Diplomarbeit

Statistical Analysis of High Dimensional Biomedical Data

ausgeführt am
Institut für Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter der Anleitung von
Ao. Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

durch
Jose Carlos Martinez Avila
Matr.Nr.: 0541186
Linzerstraße 280 11
1140 Wien

Wien, am 5 September 2012

Contents

To my wife, Edith and my two daughters, Anna and Barbara, they are my compass in my life journey.

To my parents, they gave me the life.

To Peter, who received me friendly and allows me a new beginning. Thank you for your teachings, patience and time.

"With Faith, heart and steel!"
"There is nothing like a challenge to bring out the best in man."
Sean Connery as Juan Sanchez Ramirez Villalobos
The Highlander(1986)

Summary

During the last decades the amount and accuracy of analytical methods in different fields of science has produced large numbers of measurements in just one individual or sample. Biochemical analysis methods provide a wide range of available measures which should be useful to explain a certain outcome or trait. Biomolecular methods applied to genetics allows to know how a genome is built and expressed. The association of a Phenotype with a Genotype is a main task of the Genomics and Bioinformatics field. At the -omics era we are swimming in a sea of data, but to dive in it needs more training and caution.
The aim of this master thesis is to present different methods to analyze high dimensional data with a case study focused in Biomedical Data.
The underlying theory of each method is briefly explained without lack of accuracy. Together with the theory a direct application in R code with an example using a grapevine data is presented.
The master thesis is organized in seven chapters. The first chapter is an introductory chapter which covers basic concepts and definitions about high dimensional data, genetics and the R environment. Also in this chapter the example data set of grapevine is explained.
Chapters from two to five present the theory of the different methods such as Multivariate Outlier Identification, Linear Discriminant Analysis, Principal Components Regression, Partial Least Squares Regression, Penalized Regression, Cluster Analysis, Permutation test, Significance Analysis of Microarrays and Moderated t-statistic. At the end of each chapter, the explained method is applied to the grapevine data set.
The sixth Chapter is dedicated to the case study using 8650 transcripts of 364 patients affected and non affected of Alzheimer disease.
The final chapter concludes some key issues in the statistical analysis of high dimensional biomedical data.

Chapter 1

Introduction

1.1 High Dimensional Data

John Snow in 1854 was looking for a reasonable hypothesis to explain the cholera outbreak in Soho district of London. There was a controversy between two theories, miasma and germ. The miasma theory pointed out that illness was caused by pollution, associated with poor air quality of an industrial city as London. The germ theory postulated that microorganisms are the cause of many illnesses. He was not convinced about the miasma theory explanation and irrespective of which kind of germ could produce the cholera he began a research about the spread of the cholera. The genius idea of him was to draw a map the cholera cases, see Figure 1.1.

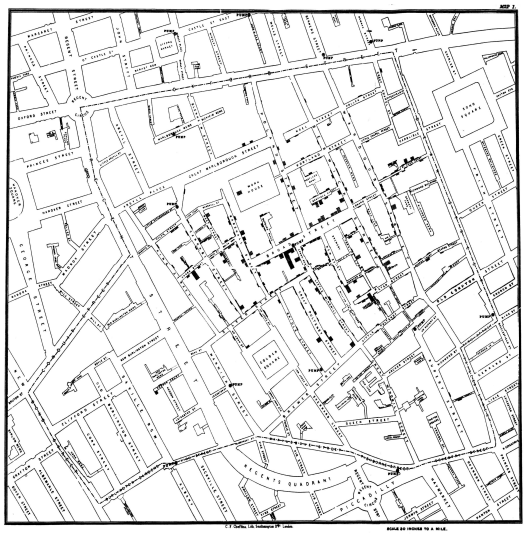

Figure 1.1: Original map by John Snow showing the clusters of cholera cases in the London epidemic of 1854.

John Snow realized that the origin of the cholera outbreak was a public water pump on Broad Street (Snow, 1854). The pump was disabled and this was the beginning of the end

of the cholera outbreak. At Snow's time the analytical methods were not developed enough as in the present days and there was lack of data, a recurrent problem in research. When Snow had lived today, he could have thousands of data of the outbreak. First he could analyze water, air, food of the cholera patients using the chemometrical approach, that will give him a big amount of data. Second, a genomic approach would be possible just with a blood drop of each patient and a water drop of the pump. He could have thousands of variables per patient but may be he could be lost without any key about the cholera outbreak.

This is like high dimensional data look, a number of observations, patients, animals, plants, materials or subjects, n, much smaller than the amount of variables, p, chosen to explain the event or outcome. This can be summarized in matrix of data, $\mathbf{A}$, as follows,

$$\mathbf{A}_{n\times p} \;\; with\; p \gg n.$$

The development of analytical tools, e.g. mass spectrometry, nuclear magnetic resonance, infrared and polymerase chain reaction gives today the opportunity to produce different measures of the same subject. Unfortunately this is not for free, new computational tools are needed but more important robust and reliable statistical methods to explain in this p-dimensional space which variables are controlling the observed reality.
High dimensional data appear in a wide range of research fields, like in agriculture, livestock ,marketing, finance, climatology, material science,...,but one of the most successful and famous is the *genomics* that has a direct application in biomedical engineering.
In biomedical research there are two issues more to take into account. The number of subjects available is smaller than in other research fields because patients are volunteers, then once you have a patient you try to get as much measures as possible of each one. Another problem more to have enough observations is also that clinical trials are under supervision of ethic commissions and it is not allowed to use a real placebo with ill patients.

We are swimming in a sea of data, but we are going to drown because we do not see the life jacket.

1.2 Genetic Background

Phenotype, P, is the observable part of a trait, e.g., the milk yield, *trait*, of a cow in kilograms per lactation *observation*. The phenotype can be decomposed in two terms,

$$P = G + E, \tag{1.1}$$

where G, is the Genotype and E is the Environment effect. This simple formula explains that an observable trait is the consequence of the genotype and the environment (Falconer and Mackay, 1998), following the example about the milk yield, the milk yield of a cow is the result of the genotype plus the effect given by herd, farm, food, and this is the environment where the individual, cow, lives. Genotype is constant and Environment can be modified during all the life of the individual.

Taking variances on both sides of equation (1.1), gives

$$var\,(P) = var\,(G) + var\,(E) + cov\,(G, E), \quad (1.2)$$

which represents the possible sources of variation in an observable trait. The term $cov\,(G, E)$ represents the interaction *genotype-environment*, that in some cases is set to zero but sometimes it can not be simplified. The infinitesimal model of Fisher (1918) proposed that genotype is the sum of infinite genes with a little effect and unknown location, this model is robust to explain most of the traits, but if the trait has a small number of regulator genes, Fisherś model cannot be applied. When in 1953 Watson and Crick discovered the structure of DNA, and later Kary Mullis, 1983, patented the polymerase chain reaction, PCR, technique a new era began, with the central dogma of molecular biology which explains the flow of information from DNA to proteins (1.3).

DNA, *deoxyribonucleic acid*, is built in form of double-stranded nucleotides which have three elements, a sugar, a phosphate and a base. There are four bases *adenine*, A, *cytosine*, C, *guanine*, G and *thymine*, T with complementarity between them. A and T bind together and C and G also. Only with these four letters, grouped in triplets, the genetic information is coded. Each triplet named codon means an aminoacid. RNA, *ribonucleic acid*, is built in form of a single-stranded nucleotides where *thymine*, T, is replaced by *uracil*, U. There are different types of RNA with different functions, transfer RNA, *tRNA*, messenger RNA, *mRNA* and ribosomal DNA *rRNA*.

(1.3)

The synthesis of proteins according to the central dogma of molecular biology, (1.3) is done in two steps, the first inside the nucleus and the second into the cytoplasm (we consider an eukaryotic cell). The DNA double-stranded is transcript from the double-straded DNA to the single-stranded RNA. This transcribed RNA suffers a post-transcriptional process to produce mRNA which leaves the nucleus towards the cytoplasm.

In the cytoplasm begins the second step of the synthesis of proteins, translation. The information carried by the mRNA in form of bases, codons, will be translated into a protein in form of aminoacids at the ribosomes. The tRNA with a complementary codon of the mRNA, brings an aminoacid. The rRNA at ribosomes work as starter of a chemical reaction that the aminoacids merge with the previous one and free the tRNA. A chain of aminoacids is elongate until the mRNA is read and the ribosome reads a codon with the instruction *"stop"*.

The process that we have outlined above is called gene expression, the process of converting a DNA sequence into a protein. The amount of DNA in the cells of a living can be considered as a constant but the amount of mRNA is different between cells, tissues, diseases or environmental conditions. The difference in the mRNA between cells and conditions is the basis of the microarray technology.
Microarray is a technical device to measure the gene expression, it works as follows:

- From a tissue, mRNA is extracted and the complementary DNA, cDNA, obtained via reverse transcription reaction.
- cDNA is fluorescently labeled.
- Thousands of different single-stranded DNA arrays are placed separately in a matrix.
- cDNA and DNA array hibridized. Under the appropriate source of light an image will be generated.
- Intensity of the generated image in each element of the matrix depends on the original amount of mRNA in the tissue.

Single nucleotide polymorphism, SNP, is an important issue regarding genetics and high dimensional data. A SNP is a single difference, in a single nucleotide between the DNA sequences of individuals the same species or paired chromosomes the same individual. Let us suppose that in a given location of the DNA, an individual has the DNA sequence, $AAATTTCCCGGG$, and another individual has $AAATTACCCGGG$. This is a SNP and in fact it is a mutation in the population. The two possibilities of a SNP are called alleles, i.e. $AAATTTCCCGGG = \ allele_1$ and $AAATTACCCGGG = \ allele_2$. Diploid organisms such as mammals have two copies of each chromosome and can have SNPs between the two copies. In our example the individual can be, TA, TT or AT. This defines a variable or factor with 3 values or levels, respectively. The next step is to use this SNPs to explain an outcome, trait or dependent variable, using thousands of SNPs.

Now we know that genes are finite and that it is possible to locate them. In fact it seems to be a combination of both models in reality, a few genes with large effect and a big amount of genes with small effect.
The gene apolipoprotein E $APOE\varepsilon 4$, in chromosome 19, has three alleles, $\epsilon 2, \epsilon 3, \epsilon 4$. Allele $\epsilon 4$ in homozigosys (Corder et al., 1993) is sufficient to cause Alzheimer Disease (AD) at age 80, but in heterozygosis it is not determinant of AD. There should be genes with small effect which can explain that some patients $\epsilon 4 - \epsilon 4$ never develop AD and one third of AD affected are $\epsilon 4$ negative.

1.3 About R

The official definition of R can be obtained in the web page of R project, `www.r-project.org`, *R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly ATT, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.*
R is Open Source, that can be defined as freely available, free access to the code and possibility to extend it. These features are the main strengths because users can programme their own code based on previous packages. Later this code will be included in an R repository in a package. In this way it is possible to find R packages focused on applied statistics or highly specialized topics and also methodological ones.

R packages are uploaded and located at the Comprehensive R Archive Network, CRAN, which is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. There is around 3500 packages in the Austrian CRAN repository, and the number of R packages grows exponentially (Fox, 2009). Figure 1.2 shows the fast development of R packages.

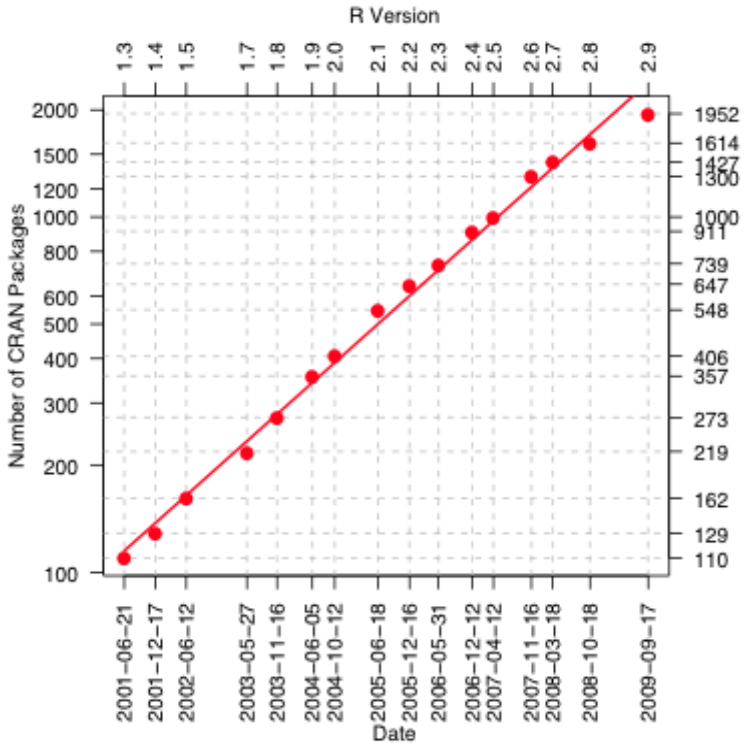

Figure 1.2: Number of R packages by date and version in a logarithm scale

A highly specialized R package considered as reference in the analysis of microarray data is `bioconductor`. The majority of the source code is written in R but also other languages

such as C, C++, Java have a place to improve the interaction between users and platforms. Looking at `www.bioconductor.org` it is possible to learn about the package, installation and features. Bioconductor cover two main purposes, a repository of software and data to develop and explore new statistical methods in genomics, and to standardize software, procedures and practices to allow portability in computer platforms around the world.

1.4 Example Data Set

The different methods will be introduced using a example data set, that contains a matrix of variables `x` and a data frame where class variables are stored in the object `xinfo`.

```
> dim(x)
```

```
[1]  81 643
```

```
> summary(xinfo)
```

```
 Pflanzennummer Behandlung Tag    Blatt
 13     : 3     Dry:33     0:15   3:27
 19     : 3     K  :48     3:30   4:27
 25     : 3                5:36   5:26
 27     : 3                       6: 1
 29     : 3
 31     : 3
 (Other):63
```

```
> x[1:5, 1:5]
```

```
         X458        X459        X460        X461        X462
1 0.003838085 0.006226730 0.008373729 0.010313984 0.011409971
2 0.003571557 0.005539229 0.007307842 0.008906148 0.009983964
3 0.001725531 0.003506811 0.005107889 0.006554793 0.007882202
4 0.001198527 0.002481380 0.003634453 0.004676493 0.005620934
5 0.000824366 0.002732455 0.004447513 0.005997421 0.007109025
```

The data set has `643` metabolic variables of `27` grapevine plants where the treatment variable is `Behandlung` divided in two `K,Dry` groups, irrigated or not irrigated. Variable `Tag` indicate three different time points `0,3,5` on which day the measure was taken.

The data set originates from an experiment carried out in a finger project by the Institut für Weinbau and *AnalitikZentrum*, IFA, Tulln.

	0	3	5
Dry	0	15	18
K	15	15	18

Table 1.1: Number of observations by treatment and time points

Chapter 2

Multivariate Outlier Identification

Multivariate Outlier Identification could be considered as the beginning or the end of high dimensional statistical analysis. The beginning because in a high dimensional data set the presence of outliers could introduce a bias in the final conclusions, or even worse a mislead, thus before starting any statistical analysis of it is necessary to clean the high dimensional data from the outliers. However an abnormal value out of any reasonable bounds will be considered as outlier even if the value corresponds to a special feature, when no outlier identification is used. Last but not least, methods to outlier detection are computationally intensive and the use in a high dimensional data set could be a problem.
The method presented here was developed by Filzmoser et al.(2008) . This procedure is implement in two phases, the first one is designed to detect location outliers calculating a "location" weight w_1 and the second one scatter outliers using a "scatter" weight w_2. The final weight of an observation i is defined as,

$$w_i = \frac{(w_{1i} + s)(w_{2i} + s)}{(1 + s)^2} \tag{2.1}$$

where s is a scaling constant with $s = 0.25$ which ensures a final weight of zero only if both phases give a low weight. A point i is classified as outlier if $w_i < 0.25$.

