su} are available, too, in the same columns as in X_i. One can then denote $X'_s = (x'_{s1}, ..., x'_{sv})^T$ and make the

Definition 4.1 X_i and X_s are neighbors if $d(X'_i, X'_s)$ is small.

and

Definition 4.2 The k rows X_s $(s \neq i)$ with the k smallest distances to X_i are the k nearest neighbors to X_i.

To estimate the missing value x_{ij} let $x_{s_1 j}, x_{s_2 j}, ..., x_{s_k j}$ be the x_{sj} such that X_s belongs to the k nearest neighbors of X_i. The missing value x_{ij} can now be estimated by the mean

$$\hat{x}_{ij}^{mean} = \frac{1}{k} \sum_{l=1}^{k} x_{s_l j}, \tag{4.6}$$

a weighted mean

$$\hat{x}_{ij}^{wmean} = \frac{1}{k} \sum_{l=1}^{k} w_{i s_l} x_{s_l j}, \tag{4.7}$$

with

$$w_{i s_l} = \frac{1}{d(X'_i, X'_{s_l}) \sum\limits_{l'=1}^{k} \frac{1}{d(X'_i, X'_{s_{l'}})}}, \tag{4.8}$$

or by the median

$$\hat{x}_{ij}^{median} = median(x_{s_1 j}, x_{s_2 j}, ..., x_{s_k j}). \tag{4.9}$$

4.1.3 Principal Component Regression

Another possibility of missing values estimation is given by using principal component (PC) regression. This approach reflects not only the relationship between pairs of proteins but between a greater set of proteins. Consider again to have the matrix X of expression values from r proteins in n samples. Furthermore, let $\tilde{X}$ be the $(r-1) \times n$ matrix X without row i and X_i the ith row. It is assumed that there is a biological relationship between all proteins that can be described by a linear model. That is the rows of $\tilde{X}$ can be seen as the independent variables and X_i as the dependent variable of the linear model

$$X_i = \tilde{X}^T \cdot b + e, \tag{4.10}$$

where the vector $b = (b_1, ..., b_{r-1})^T$ contains the regression coefficients and $e = (e_1, ..., e_n)^T$ is the error vector. The idea of estimating missing values by regression is to use the set Q_i of non missing values of X_i and the respective columns of $\tilde{X}$ to determine the regression coefficients b. The missing value x_{ij} can then be estimated by

$$\hat{x}_{ij} = \tilde{X}_j^T \cdot b \tag{4.11}$$

where $\tilde{X}_j$ is the jth column of $\tilde{X}$.

However, in the case that there are more variables than observations, there is an infinite number of solutions b which all fit equation 4.10. This is usually given for protein expression data from 2-D DIGE experiments, because the expression of hundreds or thousands of proteins are measured on only a few gels. Thus, a reduction of dimensionality is necessary. One possibility to reduce the dimension is to apply PC analysis (cf. Johnson and Wichern, 2002). In terms of the 2-D gel data, the idea of PC analysis is to extract uncorrelated principal components from the rows of $\tilde{X}$ by linear transformations

$$Y_i = a_{i,1}\tilde{X}_1^T + a_{i,2}\tilde{X}_2^T + ... + a_{i,r-1}\tilde{X}_{r-1}^T, \tag{4.12}$$

with $i = 1, ..., r-1$. The first principal component is then given by that Y_1 that maximises $Var(Y_1)$ subject to the constraint $a_1^T a_1 = 1$, where $a_1 = (a_{1,1}, a_{1,2}, ..., a_{1,r-1})^T$. The ith principal component is given by that linear combination Y_i that maximizes $Var(Y_i)$ subject to the constraint $a_i' a_i = 1$ and $Cov(Y_k, Y_i) = 0$ for $k < i$, where $a_i = (a_{i,1}, a_{i,2}, ..., a_{i,r-1})^T$. All linear combinations Y_i can now be summarised in a $(r-1) \times n$ matrix Y with the principal components in its rows.

These principal components can also be used as independent variables and X_i as dependent variable and one can again built a linear model:

$$X_i = Y^T \cdot b' + e'. \tag{4.13}$$

In this model there are still more variables than observations. But dimension can now be reduced by using only the first k principal components of $\tilde{X}$, that is the rows of Y, as independent variables, where $k \leq n$. Because n is mostly very small in 2-D gel experiments it is recommended to use $k = n$ principal components. If we denote $Y^{(k)}$ as the $k \times n$ matrix with the first $k = n$ principal components in its rows, model 4.13 becomes

$$X_i = Y^{(k)T} \cdot b'' + e''. \tag{4.14}$$

Using only the available values from X_i and the respective columns of $Y^{(k)}$ one can obtain an estimate $\hat{b}''$ for b''. The missing value x_{ij} can then be estimated by

$$\hat{x}_{ij} = Y_j^{(k)T} \cdot \hat{b}'', \tag{4.15}$$

where $Y_j^{(k)}$ is the jth columns of $Y^{(k)}$.

Another, even more robust, regression method which would also imply the covariance structure between X_i and $\tilde{X}$ is given by partial least squares (PLS) regression (cf. Geladi and Kowalski, 1986, Frank and Friedman, 1993, and Abdi, 2003). This method was also used for the estimation of missing data in DNA microarray experiments by Nguyen et al. (2004) and Brás and Menezes (2006).

4.1.4 Evaluation of Missing Values Estimation

The three methods (row mean, k nearest neighbor and PC regression) for the estimation of missing values were evaluated as follows. From a real data set of expression values from a 2-D DIGE experiment, the rows and columns with missing values were removed. Thus, a complete matrix A with $r = 526$ rows and $n = 5$ columns remained. From this data set, four incomplete data matrices B were generated, with 5, 10, 20 and 30 % of randomly chosen artificial missing values, respectively. The estimation methods were applied to each of these incomplete matrices and resulted in new, complete matrices C. To evaluate the results, each of the matrices C was compared to original matrix A by the normalised root mean square (RMS) error:

$$\textit{normalised RMS error (A, C)} = \frac{\sqrt{\sum_{i=1}^{r}\sum_{j=1}^{n}(A_{ij}-C_{ij})^2/(r*n)}}{\sum_{i=1}^{r}\sum_{j=1}^{n}A_{ij}/(r*n)}. \tag{4.16}$$

The normalised RMS error * 100 gives the average percentage deviation of the entries of C to the entries of A. The resulting errors according to each method and each portion of missing values are displayed in table 4.2. The PC regression was carried out by using $k = n$ principal components.

proportion of missing values	5%	10%	20%	30%
row mean method	0.13	0.19	0.26	0.32
$k$nn method	0.02	0.04	0.05	0.07
PC regression	0.10	0.07	0.11	0.34

Table 4.2: Normalised RMS error when applying the three methods to the incomplete data sets. The error for the k nearest neighbor method is the minimum that was achieved by this method.

The error for the k nearest neighbor method depends on the number k of neighbors, the distance measure between the protein pairs and the actual estimator. To compare the impacts of these parameters to the normalised RMS error one can plot a curve of this error in dependence of k and with different settings of estimators and distances. Figure 4.1a represents the normalised RMS error curves for different percentages of missing values. As distance measure the Euclidean distance and as

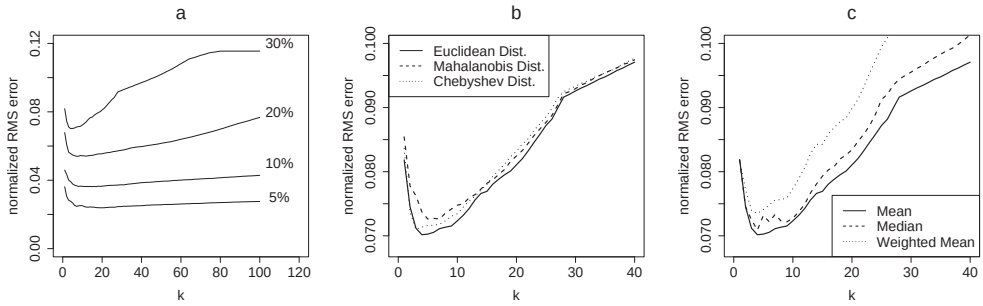


Figure 4.1: a) Normalized RMS errors in dependence of k. The $k$nn method applied a) to data with different proportions of missing values, b) with different distance measures and c) with different missing values estimators.

estimator the mean were used, here. One can see that error increases with increasing fractions of missing values. The curves have their minimum between five and twenty neighbors. The error curves for the different distance measures are displayed in figure 4.1b. Here, the k nearest neighbor method was applied to a data set with 30% of missing values and with the mean as estimator. The best performance is shown by the Euclidean distance followed by the Mahalanobis and the Chebyshev distance. However, the differences are not very big. Figure 4.1c finally shows the error curves for the different estimators, where the k nearest neighbor method was again applied with the Euclidean distance to the data set with 30% of missing values. As can be seen, the influence of the type of estimator isn't to big here, too. The best performance shows the mean. Similar results as in these three plots can also be observed for other combinations of estimators and distance measures. More error curves are displayed in appendix C.

4.1.5 Consideration of the Missing Values Problem in Sample Size Planning

The existence of missing values in 2-D gel data should also be considered in sample size planning for such experiments. As has been shown in section 4.1, the number of jointly existent protein spots on a series of replicated 2-D gels decreases with an increasing number of replicates. On the other hand, the power of statistical tests increases with an increasing number of replicates. Hence, there has always to be made a compromise between a desired statistical power and the number of proteins that remain for the analyses. The concrete planning should always be made in close cooperation of statisticians and biochemists.

4.2 Analysis of Time Dependent 2-D DIGE Data

A frequent problem in 2-D DIGE experiments is the comparison of the temporal courses of the protein expression in treated and untreated samples. In such experiments, protein expression is usually not measured at a great number of times but only at a few ones, say five to ten. Thus, the resulting data can be analysed by using analysis of variance (ANOVA) methods for longitudinal data. An outline of the design for a time dependent 2-D DIGE experiment is shown in table 4.3. It

	replication 1	replication 2	...	replication n
time 1	gel_{11}	gel_{12}	...	gel_{1n}
time 2	gel_{21}	gel_{22}	...	gel_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
time T	gel_{T1}	gel_{T2}	...	gel_{Tn}

Table 4.3: Design of a time dependent 2-D DIGE experiment with n gels at T times.

should be remarked, here, that n is the number of replications for each, treatment and control, because both are applied to the same gel in the 2-D DIGE technology.

In this section, a mixed linear model and respective F-tests for the detection of treatment/time-interactions and treatment effects in such experiments are proposed. Furthermore, F-tests for the analysis of times are presented.

4.2.1 A Mixed Model for Longitudinal Data

A method that reflects the concrete situation of a time dependent 2-D DIGE experiment like in table 4.3 is for example given in Diggle et al. (1994). In Jung et al. (2005) it was first proposed to apply this method when having such a situation. The analysis has to be done separately for each spot which has been detected on each of the $T \cdot n$ gels. To begin with, denote y_{gjt} as the standardised expression value for the current protein on the jth gel replication at the tth time and within the gth group, where $j = 1, ..., n$, $t = 1, ..., T$ and $g = 1, ..., G$. In a 2-D DIGE experiment, the number G of groups is usually 2, namely treatment and control. The design for such an experiment can be seen as a kind of split-plot design, with two main plots representing the two groups. The sub plots are the replications. Hence, this is also a hierarchical design, because each replication belongs either to the treatment or control group. Furthermore, the T levels of the time factor are not randomised to the replications in the usual sense, of course. Since the same protein is analysed over the time the model should heed the time-dependence of the measurements. This is given for the following model:

$$y_{gjt} = \beta_g + \gamma_{gt} + U_{gj} + Z_{gjt}, \tag{4.17}$$

where β_g is the main effect of the gth group, γ_{gt} is the interaction between group and time, $U_{gj} \sim N(0, \nu^2)$ is the random effect of the jth replication and $Z_{gjt} \sim N(0, \sigma^2)$ are the random errors. With these distribution assumptions for the random effects,

the vector $Y_{gj} = (Y_{gj1}, Y_{gj2}, ..., Y_{gjt})$ is normally distributed with covariance matrix

$$V = \sigma^2 \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix} + \nu^2 \begin{pmatrix} 1 & \dots & \dots & 1 \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 1 & \dots & \dots & 1 \end{pmatrix}, \tag{4.18}$$

where I and J are of size $t \times t$. That means that the correlation between two points in time is given by

$$\rho = \nu^2/(\nu^2 + \sigma^2). \tag{4.19}$$

The situation of equation (4.18) that the variances are equal and that all covariances are equal is called 'compound symmetry' (cf. Glantz and Slinker, 2000).

The most frequent question that occurs in 2-D DIGE experiments is whether there is an interaction between time and treatment in the protein expression. Such an interaction exists when the temporal courses of treatment and control are not parallel. A statistical test that examines this problem is given by testing with respect to model 4.17 the null hypothesis $\gamma_{gt} = \gamma_t$ for $g = 1, ..., G$ and for $t = 1, ..., T$. This null hypothesis reflects the situation that the mean response profiles of the groups are parallel. A test statistic for this hypothesis is given by

$$F_1 = \frac{ISS_2/[(G-1)(T-1)]}{RSS_2/[(G*n-G)(T-1)]}. \tag{4.20}$$

The ANOVA table 4.4 presents the sums of squares used in F_1. With the above satisfied assumptions for the error terms in model 4.17 and of compound symmetry, F_1 follows an F-distribution with $(G-1)\cdot(T-1)$ and $(G\cdot n-G)\cdot(T-1)$ degrees of freedom (cf. Crowder and Hand, 1995). A respective p-value for this test can be calculated by

$$p(F_1) = 1 - F_{(G-1)(T-1),(G\cdot n-G)(T-1)}(F_1), \tag{4.21}$$

where F is the distribution function of the F-distribution.

Another problem to be analysed in 2-D DIGE experiments is to test the null hypothesis that there is no treatment effect, i.e. testing $\beta_g = \beta$ for $g = 1, 2$, meaning that the mean temporal course for the expression of the current protein in the treated and untreated samples are on the same level. The respective F-statistic for testing this hypothesis is given by

$$F_2 = \frac{BTSS_1/(G-1)}{RSS_1/(G \cdot n - G)}. \tag{4.22}$$

Under the above assumptions, F_2 is F-distributed with $(G-1)$ and $(G \cdot n - G)$ degrees of freedom. The respective p-value for this test can be calculated by

$$p(F_2) = 1 - F_{(G-1),(G \cdot n - G)}(F_2). \tag{4.23}$$

The sums of squares of the F_2-statistic are also given in ANOVA table 4.4.