Detection of location outliers begins with a calculation of robustly sphering of the data as (2.2).

$$x^*_{ij} = \frac{x_{ij} - med(x_{1j}, ..., x_{nj})}{MAD(x_{1j}, ..., x_{nj})}, \ for \quad j = 1, ..., p \tag{2.2}$$

where n is the number of observations, p the number of variables, and MAD is the median absolute deviation defined as follows,

$$MAD(x_1, ..., x_n) = 1.4826 \cdot \underset{j}{med} \left| x_j - \underset{i}{med}\, x_i \right| \tag{2.3}$$

A semi robust PCA is applied in order to reduce dimensions from p to p^* and to retain only variables which explain at least 99 percent of the total variance. The resulting semi robust PCA scores are collected in the matrix $\mathbf{Z}^*$ with entries z^*_{ij} for $i = 1, ..., n$ and

$j = 1, ..., p^* \leq p$. Weights for the "dimension" outliers are calculated by the absolute value of a robust kurtosis measure.

$$\omega_j = \left| \frac{1}{n} \sum_{i=1}^{n} \frac{z_{ij}^* - med(z_{1j}^*, ..., z_{nj}^*)^4}{MAD(z_{1j}^*, ..., z_{nj}^*)^4} - 3 \right|, \quad j = 1, ..., p^* \tag{2.4}$$

This phase is continued by calculating a robust Mahalanobis distance to ensure an accurate classification between outliers and non-outliers. The second phase focuses on the detection of scatter outliers, using the semi robust PCA of the first phase and $\mathbf{Z}^*$ but the weight of the observations is done without a robust kurtosis measure.

We apply this outlier detection method to the example data set.

The left panel of Figure 2.1 shows the weights of each observation where observations with weights below 0.25 can be considered as outliers and below 0.05 as extreme outliers. 16 observations of the grapevine data set are outliers. At the right panel of the same figure weights of the metabolic variables are shown. 85 variables are considered as outliers and the weights of the non outliers are in a wide range until variable 300. The first that we can do to explain the 16 observations as outliers is to make a table plant number versus number of outlier observations.

```
> t(table(xinfo$Pflanzennummer, outliers_sample$wfinal01))
```

	13	19	25	27	29	31	32	33	34	35	36	37	38	4	42	43	47	48	49	52	53	54	55	58	59	60	7
0	1	0	0	0	0	3	3	2	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	1
1	2	3	3	3	3	0	0	1	3	2	3	3	3	2	3	3	2	3	3	2	3	3	3	1	3	3	2

It can be seen that plants 31 and 32 have all their samples as outliers and plants 33 and 58 two of three. Regarding the variables, it is easy to find which ones are outliers.

```
 [1] "X458" "X459" "X460" "X461" "X462" "X463" "X464" "X465" "X466" "X467"

 [1] "X468" "X469" "X470" "X471" "X472" "X473" "X474" "X475" "X476" "X477"

 [1] "X478" "X479" "X480" "X481" "X482" "X483" "X484" "X485" "X486" "X487"

 [1] "X488" "X489" "X490" "X491" "X492" "X493" "X494" "X495" "X496" "X497"

 [1] "X498" "X499" "X500" "X501" "X502" "X503" "X504" "X505" "X506" "X507"

 [1] "X508" "X509" "X510" "X511" "X512" "X513" "X514" "X515" "X516" "X517"

 [1] "X518" "X519" "X520" "X521" "X522" "X523" "X524" "X525" "X526" "X527"

 [1] "X528" "X529" "X530" "X531" "X532" "X533" "X534" "X535" "X536" "X537"

[1] "X538" "X539" "X540" "X541" "X542"
```

This helps to calibrate the analytical method, because these 85 were not measured in a proper way.

```
> library(mvoutlier)
> outliers_sample = pcout(x, makeplot = FALSE)
> outliers_var = pcout(t(x), makeplot = FALSE)
> par(mfrow = c(1, 2))
> par(mar = c(4, 2, 1, 1))
> plot(outliers_sample$wfinal, col = outliers_sample$wfinal01 + 1,
+ ylim = c(0, 1), xlab = "observation", ylab = "Weight", pch = 16)
> abline(h = 0.25)
> abline(h = 0.05, lt = 3)
> par(mar = c(4, 2, 1, 1))
> plot(outliers_var$wfinal, col = outliers_var$wfinal01 + 1,
+  ylim = c(0, 1), xlab = "variable",  ylab = "Weight", pch = 16)
> abline(h = 0.25)
> abline(h = 0.05, lt = 3)
```

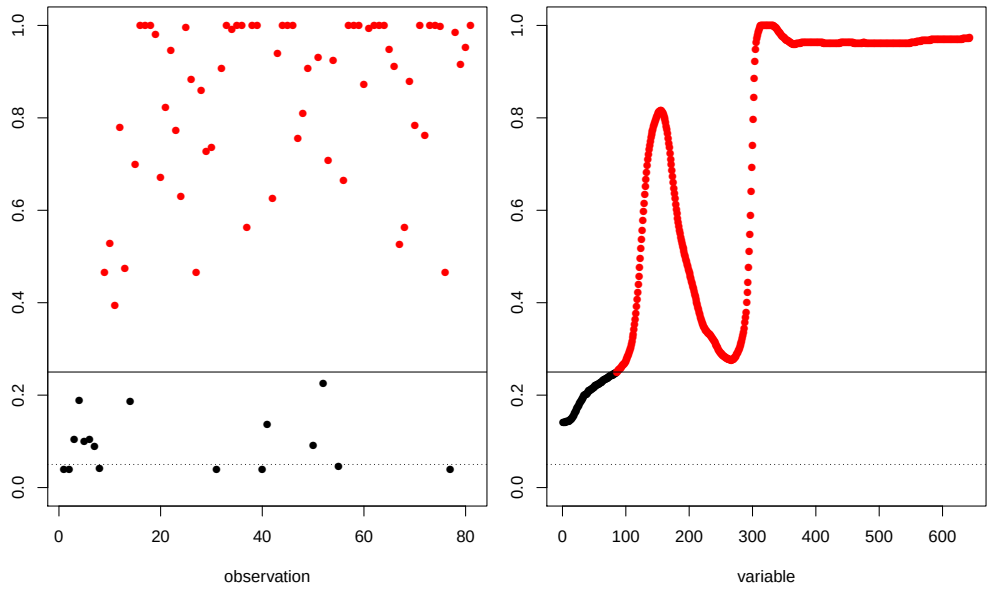

Figure 2.1: Outlier detection in Grapevine data. Red means non outlier vs black outlier. Observations , left, Variables, right

Chapter 3

Statistical Methods for Classification and Regression of High Dimensional Data

3.1 Linear Discriminant Analysis (LDA)

Given the group G which contains K class with $k = 1, 2, ..., K$ and $\mathbf{x}$ is the vector of observations, an observation belongs to a class k if the posterior probability is maximized.

$$P(G = k \mid \mathbf{x}) = \frac{h_k(\mathbf{x})\pi_k}{\sum_{l=1}^{k} h_l(\mathbf{x})\pi_l} \tag{3.1}$$

with $h_k(\mathbf{x})$ is the probability density function of $\mathbf{x}$ in the class $G = k$ and π_k is the prior probability which includes the previous information about the distribution that we know.

Assuming that $h_k(\mathbf{x})$ follows a multivariate normal distribution, with mean $\boldsymbol{\mu}_k$, variance-covariance matrix $\boldsymbol{\Sigma}_k$, and p dimensions, we rewrite $h_k(\mathbf{x})$ in terms of $\varphi(\mathbf{x})$

$$\varphi(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^p \left|\boldsymbol{\Sigma}_k\right|}} exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}{2}\right\}, \tag{3.2}$$

For LDA, we assume that the variance-covariance matrix $\boldsymbol{\Sigma}$ is equal for all groups,
i.e. $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ with $k = 1, 2, ..., K$. The log-ratio is needed to compare two groups as follows:

$$\begin{aligned} log\frac{P(G=k \mid \mathbf{x})}{P(G=l \mid \mathbf{x})} &= log\frac{\varphi_k(\mathbf{x})\pi_k}{\varphi_l(\mathbf{x})\pi_l} = log\frac{\varphi_k(\mathbf{x})}{\varphi_l(\mathbf{x})} + log\frac{\pi_k}{\pi_l} \\ &= log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\boldsymbol{\mu}_k+\boldsymbol{\mu}_l)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k-\boldsymbol{\mu}_l) + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k-\boldsymbol{\mu}_l) \end{aligned} \tag{3.3}$$

From expression (3.3) we define the linear discriminant function as follows,

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + log\pi_k \tag{3.4}$$

Equation (3.4) is a linear equation in $\mathbf{x}$ with slope $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$, and independent term $log\pi_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$ which gives a decision rule to classify an observation into group k.
One of the criticisms of this method is the estimation of $\boldsymbol{\Sigma}^{-1}$ that plays a key role in the linear discriminant function. For data with more variables than observations the inverse of the covariance matrix would not be computable. However, here a generalized inverse can be used. Moreover, the assumption of a normal distribution must hold.

In R using the grapevine data a linear discriminant analysis, `LDA` is implemented in the next code. First we need to install the R package `MASS` in order to use the function `lda`. This function uses as arguments a matrix with the variables, `x` and a vector indicating the different classes, here `Behandlung`. The option `cv=TRUE` produces a leave-one-out cross-validation, that uses a single observation as the validation data.

```
> library(MASS)
> res = lda(x, group = xinfo$Beh, cv = TRUE)
> lda.table = table(xinfo$Beh, predict(res)$class)
```

	Dry	K
Dry	33	0
K	0	48

Table 3.1: Predicted vs observed classes

Figure 3.1, shows the results of function `lda` using cross validation. The 81 samples were correctly assigned and no misclassification occurs. These results are quite perfect because all observations were used, in the next sections when other methods will be introduced we have the opportunity to compare LDA accuracy. See that discriminant values of group `Dry` are negatives and values of group `K` are positive.

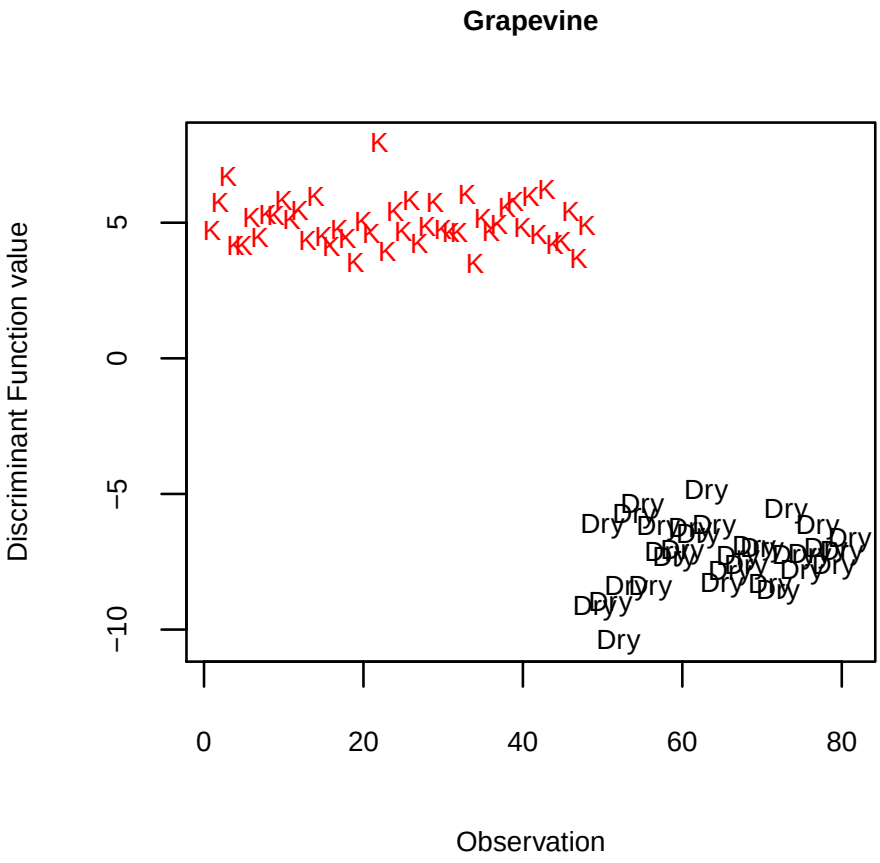

Figure 3.1: LDA for the Grapevine data.

3.2 Principal Components Regression (PCR)

Principal components analysis (PCA) is a method to find the main uncorrelated directions (components) of a multidimensional data set. This leads to a dimension reduction and allows the visualization of complex data. Only the most informative components are retained. PCA is also an initial step for other statistical procedures.

Given a mean-centered data matrix $\mathbf{X}$, with dimensions $n \times m$, the principal component transformation is defined as,

$$\mathbf{Z} = \mathbf{X} \cdot \mathbf{V},$$

where $\mathbf{V}$ is an orthogonal matrix, called loadings matrix, with columns $\mathbf{v}_i \quad i = 1, ..., m$ $\mathbf{v}_i^T \mathbf{v}_j = 0 \;\; i \neq j$, which are unit vectors, and $\mathbf{Z}$ is the transformed $n \times m$ dimensional matrix of scores.

After this transformation, we can use the first q components $1 \leq q \leq m$ in a regression model such that $\mathbf{X}$ describes a response variable $\mathbf{y}$.

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ &= \mathbf{X}\mathbf{V}\mathbf{V}^T\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ &= \mathbf{Z}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \end{aligned}$$

The regression coefficients are $\boldsymbol{\theta} = \mathbf{V}^T\boldsymbol{\beta}$ and the least squares estimators of $\boldsymbol{\theta}$ are, $\hat{\theta}_k = (\mathbf{z}_k^T\mathbf{z}_k)^{-1}\mathbf{z}_k^T\mathbf{y}$, $\hat{\boldsymbol{\theta}} = \left(\hat{\theta}_1, ..., \hat{\theta}_q\right)^T$ with $\mathbf{z}_k$ the k^{th} row of $\mathbf{Z}$. Given that the components of $\mathbf{Z}$ are orthogonal it is possible to write,

$$\begin{aligned} \hat{\mathbf{y}}_{PCR} &= \sum_{k=1}^{q} \hat{\theta}_k \mathbf{z}_k \\ \hat{\boldsymbol{\beta}}_{PCR}(q) &= \sum_{k=1}^{q} \hat{\theta}_k \mathbf{v}_k = \mathbf{V}\hat{\boldsymbol{\theta}} \end{aligned}$$

The R package **pls** will help us to fit a `PCR` model. The function performs internally a cross validation which is used to obtain predicted values. Figure 3.2 points out the reduction of *root mean squared error of prediction*, `RMSEP`, against the number of components in the `643` variables. Given a vector of observations $\mathbf{y} = (y_1, ..., y_n)^T$ and a vector of cross validated predictions $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_n)^T$

$$RMSEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$$

with $e_i = y_i - \hat{y}_i$ $i = 1, ..., n$ and n the number of predictions. Large values of RMSEP imply lack of accuracy in the predictions of the proposed model.

3.3 Partial Least Squares Regression (PLSR)

In a regression model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{y}$ is a vector of n observations, $\mathbf{X}$ is a matrix of n observations and p predictors, $\boldsymbol{\beta}$ is the vector of regression coefficients and $\boldsymbol{\varepsilon}$ the residual vector, the aim of `PLSR` is to predict $\mathbf{y}$ from $\mathbf{X}$. The model can be presented in terms of a latent variable model such that

$$\mathbf{y} = \mathbf{T}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \tag{3.5}$$

- The $\boldsymbol{\gamma}$ coefficients have $q \leq p$ entries.
- The matrix $\mathbf{T}$ is a $n \times q$ matrix.
- The dimension reduction leads on a stable regression of $\mathbf{y}$ on $\mathbf{T}$.
- $\mathbf{T}$ is not directly observable, but it can be computed sequentially, for $k = 1, 2, ..., q$, following the `PLS` criterion,

  $$\mathbf{a}_k = \underset{\mathbf{a}}{argmax} Cov\left(\mathbf{y}, \mathbf{X}\mathbf{a}\right)$$

  with $\|\mathbf{a}_k\| = 1$ and $Cov\left(\mathbf{X}\mathbf{a}_k, \mathbf{X}\mathbf{a}_j\right) = 0$ for $1 \leq j < k$

The vectors $\mathbf{a}_k$ with $k = 1, 2, ..., q$, called loadings, are stored as columns in a matrix $\mathbf{A}$. The score matrix is defined as

$$\mathbf{T} = \mathbf{X}\mathbf{A}.$$

Then the regression problem (3.5) can be written as

$$\begin{aligned} \mathbf{y} &= \mathbf{T}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \\ &= (\mathbf{X}\mathbf{A})\,\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \\ &= \mathbf{X}\underbrace{(\mathbf{A}\boldsymbol{\gamma})}_{\boldsymbol{\beta}} + \boldsymbol{\varepsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \end{aligned}$$

Now the name "*partial*" is easily understandable, because the procedure makes a regression using $\mathbf{X}$ but weighted by $\mathbf{A}\boldsymbol{\gamma}$ with a dimension reduction. The `PLS` regression uses $\mathbf{y}$ to calculate the different directions and for this reason it is better to use `PLSR` rather than `PCR` for prediction purposes.

Table 3.2 compares the prediction ability between `PCR` and `PLSR`, in the case of 10 components. Following this idea, Figure 3.3 presents the proportion of correct assignments for different numbers of components.

The accuracy and prediction ability improvement of `PLSR` over `PCR` have been evaluated. Using the same numbers of components, `PLSR` provides better assignment and lower error rates.

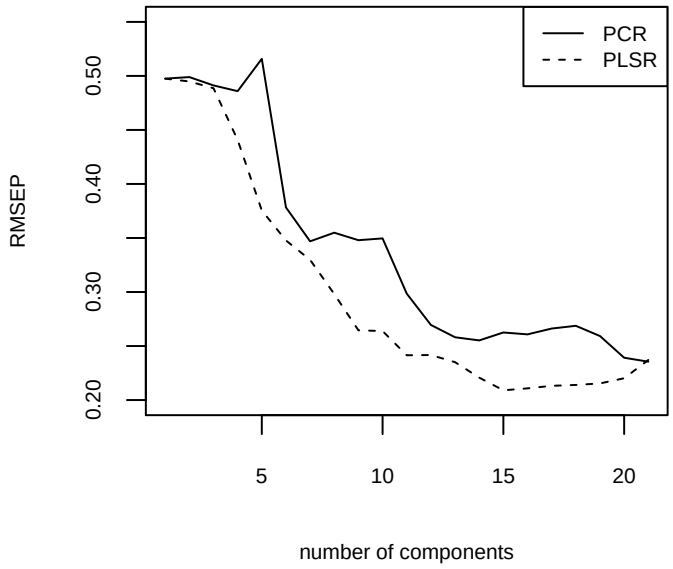

Figure 3.2: `RMSEP` vs Number of components in Grapevine data.

	PCR		PLS	
	Dry	K	Dry	K
Dry	31	2	32	1
K	2	46	0	48

Table 3.2: Prediction table using 10 components.

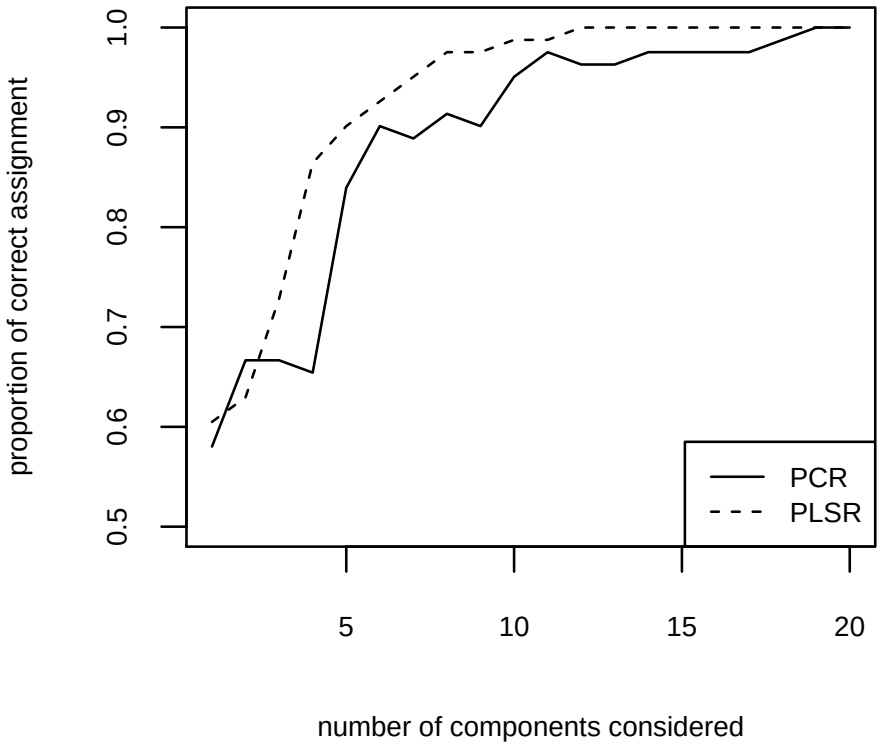

Figure 3.3: Proportion of perfect assignment `PCR`, solid, vs `PLSR`, dashed, in Grapevine data.

3.4 Sparse Partial Least Squares (SPLS)

One step further of Partial Least Squares Regression is to use a sparsity constraint in the sense of carrying out a simultaneous variable selection and dimension reduction (Le Cao et al., 2008). This is necessary in the case of high dimensional data where the number of variables, p is larger than the number of observations, n. The idea is to penalize with a sparsity constraint the loading vectors $\mathbf{a}_k$ which indicate the relative importance of the variables, if this constraint is set to zero the results will be the same as classical PLS.

Although SPLS is mainly used in regression problems it can be applied to classification problems. The qualitative response is recoded in a dummy variable that records the membership of each observation and in this way an SPLS Discriminant Analysis (SPLSDA) performs classification and variable selection in a one step procedure.

The work of (Le Cao et al., 2011) using the R package `mixOmics` proves that SPLSDA has a good classification performance in the analysis of biological data sets.

We divide the `81` grapevine data observations in two groups randomly, train, with 25 observations and test. Train will be used to calculate the parameters of `SPLSDA` and test will be used to predict. This procedure will be run several times, 20 replicates, and each time a new set of train data will be used. At the end the mean of perfect assignment is returned, showing that with only one third of the observations and 6 components reasonable results are obtained.

```
> library(mixOmics)
> ncom = 6
> nsamp = 20
> splsda.prop <- matrix(nrow = nsamp, ncol = ncom)
> for (i in 1:nsamp) {
+     train <- sample(81, 25, replace = FALSE)
+     for (j in 1:ncom) {
+         splsda.train <- splsda(x[train, ], xinfo$Beh[train], ncomp = ncom,
+         keepX = rep(30, ncom))
+         test.predict <- predict(splsda.train, x[-train, ], method = "max.dist")
+         aa <- table(xinfo$Beh[-train], test.predict$class$max.dist[, j])
+         splsda.prop[i, j] = sum(diag(aa))/sum(aa)
+     }
+ }
> round(apply(splsda.prop, 2, mean), 3)

[1] 0.558 0.608 0.654 0.736 0.805 0.838
```

The same idea is used to produce Table 3.3, where different methods are compared using the same data and number of components. For LDA, after 20 replicates we obtain a mean percentage of correct assignments of 0.846.

	1 comp	2 comp	3 comp	4 comp	5 comp	6 comp
PCR	0.53	0.54	0.53	0.58	0.77	0.80
PLSR	0.53	0.60	0.66	0.76	0.81	0.82
SPLSDA	0.62	0.54	0.77	0.81	0.87	0.89

Table 3.3: Percentage of correct assignment table using different number of components and the same observations

3.5 Penalized Regression (PLogReg)

An important issue when we are dealing with high dimensional data is to know which variables are really significant in order to explain the trait of interest. A first approach is to solve

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.6}$$

where $\mathbf{y}$ is a vector of n observations, $\mathbf{X}$ is a matrix of n observations and p predictors, $\boldsymbol{\beta}$ is the vector of regression coefficients and $\boldsymbol{\varepsilon}$ the residual vector. The least squares estimator of the regression coefficients is defined as follows,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.7}$$

Equation (3.7) in the case of high dimensional data is not appropriate to solve the problem. Some columns, $\mathbf{x}_i$, of $\mathbf{X}$ could be collinear $\mathbf{x}_i = \mathbf{x}_j + \mathbf{x}_k$ and also when the number of variables p is greater than the number of the observations n, $p \gg n$, there will be overfitting of the data because there are many, thousands, or infinitely choices of $\boldsymbol{\beta}$ which fit the data perfectly. This is more relevant in the case of genetics, where one of the main aims of a genetic study is to detect significant genes or markers from a high dimensional set of them.

The general approach consists in a modification in the sum of squared errors criterion via a penalization (Witten and Tibshirani, 2009), which can be generalized as,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{arg\,min} \left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 + \lambda_1 \left\|\boldsymbol{\beta}\right\|^{p_1} + \lambda_2 \left\|\boldsymbol{\beta}\right\|^{p_2} \tag{3.8}$$

with $\left\|\boldsymbol{\beta}\right\|^k = \sum_{i=1}^{p} \left|\boldsymbol{\beta}\right|^k$ and p the number of variables. In equation (3.8) $\lambda_i \geq 0,\ i = 1, 2$, are tuning parameters which regulate the strength of the penalty and the exponents p_1 and p_2, define the penalty, e.g. $\lambda_2 = 0,\ p_1 = 0$ gives best subset selection, $\lambda_2 = 0,\ p_1 = 1$ lasso regression (Tibshirani, 1996) and $\lambda_2 = 0,\ p_1 = 2$ ridge regression

Penalized Regression, in the different forms, e.g. lasso or ridge, is widely used when the outcome or trait is continuous such as the amount of RNA or a quantitative trait in the genomic selection of farm animals. Given the importance of these methods, they will be explained in this section but a detailed explanation is available in Hastie et al.(2009).

The estimated coefficients of the Lasso regression can be rewritten, following formula (3.8) in form of expression,

$$\hat{\boldsymbol{\beta}}^{lasso} = \underset{\boldsymbol{\beta}}{arg\,min} \left\{ \frac{1}{2}\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3.9}$$

The Lasso penalty, L_1, is the term $\sum_{j=1}^{p} |\beta_j|$. When L_1 is small enough, Lasso computes some coefficients equal to zero, then these variables are not significant in order to explain the data. λ is the variable which controls the amount of penalty or shrinkage. The larger

λ the bigger the penalization. Computational approaches are available and provide a set of solutions as λ is varied.
Using formula (3.8), the estimated coefficients of the Ridge regression are,

$$\hat{\boldsymbol{\beta}}^{ridge} = \underset{\boldsymbol{\beta}}{arg\,min} \left\{ \frac{1}{2}\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{3.10}$$

The term $\sum_{j=1}^{p} \beta_j^2$ defines the Ridge penalty, L_2. The idea is to avoid the poor determination and high variance of correlated variables in the regression. When one coefficient has a large positive value, the correlated variable has also a large coefficient but negative, which cancels the effect of both variables. Equation (3.7), under the point of view of Ridge regression can be seen as,

$$\hat{\boldsymbol{\beta}}^{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.11}$$

with $\mathbf{I}$ the $p \times p$ identity matrix. λ adds a positive value at the diagonal of $\mathbf{X}^T\mathbf{X}$ to prevent problems of singularity.
An intermediate approach is the elastic net proposed by Zou and Hastie,(2005), with penalty term as follows,

$$\lambda \sum_{j=1}^{p} (\alpha\beta_j^2 + (1-\alpha)\,|\beta_j|) \tag{3.12}$$

where α is a parameter which gives more weight to the L_1 or L_2 term. When $\alpha = 1$ the elastic net is equal to ridge regression and with $\alpha = 0$, it produces a lasso regression. The elastic net select variables as lasso and penalizes correlated variables as ridge.
When the dependent variable is qualitative and binary a logistic approach is needed. Logistic regression is a particular case of the Generalized Linear Model,

$$\mathrm{E}(\mathbf{y}) = \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta})$$

where the link function g is the logit function $\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}}\right)$ and $\boldsymbol{\mu}$ defined as,

$$\boldsymbol{\mu} = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$$

The R package `glmnet` fits elastic net problems for regression but also logistic and multinomial regression. The function `cv.glmnet` calculates the optimal λ which minimizes the deviance using by default 10-fold cross-validation. After that a penalized model is fitted with the calculated λ with the function `glmnet`. This function has as default parameter $\alpha = 1$ then a lasso penalty is used because the elastic net penalty is defined as $(1-\alpha)\beta_j^2 + \alpha\,|\beta_j|$.
Using this R package, a penalized logistic regression is fitted, with the aim to find the explanatory variables in the grapevine data set, as the following R code shows.

The optimal λ is 0.026.

The binomial deviance is defined as,

$$D = -2\ln\frac{\text{(likelihood of the fitted model)}}{\text{(likelihood of the saturated model)}} \tag{3.13}$$

```
> library(glmnet)
> res.cvpenalized <- cv.glmnet(x, xinfo$Beh, family = "binomial", maxit = 5000)
> res.penalized <- glmnet(x, xinfo$Beh, family = "binomial", maxit = 5000)
> pen.coeff = coef(res.penalized, s = res.cvpenalized$lambda.min)
> nonzerovar = which(pen.coeff != 0)
> expl.coef = pen.coeff[nonzerovar]
> allnames = colnames(x)
> nonzerovar = nonzerovar - 1
> names = c("Intercept", allnames[nonzerovar])
> markers = matrix(expl.coef, nrow = 1, ncol = length(expl.coef))
> colnames(markers) = names
> plot(res.cvpenalized)
```

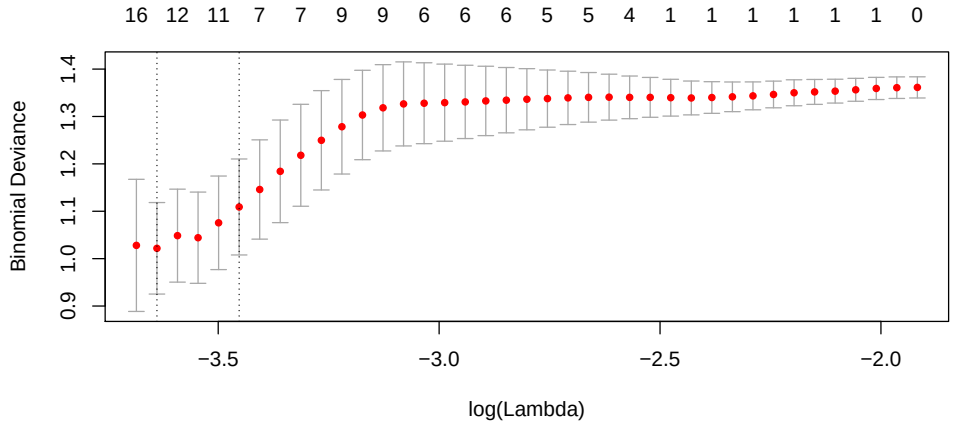

Figure 3.4: Binomial deviance vs λ for the grapevine data

Now we present the names of the variables included in the model.

```
Intercept      X468      X705      X706      X707      X708
   -0.915    73.857   -34.884   -32.858   -26.850   -10.681

   X709     X710     X747     X844     X845     X846
 -8.674   -6.558  138.050   -0.502   -1.232   -1.216

   X847     X857     X858    X1100
 -1.201  -25.279  -14.676   41.090
```

Here we obtain a drastic variable reduction from 644 to 16 variables.

Table 3.4 presents the observed versus predicted class membership using penalized logistic regression with a true assessment rate of 0.9. Sampling 26 observations of 81, 30 times gives

	Dry	K
Dry	26	7
K	1	47

Table 3.4: Observed vs predicted classes

a mean of true assessment of 0.716 From these samples the median number of variables selected by penalized logistic regression is 13.

The previous R code presented is the basis to sample again 30 times but using different values of α, from $\alpha = 0$, ridge regression, to $\alpha = 1$, lasso regression, as the function `glmnet` has been defined.

Table 3.5 shows how the modification of α gives an improvement of true assessment and a

	0	0.25	0.5	0.75	1
mean of true assessment	0.692	0.793	0.774	0.758	0.75
median number of variables	643	238	154	92	11

Table 3.5: Elastic net results as α change between 0 and 1.

larger number of variables used to explain if the grapevine plant was irrigated or not. Now we can run again the `glmnet` function, with $\alpha = 0.25$ to find which markers are important. 213 markers were found with non zero coefficients, see Figure 3.5.

When the weights of the outlier detection method is plotted against the estimated coefficients, we obtain Figure 3.6. Weights between 0 and 0.25 denote outlier. As we can see, variables classified as outliers have larger estimated coefficients than non outliers. The information coming from the elastic net approach and multivariate outlier detection has a great relevance to understand the data.