Within table 4.4 the following terms are defined as the mean of all observations in the experiment:

$$y_{...} = \sum_{g=1}^{G} \sum_{j=1}^{n} \sum_{t=1}^{T} y_{gjt}, \tag{4.24}$$

the mean of all observations in group g on gel j:

$$y_{gj\cdot} = \sum_{t=1}^{T} y_{gjt}, \tag{4.25}$$

the mean of all observations at time t:

$$y_{\cdot\cdot t} = \sum_{g=1}^{G} \sum_{j=1}^{n} y_{gjt}, \tag{4.26}$$

and the mean of all observations in group g at time t:

$$y_{g\cdot t} = \sum_{j=1}^{n} y_{gjt}. \tag{4.27}$$

4.2.2 Descriptive Analysis of Longitudinal Data

Besides the detection of treatment/time-interactions or treatment effects, it is usually of interest to describe the temporal courses of the expression of each protein more detailed by some more statistics. The above described tests only detect

source of variance	sums of squares	d.o.f.
between treatment	$BTSS_1 = T\sum_{g=1}^{G} n(y_{g\cdot\cdot} - y_{\dots})^2$	$G-1$
whole plot residual	$RSS_1 = TSS_1 - BTSS_1$	$G \cdot n - G$
whole plot total	$TSS_1 = T\sum_{g=1}^{G}\sum_{j=1}^{n}(y_{gj\cdot} - y_{\dots})^2$	$G \cdot m$
between time	$BTSS_2 = G \cdot n\sum_{t=1}^{T}(y_{\cdot\cdot t} - y_{\dots})^2$	$T-1$
treatment-time interaction	$ISS_2 = \sum_{t=1}^{T}\sum_{g=1}^{G} G \cdot n(y_{g\cdot t} - y_{\dots})^2$ $-BTSS_1 - BTSS_2$	$(G-1)\cdot$ $(T-1)$
split plot residual	$RSS_2 = TSS_2 - ISS_2$ $-BTSS_2 - TSS_1$	$(G \cdot m - 2)\cdot$ $(T-1)$
split plot total	$TSS_2 = \sum_{g=1}^{G}\sum_{j=1}^{n}\sum_{t=1}^{T}(y_{gjt} - y_{\dots})^2$	$G \cdot T \cdot n - 1$

Table 4.4: ANOVA table for the analysis of longitudinal data from 2-D DIGE experiments (d.o.f. = degrees of freedom).

whether there are effects. In order to get a more detailed impression of of the temporal courses of protein expression one can display for example the mean start level of a protein within group g:

$$y_{g\cdot 1} = \sum_{j=1}^{n} y_{gj1}, \tag{4.28}$$

or the respective mean end level:

$$y_{g\cdot T} = \sum_{j=1}^{n} y_{gjT}, \tag{4.29}$$

or their difference:

$$d(y_{g\cdot 1}, y_{g\cdot T}) = |y_{g\cdot 1} - y_{g\cdot T}|. \tag{4.30}$$

The graphical representation of longitudinal data can be done by plotting the protein expression versus time. Such a plot should include the single data points of each replication as well as the mean curves for each group.

4.2.3 Analysis of Single Times

With the model given in equation 4.17 it is also possible to explore treatment-effects at each single time t. The F-statistic for the null hypothesis that there is no treatment effect at the fix point in time t is given by

$$F_3 = \frac{BTSS/(G-1)}{RSS/(G \cdot n - G)}, \tag{4.31}$$

and follows also an F-distribution with $(G-1)$ and $(G \cdot n - G)$ degrees of freedom. The sums of squares are given in the ANOVA table 4.5. The respective p-value for

source of variance	sums of squares	d.o.f.
between treatment	$BTSS = \sum_{g=1}^{G} n(y_{g\cdot t} - y_{\cdot\cdot t})^2$	$G-1$
residual	$BTSS = TSS - BTSS$	$G-1$
total	$TSS = \sum_{g=1}^{G} \sum_{j=1}^{n} (y_{gjt} - y_{\cdot\cdot t})^2$	$G \cdot n - G$

Table 4.5: ANOVA table for the analysis of a single times.

this F-test can be calculated by

$$p(F_3) = 1 - F_{(G-1),(G\cdot n-G)}(F_3). \tag{4.32}$$

4.2.4 Example: Analysis of a Neuroblastoma Study

In this section the above ANOVA methods are applied to protein expression data from a proteome study of the neuroblastoma cell line SY5Y (cf. Sitek et al., 2005). Neuroblastoma are common solid tumors which occur in early childhood. The proteome of neuroblastoma depends on the activation of different neurotrophin receptors (TrkA and TrkB) by their ligands (cf. Nakagaware et al., 1994). Here, the proteome samples of the SY5Y cell line when the TrkA receptors are activated by

their ligand NGF (nerve growth factor) are compared to the case that the receptors are not activated. Protein expression was measured at five non-equidistant times, namely at 0, 0.5, 1, 6 and 24 hours after treatment. The number of replications was 4 for each, treatment and control. The data sets contained around 20 percent of missing values. Therefore, before doing the ANOVA tests, the missing values were estimated by the k nearest neighbor method with the Euclidean distance and the mean as estimator. The analysis was done for only those proteins for which at least three values were available, so that at most one missing value was estimated per protein. Thus, 440 spots remained for the analysis. Using the F_1-statistic and $\alpha = 0.05$ as significance level, seven protein spots with a significant treatment/time-interaction were identified (cf. table 4.6). Because the 440

rank	spot-no.	F_1	p-value	adj. p-values
1	910	9.6517	>0.0000	0.037
2	1136	4.9624	0.0046	1.000
3	941	3.6116	0.0192	1.000
4	1301	2.9517	0.0407	1.000
5	1166	2.8776	0.0444	1.000
6	2227	2.7896	0.0492	1.000
7	1787	2.7806	0.0497	1.000

Table 4.6: Proteins spots with a significant treatment/time-interaction ranked by their unadjusted p-values. The p-values adjustment in the last columns was done by the Bonferroni method.

tests were carried out simultaneously, here, an adjustment for multiple hypothesis testing is required. Some special characteristics of multiple hypothesis testing in 2-D DIGE experiments will be discussed in the next section. The mean temporal courses of the most significant spot, i.e. 910, are plotted in figure 4.2. As can be seen, the interaction takes place only within the first hour after treatment, here. The plots of the other significant spots are displayed in appendix D.

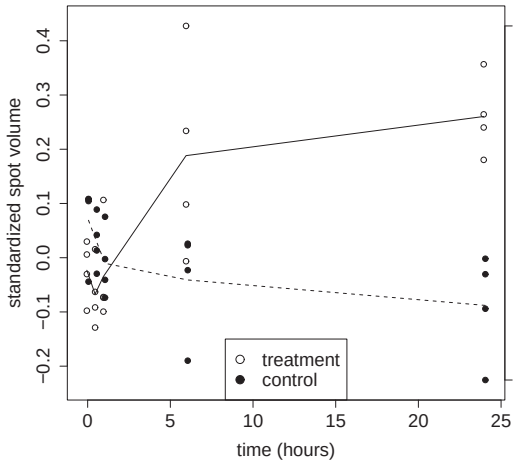


Figure 4.2: Mean temporal courses of protein 910 in the treatment group (solid line) and control group (dashed line), respectively. The single points show the measurements of the replications.

Next, treatment effects were discovered by using the F_2-statistic. This test resulted in 53 significant spots, the top 5 of them are displayed in table 4.7. The

rank	spot-no.	F_2	p-value	adj. p-values
1	2363	56.0883	0.0002	0.088
2	2502	45.4407	0.0005	0.220
3	935	43.0781	0.0005	0.220
4	1266	42.1088	0.0006	0.264
5	2123	28.2813	0.0017	0.748

Table 4.7: The five most significant spots of 53 spots with significant treatment effect, ranked by their unadjusted p-values.

mean expression profiles of the most significant protein, represented by spot 2363, are plotted in figure 4.3. The plot shows that after one hour, the mean expression profiles are nearly parallel, where the line for the treatment appears on a higher level.

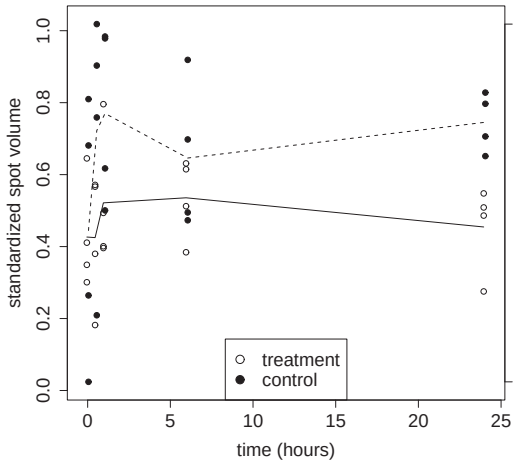


Figure 4.3: Mean temporal courses of spot 2363 in the treatment group (solid line) and control group (dashed line), respectively.

Additionally, the single times were analysed for treatment effects by using the F_3-statistic. The most significant protein spots are displayed in tables 4.8. Specifically, there were 20, 29, 9, 16 and 440 significant protein spots at the single points in time. How the Bonferroni adjustement reduces these numbers is discussed in the following section.

4.3 The Multiple Testing Problem in 2-D DIGE Experiments

Like in DNA microarray experiments, 2-D DIGE experiments also imply not only a single statistical test but hundreds or sometimes thousands of them simultaneously. Hence, it is necessary to adjust the testing levels with respect to some error rate like the *family-wise error rate* or the *false discovery rate* (compare section 3.1.1). Using the Bonferroni-adjustement for the F-tests in above the study of protein expression in neuroblastoma there remain only very few spots which are called significant. Specifically, only one spot remains with a significant

treatment/time-interaction and no spot remains with a treatment effect. The result of the Bonferroni-adjustment for the single points in time can be seen in tables 4.9.

However, multiple hypothesis testing shapes up as a general problem in 2-D DIGE experiments. Many of the detected gel spots don't represent proteins but background staining. Common image analysis software also regards them automatically as protein spots. Furthermore, some gel spots represent more than one protein. Not until after statistical analysis an experienced biochemist eyes up the significant spots on the gel and sorts them out if he assumes that they were only stains or represent more than a single protein. Thus, the set of spots to be analysed reduces, meaning that the parameter r gets smaller. Hence, the adjustment for multiple hypothesis testing has to be renewed. But even after such an readjustment the real number of actual proteins on the gel persists unclear. Therefore, it is recommended to always print both the unadjusted and the ajusted p-values in the results and to readjust as often as possible.

	time 1 (0 h)			time 2 (0.5 h)		
rank	spot-no.	F_3	p-value	spot-no.	F_3e	p-value
1	2502	17.8709	0.0055	3081	36.7425	0.0009
2	2568	17.6444	0.0056	641	20.8458	0.0038
3	2420	16.5422	0.0066	1702	19.1797	0.0047
4	1914	15.4450	0.0077	935	17.6755	0.0057
5	2123	14.0516	0.0095	2162	16.8531	0.0063

	time 3 (1 h)			time 4 (6 h)		
rank	spot-no.	F_3	p-value	spot-no.	F_3	p-value
1	1277	14.7812	0.0085	2577	25.0578	0.0024
2	1543	11.3998	0.0149	955	22.4686	0.0031
3	2007	9.8881	0.0200	1136	21.7405	0.0035
4	1054	9.8566	0.0201	941	16.2100	0.0069
5	1136	9.1546	0.0232	1850	13.5863	0.0103

	time 5 (24 h)		
rank	spot-no.	F_3	p-value
1	1136	110.2500	0.00004
2	1166	36.3110	0.0009
3	910	31.8153	0.0013
4	935	25.7956	0.0023
5	11125	23.9793	0.0027

Table 4.8: Most significant proteins at single times, ranked by their unadjusted p-values.

	time 1 (0 h)			time 2 (0.5 h)		
rank	spot-no.	F_3	adj. p-value	spot-no.	F_3e	adj. p-value
1	2502	17.8709	1.0000	3081	36.7425	0.4022
2	2568	17.6444	1.0000	641	20.8458	1.0000
3	2420	16.5422	1.0000	1702	19.1797	1.0000
4	1914	15.4450	1.0000	935	17.6755	1.0000
5	2123	14.0516	1.0000	2162	16.8531	1.0000

	time 3 (1 h)			time 4 (6 h)		
rank	spot-no.	F_3	adj. p-value	spot-no.	F_3	adj. p-value
1	1277	14.7812	1.0000	2577	25.0578	1.0000
2	1543	11.3998	1.0000	955	22.4686	1.0000
3	2007	9.8881	1.0000	1136	21.7405	1.0000
4	1054	9.8566	1.0000	941	16.2100	1.0000
5	1136	9.1546	1.0000	1850	13.5863	1.0000

	time 5 (24 h)		
rank	spot-no.	F_3	adj. p-value
1	1136	110.2500	0.0193
2	1166	36.3110	0.4149
3	910	31.8153	0.5854
4	935	25.7956	0.9979
5	11125	23.9793	1.0000

Table 4.9: Most significant proteins at the single times and their Bonferroni adjusted p-values.

5 Analysis of Protein Expression Data from MS Experiments

After regarding protein expression data from 2-D DIGE experiments in the previous chapter, this chapter now deals with protein expression data from mass spectrometry (MS) experiments. While the specific problem of protein expression data from 2-D DIGE experiments are missing values, data from MALDI-TOF mass spectrometry confronts statisticians with outliers. Such outliers occur in the repeated measurements of the a sample from the same patient. Up to now, practitioners of mass spectrometry detect outliers only by visual judgement, being quote time-consuming when having mass spectra from a lot of patients. Here, a statistical approach for the detection of outliers is proposed. This will be the topic of section 5.1.

Another characteristic of MS data is that in some experiments the number of proteins in the samples is considerable smaller than in samples used in experiments with 2-D gel electrophoresis or DNA microarrays. For example, in some MS experiments certain sets of proteins are preselected from the original sample by some kind of magnetic beads, yielding a new sample of only around 50 to 500 proteins. If it is now desired to detect differentially expressed proteins using the nonparametric approach of chapter 3.1, the set observations to use for the kernel density estimation is small, too. A proposal for the improvement of the kernel density estimation when having only a small set of observations will be discussed in section 5.2.

Finally, the methods for the detection of outliers and differentially expressed proteins are applied to MS data from a study of human thyroids in section 5.3.

5.1 Outliers in MS Data

In some MS experiments the sample from each patient is analysed repeatedly by mass spectrometry resulting in more than one spectrum for each sample. The number k of repetitions is usually not very big, for example 4-10. Let us index these repetitions by l with $l = 1, ..., k$. As is known by mass spectrometry practitioners some of the multiple measurements can results in 'bad' spectra. These bad spectra are either characterised by strong noise or they show changes in the relative intensity patterns of the peaks. Sorting these bad spectra out only by visual judgement (as it is common practice) is quite time-consuming and can be influenced by subjective criteria. A more objective way for detecting these bad spectra is to apply some standardised statistical method. Here, a method for the detection of multivariate outliers is adapted for the situation of repeated measurements in MS experiments to obtain a uniform criterion for the removal of bad spectra. This approach is based on a method that was proposed by Egan and Morgan (1998) for multivariate outlier detection in analytical chemical data. Their method was designed for the situation of having a large number of variables measured on a large number of individuals as well. In the situation of protein expression data from MS experiments, however, the sample from each individual is measured multiple times, as stated above. Therefore the method of Egan and Morgen (1998) will be modified in some parts.