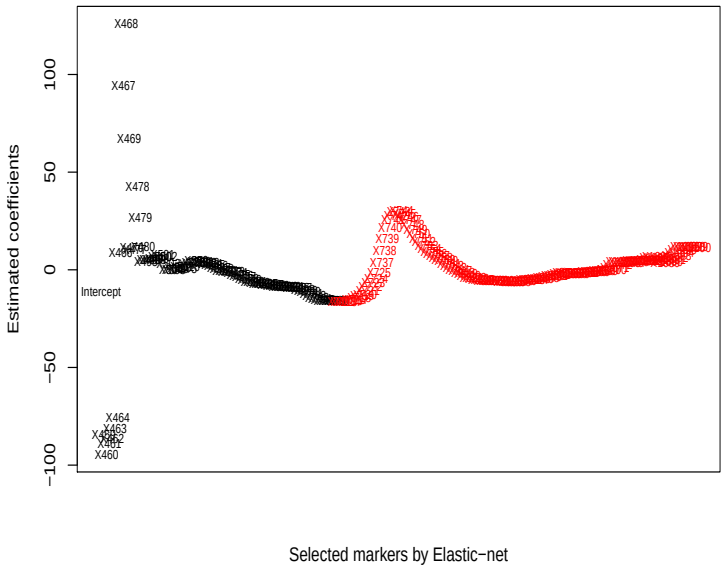

Figure 3.5: Selected markers by elastic-net with $\alpha = 0.25$. Black markers were classified as outliers in a previous section

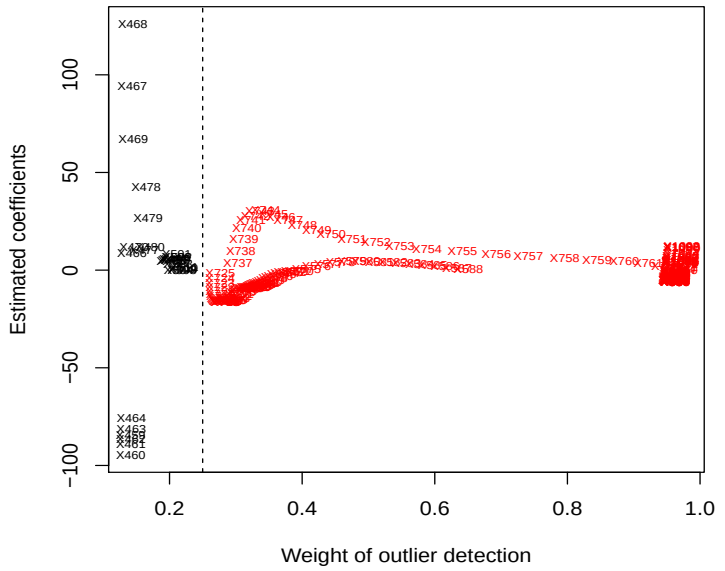

Figure 3.6: Selected markers by elastic-net with $\alpha = 0.25$ vs Weight. Black markers were classified as outliers in a previous section

Chapter 4

Cluster Analysis

Cluster analysis is a group of methods that mimics the human thought which tries to makes groups, clusters, to decompose the reality that is difficult to explain. Each of these groups are more easily understandable than the original reality. In this chapter we make a brief explanation of cluster analysis methods for further details the book Draghici,(2011) covers in detail this topic.
Cluster analysis is appropriate when there is no previous knowledge about the data. This method needs two things to be implemented, a measure of similarity and a procedure (algorithm) to make the groups.

The measure of similarity is made using a distance metric, d. A distance metric is a function in a n-dimensional space $\mathbb{R}^n$ of two points $\mathbf{x}$ and $\mathbf{y}$ which holds the following properties:

Symmetry. The distance should be symmetric

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$$

Positivity. The distance should be equal or greater than zero

$$d(\mathbf{x}, \mathbf{y}) \geq 0$$

Triangle Inequality. Given 3 points, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$$

There are many different distances that hold the previous three properties and we are going to present some of them. In two n dimensional vectors, $\mathbf{x} = (x_1, x_2, ..., x_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_n)$ the following distances can be applied.

Euclidean distance. $d_E(\mathbf{x},\mathbf{y}) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$

Manhattan distance. $d_M(\mathbf{x},\mathbf{y}) = |x_1-y_1|+|x_2-y_2|+...+|x_n-y_n| = \sum_{i=1}^{n}|x_i-y_i|$

Correlation distance. $d_R(\mathbf{x},\mathbf{y}) = 1-r_{xy}$ where r_{xy} is the Pearson correlation coefficient of the vectors $\mathbf{x}$ and $\mathbf{y}$.

Mahalanobis distance. $d_{Ml} = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})}$ where $\mathbf{S}$ is the variance covariance matrix. If $\mathbf{S}$ is the identity matrix, I, the Mahalanobis is equal to the classical Euclidean distance.

We have presented the most usual distances used in cluster analysis. Now we need to define a procedure, algorithm, to make clusters. Similar patterns grouped together by the algorithm form clusters and the main criticism of this method is that given enough genes (variables) the genes will always cluster. Also clustering is not deterministic, in the sense that the same clustering algorithm applied to the same data may produce different results.

The k-means clustering is one of the most widely used algorithms due to their simplicity and fast. The algorithm needs the number of clusters, k, as an input and after this the algorithm chooses randomly k points as the centers of the clusters. The algorithm will calculate the distance from each point to the cluster centers and the points are included into the clusters looking at the closest distance. The cluster centers will be updated with the elements of each cluster. Since the centers have been recalculated it is necessary to update the memberships into the clusters. This algorithm will terminate when no point moves from one cluster to another.

The hierarchical clustering algorithm became popular at the beginning of the microarray area. As the name indicates, this algorithm not only produces clusters but also a hierarchy. The result is a tree where the leaves are the individual patterns (variables, genes or experiments) and the root the convergence point of all branches. The tree can be built from bottom to top, bottom-up, or from top to bottom, top-down. There are defined four distances between clusters, single linkage clustering, distance between the closest neighbors, complete linkage, farthest neighbors, centroid linkage, distance between the centers of the clusters or average linkage, the average distance of all elements in each cluster. The speed of the algorithm depends on their complexity and then on the linkage choice. The hierarchy is presented as a dendrogram and in a heatmap which is a color chart where colors represent the values of the objects for each case.

The R package `mixOmics` with the function `cim` plots a heat map using as distance the default settings of `dist`, Euclidean, and hierarchical cluster algorithm the default of `hclust`, complete. This function applied to the grapevine data produces the following heatmap.

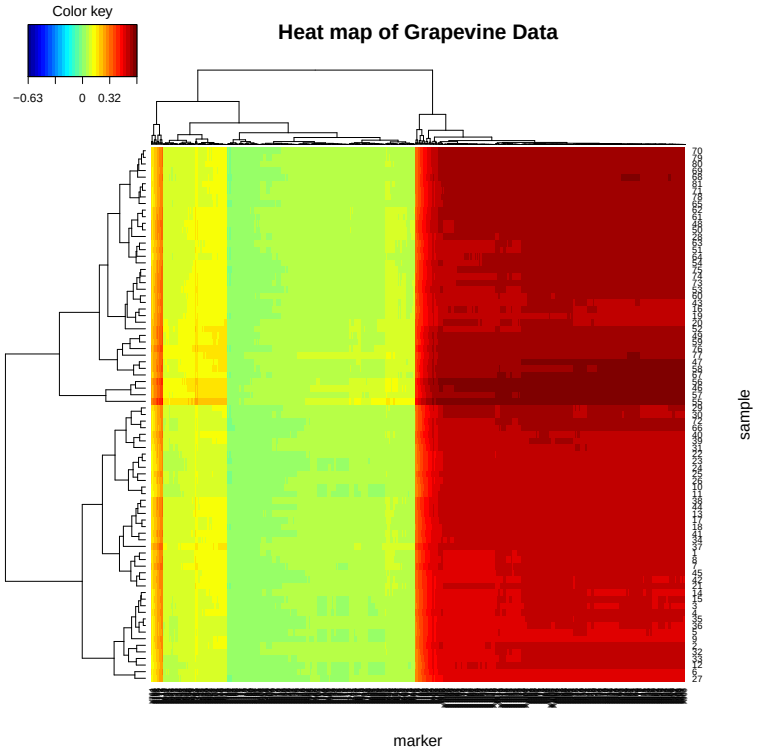

Figure 4.1: Heat map of Grape Vine data.

In Figure 4.1, first there are three different areas looking at markers. Remember the left panel of Figure 2.1 where after variable number 300 weights are close to one. It seems that the heatmap is a bird's eye view of Figure 2.1. There is a dark red area at the top bottom of the figure. The next two figures are made changing distance and hierarchical cluster method in order to see the difference.

Using the cophenetic correlation coefficient we can compare dendrograms (Sneath and Sokal, 1973). 0.704 is the correlation between the dendrogram in Figure 4.1 and Figure 4.2, and 0.977 the correlation with the dendrogram in Figure 4.3. Between the dendrogram in Figure 4.2 and the dendrogram in Figure 4.3, the correlation is 0.69.
The dendrograms in Figure 4.1 and in Figure 4.3 are more similar than the dendrogram in Figure 4.2.

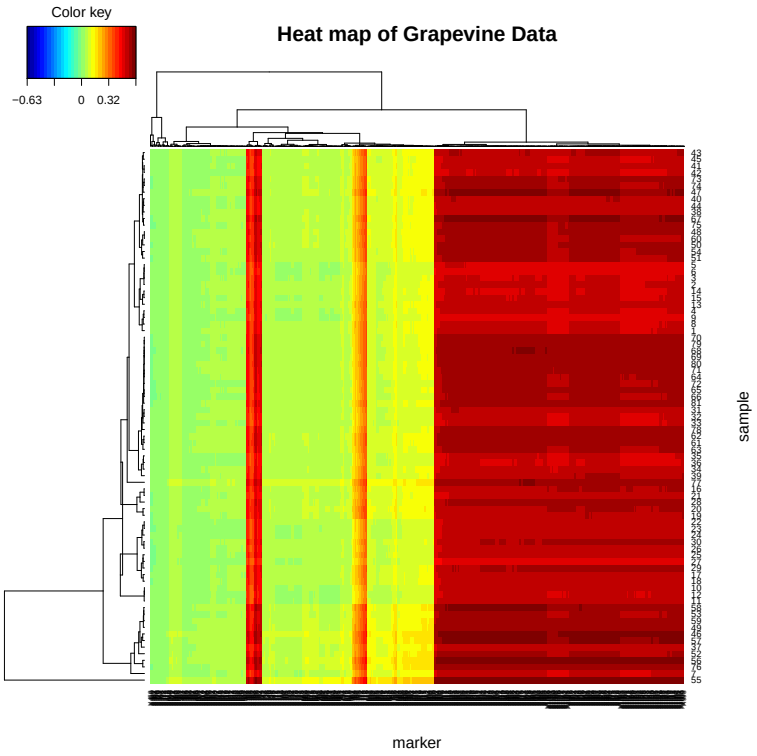

Figure 4.2: Heat map of Grape Vine data, changing cluster algorithm (centroid) and distance (correlation).

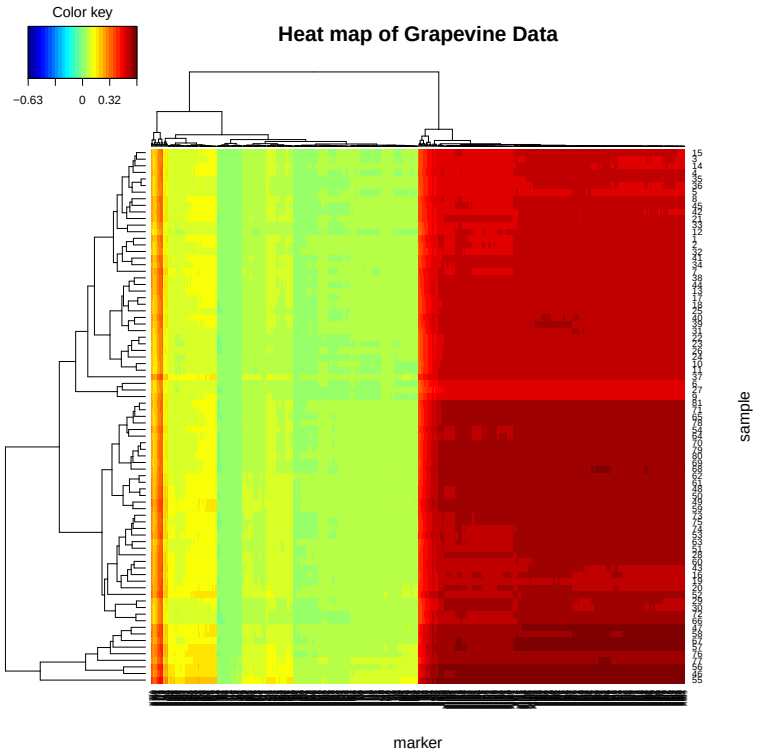

Figure 4.3: Heat map of Grape Vine data, changing cluster algorithm (average) and distance (Manhattan).

Chapter 5

Freak variables detection

Research groups are always looking for special variables with a large effect in the variable of interest that they study. These kinds of variables are like the philosopher's stone in each research field and usually lead to the researcher's eureka effect. Although the methods presented in this chapter came from the genomics field and focused in gene expression, they can be straightforwardly applied to other research fields.
Given a matrix of data $\mathbf{X}_{n\times p}$ where each column, p, represents a variable and each row, n, is an observation and a vector of features, treatments or disease, $\mathbf{Y}_{n\times 1}$, the first idea could be compare by columns a mean difference between the elements groups or levels of $\mathbf{Y}$.

The next R code simulates this situation.

First we simulate in the object `d` 100×500 values of a normal distribution $N(0, 1)$. These values are allocated in a matrix called `cosa`. The matrix `cosa` is our data matrix, with 100 rows and 500 columns. Afterwards a vector with two groups is simulated in `feature` which could represent e.g. two different treatments, disease or no disease. Each column of `cosa` is labeled and finally we have our simulated data.

```
> d = rnorm(100 * 500)
> cosa = matrix(d, nrow = 100, ncol = 500)
> feature = c(rep("A", 50), rep("B", 50))
> feature = as.factor(feature)
> nombre = paste(seq(1, 500, 1), c("M"), sep = "")
> colnames(cosa) = nombre
> findx = function(x) {
+     A = x[feature == "A"]
+     B = x[feature == "B"]
+     p = t.test(A, B, var.equal = FALSE)$p.value
+     foldchange = mean(A) - mean(B)
+     c(foldchange, p)
+ }
> results = t(apply((cosa), 2, findx))
```

```
> colnames(results) = c("FC", "p")
> results = results[order(results[, 2]), ]
> head(results, 10)
```

```
             FC           p
32M  -0.6237323 0.001517900
497M  0.5977804 0.001870183
449M  0.5269343 0.004121771
384M  0.5680131 0.004180301
317M -0.5671985 0.005549734
395M -0.5669798 0.007358838
452M  0.4626617 0.012186182
274M  0.5221724 0.016454818
467M  0.4855616 0.017254258
351M  0.4747705 0.019282158
```

The function `findx` performs a t-test assuming different variances per group, *Welch*-test, returns the p-value, `p` and the difference of the means, `FC`.
The null hypothesis of the t-test, H_0, assumes no differences in values between groups, in our example A and B, on the other side the alternative hypothesis, H_1, assumes differences in values between groups. The p-value is the probability to reject the H_0 when it is true and this is the Type I error and also defined as family-wise error rate, FWER.
Looking at the first 10 results ordered by p-value, we can see that we have found some variables with statistical difference within the two groups.