The concrete situation to start with is given in table 5.1. For the sample from each of n patients k mass spectra are generated containing each the intensities of r proteins. Let us denote the $r \times k$ data matrix of patient j by $X^{(j)}$ with the peak intensities as entries, where $x_{il}^{(j)}$ is the intensity of the ith protein of the lth

	patient 1			...	patient n		
	spectr. 1	...	spectr. k	...	spectr. 1	...	spectr. k
protein 1	37.5	...	37.2	...	32.0	...	34.0
protein 2	11.1	...	10.8	...	12.2	...	12.4
⋮	⋮	⋱	⋮	⋱	⋮	⋱	⋮
protein r	23.4	...	21.9	...	22.9	...	24.5

Table 5.1: Protein expression data from MALDI-TOF MS with k spectra for each of n patients.

spectrum ($i = 1, ..., r$, $l = 1, ..., k$, $j = 1, ..., n$). For the data matrix $X^{(j)}$ of each patient one can derive the $k \times k$ distance matrix $D^{(j)}$ using for some multivariate distance measure, for example the Euclidean distance. The matrix $D^{(j)}$ consists then of all pairwise distances between the l spectra of patient j. Assume that half of the l spectra from each patient are 'good' ones. This assumption is confirmed by experienced mass spectrometry practitioners. Hence, one can use those $k/2$ spectra of a certain patient with the smallest distances among each other to calculate a robust multivariate mean

$$m^{(j)} = \frac{2}{k} \sum_{l \in \Lambda} X_l^{(j)}, \tag{5.1}$$

where Λ is the set of indexes l which belong to the $k/2$ closest spectra of patient j. Next, the Euclidean distances $d_l^{(j)}$ from all l spectra (from the jth patient) to $m^{(j)}$ are calculated. It is now of interest to get an impression of how big a distance of a certain spectrum to the robust centroid should be to call this particular spectrum an outlier. In order to answer this question we regard the distribution of all $k * n$ distances $d_l^{(j)}$, with $l = 1, ..., k$ and $j = 1, ..., n$. Now, one has to choose a threshold within this distribution of the distances where spectra with distances beyond this threshold are called outliers. This step should be treated very carefully, however. It is not advisable to just arbitrary call the spectra with the $\alpha\%$ biggest distances to be outliers. This threshold should be chosen in close cooperation with mass spectrometry practitioners. For each possible threshold they should reconcile the

set of outliers with the set of 'bad' spectra found by their own visual criteria. After outlier detection and removal one can summarise the spectra of each patient for example by their multivariate mean.

5.2 Detection of Differentially Expressed Proteins Using MS Data

As in 2-D DIGE experiments, the detection of differentially expressed proteins (or peptides) is also a subject in MS experiments. In general, this problem can be treated by using the nonparametric method for multiple hypothesis testing that was discussed in section 3.1. However, protein expression data from MS experiments often consists of values for only around 50-500 proteins or peptides, especially in those experiments, where certain proteins have been selected by magnetic beads (cf. Zhang et al., 2004 and Baumann et al., 2005). Thus, the kernel density estimation that is applied within the nonparametric method is based on only very few observations. According to Gannoun et al. (2004), it often occurs that the tails of a density are not well estimated by kernel estimation because of too few observations. They proposed therefore to use a reflection method, where $\beta * 100\%$ of artificial observations are added in the tails of the ordered list of the original observations. This reflection approach is executed as follows. Let $z_{(1)}, ..., z_{(B)}$ be the initial ordered data from which one wants to estimate the density function. Then, the artificial observations in the left and the right tail of the density are generated by

$$\tilde{z}_{(b+1)} = z_{(1)} - (z_{(b+1)} - z_{(1)}) \tag{5.2}$$

and

$$\hat{z}_{(b+1)} = z_{(B)} + (z_{(B)} - z_{(B-b)}), \tag{5.3}$$

respectively, where $b = 1, ..., [B\beta/2]$ and $[\eta]$ denotes the integer part of η. As remarked in Gannoun et al. (2004), making β very small, around 0.5%, suffices

when the number of original observations is large. They also mention that the rejection region of the nonparametric approach is very sensitive to the amount of artificial observations used in the kernel estimation.

In order to get a closer view to the effect of using this reflection approach in kernel density estimation, this method is applied to B random samples from a N(0,1)-distribution, here. The amount of artificial observations was set to 0, 0.5, 1, 5, 10 and 20%, respectively. The estimated densities $f_B(z)$ were then compared to the true density $f(z)$ of the N(0,1)-distribution by the integrated mean square error

$$IMSE(f_B(z), f(z)) = \int_{-\infty}^{\infty} E\{[f_B(z) - f(z)]^2\}dz \tag{5.4}$$

These errors are presented in table 5.2. One can see, that the error increases

	B			
β	50	100	500	1000
0.000	0.011	0.007	0.002	0.001
0.005	0.019	0.013	0.006	0.007
0.010	0.019	0.012	0.009	0.011
0.050	0.019	0.016	0.018	0.020
0.100	0.024	0.025	0.029	0.032
0.200	0.040	0.040	0.048	0.051

Table 5.2: Integrated mean square error of kernel density estimation when using a reflection approach to add $\beta*100\%$ of artificial observations to B original observations.

with an increasing amount of artificial observations. The effect of the reflection approach can also be seen in figures 5.1 and 5.2. The first figure shows the density estimation without reflection approach, the second one the case that 20% of artificial observations have been added before kernel estimation. From the second figure it can be seen that the reflection approach gives too much weight to the tails of the density. From this simulation it can be deduced that the reflection approach is not

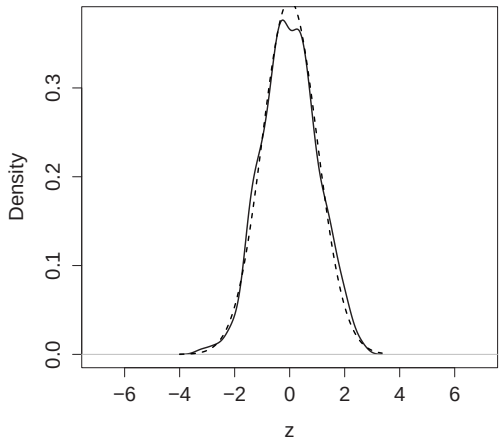


Figure 5.1: True (solid line) and estimated (dashed line) density functions without reflection approach.

necessary in the case of a standard normal distribution. In fact, the more artificial observations are added, the worse the estimation becomes. Thus, one can follow the recommendation given by Gannoun et al. (2004) to be careful with this reflection approach. It's effect on the set of differentially expressed proteins in an MS study is also focus of the following subsection.

5.3 Example: Analysis of a Thyroid Study

In this section, the above detailed methods are applied to protein expression data that was surveyed in an MS study of the protein expression in human thyroids. Altogether, there were 2827 spectra of 738 patients, where the sample of each patients was measured three or four times by MALDI-TOF mass spectrometry. The algorithm for outlier detection was applied to these data and the distances $d_l^{(j)}$ were calculated. The distribution of these distances is plotted in figure 5.3. In order to find an appropriate threshold for these distances, where spectra with distances greater than this threshold are seen as outliers, each possible threshold was compared with the set of outliers found by the visual inspections of the MS prac-

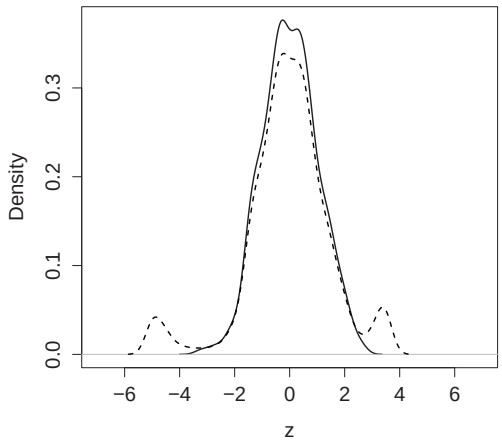


Figure 5.2: True (solid line) and estimated (dashed line) density functions with reflection approach.

titioners. This let to a threshold of 193683, here, meaning that around 20% of the spectra were determined as outliers. Among this set around 80% of the spectra agreed with the practitioners criteria to be an outlier. This outlier procedure was firstly carried out with the raw spectra and secondly with the preprocessed spectra. The results were nearly the same for both, meaning that the outlier detection can be done before data preprocessing. Thus, calculation time can be reduced.

After removing the outliers, the remaining spectra were used to find differentially expressed proteins using the nonparametric method in combination with the reflection approach for the density estimation. First, the spectra for each patient were summarised by their multivariate mean. Next, the spectra were divided by sex, where 300 patients were randomly chosen for each class, men and women. The number of proteins in this data set was very small, i.e. 37, due to selection of proteins by magnetic beads. With these data, the nonparametric method was carried out to find differentially expressed proteins. The alpha-level was 0.05, here. In order to see the effect of applying the reflection approach to the kernel estimation within this procedure, it was carried out with different percentages $\beta * 100$ of ar-

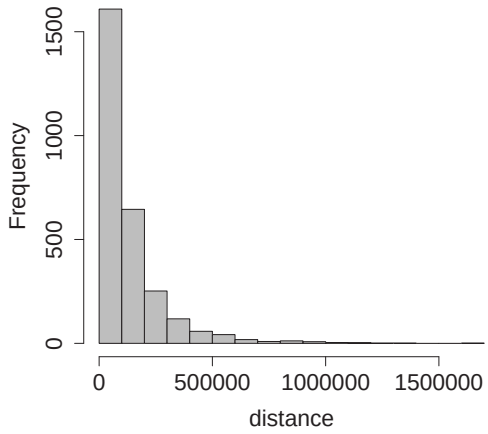


Figure 5.3: Histogram of the distances of the spectra to the mean of the spectra from the respective patient.

tificial observations. The number of differentially expressed proteins with respect to the amount of artificial observations in the kernel estimation can be found in table 5.3. As can be seen, the reflection approach gives strong weight to the tails of

β	rejection region	# diff. expr. proteins	# artificial obs.
0	[-2.41, 0.85]	14	0
0.1	[-2.50, 1.05]	14	2
0.15	[-2.61, 1.35]	13	4
0.2	[-2.76, 1.58]]	10	6

Table 5.3: Amount of artificial observations for density estimation and its influence on the number of differentially expressed proteins.

the density and expands thus rejection region. The consequence is that the number of differentially expressed proteins becomes smaller. As mentioned in the previous subsection, one can again deduce that the use of the reflection approach has to be considered very carefully.

6 Summary and Outlook

The demand for statistical analyses in bioanalytical research is high. Especially those experiments with high-throughput technologies which were developed or enhanced since the middle of the 1990s produce data sets of enormous size from which correct statistical inferences have to be made. Three of these high-throughput technologies are DNA microarrays, two-dimensional gel electrophoresis (2-DE) and several types of mass spectrometry (MS). In this work, some new approaches for the statistical analysis of gene and protein expression data from experiments with these technologies are presented and discussed.

First, in chapter 2, the principles of experiments with DNA mciroarrays, 2-DE and MS were explained as well as the resulting data structures. Especially, the statistical preprocessing, like signal calculation, normalisation, standardisation and calibration of such data were discussed. On the one hand, preprocessing is needed due to technical inaccuracies in the experiments, on the other hand it is necessary to bring the data in a form needed for the statistical methods for the actual analysis.

Chapter 3 focused on the analysis of gene expression data from DNA microarrays. In particular, a nonparametric method of multiple hypothesis testing for the detection of differentially expressed genes, developed by Pan et al. (2001) and Gannoun et al. (2004), was improved and the properties of this new version were discussed. These improvements consisted of making the kernel estimation more precise and of the presentation of a new fast algorithm for finding the cut-off point

for the likelihood ratio. Furthermore, it was shown how to calculate p-values for each gene when using this method. In addition a software implementation of this renewed method was introduced.
Compared with an alternative method for the detection of differentially expressed genes, a permutation test, the improved nonparametric method was shown to be very fast but to have less statistical power. It was also shown that the nonparametric method breaks down when there are no differentially expressed genes in the biological samples.

Next, in chapter 4, a complete strategy for the differential analysis of data from 2-D DIGE experiments was presented and evaluated. Originally the idea was to simply apply the nonparametric method for gene expression data to those 2-D DIGE data in order to find differentially expressed proteins. However, it was found that there are great amounts of missing values in these data sets. Therefore, several methods for the estimation of missing values in gene expression data were applied to 2-D DIGE data and evaluated by the normalised root mean square error. These methods were the row mean method, the k nearest neighbor method and principal component regression. It has been seen that the k nearest neighbor method performs best and is therefore recommended for further experiments. After estimation of missing values the whole range of multivariate methods which have already been used in DNA microarray experiments (e.g. clustering, classification, etc.) can be applied to protein expression data from 2-D DIGE experiments.
Oftentimes, 2-D DIGE experiments include measurements of protein expression over several times in order to find treatment effects or treatment/time-interactions. An analysis of variance model for longitudinal data, originally given in Diggle et al. (1994), was corrected and applied to such time-dependent DIGE data.
The special problem of multiple hypothesis testing in 2-D DIGE experiments was discussed and a recommendation of how to handle this problem was made.

Last, in chapter 5, the statistical analysis of protein expression data from MS experiments was discussed. In those experiments, the sample from each patient is measured multiple times by MS resulting in multiple mass spectra for each patient. However, MS practitioners expect some of these spectra to be outliers but they can detect them only by visual judgement. Therefore, a method for multivariate outlier detection, given in Egan and Morgen (1998), was modified to find outlier spectra in those multiple measurements.

Like with the protein expression data from 2-D DIGE experiments, it was originally intended to apply the nonparametric method for multiple hypothesis testing from chapter 3 to MS data, too. Such data consist oftentimes of values for only 50-500 proteins. This specific situation and its meaning for the kernel estimation within the nonparametric method was evaluated. It has been found that the reflection approach, proposed in Gannoun et al. (2004) for the generation of artificial observations as basis for the kernel estimation has to be handled carefully.

*

Genomics and proteomics are still in their infancy. New analytical technologies or enhancements of existing technologies will produce data sets of new shape and there will subsequently follow new statistical challenges.

Protein expression, for example, can also be measured by a combination of ICAT (isotope-coded affinity tags) with mass spectrometry (cf. Gygi et al., 1999). This technique labels the proteins from the different groups not by different dyes but by different specific masses. In addition, like DNA microarrays, protein arrays have been developed to measure protein expression, too (cf. Sydor and Nock, 2003). Statistical methods have to be evaluated how they fit with the respective resulting data structures.

Furthermore, statistical methods are also necessary for many other techniques of biochemical research. Proteins spots that have been analysed by gel electrophoresis need to be identified by mass spectrometry. Therefore, the masses of an analyte are compared with databases and the probabilities that a mass represents a certain pro-

tein is derived by statistical methods (cf. Nesvizhskii et al., 2002). Very important is also the detection of genetic networks or regulatory pathways (cf. Grzegorczyk and Urfer, 2004). A new approach to analyse the binding activity of interacting biomolecules by PLS regression is given in Kirschbaum and Urfer (2006).
A new biological area, called systems biology (cf. Spivey, 2004), intends to study not only the expression of genes or proteins but of all metabolites of an organism (cf. Nicholson and Wilson, 2003) and tries find the interactions between genomes, proteomes, metabolomes and signal pathways. It is therefore also of interest to statistically compare the expression of metabolites in different biological samples, like normal and cancerous tissues.

Concerning this thesis, a starting point for further research are classification problems in MS experiments. A great benefit of MS is that the analytes may also be several body fluids which are easy to collect, like serum, blood, urine or saliva instead of tissue samples (cf. Villanueva et al., 2004). Mass spectra of these analytes can be used for medical diagnosis. Statistical methods for the classification of new spectra to known disease classes are for example support vector machines, genetic algorithms (cf. Jeffries, 2004) and combinations of principal component analysis or partial least squares analysis with linear discriminant analysis (Boulesteix, 2004 and Lilien et al., 2003). Also the classification method by nearest shrunken centroids (Tibshirani et al., 2002) should be considered in such classification problems. In this context, an important question is, how many training samples should be used for these algorithms to improve the classification of spectra. Therefore, one can model the classification error in dependence of the training set size. Mukherjee et al. (2003) for example proposed to model the classification error e in dependence of the training set size n as

$$e(n) = a_1 + a_2 n^{-\alpha}, \tag{6.1}$$

where a_2 is the learning rate of the classification method, α is the decay rate and a_1 the minimum error rate that can be achieved. Based on observed classification

errors one can thus find the relation between training size and classification error and use it for prognoses of e. Of interest for classification problems in MS experiments may also be sequential approaches of sample size planning like presented in Fu et al. (2005).

Appendix

A: R-package 'degenes'

In this appendix the functions of the R-package 'degenes' is given (cf. Jung et al., 2006b).

'common.genes'

R-code:

```
common.genes <- function(file1, file2, h1 = 1, h2 = 1) {
    a <- read.table(file1, sep = "", row.names = NULL,
    header = FALSE, skip = h1)
    pvalues.a <- data.matrix(a[, seq(3, dim(a)[2], 2)])
    median.a <- rep(0, (dim(pvalues.a)[1]))
    median.a <- apply(pvalues.a, 1, median)
    subset.a <- which(median.a < 0.05)
    b <- read.table(file2, sep = "", row.names = NULL,
    header = FALSE, skip = h2)
    pvalues.b <- data.matrix(b[, seq(3, dim(b)[2], 2)])
    median.b <- rep(0, (dim(pvalues.b)[1]))
    median.b <- apply(pvalues.b, 1, median)
    subset.b <- which(median.b < 0.05)
    g <- sort(union(subset.a, subset.b))
    return(g)
}
```

Arguments:
`file1, file2`: Character string, framed by "", naming the source of the treatment and control data set, respectively.
`h1, h2`: Have to set 0 if there are no column names in the .txt-files, and 1 if there are column names. Default value is 1.

Value:
g: Vector of indices of those genes, for which the expression values were reliable.

'deg'

R-code:

```
deg <- function(treatment, control, ref = 0, alpha = 0.05) {
    n <- dim(treatment)[1]
    J1 <- dim(treatment)[2]
    J2 <- dim(control)[2]
    m1 <- J1/2
    m2 <- J2/2
    X1 <- (treatment[, 1:m1] - treatment[, (m1 + 1):J1])/2
    X2 <- (control[, 1:m2] - control[, (m2 + 1):J2])/2
    Z.big <- Z.calc(treatment, control, X1, X2, m1, m2)
    z.small <- z.calc(X1, X2, m1, m2)
    if(ref == 0) {
        new1 <- Z.big
        new2 <- z.small }
    if(ref != 0) {
    new1 <- reflection(sort(Z.big), ref)
    new2 <- reflection(sort(z.small), ref) }
    z <- seq(1.5 * min(min(new1), min(new2)), 1.5 *
    max(max(new1), max(new2)), 0.01)
    lz <- length(z)
    kern1 <- kern(new1, z, lz)
    kern2 <- kern(new2, z, lz)
    if (max(kern2) <= max(kern1)) {
        print("No differentially genes.", quote=FALSE) }
    if (max(kern2) > max(kern1)) {
        cat("Determination of the rejection region", fill=T)
        Tk.hat <- kern2[which(kern1 > 0)]/kern1[which(kern1 >
        0)]
        single.alpha <- alpha/n
        integral.factor <- 0.005
        help <- 0
        c0 <- 0
        c1 <- max(Tk.hat)
        repeat {
            c <- (c1 + c0)/2
            Ac.hat <- which(Tk.hat < c)
```

```
            Ac.hat.compl <- which(Tk.hat >= c)
            integral <- 0
            integral <- (kern2[min(Ac.hat)] +
            sum(kern2[(min(Ac.hat) + 1):(min(Ac.hat.compl) - 2)]
            * 2) + kern2[min(Ac.hat.compl) - 1])
            * integral.factor
            integral <- integral + (kern2[max(Ac.hat.compl) + 1]
            + sum(kern2[(max(Ac.hat.compl) + 2):(max(Ac.hat) -
            1)] * 2) + kern2[max(Ac.hat)]) * integral.factor
            if(integral > single.alpha)
            c1 <- c
            if(integral < single.alpha)
            c0 <- c
            if(abs(integral - single.alpha) <= 1e-008) break
            if(c <= 2e-006) break
            if(help > 100) break
            help <- help + 1 }
        z <- z[which(kern1 > 0)]
        f <- which(Tk.hat >= c)
        region <- c(z[min(f)], z[max(f)])
        a <- which(Z.big < region[1])
        b <- which(Z.big > region[2])
        values <- sort(c(a, b))
        output <- list(values=values, region=region,
        single.alpha=single.alpha)
        cat(date(), fill=T)
        return(output) }
}
```

Arguments:
`treatment, control`: Data sets of gene expression.
`ref`: Parameter ≥ 0 giving the percentage of artificial observations to be added for before kernel estimation. Default value is 0.
`alpha`: Specification of global testing level. Default value is 0.05.

Value:
A list with following components:
`single.alpha`: Adjusted alpha level.
`region`: Rejection region for Z statistics.
`values`: Indexes of differentially expressed genes.

'kern'

R-code:

```
\begin{verbatim}
kern <- function(a, z, lz) {
    n2 <- length(a)
    band1 <- (sqrt(var(a)) * (n2^(-1/5))) * 1.144
    kernx <- rep(0, lz)
    for(k in 1:lz) {
        zk <- rep(z[k], n2)
        kernxx <- dnorm(((zk - a)/band1), 0, 1)
        kernx[k] <- sum(kernxx)/(n2 * band1) }
    return(kernx)
}
```

Arguments:
`a`: Vector of observations from the density to be estimated.
`z`: Vector of data points were the density is estimated.
`lz`: Length of z.

Value:
`kernx`: Vector with values of estimated density.

'power.plot'

R-code:

```
power.plot <- function(treatment, control, ref = 0, alpha
= 0.05) {
    n <- dim(treatment)[1]
    J1 <- dim(treatment)[2]
    J2 <- dim(control)[2]
    m1 <- J1/2
    m2 <- J2/2
    X1 <- (treatment[, 1:m1] - treatment[, (m1 + 1):J1])/2
    X2 <- (control[, 1:m2] - control[, (m2 + 1):J2])/2
    Z.big <- Z.calc(treatment, control, X1, X2, m1, m2)
    z.small <- z.calc(X1, X2, m1, m2)
    if(ref == 0) {
        new1 <- Z.big
        new2 <- z.small }
    if(ref != 0) {
        new1 <- reflection(sort(Z.big), ref)
        new2 <- reflection(sort(z.small), ref) }
    z <- seq(1.5 * min(min(new1), min(new2)), 1.5 *
```

```
max(max(new1), max(new2)), 0.01)
lz <- length(z)
kern1 <- kern(new1, z, lz)
kern2 <- kern(new2, z, lz)
Tk.hat <- kern2[which(kern1 > 0)]/kern1[which(kern1 > 0)]
single.alpha <- alpha/n
integral.factor <- 0.005
help <- 0
c0 <- 0
c1 <- max(Tk.hat)
repeat {
    c <- (c1 + c0)/2
    Ac.hat <- which(Tk.hat < c)
    Ac.hat.compl <- which(Tk.hat >= c)
    integral <- 0
    integral <- (kern2[min(Ac.hat)] + sum(kern2[(min(Ac.hat)
    + 1):(min(Ac.hat.compl) - 2)] * 2) +
    kern2[min(Ac.hat.compl) - 1]) * integral.factor
    integral <- integral + (kern2[max(Ac.hat.compl) + 1] +
    sum(kern2[(max(Ac.hat.compl) + 2):(max(Ac.hat) - 1)] *
    2) + kern2[max(Ac.hat)]) * integral.factor
    if(integral > single.alpha)
    c1 <- c
    if(integral < single.alpha)
    c0 <- c
    if(abs(integral - single.alpha) <= 1e-008) break
    if(c <= 2e-006) break
    if(help > 100) break
    help <- help + 1 }
z <- z[which(kern1 > 0)]
f <- which(Tk.hat >= c)
region <- c(z[min(f)], z[max(f)])
limit1 <- rep(0, 100)
limit2 <- rep(0, 100)
power <- rep(0, 100)
d <- seq(min(z)-region[1], max(z)- region[2], length=100)
for(i in 1:100) {
    limit1[i] <- d[i] + region[1]
    limit2[i] <- d[i] + region[2]
    power[i] <- 0 }
for(i in 1:100) {
    lim1 <- max(which(z <= limit1[i]))
    lim2 <- min(which(z >= limit2[i]))
    for(j in 1:lim1) {
```

```
            power[i] <- power[i] + ((kern1[j] + kern1[j + 1])
            * integral.factor) }
        for(l in lim2:(length(z) - 1)) {
            power[i] <- power[i] + ((kern2[l] + kern2[l + 1])
            * integral.factor) }}
    plot(d, power, xlab = "expression change d", ylab = "power",
    type = "l")
}
```

Arguments:
`treatment, control`: Data sets of gene expression.
`ref`: Parameter ≥ 0 giving the percentage of artificial observations to be added for before kernel estimation. Default value is 0.
`alpha`: Specification of global testing level. Default value is 0.05.

'pvalue'

R-code:

```
pvalue <- function(treatment, control, ref = 0, alpha = 0.05) {
    n <- dim(treatment)[1]
    J1 <- dim(treatment)[2]
    J2 <- dim(control)[2]
    m1 <- J1/2
    m2 <- J2/2
    X1 <- (treatment[, 1:m1] - treatment[, (m1 + 1):J1])/2
    X2 <- (control[, 1:m2] - control[, (m2 + 1):J2])/2
    Z.big <- Z.calc(treatment, control, X1, X2, m1, m2)
    z.small <- z.calc(X1, X2, m1, m2)
    if(ref == 0) {
        new1 <- Z.big
        new2 <- z.small }
    if(ref != 0) {
        new1 <- reflection(sort(Z.big), ref)
        new2 <- reflection(sort(z.small), ref) }
    z <- seq(1.5 * min(min(new1), min(new2)), 1.5
    * max(max(new1), max(new2)), 0.01)
    lz <- length(z)
    integral.factor <- 0.005
    kern1 <- kern(new1, z, lz)
    kern2 <- kern(new2, z, lz)
    if (max(kern2) <= max(kern1)) {
        print("No differentially genes.", quote=FALSE) }
    if (max(kern2) > max(kern1)) {
```

```
        p.value <- rep(0, n)
        c <- 0
        n2 <- length(new1)
        band1 <- (sqrt(var(new1)) * (n2^(-1/5))) * 1.144
        band2 <- (sqrt(var(new2)) * (n2^(-1/5))) * 1.144
        Tk.hat <- kern2[which(kern1 > 0)]/kern1[which(kern1 >
        0)]
        for(i in 1:n) {
            zkk <- rep(Z.big[i], n)
            pkern <- dnorm(((zkk - new1)/band1), 0, 1)
            pkernel <- sum(pkern)/(n2 * band1)
            zkk <- rep(Z.big[i], n)
            pkern <- dnorm(((zkk - new2)/band2), 0, 1)
            pkernel2 <- sum(pkern)/(n2 * band2)
            if (i%%1000==0) cat("number of estimated p-values:",
            i, fill=T)
            c <- pkernel2/pkernel
            Ac.hat <- which(Tk.hat < c)
            Ac.hat.compl <- which(Tk.hat >= c)
            if(c <= min(Tk.hat)) p.value[i] <- 0
            if(c >= max(Tk.hat)) p.value[i] <- 1
            if(c > min(Tk.hat) && c < max(Tk.hat)) {
            integral <- (kern2[min(Ac.hat)] +
            sum(kern2[(min(Ac.hat) + 1):(min(Ac.hat.compl) - 2)]
            * 2) + kern2[min(Ac.hat.compl) - 1])
            * integral.factor
            integral <- integral + (kern2[max(Ac.hat.compl) + 1]
            + sum(kern2[(max(Ac.hat.compl) + 2):(max(Ac.hat) -
            1)] * 2) + kern2[max(Ac.hat)]) * integral.factor
            p.value[i] <- integral }}
        unadjusted <- p.value
        adjusted <- rep(0, n)
        for (i in 1:n) {
            adjusted[i] <- min(unadjusted[i]*n, 1) }
        output <- list(unadjusted = unadjusted, adjusted = adjusted)
        return(output)
}}
```

Arguments:
`treatment, control`: Data sets of gene expression.
`ref`: Parameter ≥ 0 giving the percentage of artificial observations to be added for before kernel estimation. Default value is 0.
`alpha`: Specification of global testing level. Default value is 0.05.

Value:
A list with following components:
`unadjusted`: Vector of unadjusted p-values.
`adjusted`: Vector of adjusted p-values.

'read.values'

R-code:

```
read.values <- function(file, g, h = 1, pval = FALSE) {
    a <- read.table(file, sep = "", row.names = NULL, header =
    FALSE, skip = h)
    b <- 1
    if (pval==TRUE) b <- 2
    expr.values <- data.matrix(a[g, seq(2, dim(a)[2] - 1, b)])
    return(expr.values)
}
```

Arguments:
`file`: Character string, framed by "", naming the source of the treatment or control data set.
`h`: Has to set 0 if there are no column names in file, and 1 if there are column names. Default value is 1.
`g`: Vector, returned from the function 'common.genes'.
`pval`: A logical value indicating whether there are detection p-values in the data set or not. Default is FALSE.

Value:
`expr.values`: Matrix, containing expression values, with the genes in the rows and the arrays in the columns.

'reflection'

R-code:

```
reflection <- function(a, ref) {
    long <- (ref * n) %/% 2
    one <- rep(a[1], long)
    two <- a[2:(long+1)]
    left <- one - (two - one)
    one <- rep(a[n], long)
    two <- a[(n-1):(n-long)]
    right <- one + (one - two)
```

```
    return(c(left, a, right))
}
```

Arguments:
`a`: Vector with original values to be used for kernel density estimation.
`ref`: Percentage of artificial observations to be calculated before kernel density estimation.