Given that our level of significance α is equal to 0.05, there is a probability of 5% to find variables with differences between groups even if this difference is not true. Let us note that we have simulated 500 variables, but in a high dimensional problem with thousands of variables the amount of falsely discovered variables will be out of control. We should make a false discovery rate, FDR, (Benjamini and Hochberg, 1995) correction, which is a multiple comparisons correction of the p-values.
The FDR correction procedure is used in microarray data where there are dependencies between genes and FDR takes into account some dependencies. The procedure works as follows:

1. Calculate the individual p-value for each of the p variables at a significance level α.

2. Sort the p-values in increasing order

$$p_1 < p_2 < ... < p_j < ... < p_p$$

3. Compare the p-values of each gene with a threshold such as,

$$p_1 < \frac{1}{p}\alpha, \;\; p_2 < \frac{2}{p}\alpha, \;\; ..., \;\; p_j < \frac{j}{p}\alpha, \;\; ... \;\; p_p < \frac{p}{p}\alpha$$

4. Find the largest value of j for which $p_j < \frac{j}{p}\alpha$ holds.

5. The null hypotheses of variables from 1 to j should be rejected.

There is an R function `p.adjust` which makes the correction using different methods, for FDR we use `fdr`.

```
> adjustedp = p.adjust(results[, 2], method = "fdr")
> results = cbind(results, adjustedp)
> head(results, 10)
```

```
             FC           p adjustedp
32M  -0.6237323 0.001517900 0.4675459
497M  0.5977804 0.001870183 0.4675459
449M  0.5269343 0.004121771 0.5225377
384M  0.5680131 0.004180301 0.5225377
317M -0.5671985 0.005549734 0.5549734
395M -0.5669798 0.007358838 0.6132365
452M  0.4626617 0.012186182 0.7951304
274M  0.5221724 0.016454818 0.7951304
467M  0.4855616 0.017254258 0.7951304
351M  0.4747705 0.019282158 0.7951304
```

Unfortunately after the correction no significant variable was found as we expected because the data was simulated in this way. The example wanted to warn about the *black-box* use of statistics and also presents a concept that other methods use to detect significant variables.

We apply the previous procedure to our grapevine data. After this correction we do not find any significant features.

```
> mytest = function(x) {
+     aK = x[xinfo$Behandlung == "K"]
+     aDry = x[xinfo$Behandlung == "Dry"]
+     p = t.test(aK, aDry, var.equal = FALSE)$p.value
+     foldchange = mean(aK) - mean(aDry)
+     c(foldchange, p)
+ }
> results = t(apply(x, 2, mytest))
> colnames(results) = c("FC", "p")
> results = results[order(results[, 2]), ]
> adjustedp = p.adjust(results[, 2], method = "fdr")
> results = cbind(results, adjustedp)
> head(results, 10)
```

```
              FC           p adjustedp
X468 0.003167703 0.004140455 0.6363523
X467 0.002943598 0.004389449 0.6363523
X469 0.003093413 0.005617367 0.6363523
X466 0.002620901 0.005631598 0.6363523
X470 0.003026154 0.007541757 0.6363523
X465 0.002269707 0.008630089 0.6363523
X471 0.002965001 0.009957671 0.6363523
X463 0.001733794 0.012001435 0.6363523
X464 0.001959100 0.012384382 0.6363523
X462 0.001486424 0.012425251 0.6363523
```

5.1 Permutation Test

The previous introduction of this chapter has shown the importance of controlling the Type I error. Another issue to consider is the dependence between variables when we are dealing with high dimensional data.
The analysis of biological data such as gene expression, metabolic pathways or other complex processes needs to consider the possible correlation between variables.
The Westfall and Young step down correction (Westfall and Young, 1993) adjusts the p-value in a general way taking into account the correlation. This correction begins with a permutation of the levels of the factor of interest. The word permutation came from the Latin word *permutatio*, which means *change, exchange*, and this is the point, the random exchange of the levels of the factor to analyze and the test of each one. The test used here is a t-test with a correction for multiple comparisons.
Repeating the process thousand or tens of thousand times, we will have a p-value which is the proportion of times the value of the random permutation test is less or equal to the value of the original test.
This method takes into account the correlation between genes (variables) and this is the main advantage. However the computational cost of the random exchange of the levels of the factor of interest and the empirical nature of the procedure are the two main objections.

The `multtest` package in `bioconductor` performs the Westfall and Young step down multiple testing procedure with the function `mt.maxT`.
This function has as arguments the data matrix, the class labels as vector and the number of permutations. We apply this to our Grapevine data.

```
> library(multtest)
> res = mt.maxT(X = t(x), classlabel = xinfo$Behandlung, B = 1000)
> head(res, 10)

     index teststat  rawp  adjp
X468    11 2.953525 0.008 0.029
X467    10 2.933368 0.008 0.032
X469    12 2.848444 0.009 0.036
X466     9 2.846696 0.009 0.036
X470    13 2.744728 0.012 0.043
X465     8 2.694004 0.010 0.051
X471    14 2.644475 0.017 0.057
X463     6 2.571513 0.017 0.072
X464     7 2.559976 0.015 0.075
X462     5 2.558981 0.015 0.075
```

As we can see, only few variables are considered significant after the Westfall and Young correction. Figure 5.1 gives aditional information about the significant variables. The few variables reported as significant were in a previous analysis clasified as outliers.

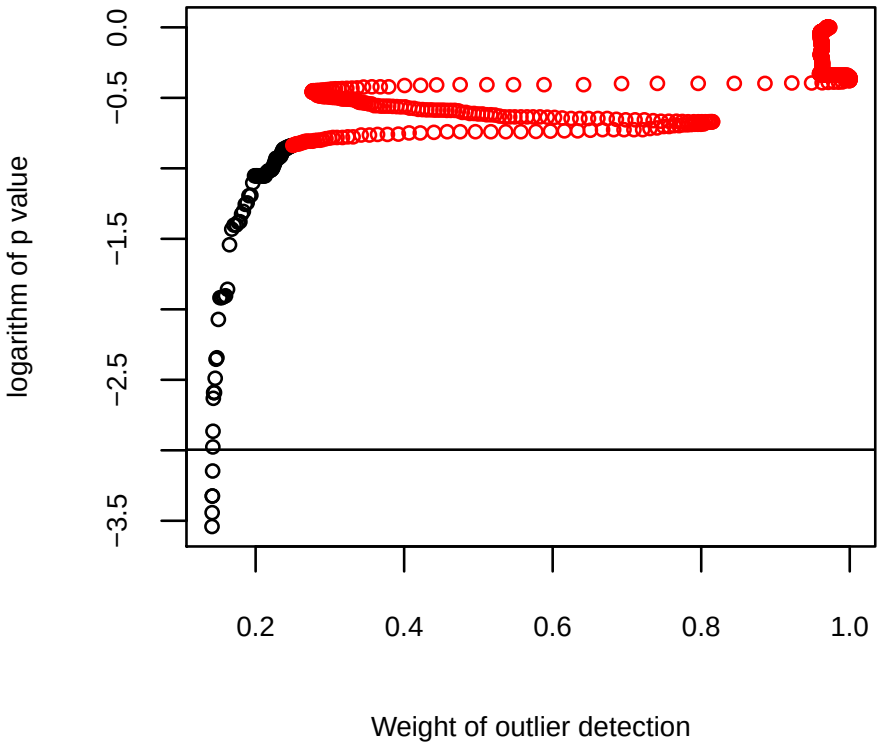

Figure 5.1: Multivariate outlier weights vs corrected p-values, `mt.maxT`, grapevine data. Outliers in black and non outliers in red.

5.2 Significance Analysis of Microarrays (SAM)

The significance analysis of microarrays method, SAM, (Tusher et al., 2001) was develop as response to the rigorous approach of Westall and Young (1993). SAM is based on a statistic which uses the difference between means divided by an estimate of the standard deviation which is the basic idea of a t-test.
The statistic calculated by SAM for a gene i is defined as follows,

$$d_i = \frac{\bar{x}_{iA} - \bar{x}_{iB}}{s_i + s_0} \tag{5.1}$$

where the numerator is the difference between the means of the A and B group and the denominator is the sum of a standard deviation and a tunning term, s_i and s_0. The standard deviation, s_i, is the square root of the pooled sample variance, as in a t-test asuming homoscedasticity.
s_0 is a tunning term to avoid $d_i \longrightarrow \infty$ when s_i is becoming too small. If s_0 is equal to zero, d_i becomes a t-test.

SAM selects genes taking into account their variance and the expression levels, the difference between means has less influence in the calculated statistic. In this way SAM has a statistic less constrained but the problem about multiple comparisons exists and thus a permutation procedure is used to control the false discovery rate.
The R package `samr` performs a SAM analysis. The first step is to provide the data in a list with the data matrix, `x`, of p genes in the rows and n samples or observations in the columns. The observed trait or outcome given in a vector `y` is the second element of the list. Names of genes and a logical value indicating if the matrix `x` is log2 transformed or not shall also be provided.
The function `samr` computes the d_i statistic using the data provided and the number of permutations in `nperms`, but `samr` computes the maximum number of possible permutations.

```
> library(samr)
> Dat = x
> data = list(x = t(Dat), y = xinfo$Behandlung, genenames = rownames(Dat),
+ logged2 = F)
> samr.obj = samr(data, resp.type = "Two class unpaired", nperms = 20,
+  random.seed = 123)
> delta.table = samr.compute.delta.table(samr.obj, min.foldchange = 1)
```

In the area of life sciences a variable, gene, with a very low p-value but with small fold change needs caution. For this reason a minimum fold change should be provided and this is given by `min.foldchange` in the computation of Δ.
The value of Δ is a threshold of significance in the sense of a distance from the statistic d_i to the same value computed via permutations.
In the case of our grapevine data the Δ value reported by `samr.compute.delta.table` when the median FDR was equal to zero, was 0.83. Now with the value of minimum fold change

and Δ we can plot and detect significant variables (genes).

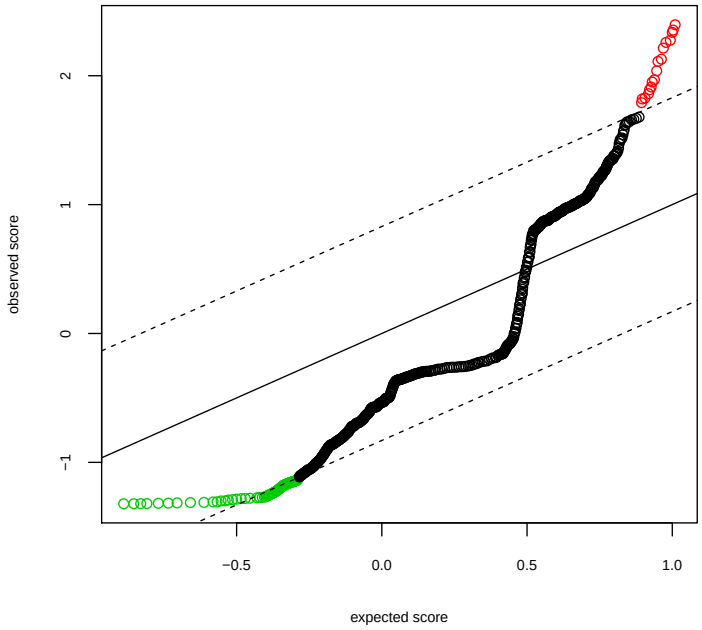

Figure 5.2: SAM plot of significance with Δ equal to 0.83. Variables in color have a difference greater than Δ. Green down regulation and red up regulation.

Figure 5.2 presents variables up or down regulated, in color, and no significant variables in black. Using the grapevine data we found 17 variables up regulated and 85 down regulated. The next step is to know which variables were colored and this is performed by the function `samr.compute.siggenes.tables`. When we compare the up regulated variables with the results of multivariate outlier detection, we see that these were reported as outliers. However, down regulated variables were not reported as ouliers.

5.3 Moderated t-statistic

The main idea of this method is to fit a linear model for each gene, g. Given a expression vector $\mathbf{y_g}$ a linear model can be applied as follows,

$$\begin{aligned} E(\mathbf{y}_g) &= \mathbf{X}\boldsymbol{\alpha}_g \\ var(\mathbf{y}_g) &= \mathbf{W}_g\boldsymbol{\sigma}_g^2 \end{aligned} \tag{5.2}$$

where $\mathbf{X}$ is a design matrix, $\boldsymbol{\alpha}_g$ a coefficient vector and $\mathbf{W}_g$ a known weight matrix.
Once the linear model proposed in equation (5.2) is fitted, estimators of $\boldsymbol{\alpha}_g$, $\boldsymbol{\sigma}_g^2$ such as $\widehat{\boldsymbol{\alpha}}_g$, $var(\widehat{\boldsymbol{\alpha}}_g)$ and $\mathbf{s}_g^2$ are generated. Using a contrast matrix $\mathbf{C}$, any contrast of biological, research, interest can be calculated with the expression,

$$\boldsymbol{\beta}_g = \mathbf{C}^T\boldsymbol{\alpha}_g \tag{5.3}$$

from estimators above.
We should make two assumptions, $\widehat{\boldsymbol{\beta}}_g$, the contrast estimators are normally distributed and $\mathbf{s}_g^2$, the residual variances follow a scaled χ^2 distribution. A hierarchical Bayes' model is set up to use this information. An inverse χ^2 prior for the $\boldsymbol{\sigma}_g^2$ is assumed by the empirical Bayes method (Smyth, 2004), with mean s_0^2 and degrees of freedom f_0. The posterior values for the residual variances are,

$$\tilde{s}_g^2 = \frac{f_0 s_0^2 + f_g s_g^2}{f_0 + f_g} \tag{5.4}$$

where $\dfrac{f_0}{f_0 + f_g}$ is the weight coefficient associated with all probes and $\dfrac{f_g}{f_0 + f_g}$ is associated with gene g.
Now a t-test is performed with denominator $\tilde{s}_g^2$ and numerator the estimator of the contrast of interest, $\widehat{\boldsymbol{\beta}}_g$. This is a moderated t-statistic.
The R package `limma` performs a moderated t-statistic as described above. The following R code will perform the analysis using our grapevine data.

```
> XX = t(x)
> library(limma)
> desing = model.matrix(~0 + factor(xinfo$Behandlung))
> colnames(desing) = c("K", "Dry")
> fit = lmFit(XX, desing)
> cont.matrix = makeContrasts(NovsYes = K - Dry, levels = desing)
> fit2 = contrasts.fit(fit, cont.matrix)
> fit2 = eBayes(fit2)
> options(digits = 3)
```

`model.matrix` function creates the design matrix and `lmFit` estimates the parameters of the associated linear model.
The function `makeContrast` calculates the contrasts of interest following by `contrast.fit`

Row	ID	logFC	AveExpr	t	P.Value	adj.P.Val	B
11	X468	-0.0032	0.0086	-2.5180	0.0138	0.6772	-4.8926
10	X467	-0.0029	0.0074	-2.4628	0.0159	0.6772	-5.0216
12	X469	-0.0031	0.0099	-2.4559	0.0162	0.6772	-5.0376
13	X470	-0.0030	0.0112	-2.3928	0.0190	0.6772	-5.1820
9	X466	-0.0026	0.0062	-2.3438	0.0215	0.6772	-5.2916
14	X471	-0.0030	0.0123	-2.3298	0.0223	0.6772	-5.3226
15	X472	-0.0029	0.0138	-2.2396	0.0278	0.6772	-5.5183
8	X465	-0.0023	0.0049	-2.1680	0.0331	0.6772	-5.6690
16	X473	-0.0027	0.0154	-2.1452	0.0349	0.6772	-5.7159
17	X474	-0.0026	0.0168	-2.0548	0.0431	0.6772	-5.8980

Table 5.1: First 10 variables comparing K vs Dry in the grapevine data.

which computes $\widehat{\boldsymbol{\beta}}_g$ and standard errors. Finally `eBayes` solves the hierarchical Bayes's model.

Table 5.1 presents the ouput of `topTable` with information about the moderated t-test, t, the associated p-value, *P.Value*, and the adjusted p-value (Benjamini and Hochberg, 1995), *adj.P.Val*.
B is the B-statistic which is the log-odds that the gene is differentially expressed e.g., for *X468* $exp(-4.8926)/1 + exp(-4.8926) = 0.00744$.
An approach to summarize the statistical significance via p-value and the biological or technical importance, fold change, is to draw both in a graph. This graph is called *volcano* plot and Figure 5.3 shows the results for the grapevine data where the name of the first ten variables were added. As in section 5.1 the most promising variables were classified as outliers in chapter 2.