Value:
Vector, containing the original and the artificial observations.

'z.calc'

R-code:

```
 z.calc <- function(a, b, m1, m2) {
    cat("Calculation of z", fill=T)
    z <- ((apply(a, 1, sum)/m1) - (apply(b, 1, sum)/m2))
    / sqrt ((apply(a, 1, var)/m1) + (apply(b, 1, var)/m1))
    return(z)
}
```

Describtion:
Function, that is called by the function 'deg' to calculate the z-statistics.

'Z.calc'

R-code:

```
Z.calc <- function(a, b, c, d, m1, m2) {
    cat("Calculation of Z", fill=T)
    Z <- (apply(a, 1, mean) - apply(b, 1, mean)) /
    sqrt ((apply(c, 1, var)/m1) + (apply(d, 1, var)/m1))
    return(Z)
}
```

Describtion:
Function, that is called by the function 'deg' to calculate the Z-statistics.

'zmk'

R-code:

```
zmk <- function(k = 2, treatment, control, ref = 0, alpha =
0.05) {
    n <- dim(treatment)[1]
    J1 <- dim(treatment)[2]
    J2 <- dim(control)[2]
    m1 <- J1/2
    m2 <- J2/2
    X1 <- (treatment[, 1:m1] - treatment[, (m1 + 1):J1])/2
    X2 <- (control[, 1:m2] - control[, (m2 + 1):J2])/2
    Z.big <- Z.calc(treatment, control, X1, X2, m1, m2)
    z.small <- z.calc(X1, X2, m1, m2)
    zmk <- rep(0, n)
    e <- 0
    for(i in 1:(dim(treatment)[1])) {
        e <- 0
        e <- round(runif(k, 1, (dim(treatment)[1])))
        e <- z.small[e]
        zmk[i] <- sum(e)/k }
    if(ref == 0) {
        new1 <- Z.big
        new2 <- zmk }
    if(ref != 0) {
        new1 <- reflection(sort(Z.big), ref)
        new2 <- reflection(sort(z.small), ref) }
    z <- seq(1.5 * min(min(new1), min(new2)), 1.5 *
    max(max(new1), max(new2)), 0.01)
    lz <- length(z)
    kern1 <- kern(new1, z, lz)
    kern2 <- kern(new2, z, lz)
    Tk.hat <- kern2[which(kern1 > 0)]/kern1[which(kern1 > 0)]
    single.alpha <- alpha/n
    integral.factor <- 0.005
    help <- 0
    c0 <- 0
    c1 <- max(Tk.hat)
    repeat {
        c <- (c1 + c0)/2
        Ac.hat <- which(Tk.hat < c)
        Ac.hat.compl <- which(Tk.hat >= c)
        integral <- 0
        integral <- (kern2[min(Ac.hat)] + sum(kern2[(min(Ac.hat)
        + 1):(min(Ac.hat.compl) - 2)] * 2) +
        kern2[min(Ac.hat.compl) - 1]) * integral.factor
        integral <- integral + (kern2[max(Ac.hat.compl) + 1] +
```

```
        sum(kern2[(max(Ac.hat.compl) + 2):(max(Ac.hat) - 1)] *
        2) + kern2[max(Ac.hat)]) * integral.factor
        if(integral > single.alpha)
        c1 <- c
        if(integral < single.alpha)
        c0 <- c
        if(abs(integral - single.alpha) <= 1e-008) break
        if(c <= 2e-006) break
        if(help > 100) break
        help <- help + 1 }}
    z <- z[which(kern1 > 0)]
    f <- which(Tk.hat >= c)
    region <- c(z[min(f)], z[max(f)])
    limit1 <- rep(0, 100)
    limit2 <- rep(0, 100)
    power <- rep(0, 100)
    d <- seq(min(z)-region[1], max(z)- region[2], length=100)
    for(i in 1:100) {
        limit1[i] <- d[i] + region[1]
        limit2[i] <- d[i] + region[2]
        power[i] <- 0 }
    for(i in 1:100) {
        lim1 <- max(which(z <= limit1[i]))
        lim2 <- min(which(z >= limit2[i]))
        for(j in 1:lim1) {
            power[i] <- power[i] + ((kern1[j] + kern1[j + 1])
            * integral.factor) }
        for(l in lim2:(length(z) - 1)) {
            power[i] <- power[i] + ((kern2[l] + kern2[l + 1])
            * integral.factor)}}
    plot(d, power, xlab = "expression change d", ylab = "power",
    type = "l")}
```

Arguments:
`treatment, control`: Data sets of gene expression.
`ref`: Parameter ≥ 0 giving the percentage of artificial observations to be added for before kernel estimation. Default value is 0.
`alpha`: Specification of global testing level. Default value is 0.05.
`k`: Factor, specifying the number of replicates for the power calculation by k*dim(treatment)[2].

B: Permutation Algorithm

In this appendix the permutation algorithm for adjusted p-values (c.f. Dudoit et al. (2003) used in section 3.3.1 and it's R-implementation are given.

Permutation Algorithm:

For the bth permutation, $b = 1, ..., B$:

1. Permute the m columns of the data matrix X.

2. Compute realisations of test statistics

$$t_{i,b},$$

   for $i = 1, ..., n$.

3. Next, compute succesive maxima of the test statistics

$$u_{mb} = |t_{r_n,b}|,$$

   and

$$u_{ib} = \max(u_{i+1,b}, |t_{r_i,b}|),$$

   for $i = n-1, ..., 1$, where r_i are such that $|t_{r_1}| \geq |t_{r_2}| \geq ... \geq |t_{r_n}|$ for the *original* data.

4. The permutation adjusted p-values are

$$p_{r_i} = \frac{\sum_{b=1}^{B} I(u_{i,b} \geq |t_{r_i}|)}{B}.$$

R-Implementation:

```
X = cbind(treatment, control)
n = dim(X)[1]
m = dim(X)[2]
```

```
B = 500
perm.matrix = matrix(NA, ncol=m, nrow=B)
for (b in 1:B) {
   perm.matrix[b,] = sample(1:m, m, replace=FALSE)
}
t.i = rep(0, n)
for (i in 1:n) {
   t.i[i] = (mean(treatment[i,]) - mean(control[i,]))
   / sqrt(var(treatment[i,])/m1 + var(control[i,])/m2)
}
t.ib = matrix(0, ncol=B, nrow=n)
for (b in 1:B) {
   perm.b = X[,perm.matrix[b,]]
   for (i in 1:n) {
     t.ib[i,b] = (mean(perm.b[i,1:m1])
     - mean(perm.b[i,((m1+1):(m1+m2))]))
     / sqrt(var(perm.b[i,1:m1])/m1
     + var(permu.b[i,((m1+1):(m1+m2))])/m2)
}}
u.mb = t.ib[order(t.i),]
for (b in 1:B) {
   u.mb[n,b] = abs(u.mb[n,b])
   for (i in (n-1):1) {
     u.mb[i,b] = max(u.mb[i+1,b], abs(u.mb[i,b]))
}}
t.ri = t.i[order(t.i)]
p.i = rep(0, n)
for (i in 1:n) {
   p.i[i] = length(which(u.mb[i,]>=abs(t.ri[i]))) / B}
```

C: List of Differentially Expressed Genes

In this appendix the differentially expressed genes from the example in section 3.4 are given. The four columns represent the ranks, the row numbers in the data set, the Affymetrix' names and the adjusted p-values.

rank	row-no.	name	p-value	rank	row-no.	name	p-value
1	17678	217773_s_at	> 0	26	8901	208866_at	$1.7*10^{-62}$
2	18389	218484_at	> 0	27	11986	212060_at	$9.0*10^{-62}$
3	19262	219358_s_at	$5.3*10^{-260}$	28	13453	213532_at	$3.7*10^{-61}$
4	3108	203039_s_at	$8.0*10^{-257}$	29	12279	212354_at	$3.8*10^{-59}$
5	17888	217983_s_at	$3.6*10^{-211}$	30	1357	201288_at	$1.6*10^{-57}$
6	17889	217984_at	$2.1*10^{-204}$	31	1733	201664_at	$2.9*10^{-57}$
7	1247	201178_at	$1.8*10^{-171}$	32	3741	203674_at	$1.3*10^{-56}$
8	1047	200978_at	$6.1*10^{-160}$	33	21026	221123_x_at	$1.3*10^{-56}$
9	973	200904_at	$3.4*10^{-131}$	34	11907	211980_at	$5.5*10^{-56}$
10	11613	211671_s_at	$2.3*10^{-126}$	35	10522	210512_s_at	$1.1*10^{-55}$
11	2286	202217_at	$1.3*10^{-110}$	36	17779	217874_at	$1.1*10^{-54}$
12	2394	202325_s_at	$5.0*10^{-109}$	37	19038	219134_at	$1.7*10^{-53}$
13	9247	209213_at	$3.9*10^{-108}$	38	11967	212041_at	$7.4*10^{-50}$
14	8435	208394_x_at	$3.4*10^{-106}$	39	2277	202208_s_at	$9.6*10^{-48}$
15	108	31845_at	$6.1*10^{-106}$	40	5592	205525_at	$1.2*10^{-46}$
16	1934	201865_x_at	$3.4*10^{-104}$	41	3003	202934_at	$4.6*10^{-46}$
17	9506	209473_at	$3.7*10^{-103}$	42	8747	208712_at	$4.6*10^{-46}$
18	1972	201903_at	$1.8*10^{-99}$	43	13115	213193_x_at	$8.8*10^{-46}$
19	4722	204655_at	$3.2*10^{-93}$	44	10902	210915_x_at	$1.7*10^{-45}$
20	2873	202804_at	$1.4*10^{-91}$	45	21878	221978_at	$7.0*10^{-45}$
21	9105	209071_s_at	$3.6*10^{-89}$	46	11917	211990_at	$7.8*10^{-42}$
22	2891	202822_at	$6.6*10^{-85}$	47	5550	205483_s_at	$5.1*10^{-41}$
23	18258	218353_at	$4.7*10^{-75}$	48	3792	203725_at	$1.7*10^{-40}$
24	1621	201552_at	$8.8*10^{-64}$	49	17361	217456_x_at	$5.3*10^{-40}$
25	16149	216237_s_at	$8.8*10^{-64}$	50	13270	213349_at	$2.*10^{-39}$

rank	row-no.	name	p-value	rank	row-no.	name	p-value
51	21106	221203_s_at	$2.0*10^{-38}$	81	17989	218084_x_at	$3.9*10^{-26}$
52	5639	205572_at	$2.2*10^{-37}$	82	20946	221042_s_at	$7.2*10^{-26}$
53	8930	208895_s_at	3.9*10−37	83	211	40420_at	$4.7*10^{-24}$
54	9121	209087_x_at	$7.0*10^{-37}$	84	10119	210095_s_at	$8.5*10^{-24}$
55	155	35666_at	$4.0*10^{-36}$	85	1048	200979_at	$1.9*10^{-23}$
56	1960	201891_s_at	$1.4*10^{-35}$	86	1809	201740_at	$4.8*10^{-23}$
57	11901	211974_x_at	$3.9*10^{-34}$	87	17872	217967_s_at	$7.6*10^{-23}$
58	293	1405_i_at	$1.2*10^{-33}$	88	21172	221269_s_at	$1.9*10^{-22}$
59	2925	202856_s_at	$1.2*10^{-33}$	89	734	200665_s_at	$2.9*10^{-22}$
60	13048	213125_at	$2.1*10^{-33}$	90	91	41037_at	$4.6*10^{-22}$
61	18822	218918_at	$7.1*10^{-33}$	91	19796	219892_at	$7.2*10^{-22}$
62	2023	201954_at	$9.8*10^{-32}$	92	12278	212353_at	$1.8*10^{-21}$
63	8864	208829_at	$9.8*10^{-32}$	93	10973	210986_s_at	$2.7*10^{-21}$
64	199	38671_at	$1.1*10^{-31}$	94	11322	211366_x_at	$6.5*10^{-21}$
65	12622	212697_at	$2.9*10^{-31}$	95	3399	203332_s_at	$1.4*10^{-20}$
66	21832	221932_s_at	$2.9*10^{-31}$	96	2271	202202_s_at	$1.6*10^{-20}$
67	17750	217845_x_at	$2.4*10^{-30}$	97	17930	218025_s_at	$3.4*10^{-20}$
68	4362	204295_at	$1.2*10^{-29}$	98	2572	202503_s_at	$3.7*10^{-20}$
69	368	59625_at	$1.3*10^{-29}$	99	9798	209770_at	$3.7*10^{-20}$
70	473	60471_at	$2.2*10^{-29}$	100	21629	221729_at	$1.8*10^{-19}$
71	11298	211340_s_at	$6.3*10^{-29}$	101	13878	213959_s_at	$2.0*10^{-19}$
72	2956	202887_s_at	$1.6*10^{-28}$	102	18043	218138_at	$2.0*10^{-19}$
73	974	200905_x_at	$2.6*10^{-28}$	103	1688	201619_at	$4.6*10^{-19}$
74	2017	201948_at	$4.4*10^{-28}$	104	19186	219282_s_at	$4.6*10^{-19}$
75	1361	201292_at	$3.3*10^{-27}$	105	17638	217733_s_at	$6.4*10^{-19}$
76	20136	220232_at	$3.3*10^{-27}$	106	5017	204950_at	$1.6*10^{-18}$
77	21491	221589_s_at	$3.3*10^{-27}$	107	11516	211571_s_at	$2.4*10^{-18}$
78	14659	214743_at	$5.4*10^{-27}$	108	9471	209438_at	$4.9*10^{-18}$
79	2728	202659_at	$2.4*10^{-26}$	109	1081	201012_at	$5.3*10^{-18}$
80	21796	221896_s_at	$2.4*10^{-26}$	110	6858	206792_x_at	$7.3*10^{-18}$