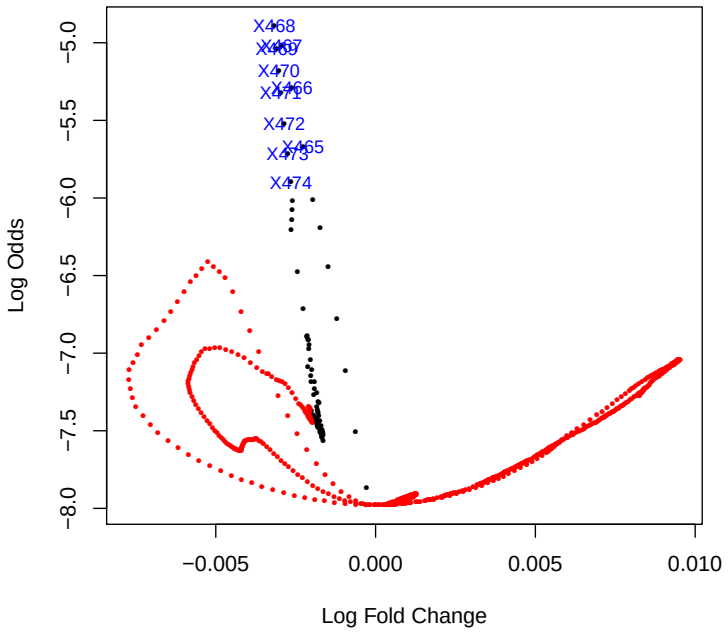

Figure 5.3: Volcano plot of grapevine data. Outliers in black and non outlier in red

Chapter 6

Case Study

This section tries to develop new techniques based on the data of Webster and Gibbs (2009) . We used here rank-invariant normalized intensities for expression data of 364 patients (188 non affected, 176 affected) with a confirmed pathologic diagnosis of late-onset Alzheimer disease (LOAD). Transcripts that were detected in less than 90% of cases or 90% of controls are not included. Any intensity where the Illumina detection score was < 0.99 was coded as `NaN`

This data was read in R and stored in a matrix called `x`. Missing values were recoded with column mean imputation, after this the values were log_{10} transformed, and renamed as `X` and this will be used for the further analysis here.

Information about important covariates are in another file called `samples.covar`. In order to see the structure of the data, the following R code can help.

```
> covlist = read.table("samples.covar", head = TRUE)
> str(covlist)

'data.frame':        364 obs. of  9 variables:
 $ Group         : Factor w/ 2 levels "WGAAD","WGACON": 1 1 1 1 1 1 1 1 1 1 ...
 $ Ind           : int  15 18 20 24 25 28 29 31 10 35 ...
 $ Diagnosis     : int  2 2 2 2 2 2 2 2 2 2 ...
 $ age           : int  84 80 81 91 80 84 75 82 85 85 ...
 $ apoe          : int  44 34 34 34 44 34 33 34 34 34 ...
 $ region        : int  3 3 3 3 3 3 3 3 3 1 ...
 $ pmi           : num  4.33 3.25 3 2 1.33 2.33 2 2.25 2.66 10 ...
 $ site          : int  1 1 1 1 1 1 1 1 1 2 ...
 $ hybridization: int  4 4 4 4 4 4 4 4 7 4 ...

> table(covlist$Group)

 WGAAD WGACON
   176    188
```

```
> table(covlist$Diagnosis)

  1   2
188 176

> table(covlist$region)

  1   2   3   4
 71  20 242  31

> table(covlist$apoe)

 22  23  24  33  34  44
 13  18  11 172 113  37

> table(covlist$Diagnosis, covlist$apoe)

     22  23  24  33  34  44
  1  13  15   2 120  34   4
  2   0   3   9  52  79  33
```

Group and Diagnosis code the same information where $Diagnosis = 1$ means unaffected and $Diagnosis = 2$ affected, Region has four levels as $frontal = 1$, $parietal = 2$, $temporal = 3$ and $cerebellar = 4$. Apoe is the allele dose of the gene apolipoprotein E $APOE\varepsilon 4$, pmi is the postmortem interval and hybridization the day of expression hybridization.

6.1 Multivariate Outlier Detection

Using the proposed method for multivariate outlier detection in the Alzheimer data the results are shown in Figure 6.1 with 3267 transcripts as *outliers*.

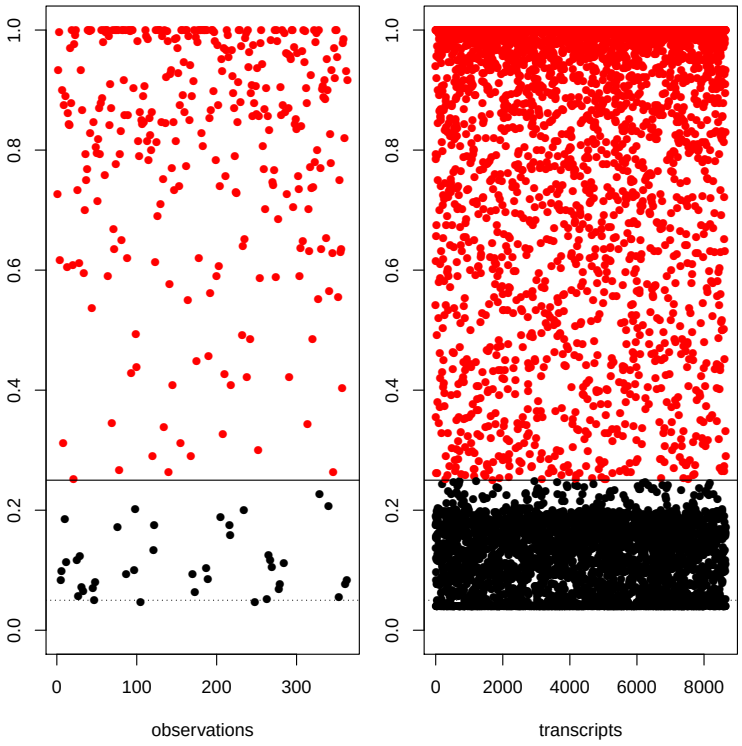

Figure 6.1: Multivariate Outlier Detection in Alzheimers Disease data. Observations, left, Transcripts, right.

Now we can split the matrix $\mathbf{X}$ of transcripts in two pieces, one which contains only outliers, $\mathbf{X}_{out}$ with dimensions 364×3267 and another matrix with non outliers $\mathbf{X}_{no}$ with dimensions 364×5383.

The relation between weights and p-values from a Welch t-test are shown in Figure 6.2 where the areas are proportional to the frequencies. In each p-value category the bigger areas of the rectangles correspond to non outliers and the areas of non outliers are increasing as soon the p-value lost significance.

When we apply the multivariate outlier detection method in order to look for outlier patients or samples, we find 40 patients classified as outliers. This is 0.89 percent of non outliers in the data set. Looking at the left panel of Figure 6.1 we can see that there is no weight below 0.05, bound of the extreme outliers. On the other hand we want to keep as much observations as possible, then we keep these observations but we try to give a reasonable explanation about these outliers using the demographic variables of `covlist`.
Making a Welch t-test to find a difference in age between outlier and non ouliers group, gives a p-value of 0.044 and age means of 79.9 and 82.6 respectively. Outliers patients died 3 years

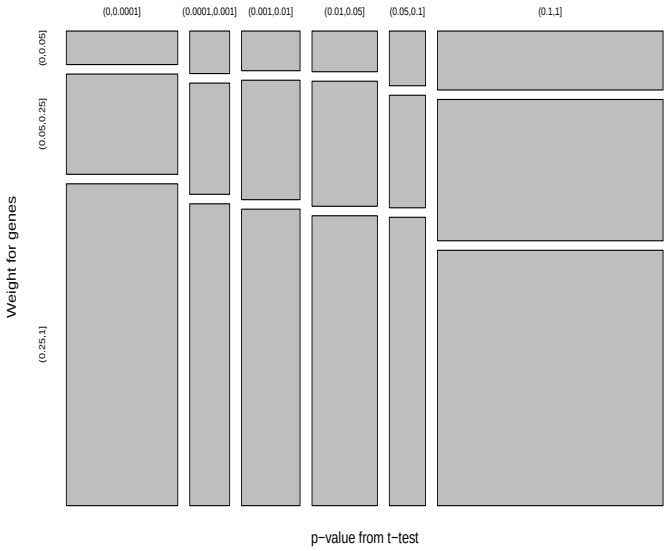

Figure 6.2: Mosaic plot. Horizontal, p-values from a Welch t-test. Vertical, weights of the multivariate outlier detection method.

earlier as non outliers. The same analysis with the variable postmortem interval gaves a not significant result.

	Outlier	Non outlier
Unaffected	19	169
Affected	21	155

Table 6.1: Number of patients in diagnosis levels versus outlier classification

	Outlier	Non outlier
Frontal	6	65
Parietal	1	19
Temporal	30	212
Cerebellar	3	28

Table 6.2: Number of patients in region levels versus outlier classification

Table 6.1 shows how outliers are distributed between Unaffected and Affected patients, then we can not find any difference. Regarding the brain region where the sample was collected, at the temporal region there is the higher outlier frequency as Table 6.2 presents.

6.2 LDA in Practice

We show in Figure 6.3, the misclassified observations,which are the 1 in red or 2 in black.

Figure 6.3: LDA in Alzheimers Disease data. Numbers denote diagnosis, 1,unaffected and 2, affected.

Sampling only 121 observations from 364 and using these as training data set, leads on Table 6.3. The predicted classification can be seen in Figure 6.3. The proportion of correct assignments is 0.855.

	1	2
1	98	42
2	10	110

Table 6.3: Observed vs Predicted using LDA with 121 observations as train data set

A interesting question to answer is, given the transcripts information of one patient, could we assess him affected or non affected using our previous information (database)? The answer will be equivalent to sample one patient of our 364, calculate the LDA with the remaining 363 and finally predict for this patient. We have done this 20 times and the proportion of correct assignments was 0.95, then it is possible to assess a diagnostic only with the transcripts information.

An LDA performed separately for $\mathbf{X}_{out}$ and $\mathbf{X}_{no}$ gives a proportion of correct assignments of 0.7654 and 0.7983 respectively.

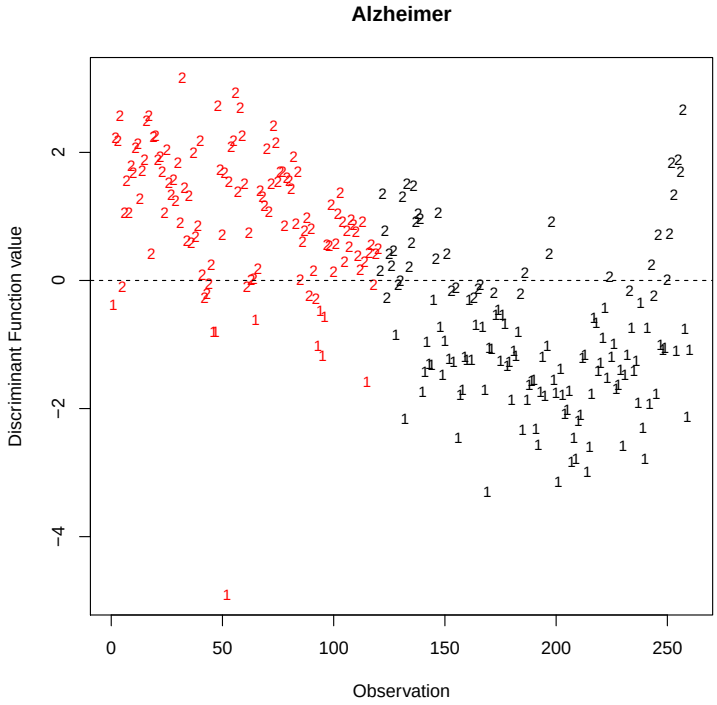

Figure 6.4: LDA in Alzheimers Disease data with 121 observations as train data set. Numbers denote diagnosis, 1,unaffected and 2, affected.

6.3 PCR and PLSR

In the second chapter these methods have been explained and outlined their implementation in R. Table 6.4 present the results in terms of proportion of correct assignments of 15 replicates based on 121 observations as train set and the remaining 244 were used to predict unaffected or affected with different numbers of components considered. For the same replicates, using LDA gives a 0.8 proportion of correct assignments.

	1	2	3	4	5	6
PCR	0.51	0.60	0.62	0.63	0.69	0.69
PLSR	0.68	0.74	0.77	0.79	0.80	0.81
SPLSDA	0.75	0.79	0.79	0.78	0.78	0.77

Table 6.4: Proportion of correct assignments after 15 replicates with 121 observations as train data set

An important issue regarding these two methods is how to choose the number of components to be used. This problem could be solve if we follow a minimum RMSEP criteria. Figure 6.5 shows no more improvement in RMSEP after 15 components and also better RMSEP with less components for PLSR. In fact there is no more RMSEP reduction when the number of components is between 5 and 20 using PLSR.

The importance of the reduction of RMSEP is clear but also a good predictive ability will be needed to use the method in a practical way. From a clinical point of view the method should assess correctly if a patient has Alzheimer disease or not and Figure 6.6 points out, the better prediction ability of PLSR. Again here it can be seen that the use of more than 5 components will not lead to a prediction ability improvement in case of PLSR. On the other hand, PCR needs 10 components more to achive the same proportion of correct assignments than PLSR.

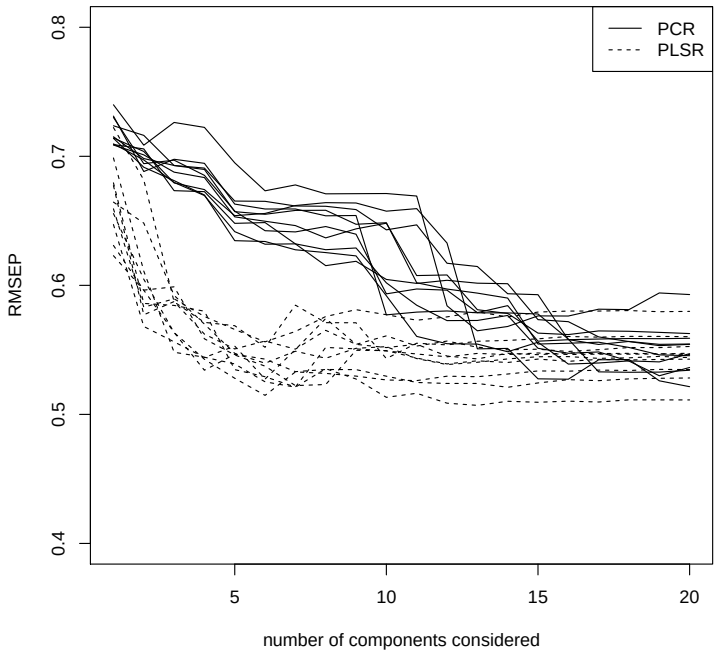

Figure 6.5: RMSEP in Alzheimer Disease data using 10 replicates with 121 observations as train data set.

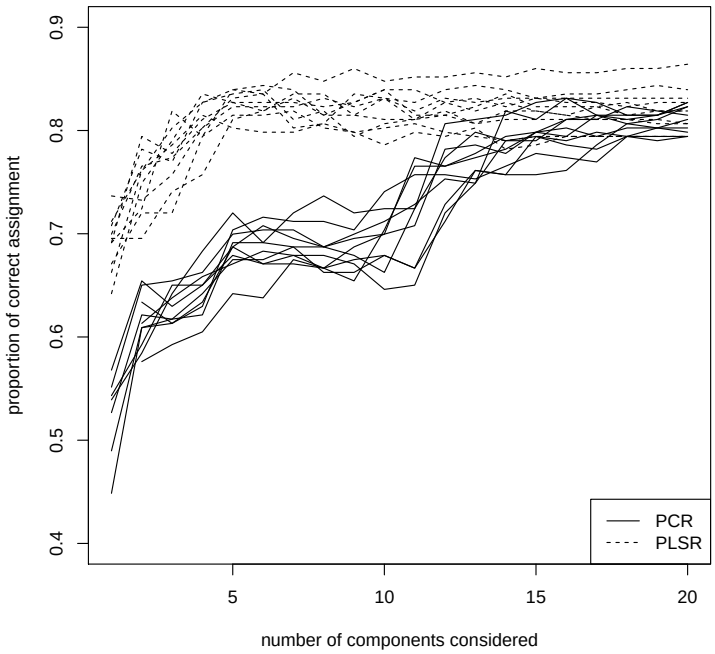

Figure 6.6: Proportion of correct assignments in Alzheimer Disease data using 10 replicates with 121 observations as train data set.

6.4 PLogReg

The interesting performance of this method that we have presented and outlined before in comparison to LDA, PCR and PLSR is the possibility to detect which variables, genes or measures have more importance in order to explain the dependent variable. A penalized logistic regression applied to all the observations in the data set reveals 68 significant transcripts from the original 8650. This is an approach to answer the question: *"Are there differential genes expressed?"*

As in the previous sections a sampling procedure has been implemented in order to check the prediction ability and to compare the penalized logistic regression with other methods. From the 364 observations, 121 are sampled randomly and used as training set to fit the penalized model, and the remaining 243 will be used to predict the memberships into the classes unaffected or affected. This procedure was repeated 20 times. The value of the optimal λ or λ_{min} is essential in a penalized logistic regression. Table 6.5 presents descriptive statistics of λ_{min} in the 20 replicates.