rank	row-no.	name	*p*-value	rank	row-no.	name	*p*-value
111	12521	212596_s_at	$1.6*10^{-17}$	141	9015	208981_at	$8.4*10^{-12}$
112	8944	208909_at	$2.4*10^{-17}$	142	4786	204719_at	$1.2*10^{-11}$
113	8883	208848_at	$2.6*10^{-17}$	143	5450	205383_s_at	$1.2*10^{-11}$
114	3154	203085_s_at	$5.6*10^{-17}$	144	12597	212672_at	$1.2*10^{-11}$
115	6732	206666_at	$5.7*10^{-17}$	145	16938	217028_at	$1.7*10^{-11}$
116	3717	203650_at	$1.2*10^{-16}$	146	22215	222316_at	$5.9*10^{-11}$
117	13458	213537_at	$1.2*10^{-16}$	147	17596	217691_x_at	$7.5*10^{-11}$
118	2049	201980_s_at	$2.7*10^{-16}$	148	13279	213358_at	$1.0*10^{-10}$
119	2179	202110_at	$2.7*10^{-16}$	149	1458	201389_at	$2.0*10^{-10}$
120	11839	211911_x_at	$2.7*10^{-16}$	150	12127	212201_at	$2.5*10^{-10}$
121	1195	201126_s_at	$3.6*10^{-16}$	151	14000	214081_at	$2.5*10^{-10}$
122	2817	202748_at	$3.9*10^{-16}$	152	11142	211160_x_at	$2.7*10^{-10}$
123	4145	204078_at	$5.7*10^{-16}$	153	2924	202855_s_at	$3.6*10^{-10}$
124	19410	219506_at	$1.1*10^{-15}$	154	12389	212464_s_at	$4.5*10^{-10}$
125	21770	221870_at	$1.8*10^{-15}$	155	472	59999_at	$8.0*10^{-10}$
126	9790	209762_x_at	$1.1*10^{-14}$	156	1632	201563_at	$8.0*10^{-10}$
127	11660	211719_x_at	$2.1*10^{-14}$	157	5266	205199_at	$8.0*10^{-10}$
128	13604	213684_s_at	$2.1*10^{-14}$	158	9104	209070_s_at	$8.0*10^{-10}$
129	10864	210869_s_at	$2.9*10^{-14}$	159	11847	211919_s_at	$1.1*10^{-9}$
130	5119	205052_at	$5.9*10^{-14}$	160	138	34210_at	$1.4*10^{-9}$
131	1887	201818_at	$6.4*10^{-14}$	161	11732	211796_s_at	$1.4*10^{-9}$
132	20700	220796_x_at	$1.8*10^{-13}$	162	1959	201890_at	$1.9*10^{-9}$
133	6574	206508_at	$2.4*10^{-13}$	163	18412	218507_at	$2.0*10^{-9}$
134	12069	212143_s_at	$2.4*10^{-13}$	164	74	38241_at	$2.7*10^{-9}$
135	9129	209095_at	$9.1*10^{-13}$	165	11	AFFX	$3.3*10^{-9}$
136	17878	217973_at	$9.1*10^{-13}$			-DapX-M_at	
137	711	200642_at	$1.3*10^{-12}$	166	11448	211501_s_at	$4.4*10^{-9}$
138	11830	211902_x_at	$3.4*10^{-12}$	167	14535	214617_at	$5.7*10^{-9}$
139	3402	203335_at	$4.7*10^{-12}$	168	14638	214722_at	$5.7*10^{-9}$
140	9614	209584_x_at	$6.5*10^{-12}$	169	1786	201717_at	$6.1*10^{-9}$

rank	row-no.	name	*p*-value	rank	row-no.	name	*p*-value
170	5793	205726_at	$7.5*10^{-9}$	200	17667	217762_s_at	$1.4*10^{-6}$
171	9493	209460_at	$7.5*10^{-9}$	201	5039	204972_at	$1.5*10^{-6}$
172	994	200925_at	$9.9*10^{-9}$	202	10506	210495_x_at	$1.5*10^{-6}$
173	13658	213738_s_at	$1.3*10^{-8}$	203	687	200618_at	$1.7*10^{-6}$
174	20884	220980_s_at	$1.7*10^{-8}$	204	3796	203729_at	$1.7*10^{-6}$
175	9485	209452_s_at	$2.2*10^{-8}$	205	6092	206025_s_at	$1.7*10^{-6}$
176	9211	209177_at	$4.8*10^{-8}$	206	4506	204439_at	$2.1*10^{-6}$
177	2181	202112_at	$6.2*10^{-8}$	207	20657	220753_s_at	$2.2*10^{-6}$
178	11938	212012_at	$6.2*10^{-8}$	208	17931	218026_at	$2.3*10^{-6}$
179	17454	217549_at	$6.2*10^{-8}$	209	7774	207713_s_at	$2.7*10^{-6}$
180	22270	222371_at	$8.0*10^{-8}$	210	2718	202649_x_at	$4.2*10^{-6}$
181	4769	204702_s_at	$1.0*10^{-7}$	211	4324	204257_at	$4.2*10^{-6}$
182	13851	213932_x_at	$1.0*10^{-7}$	212	4864	204797_s_at	$4.2*10^{-6}$
183	5879	205812_s_at	$1.3*10^{-7}$	213	3927	203860_at	$5.3*10^{-6}$
184	13712	213792_s_at	$1.6*10^{-7}$	214	12023	212097_at	$6.5*10^{-6}$
185	12097	212171_x_at	$1.7*10^{-7}$	215	18448	218543_s_at	$6.5*10^{-6}$
186	11407	211458_s_at	$2.1*10^{-7}$	216	4198	204131_s_at	$8.1*10^{-6}$
187	4958	204891_s_at	$2.8*10^{-7}$	217	5358	205291_at	$8.1*10^{-6}$
188	21631	221731_x_at	$3.5*10^{-7}$	218	8903	208868_s_at	$1.2*10^{-5}$
189	18502	218597_s_at	$4.3*10^{-7}$	219	14622	214706_at	$1.2*10^{-5}$
190	2981	202912_at	$5.4*10^{-7}$	220	18787	218883_s_at	$1.2*10^{-5}$
191	6302	206236_at	$5.7*10^{-7}$	221	19723	219819_s_at	$1.2*10^{-5}$
192	2744	202675_at	$7.2*10^{-7}$	222	840	200771_at	$1.5*10^{-5}$
193	7569	207507_s_at	$7.2*10^{-7}$	223	11891	211964_at	$1.5*10^{-5}$
194	4303	204236_at	$8.7*10^{-7}$	224	19766	219862_s_at	$1.5*10^{-5}$
195	20480	220576_at	$8.7*10^{-7}$	225	10	AFFX-DapX	$1.8*10^{-5}$
196	839	200770_s_at	$9.1*10^{-7}$			-5_at	
197	18105	218200_s_at	$9.1*10^{-7}$	226	15890	215978_x_at	$1.9*10^{-5}$
198	3989	203922_s_at	$1.1*10^{-6}$	227	3390	203323_at	$2.8*10^{-5}$
199	21433	221530_s_at	$1.1*10^{-6}$	228	8061	208012_x_at	$3.3*10^{-5}$

rank	row-no.	name	p-value	rank	row-no.	name	p-value
229	9016	208982_at	$3.3*10^{-5}$	260	18095	218190_s_at	0.0013
230	3031	202962_at	$3.4*10^{-5}$	261	71	38149_at	0.0016
231	13718	213798_s_at	$3.4*10^{-5}$	262	9699	209670_at	0.0016
232	18378	218473_s_at	$3.4*10^{-5}$	263	15360	215446_s_at	0.0016
233	12605	212680_x_at	$4.8*10^{-5}$	264	18142	218237_s_at	0.0016
234	7261	207196_s_at	$6.2*10^{-5}$	265	20191	220287_at	0.0016
235	1106	201037_at	$7.5*10^{-5}$	266	21570	221669_s_at	0.0016
236	13400	213479_at	$7.5*10^{-5}$	267	4600	204533_at	0.0018
237	15158	215244_at	$9.0*10^{-5}$	268	11143	211161_s_at	0.0020
238	9283	209249_s_at	$1.1*10^{-4}$	269	2154	202085_at	0.00251
239	9417	209384_at	$1.3*10^{-4}$	270	6389	206323_x_at	0.0025
240	17944	218039_at	$1.3*10^{-4}$	271	17745	217840_at	0.0025
241	4662	204595_s_at	$1.5*10^{-4}$	272	17787	217882_at	0.0025
242	13003	213080_x_at	$1.5*10^{-4}$	273	37	AFFX-HSAC07/	0.0028
243	14631	214715_x_at	$1.5*10^{-4}$			X00351_3_at	
244	161	36030_at	$1.8*10^{-4}$	274	9345	209311_at	0.0029
245	4967	204900_x_at	$2.6*10^{-4}$	275	15513	215600_x_at	0.0033
246	12177	212251_at	$2.6*10^{-4}$	276	536	AFFX-r2-Ec-	0.0037
247	19136	219232_s_at	$2.6*10^{-4}$			bioB-M_at	
248	19424	219520_s_at	$3.5*10^{-4}$	277	544	AFFX-r2-	0.0037
249	9282	209248_at	$3.7*10^{-4}$			Bs-dap-5_at	
250	13978	214059_at	$3.7*10^{-4}$	278	4744	204677_at	0.0039
251	12592	212667_at	$4.2*10^{-4}$	279	7570	207508_at	0.0039
252	18792	218888_s_at	$4.2*10^{-4}$	280	9157	209123_at	0.0043
253	21470	221567_at	$4.2*10^{-4}$	281	11473	211527_x_at	0.0043
254	20913	221009_s_at	$4.4*10^{-4}$	282	8764	208729_x_at	0.0044
255	18054	218149_s_at	$8.2*10^{-4}$	283	21554	221653_x_at	0.0051
256	15022	215108_x_at	$9.7*10^{-4}$	284	3067	202998_s_at	0.0057
257	5723	205656_at	$1.0*10^{-3}$	285	2	AFFX-BioB-	0.0065
258	2099	202030_at	$1.1*10^{-3}$			M_at	
259	2492	202423_at	$1.1*10^{-3}$	286	7775	207714_s_at	0.0065

rank	row-no.	name	p-value	rank	row-no.	name	p-value
287	9872	209846_s_at	0.0065	312	8798	208763_s_at	0.0201
288	3	AFFX-BioB-3_at	0.0085	313	14352	214433_s_at	0.0201
				314	15227	215313_x_at	0.0201
289	34	AFFX-HUMGAPDH/M33197_3_at	0.0085	315	18106	218201_at	0.0201
				316	21933	222033_s_at	0.0201
290	1319	201250_s_at	0.0085	317	1132	201063_at	0.0220
291	4887	204820_s_at	0.0085	318	3103	203034_s_at	0.0227
292	21417	221514_at	0.0085	319	9344	209310_s_at	0.0227
293	8725	208690_s_at	0.0097	320	122	32137_at	0.0278
294	16353	216442_x_at	0.0097	321	608	200046_at	0.0278
295	1681	201612_at	0.0111	322	7822	207761_s_at	0.0278
296	18110	218205_s_at	0.0111	323	12925	213002_at	0.0278
297	19604	219700_at	0.0111	324	5365	205298_s_at	0.0286
298	535	AFFX-r2-Ec-bioB-5_at	0.0122	325	694	200625_s_at	0.0312
				326	4403	204336_s_at	0.0312
299	1732	201663_s_at	0.0126	327	4687	204620_s_at	0.0312
300	2999	202930_s_at	0.0139	328	3243	203175_at	0.0349
301	7137	207071_s_at	0.0143	329	14386	214467_at	0.0349
302	15559	215646_s_at	0.0162	330	20943	221039_s_at	0.0349
303	393	47608_at	0.0178	331	12619	212694_s_at	0.0390
304	4346	204279_at	0.0178	332	18937	219033_at	0.0423
305	5337	205270_s_at	0.0178	333	2865	202796_at	0.0435
306	8475	208436_s_at	0.0178	334	7428	207365_x_at	0.0435
307	12933	213010_at	0.0178	335	11895	211968_s_at	0.0435
308	14768	214853_s_at	0.0178	336	13525	213605_s_at	0.0435
309	18786	218882_s_at	0.0178	337	1495	201426_s_at	0.0484
310	546	AFFX-r2-Bs-dap-3_at	0.0201	338	9635	209605_at	0.0489
				339	12358	212433_x_at	0.0489
311	2872	202803_s_at	0.0201				

D: Error Curves for Estimation of Missing Values

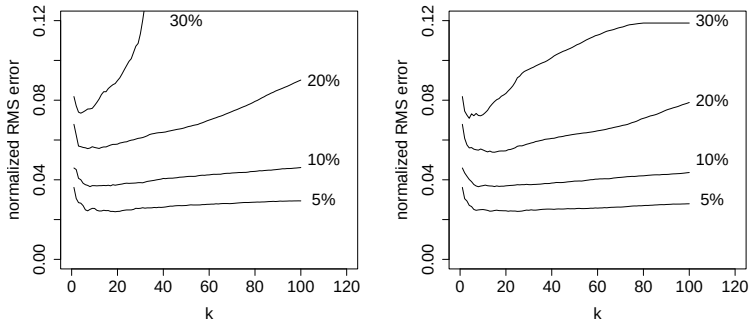


Figure C.1: Error curve when applying the imputation method with the Euclidean distance and with the weighted mean (left) or the median (right) to data sets with different percentages of missing values.

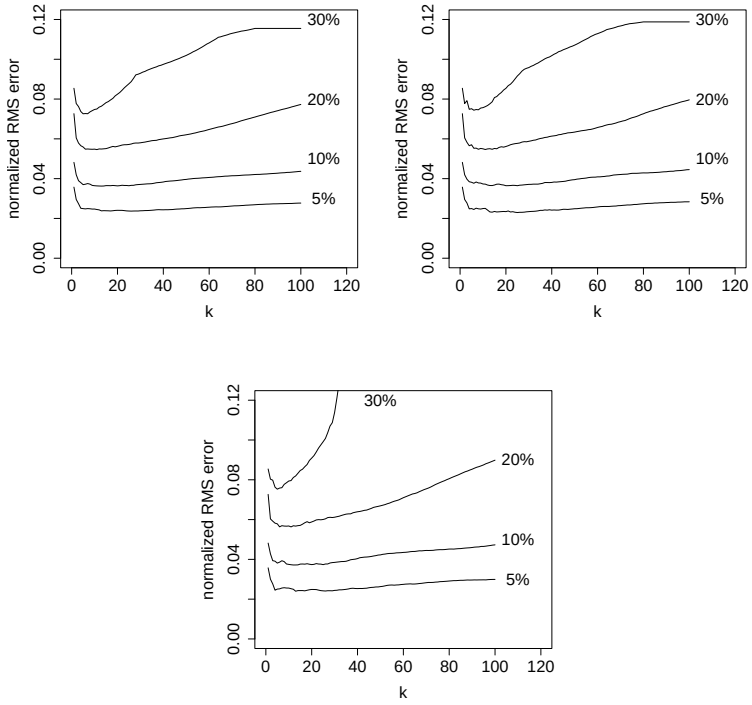


Figure C.2: Error curve when applying the imputation method with the Mahalanobis distance and with the mean (top left), the weighted mean (top right) or the median (bottom) to data sets with different percentages of missing values.