	Minimum	1st Qu.	Median	Mean	3rd Qu.	Maximum
λ_{min}	0.017	0.033	0.041	0.047	0.068	0.085

Table 6.5: λ_{min} after 20 replicates with 121 observation as training data set

The percentage of correct assignments was 0.797 and the median number of selected variables 31.

In order to find a reliable value of λ_{min} we have sampled 500 times with 121 observations from the training data set. The results of these 500 samples are presented in Table 6.6 and a density plot is available in Figure 6.7.

	Minimum	1st Qu.	Median	Mean	3rd Qu.	Maximum
λ_{min}	0.0046	0.0307	0.0429	0.0446	0.0574	0.1024

Table 6.6: λ_{min} after 500 replicates with 121 observation as training data set

We have performed a separate penalized logistic regression using $\mathbf{X}_{out}$ and $\mathbf{X}_{no}$. The percentage of correct assignment when only variable classified as outliers were used was 0.769 with a median number of variables considered of 38.5. In the case of non outlier variables the proportion of correct assignments and median number of variables considered was 0.799 and 32, respectively.

Table 6.7 presents the values of λ_{min} when outlier or non outlier variables performed a penalized logistic regression. As the second row of the table shows the values of λ_{min} are

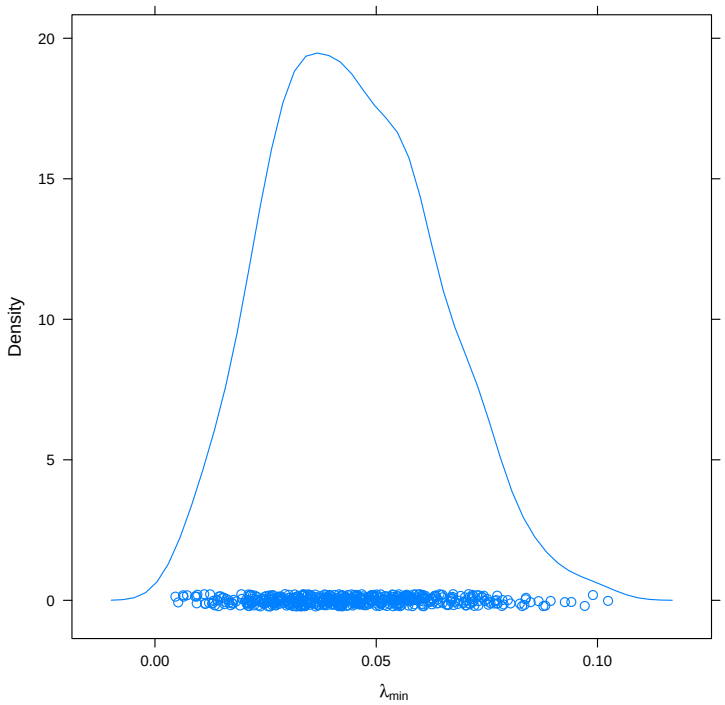

Figure 6.7: Density plot of λ_{min} sampling 500 times with 121 observations as training data set.

	Minimun	1st Qu.	Median	Mean	3rd Qu.	Maximun
Outlier λ_{min}	0.0135	0.0221	0.0344	0.0351	0.0470	0.0587
Non outlier λ_{min}	0.0146	0.0282	0.0380	0.0409	0.0523	0.0755

Table 6.7: λ_{min} after 20 replicates with 121 observation as training data set, using outlier or non outlier variables.

	0	0.125	0.25	0.375	0.5	0.625	0.75	1
mean of true assessment	0.817	0.819	0.812	0.808	0.805	0.802	0.803	0.798
median number of variables	8650	500	235	148	103	76	60	36

Table 6.8: Elastic net results as α change between 0 and 1, after 30 replicates with 121 observations as training data set.

closer to Table 6.6 indicating a better fit than in the case of outlier variables. Also the predictive ability was higher in non outlier variables.

The elastic net approach was used to find α and the results are presented in Table 6.8.

Remember that an α equal to one is a lasso logistic regression. α with the best predictive ability is within the interval $0.125 - 0.25$. The use of this information allows us finally to

find which transcripts have an effect in the Alzheimer disease. 546 transcripts were found by elastic net regression, $\alpha = 0.2$. This is what Figure 6.8 shows.

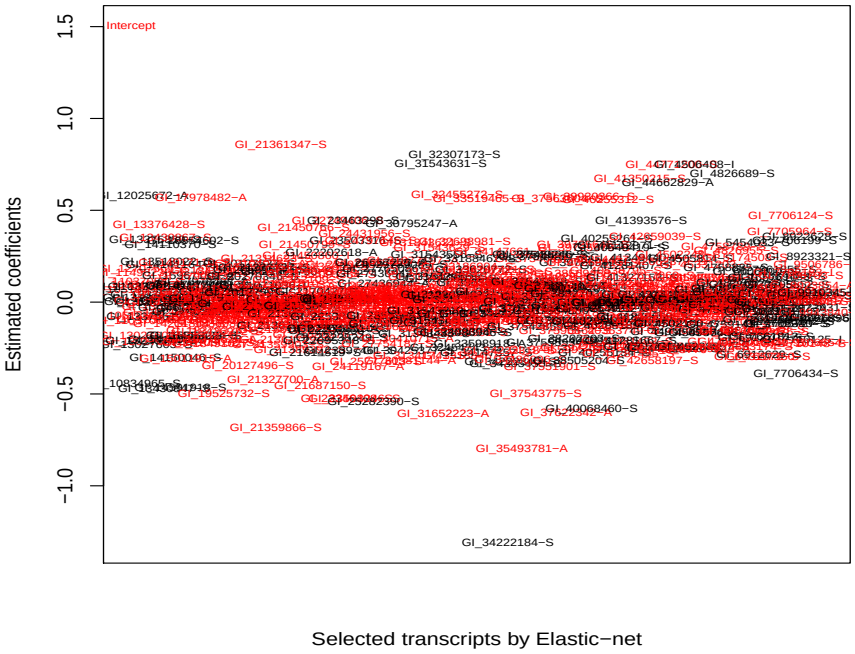

Figure 6.8: Selected transcripts, $\alpha = 0.2$, of 8650. Outliers in black and non outliers in red.

Finally plotting the estimated coefficients of the selected transcripts against their obtained weights in the multivariate outlier detection method, produces Figure 6.9. Two areas are easily distinguishable with the value of 0.25 at the vertical axis. An area where outlier variables are concentrated around $[-0.5, 0.5]$ estimated coefficients value and another one where non outliers in a wide range. The use of both methods, elastic-net and multivariate outlier detection has good performance in order to detect variables of interest.

Penalized logistic regression has pointed out similar predictive abilities as LDA, PCR, and PLSR using the same sampling procedure with the advantage of a possibility to know which are the main variables, in this case, transcripts that play a key role in the event of interest.

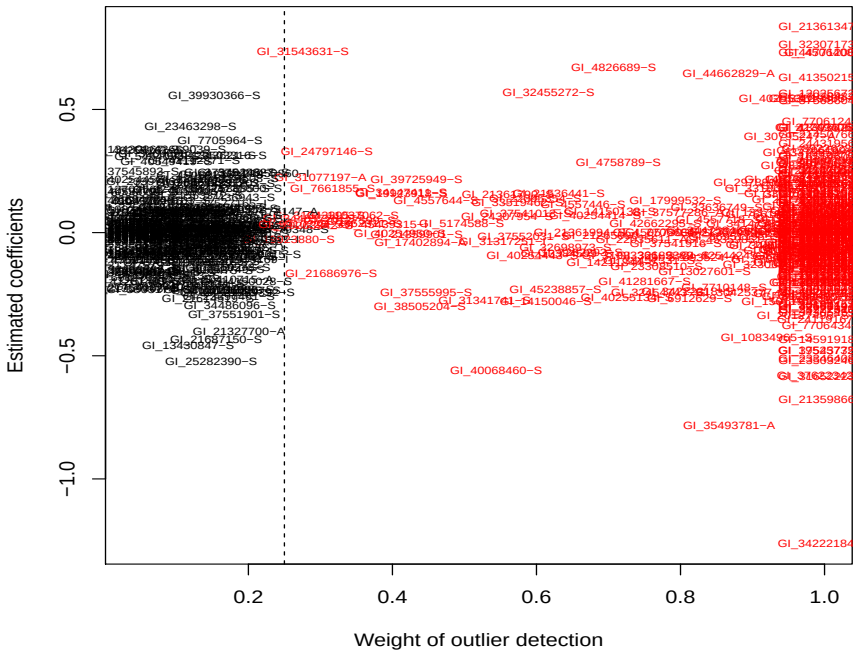

Figure 6.9: Selected transcripts against weights from multivariate outlier detection. Outliers in black and non outliers in red.

6.5 Cluster Analysis

The results obtained clustering the Alzheimer data, 364 *patients* $\times$ 8650 *transcripts*, are presented here. As in the previous chapter where cluster analysis was outlined, we use here the R package `mixOmics` to analyze the data. The function `cim` was used with the default parameters, Euclidean distance and the hierarchical clustering algorithm based on complete linkage, to produce a first heat map as Figure 6.10 shows.

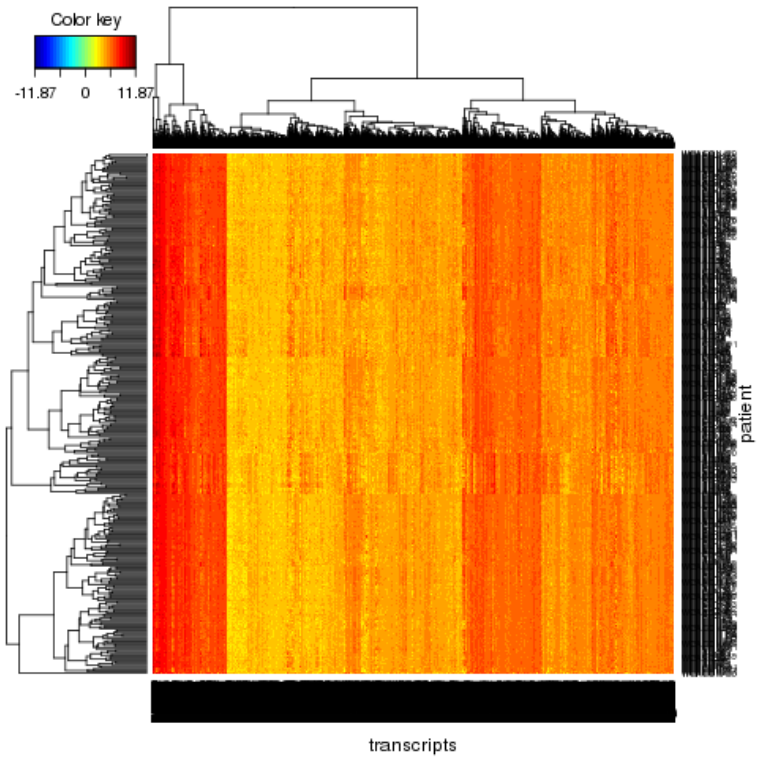

Figure 6.10: Heat map of Alzheimer data

The pattern in Figure 6.10 can be compared with Figure 6.11 where only outlier variables were used and Figure 6.12 where only non outlier variables were used.

When we compare the heat maps of Figure 6.11 and Figure 6.12 we observe clear differences. First the definition of the map is worse in the case of outlier variables than in non outliers. This is just due to the amount of data, 3267 outliers versus 5383 non outliers.
The second difference is more important because the patterns are totally different between Figure 6.12 and Figure 6.11. When the pattern in Figure 6.10 is taking as reference, Figure 6.12 is more similar than Figure 6.11.

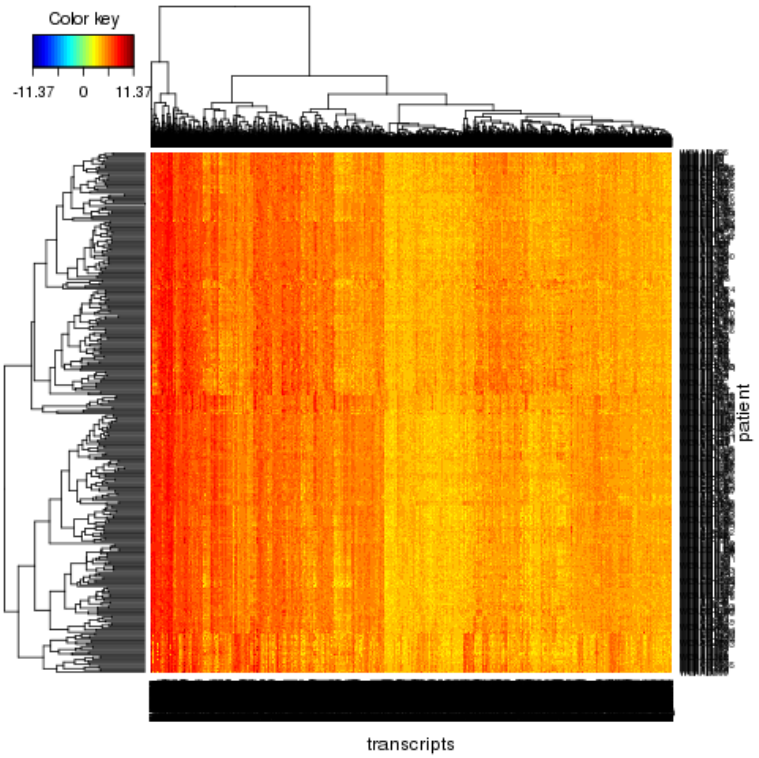

Figure 6.11: Heat map of Alzheimer data using outlier variables

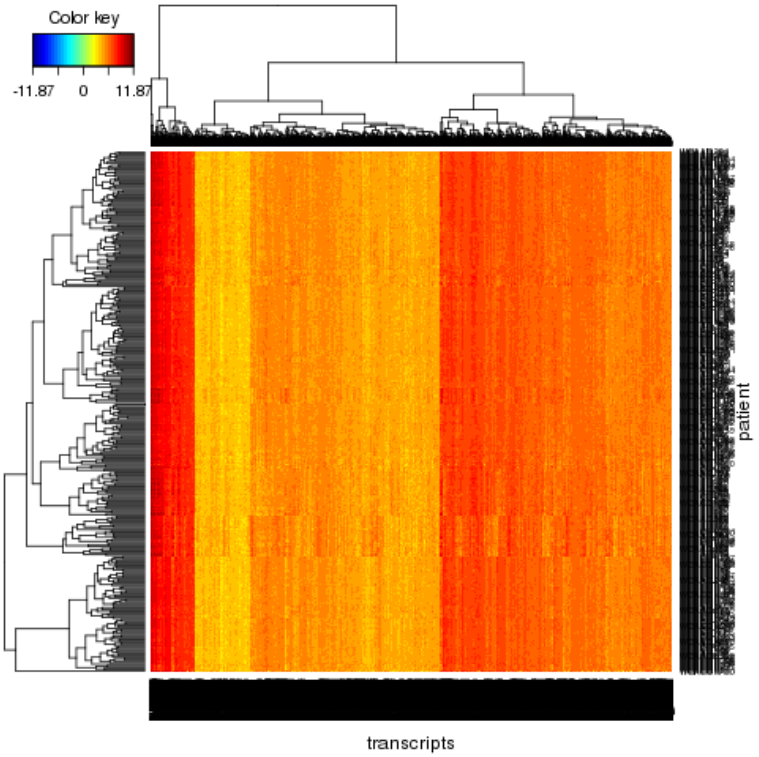

Figure 6.12: Heat map of Alzheimer data using non outlier variables

6.6 Permutation Test

Here we apply the method outlined in section 5.1 to the Alzheimer data using the `multtest` package in `bioconductor` to perform the Westfall and Young step down multiple testing procedure with the function `mt.maxT`.
The top ten significant transcripts are presented in Table 6.9, and Figure 6.13 shows p-values vs outlier weights. Using these two sources of information, 852 genes were found significant and non outliers, see right botton quadrant of Figure 6.13.

transcript	index	teststat	rawp	adjp
GI_23503246-S	2633	12.8181	0.0010	0.0010
GI_24307954-S	2713	-10.6652	0.0010	0.0010
GI_18426972-S	1218	10.5544	0.0010	0.0010
GI_21361595-S	1965	9.8388	0.0010	0.0010
GI_21536438-A	2155	9.8325	0.0010	0.0010
GI_38045920-S	5444	-9.5698	0.0010	0.0010
GI_33188462-S	4256	-9.4639	0.0010	0.0010
GI_11321580-S	127	-9.3891	0.0010	0.0010
GI_27413907-S	3079	-9.3278	0.0010	0.0010
GI_32306535-S	4107	-9.2939	0.0010	0.0010

Table 6.9: Top ten transcripts using Westfall and Young correction in Alzheimer data

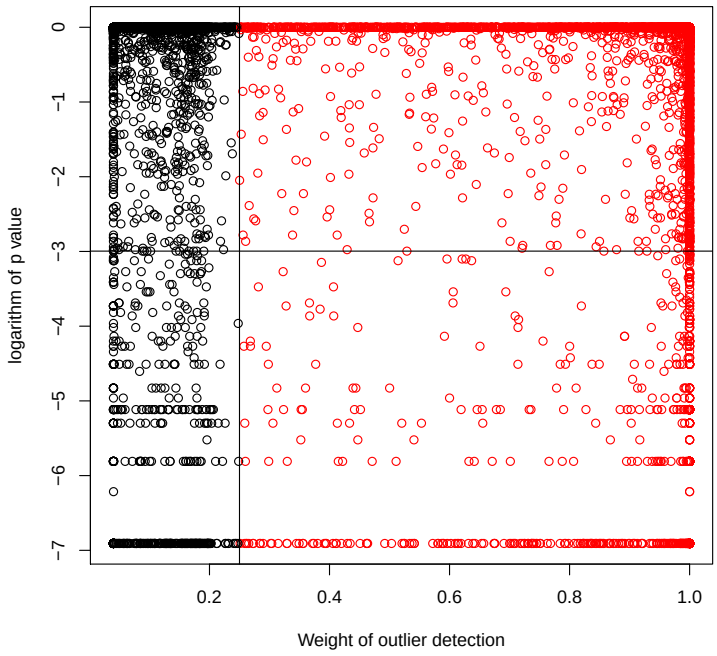

Figure 6.13: Multivariate outlier weights vs corrected p-values, `mt.maxT`, Alzheimer data. Outliers in black and non outlier in red

6.7 SAM

Using a $\Delta = 1.561$ and fold change equal to one results in 1795 significant transcripts with a $90th$ percentile FDR less than 0.5%. From the significant transcripts 989 were up regulated and 806 down regulated.

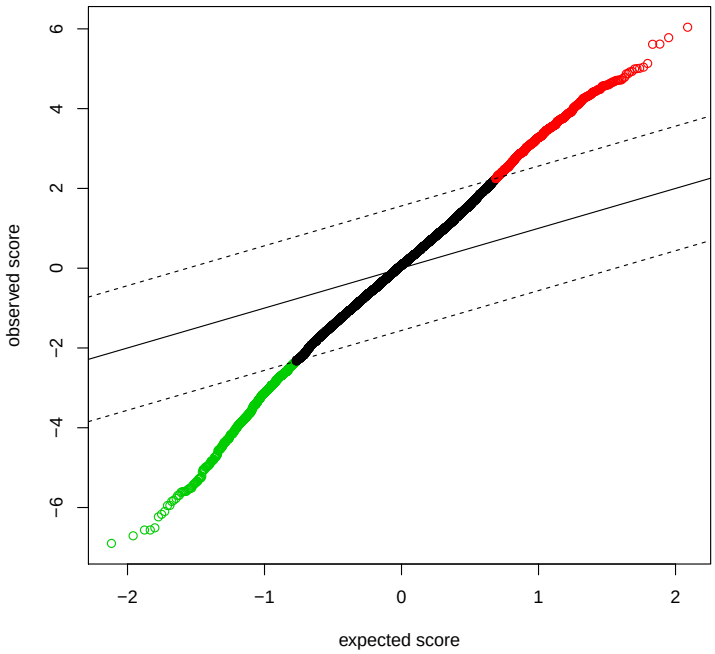

Figure 6.14: SAM plot of significance with Δ equal to 1.561. Variables in color have a difference greater than Δ. Green down regulation and red up regulation.

Figure 6.14 presents variables up or down regulated, in color, and no significant transcripts in black. The next step is to know which variables were colored and this is performed by the function `samr.compute.siggenes.tables`. When we compare the significant transcripts with the results of multivariate outlier detection, we find that 655 were reported as ouliers.

6.8 Moderated t-test

Performing a moderated t-test using the R package `limma` comparing "Affected" against "Non affected" produces results as in Table 6.10

Row	ID	logFC	AveExpr	t	P.Value	adj.P.Val	B
2633	GI_23503246-S	0.290	7.030	12.830	0.000	0.000	60.265
2713	GI_24307954-S	-0.366	6.083	-10.622	0.000	0.000	41.787
1218	GI_18426972-S	0.328	6.213	10.500	0.000	0.000	40.812
2155	GI_21536438-A	0.245	7.138	9.865	0.000	0.000	35.858
1965	GI_21361595-S	0.197	7.407	9.766	0.000	0.000	35.098
5444	GI_38045920-S	-0.553	5.128	-9.757	0.000	0.000	35.031
127	GI_11321580-S	-0.235	6.037	-9.406	0.000	0.000	32.384
979	GI_16753217-S	-0.564	5.395	-9.380	0.000	0.000	32.190
4107	GI_32306535-S	-0.326	5.638	-9.377	0.000	0.000	32.166
4256	GI_33188462-S	-0.212	4.940	-9.373	0.000	0.000	32.136

Table 6.10: First 10 transcripts comparing "Affected" vs "Non affected" in the Alzheimer data.

The information about B and logFC from Table 6.10 allows us to draw a volcano plot were the weights of multivariate outlier detection make a better view of the results. In addition to this the names of the most significant transcripts are written in blue.

When the Figure 6.15 is drawn with an extra axis to allocate the values of the outlier detection weights we obtain a three dimensional plot as Figure 6.16 shows. The previous volcano plot is transformed to a pointed barrel vault with longitudinal axis the oulier detection weights and four different sections.

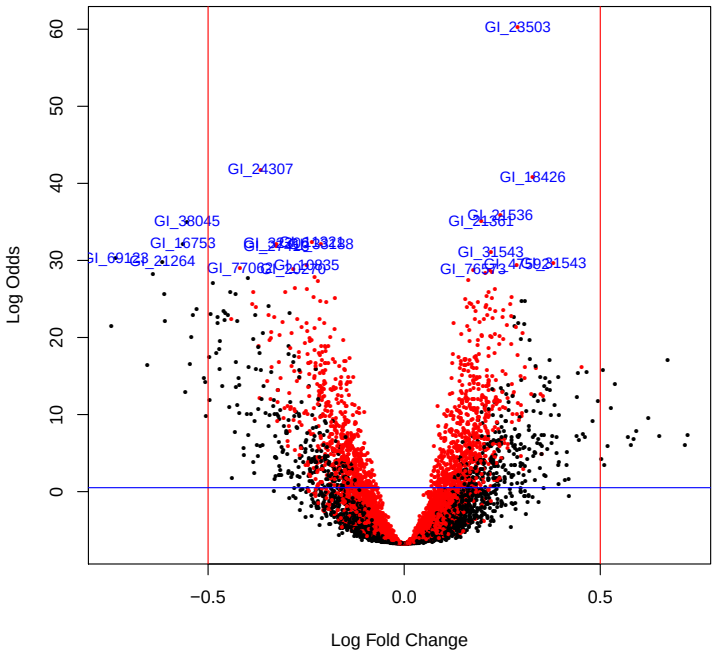

Figure 6.15: Volcano plot of Alzheimer data. Outliers in black and non outliers in red. Most significant transcripts in blue.

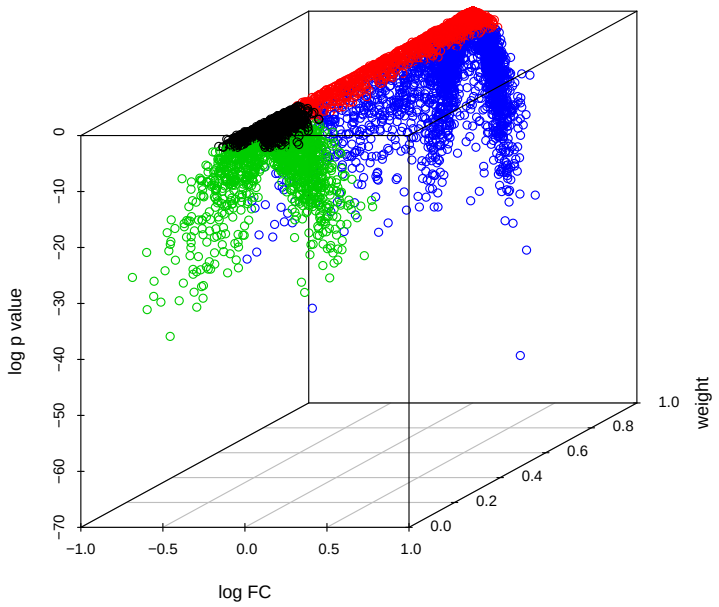

Figure 6.16: 3D Volcano plot of Alzheimer data. Outliers no significant in black, non outliers no significant in red, outliers significant in green, non outliers and significant blue

6.9 Summary of Results

Two main questions have been answered in the previous section using different methods.
Given the data, is it possible to classify future patients based on our transcripts data?
Which genes are differentially expressed?
The next subsections summarizes the information in order to have a better understanding of the results.

6.9.1 Classification

The methods used for classification were LDA, PCR, PLSR and SPLSDA.
Sampling 121 observations as training data set and using the remaining 244 to predict as we did in Table 6.4 of section 6.3 the results are presented in the next table.

LDA	PCR	PLSR	SPLSDA
0.80	0.69	0.81	0.77

Table 6.11: Proportion of correct assignments after 15 replicates with 121 observations as training data set. 6 components in PCR, PLSR, SPLSDA.

Could we assess a patient as affected or non affected using our previous information?
We sample 20 times 363 observations as training data set and the remaining patients will be predicted using LDA, PCR, PLSR and SPLSDA.

LDA	PCR	PLSR	SPLSDA
0.9	0.6	0.9	0.8

Table 6.12: Proportion of correct assignments after 20 replicates with 363 observations as training data set and one observation to predict. 6 components in PCR, PLSR, SPLSDA.

6.9.2 Differentially Expressed Genes

We are trying to answer the same question under different points of view; *Which transcripts explain the Alzheimer disease?*

Regarding the three methods widely used in microarray data, Permutation test, SAM and moderated t-statistic, their results were pointed out in Sections 6.6, 6.7 and 6.8. We can use a Venn diagram to see the overlap between these methods, as Figure 6.17 shows. There are 294 genes that have been found by all three methods, which are likely to be truly differential expressed genes. 3818 were found not significant.

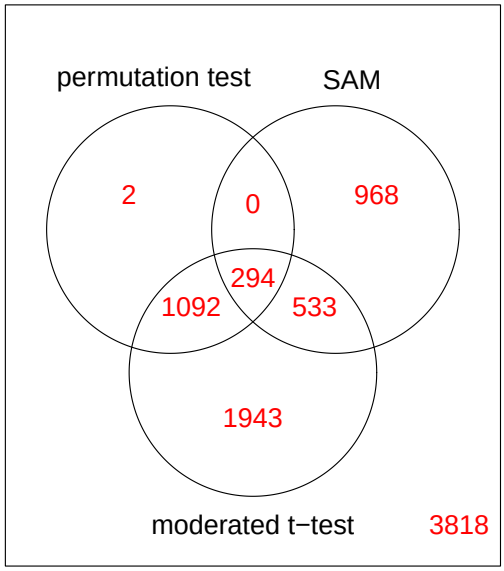

Figure 6.17: Venn diagram of Alzheimer data

Methods	Outlier	Non outlier
0	1613	2205
1	1075	1838
2	494	1131
3	85	209

Table 6.13: Number of transcripts found significant in at least 1, 2, or 3 methods such as, SAM, moderated t-test and permutation test.

A step forward is to include the information of the outlier detection section which Table 6.13

shows. While 209 genes meet the four requirements, SAM, moderated t-test and permutation test were positive and non outlier. This implies that only 2.41% of the original genes were useful.

Methods	Outlier	Non outlier
0	1528	2124
1	1077	1831
2	528	1122
3	120	273
4	14	33

Table 6.14: Number of transcripts found significant in at least 1, 2, 3 or 4 different methods, SAM, moderated t-test, permutation test and elastic net

When we need to be more strict in order to find significant genes, we can include the information coming from the elastic net approach. Table 6.14 presents the results where only 33 genes from 8650 meet the requirements.

Chapter 7

Conclusions

- A simple task before to start to analyze high dimensional data is to know what we are looking for. *Are we interested to classify future subjects, observations, patients, animals, using our current high dimensional data?* Or *are we focused to find the most significant variables which explain the outcome, trait or event of interest?*

- High dimensional data analysis can not rely on a single approach, multiple methods should be used. This means a challenge to the researcher who should be ready to improve his/her statistical skills and to see the problem from different points of view. A multidisciplinary approach is not an option, it should be a must.

- False discovery rate, FDR, and over fitting should be kept in mind when we are dealing with high dimensional data, which is related with the idea that only small amounts of variables or genes play a determinant role in the outcome or trait of interest. Even if these determinant variables are present in our data set, their values will be not so much different from the other variables and usually hidden under a noise.

- The amount of data and the multidisciplinary approach lead to the next issue to take into account, the computational feasibility. Once the methodological solution is found, the computational feasibility should be checked. Methods which quickly solve our algorithm in non high dimensional problems can drive the computer in an infinite loop or it can take weeks in the best case.

- High dimensional data solutions need to be validated. This is more important if we are focused to use the high dimensional data for future predictions. Over fitting can provide a good model but poor predictions. Dividing (randomly) the data set into training and test data, building a model only with the training data, and evaluating the model at the test data is a useful strategy. The use of different methods will point to the most relevant variables.

- For classification purposes our findings result in a better performance of PLSR, SPLSDA and LDA than PCR.

- The use of penalized regression has similar predictive abilities as LDA, PCR and PLSR with the advantage of a possibility to know which variables are involved.
- For finding the most significant variables, permutation tests, SAM and Moderated t-tests, in combination with outlier detection and/or penalized regression turned out to be useful.

Bibliography

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B Methodological*, 57(1):289–300, 1995.

E.H. Corder, A.M. Saunders, W.J. Strittmatter, D.E. Schmechel, P.C. Gaskell, G.W. Small, A.D. Roses, J.L. Haines, and M.A. Pericak-Vance. Gene dose of apolipoprotein type 4 allele and the risk of alzheimer disease in late onset families. *Science*, 261:921–923, 1993.

S. Draghici. *Statistics and Data Analysis for Microarrays Using R and Bioconductor, Second Edition*. Chapman & Hall/CRC Mathematical & Computational Biology. Taylor & Francis, Boca Raton, Florida, 2011.

D.S. Falconer and T.F.C. Mackay. *Introduction to Quantitative Genetics, 4th edition*. Longman Group Ltd, Essex, England, 1998.

P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Comput. Stat. Data Anal.*, 52:1694–1711, 2008.

R.A. Fisher. The correlation between relatives on the supposition of mendelian inheritance. *Transactions of the Royal Society of Edinburgh*, 52:399–433, 1918.

J. Fox. Aspects of the Social Organization and Trajectory of the R Project. *The R Journal*, 1(2):5–13, 2009.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction. 2nd edition*. Springer, New York, 2009.

K. Le Cao, D. Rossouw, C. Robert-Granié, and P. Besse. A sparse pls for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, 7 (1):Article 35, 2008.

K. Le Cao, S. Boitard, and P. Besse. Sparse pls discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, 12 (1):253, 2011.

G.K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3 (1), 2004.

P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. Freeman, San Francisco, 1973.

J. Snow. The cholera near golden square, and at deptford. *Medical Times and Gazette*, 9: 321–322, 1854.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

V.G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9):5116–5121, 2001.

J.D. Watson and F.H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

J.A. Webster and J.R. Gibbs. Genetic control of human brain transcript expression in alzheimer disease. *Am. J. Hum. Genet.*, 84:445–458, 2009.

P.H. Westfall and S.S. Young. *Resampling-based multiple testing : examples and methods for p-value adjustment*. Wiley, New York, 1993.

D.M. Witten and R. Tibshirani. Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):615–636, 2009.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.