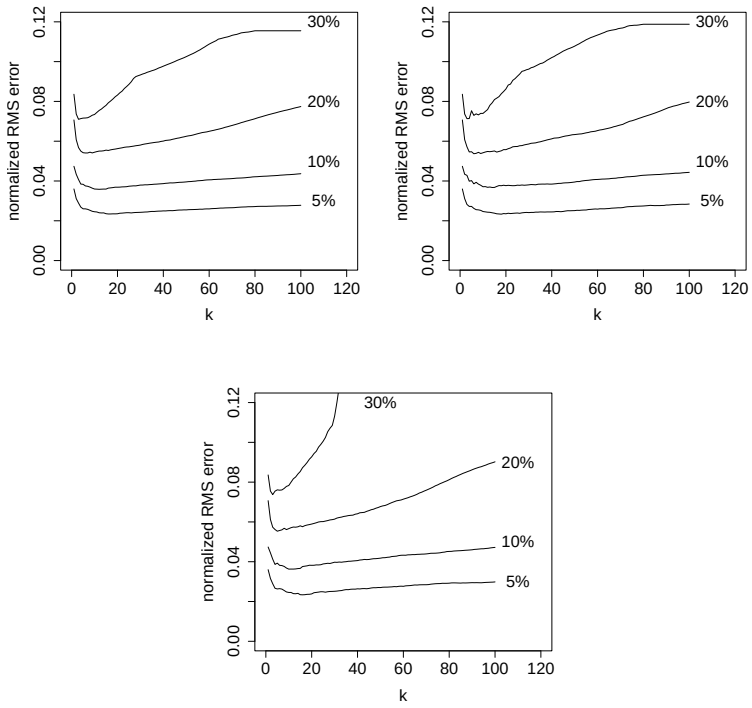


Figure C.3: Error curve when applying the imputation method with the Chebyshev distance and with the mean (top left), the weighted mean (top right) or the median (bottom) to data sets with different percentages of missing values.

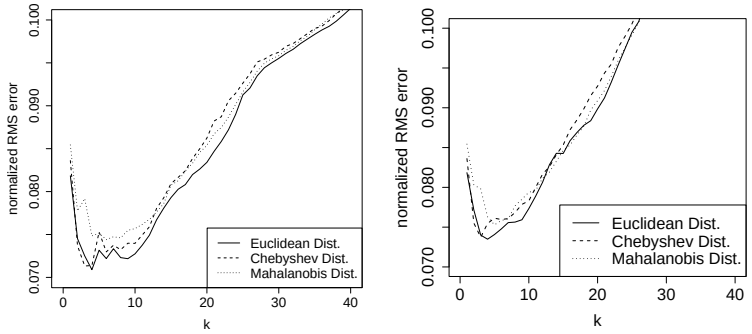


Figure C.4: Error curve for a data set with 30% of missing values. Imputation method was applied with different distances and with the median (left) or the weighted mean (right).

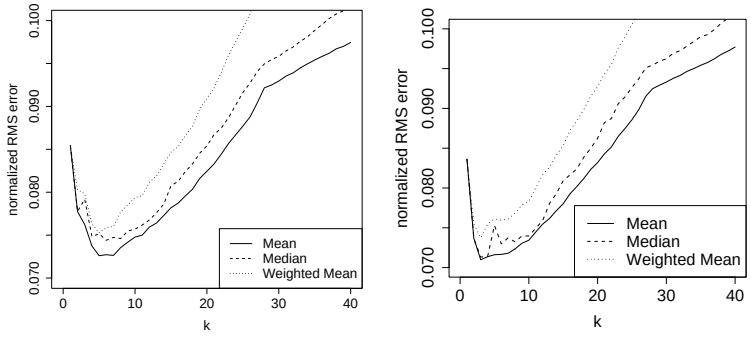


Figure C.5: Error curve for a data set with 30% of missing values. Imputation method was applied with different estimators and with the Mahalanobis distance (left) or the Chebyshev distance (right).

E: Gel Spots with Time/Treatment-Interactions

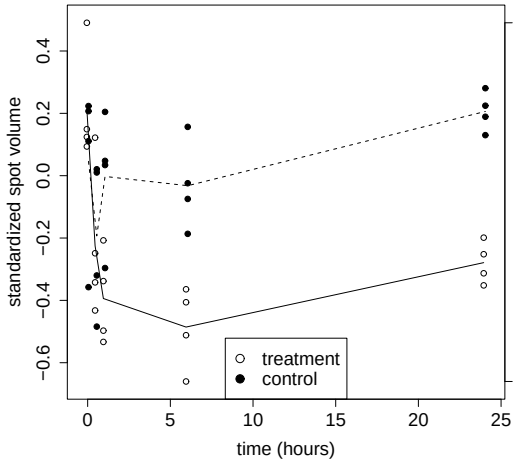


Figure D.1: Mean temporal course of spot 1136 in the treatment group (solid line) and control group (dashed line), respectively.

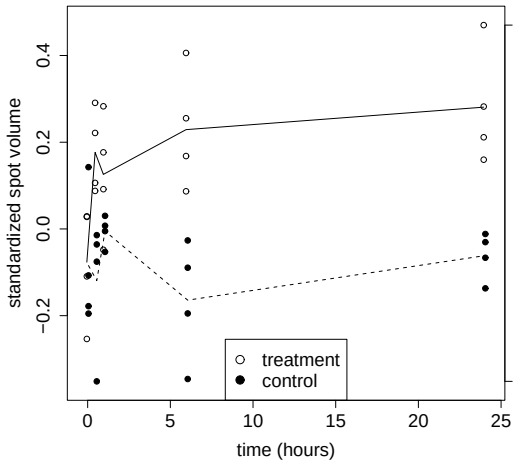


Figure D.2: Mean temporal course of spot 941 in the treatment group (solid line) and control group (dashed line), respectively.

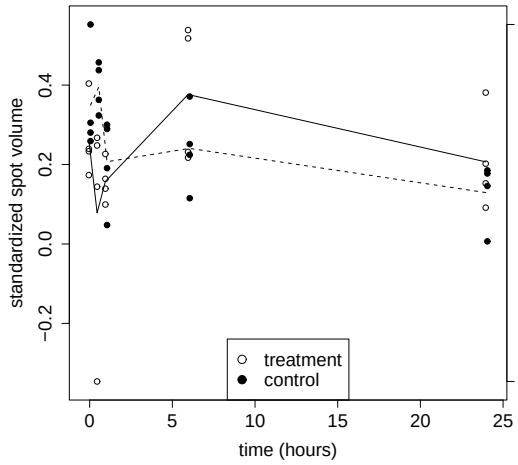


Figure D.3: Mean temporal course of spot 1301 in the treatment group (solid line) and control group (dashed line), respectively.

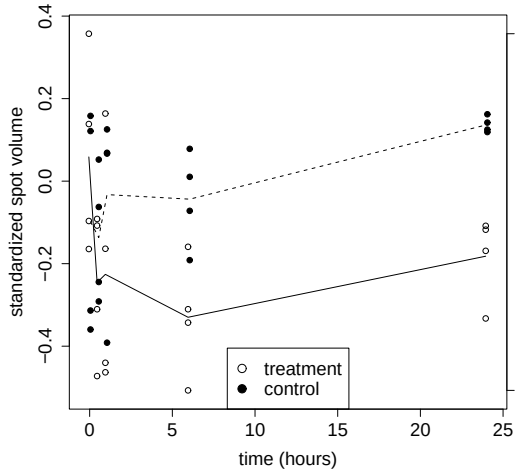


Figure D.4: Mean temporal course of spot 1166 in the treatment group (solid line) and control group (dashed line), respectively.

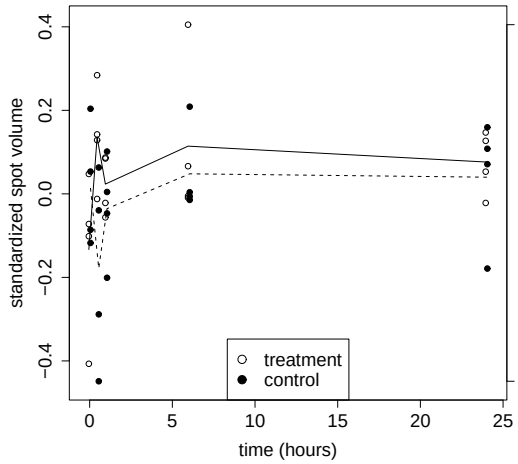
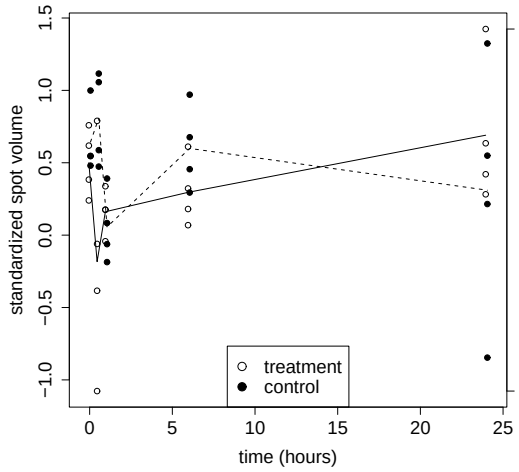


Figure D.5: Mean temporal course of spot 2227 in the treatment group (solid line) and control group (dashed line), respectively.



g

Figure D.6: Mean temporal course of spot 1787 in the treatment group (solid line) and control group (dashed line), respectively.

F: Gel Spots with Treatment Effects

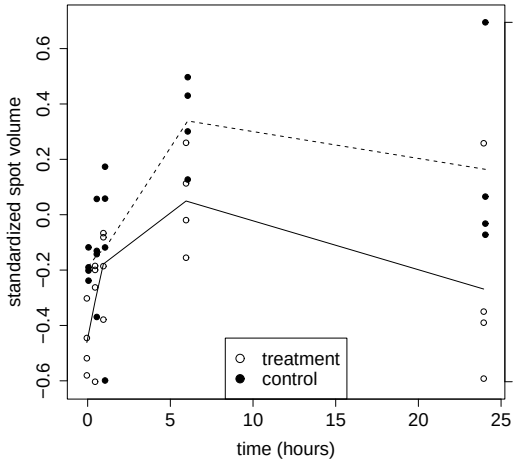


Figure E.1: Mean temporal course of spot 2502 in the treatment group (solid line) and control group (dashed line), respectively.

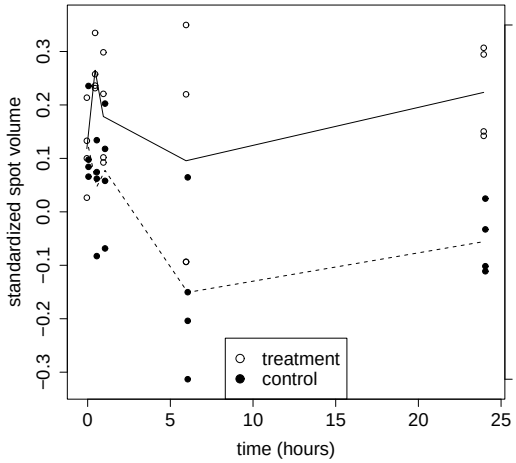


Figure E.2: Mean temporal course of spot 935 in the treatment group (solid line) and control group (dashed line), respectively.

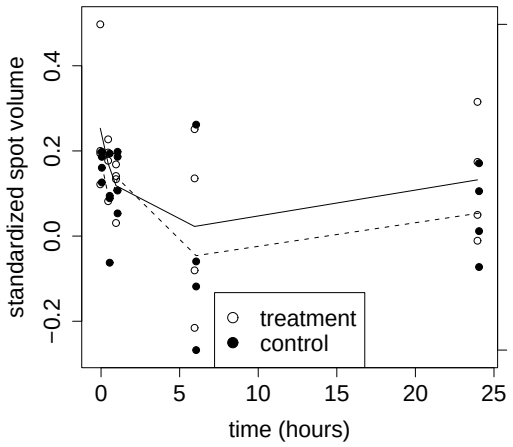


Figure E.3: Mean temporal course of spot 1266 in the treatment group (solid line) and control group (dashed line), respectively.

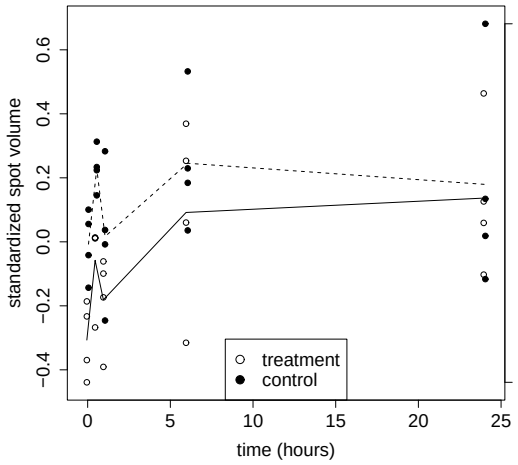


Figure E.4: Mean temporal course of spot 2123 in the treatment group (solid line) and control group (dashed line), respectively.

Notation

Chapter 2

i	Gene or protein ($i = 1, ..., r$).	13
j	Microarray, gel or spectrum ($j = 1, ..., n$).	13
n	Number of observations (microarrays, 2-D gels or spectra).	14
r	Number of variables (genes, gel spots, m/z-values).	14
m	Mass of a protein or a peptide.	15
z	Charge of a protein or a peptide.	15
h	Fluorescent dye (h=1: Cy2, h=2: Cy3 and h=3: Cy5).	18
y_{ih}	Intensity of gel spot.	18
a_h, b_h	Calibration coefficients.	18

Chapter 3

r	Number of variables (genes).	28
n	Number of observations (microarrays).	28
X	$(r \times n)$-matrix with expression values.	28
i	Gene ($i = 1, ..., r$).	28
j	Microarray ($j = 1, ..., n$).	28
e_1	Number of *true negative decisions*.	29
e_2	Number of *false positive decisions* (*type I errors*).	29

e_3	Number of *false negative decisions* (*type II errors*).	29
e_4	Number of *true positive decisions*.	29
Z_i, z_i	Test statistics for gene i.	31
f	Distribution of Z_i's.	31
f_0	Distribution of z_i's.	31
$: K$	Kernel function.	33
δ	Magnitude of expression change.	37
$\tilde{B}$	Number of runs of the permutation algorithm.	42

Chapter 4

r	Number of variables (gel spots, proteins).	52
n	Number of observations (gels).	52
X	$(r \times n)$-matrix with expression values.	52
i	Gel spot $(i = 1, ..., r)$.	52
j	2-D Gel $(j = 1, ..., n)$.	52
k	Number of neighbors or principal components.	53
t	Time $(t = 1, ..., T)$.	60
T	Number of points in time.	60
g	Group (g=1: treatment, g=2: control).	60
G	Number of groups.	60

Chapter 5

l	Spectrum from patient j $(l = 1, ..., k)$.	72
$X^{(j)}$	$(r \times k)$-matrix with spectra from patient j.	72
j	Patient $(j = 1, ..., n)$.	72
i	m/z-value $(i = 1, ..., r)$.	72
r	Number of variables (m/z-values, proteins, peptides).	72
n	Number of observations (spectra).	73

$D^{(j)}$	$(n \times n)$-distance-matrix for spectra from patient j.	73
$m^{(j)}$	Multivariate mean of replications of spectra from patient j.	73
$z_{(b)}$	Ordered values for density estimation $(b = 1, ..., B)$.	74

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Bibliography

Abdi, H. (2003): Partial least squares regression (PLS-regression). In: M. Lewis-Beck, A. Bryman, T. Futing (Eds): *Encyclopedia for research methods for the social sciences.* Sage Publications, Thousand Oaks (CA), 792–795.

Aebersold, R. and Goodlett, D.R. (2001): Mass spectrometry in proteomics. *Chemical Reviews* ,**101**, 269–295.

Affymetrix (2001): *Microarray Suite User's guide. Version 5.0.* Affymetrix Inc., Santa Clara, California.

Affymetrix (2001b): *Statistical Algorithms Reference Guide.* Affymetrix Inc., Santa Clara, California.

Affymetrix (2002): *Statistical Algorithms Description Document.* Affymetrix Inc., Santa Clara, California.

Amersham Biosciences (2003): *DeCyder Differential Analysis Software, Version 5.0, User Manual.* Amersham Biosciences, Sweden.

Baumann, S., Ceglarek, U., Fiedler, G.M., Lembcke, J., Leichtle, A. and Thiery, J. (2005): Standardized approach to proteome profiling of human serum based on magnetic bead separation and matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Clinical Chemistry*, **51**, 1–8.

Brás, L.P. and Menezes, J.C. (2006): Estimating gene expression missing data using PLS regression. In: Urfer, W. and Turkman, A. (Eds.): *Proceedings of the Workshop on Statistics in Genomics and Proteomics, October 5-8, 2005, Monte Estoril.* Centro Internacional de Mathématica, Coimbra, Portugal, 55–64.

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

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

Boulesteix, A.-L. (2004): PLS Dimension Reduction for Classification with Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, **3**, Issue 1, Article 33.

Brown, P.O. and Botstein, D. (1999): Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, **21**, 33–37.

Crowder, M.J. and Hand, D.J. (1995): *Analysis of repeated measures.* Chapman & Hall, London.

Davis, P.J. and Rabinowitz, P. (1984): *Methods of Numerical Integration.* Academic Press Inc., Orlando, Florida.

Devroye, L.P. (1978): The uniform convergence of nearest neighbor regression function estimators and their application in optimization. *IEEE Transactions on Information Theory*, **24**, 142–151.

Diggle, P.J., Liang, K.Y. and Zeger, S.L. (1994): *Analysis of Longitudinal Data.* Clarendon Press, Oxford.

Dudoit, S., Shaffer, J.P. and Boldrick, J.C. (2003): Multiple hypothesis testing in microarray experiments. *Statistical Science*, **18**, 71–103.

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001): Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, **96**, 1151–1160.

Egan, M.B. and Morgan, S.L. (1998): Outlier detection in multivariate analytical chemical data. *Analytical Chemistry*, **70**, 2372–2379.

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

Frank, I.E. and Friedman, J.H. (1993): A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–148.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000): Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**, 601–620.

Fu, W.J., Dougherty, E.R., Mallick, B. and Carroll, R.J. (2005): How many samples are needed to build a classifier: a general sequential approach. *Bioinformatics*, **21**, 63–70.

Gannoun, A., Saracco, J., Urfer, W. and Bonney, G.E. (2002): Nonparametric modelling approach for discovering differentially expressed genes replicated microarray experiments. *Technical Report*, **41/2002**, SFB 475, University of Dortmund.

Gannoun, A., Saracco, J., Urfer, W. and Bonney, G.E. (2004): Nonparametric analysis of replicated microarray experiments. *Statistical Modelling*, **4**, 195–209.

Geladi, P. and Kowalski, B.R. (1986): Partial Least-Squares Regression: A tutorial. *Analytica Chimica Acta*, **185**, 1–17.

Glantz, S.A. and Slinker, B.K. (2000): *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, New York.

Gross, J.H. (1984): *Mass Spectrometry*. Springer-Verlag, Heidelberg.

Grzegorczyk, M. and Urfer, W. (2004): Determination of interacting genes in kidney tissues using Bayesian networks. *Research Report*, **2004/3**, Department of Statistics, University of Dortmund.

Guo, X. and Pan, W. (2004): Using weighted permutation scores to detect differential gene expression with microarray data. *Research Report*, **2004-022**, Division of Biostatistics, University of Minnesota.

Gygi, S.P., Rist, B., Gerber, S.A., Turecek, F., Gelb, M.H. and Aebersold, R. (1999): Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature Biotechnology*, **17**, 994–999ns.

Hastie, T., Tibshirani, R., Eisen, M.B., Alizadeh, A., Levy, R., Staudt, L., Chan, W.C., Botstein, D. and Brown, P. (2000): 'Gen shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, **1**, 1–21.

Huber, P. (1981): *Robust Statistics*. John Wiley & Sons Inc., New Jersey.

Huber, W., von Heydebreck, A., Sültman, H., Poustka, A. and Vingron, M. (2002): Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96–S104.

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

Jeffries, N.O. (2004): Performance of a genetic algorithm for mass spectrometry proteomics. *BMC Bioinformatics*, **5**:180.

Jeffries, N. (2005): Algorithms for alignment of mass spectrometry proteomic data. *Bioinformatics*, **21**, 3066–3073.

Johnson, R.A. and Wichern, D.W. (2002): *Applied multivariate statistical analysis. 5th Edition.* Prentice Hall, New Jersey.

Jong, H.d. (2002): Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, **9**, 67-103.

Jung, K., Gannoun, A., Sitek, B., Meyer, H.E., Stühler, K. and Urfer, W. (2005): Analysis of dynamic protein expression data. *RevStat-Statistical Journal*, **3**, 99–111.

Jung, K., Gannoun, A., Sitek, B., Apostolov, O., Schramm, A., Meyer, H.E., Stühler, K. and Urfer, W. (2006a): Statistical evaluation of methods for the analysis of dynamic protein expression data from a tumor study. *RevStat-Statistical Journal*, **4**, 67–80.

Jung, K., Quast, K., Gannoun, A. and Urfer, W. (2006b): A renewed approach to the nonparametric analysis of replicated microarray experiments. *Biometrical Journal*, **48**, 245–254.

Karp, A.N., Kreil, D.P. and Lilley, K.S. (2004): Determining a significant change in protein expression with DeCyder during a pairwise comparison and to the quantification of differential expression. *Proteomics*, **4**, 1421–1432.

Ketskemety, L. (2004): Effectiveness of nearest neighbor classification with optimal scaling. *Alkalmazott Matematikai Lapok*, **21**, 81-97.

Kirschbaum, N. and Urfer, W. (2006): Partial least squares regression and its application in biomolecular interactions's analysis. *Research Report*, **2006/1**, Department of Statistics, University of Dortmund.

Klose, J. and Kobalz, U. (1995): Two-dimensional electrophoresis of proteins: an updated protocol and implications for a functional analysis of the genome. *Electrophoresis*, **16**, 1034–1059.

Knowles, M.R., Cervino, S., Skynner, H.A., Hunt, S.P., Felipe, C.de, Meneses-Lorente, G., McAllister, G., Guest, P.C. (2003): Multiplex proteomic analysis by two-dimensional differential in-gel electrophoresis. *Electrophoresis*, **3**, 1162–1171.

Kreil, D.P., Karp, A.N. and Lilley, K.S. (2004): DNA microarray normalization methods can remove bias from differential protein expression analysis of 2D difference gel electrophoresis results. *Bioinformatics*, **20**, 2026–2034.

Lehmann, E.O. (1986): *Testing statistical hypothesis. 2nd Edition.* Springer-Verlag, New York.

Li, C. and Wong, W.H. (2004): Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences*, **98**, 31–36.

Lilien, R.H., Farid, H. and Donald, B.R. (2003): Probabilistic disease classifcation of expression-dependent proteomic data from mass spectrometry of human serum. *Journal of computational biology*, **10**, 925–946.

Lodish, H., Berk, A., Zipursky, S.L., Matsudaira, P., Baltimore, D. and Darnwell, J.E. (2001): *Molekulare Zellbiologie. Vierte Auflage.* Spektrum Akademischer Verlag, Heidelberg, 917–983.

Lorkowski, S. and Cullen, P. (2001): *Analysing gene expression.* Wiley-VCH, Weinheim.

Marcus, K. and Meyer, H.E. (2003): Fallstricke in der Durchführung von Proteomanalysen. *Biospektrum*, **9**, 492–494.

Mood, A.M., Graybill, F.A. and Boes, D.C. (1974): *Introduction to the theory of statistics. Third edition.* McGraw-Hill, Singapore.

Morandell, S., Stasyk, T., Huber, L.A., Feuerstein, I., Huck, C.W., Bakry, R., Bonn, G.K., Roitinger, E. and Mechtler, K. (2005): Strategien zur funktionellen Proteom-Analyse von Signaltransduktionswegen. *Laborwelt*, **4**, 26–29.

Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., Golub, T.R. and Mesirov, J.P. (2003): Estimating dataset size requirements for classifying DNA microarray data. *Journal of Computational Biology*, **10**, 119–142.

Müller, P., Parmigiani, G., Robert, C. and Rousseau, J. (2004): Optimal sample size for multiple hypothesis testing: the case of gene expression microarrays. *Journal of the American Statistical Association*, **99**, 990–1001.

Nakagawara, A., Azar, C.G., Scavarda, N.J. and Brodeur, G.M. (1994): Expression and function of Trk-B and BDNF in human neuroblastomas. *Molecular and Cellular Biology*, **14**, 759–767.

Nesvizhskii, A.I., Keller, A., Kolker, E. and Aebersold, R. (2002): A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, **75**, 4646–4658.

Nicholson, J.K. and Wilson, I.D. (2003): Understanding 'global' systems biology: metabonomics and the continuum of metabolism. *Nature Reviews*, **2**, 668–677.

Nguyen, D.V., Arpat, A.B., Wang, N. and Carroll, R.J. (2002): DNA microarray experiments: biological and technical aspects. *Biometrics*, **58**, 701–717.

Nguyen, D.V. and Rocke, D.M. (2002): Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, **18**, 39–50.

Nguyen, D.V., Wang, N. and Carroll, R.J. (2004): Evaluation of missing value estimation for microarray data. *Journal of Data Science*, **2**, 347–370.

Pan, W., Lin, J. and Le, C. (2001): A mixture model approach to detecting differentially expressed genes with microarray data. *Technical Report*, Division of Biostatistics, University of Minnesota.

Pan, W., Lin, J. and Le, C. (2002): How many replicates of arrays are required to detect gene expression changes in microarray experiments? A mixture model approach. *Technical Report*, Division of Biostatistics, University of Minnesota.

Patterson, S.D. (2003): Data analysis – the Achilles heel of proteomics. *Nature Biotechnology*, **21**, 221–222.

Pavlidis, P., Li, Q. and Noble, W.S. (2003): The effect of replication on gene expression microarray experiments. *Bioinformatics*, **19**, 1620–1627.

Pepe, M.S., Longton, G., Anderson, G.L. and Schummer, M. (2003): Selecting differentially expressed genes from microarray experiments. *Biometrics*, **59**, 133–142.

Pusch, W., Flocco, M.T., Leung, S.-M., Thiele, H. and Kostrzewa, M. (2003): Mass spectrometry-based clinical proteomics. *Pharmacogenomics*, **4**, 463-476.

Rajagopalan, D. (2003): A comparison of statistical methods for the analysis of high density oligonucleotide array data. *Bioinformatics*, **19**, 1469–1476.

RheaCorporation (1995): *KestrelSpec, Software Manual, Version 2.4.* http://www.rheacorp.com/KesST6m.pdf.

Ripley, B.D. (1996): *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge.

Rosenblatt, M. (1979): Global measures of deviation for kernel and nearest neighbor density estimates. In: Gasser, Th. and Rosenblatt, M. (Eds.): *Smoothing techniques for curve estimation.* Springer-Verlag, Berlin, 181–190.

Schulze, A. and Downward, J. (2001): Navigating gene expression using microarrays – a technology review. *Nature Cell Biology*, **3**, E190–E195.

Schweigert, F., Rawel, H. and Raila, J. (2005): Biomarker als Indikatoren für den gesundheitsfördernen Effekt von Pflanzenmetaboliten. *Laborwelt*, **4**, 29–31.

Shaffer, J.P. (1995): Multiple hypothesis testing: A review. *Annual Review of Psychology*, **46**, 561–584.

Sitek, B., Apostolov, O., Stühler, K., Pfeiffer, K., Meyer, H.E., Eggert, A. and Schramm, A. (2005): Identification of dynamic proteome changes upon ligand activation of Trk-receptors using two-dimensional fluorescence difference gel electrophoresis and mass spectrometry. *Molecular & Cellular Proteomics*, **4**, 291–299.

Sitek, B., Scheibe, B., Jung, K., Schramm, A. and Stühler, K. (2006): Difference Gel Electrophoresis (DIGE): the next generation of two-dimensional gel electrophoresis for clinical research. In: Hamacher, M., Marcus, K., Stühler, K., van Hall, A., Wahrscheid, B. and Meyer, H.E. (Eds.): *Proteomics in Drug Research*, Whiley-VCH, Weinheim, 33–56.

Spivey, A. (2004): Systems biology: the big picture. *Environmental Health Perspectives*, **112**, A939–A943.

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

Storey, J.D. (2003): The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, **31**, 2013–2035.

Terrell, G.R. (1990): The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, **85**, 470–476.

Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, Gilbert (2002): Diagnosis of multiple cancer types shrunken centroids of gen expression. *Proceedings of the National Academy Science*, **99**, 6567–6572.

Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, Gilbert (2003): Class prediction by nearest shrunken centroids, with application to DNA microarrays. *Statistical Science*, **18**, 104–117.

Tilstone, C. (2003): DNA microarrays: vital statistics. *Nature*, **424**, 610–612.

Troyanskaya, O.G., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R.B. (2001): Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Troyanskaya, O.G., Garber, M.E., Brown, P.O, Botstein, D. and Altman, R.B. (2003): Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics*, **18**, 1454–1461.

Tusher, G.T., Tibshirani, R. and Chu, G. (2001): Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy Science*, **98**, 5116–5121.

Villanueva, J., Philip, J., Entenberg, D., Chaparrom C.A., Tanwar, M.K., Holland, E.C. and Tempst, P. (2004): Serum peptide profiling by magnetic particle-assisted, automated sample processing and MALDI-TOF mass spectrometry. *Analytical Chemistry*, **76**, 1560–1570.

Villanueva, J., Philip, J., Chaparro, C.A., Li, Y., Toledo-Crow, R., DeNoyer, L., Fleisher, M., Robbins, R.J. and Tempst, P. (2005): Correcting common errors in identifying cancer-specific serum peptide signatures. *Journal of Proteome Research*, **4**, 1060–1072.

Weckwerth, W., Willmitzer, L. and Fiehn, O. (2000): Comparative quantification and identification of phosphoproteins using stable isotope labeling and liquid chromatography/mass spectrometry. *Rapid Communications in Mass Spectrometry*, **14**, 1677–1681.

Zhan X. and Desiderio, D.M. (2003): Differences in the spatial and quantitative reproducibility between two second-dimensional gel electrophoresis systems. *Electrophoresis*, **24**, 1834–1846.

Zhang, X., Leung, S.-M., Morris, C.R. and Shingenaga, M.K. (2004): Evaluation of a novel, integrated approach using funcionalized magnetic beads, bench-top MALDI-TOF-MS with prestructured sample supports, and pattern recognition software for profiling potential biomarkers in human plasma. *Journal of Biomolecular Techniques*, **15**, 167–175.

Zhao, Y., and Pan, W. (2003): Modified nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments. *Bioinformatics*, **19**, 1046–1054.