Introduction to Statistical Methods for Microarray Data Analysis

T. Mary-Huard, F. Picard, S. Robin

Institut National Agronomique Paris-Grignon
UMR INA PG / INRA / ENGREF 518 de Biométrie
16, rue Claude Bernard, F-75005 Paris, France
(maryhuar)(picard)(robin)@inapg.inra.fr

June 30, 2004

Contents

Chapter 1

Introduction

1.1 From genomics to functional genomics

1.1.1 The basics of molecular genetic studies

The basics of molecular biology has been summarized in a concept called the Central Dogma of Molecular Biology. DNA molecules contain biological informations coded in an alphabet of four letters, A (Adenosine), T (Thymine), C (Cytosine), G (Guanine). The succession of these letters is referred as a sequence of DNA that constitutes the complete genetic information defining the structure and function of an organism.

Proteins can be viewed as effectors of the genetic information contained in DNA coding sequences. They are formed using the genetic code of the DNA to convert the information contained in the 4 letter alphabet into a new alphabet of 20 amino acids. Despite an apparent simplicity of this translation procedure, the conversion of the DNA-based information requires two steps in eucariotyc cells since the genetic material in the nucleus is physically separated from the site of protein synthesis in the cytoplasm of the cell. Transcription constitutes the intermediate step, where a DNA segment that constitutes a gene is read and transcribed into a single stranded molecule of RNA (the 4 letter alphabet remains with the replacement of Thymine molecules by Uracyle molecules). RNAs that contain information to be translated into proteins are called messenger RNAs, since they constitute the physical vector that carry the genetic information form the nucleus to the cytoplasm where it is translated into proteins via molecules called ribosomes (figure 1.1).

Biological information is contained in the DNA molecule that can be viewed as a template, then in the RNA sequence that is a vector, and in proteins which constitute effectors. These three levels of information constitute the fundamental material for the study of the genetic information contained in any organism:

1 - Finding coding sequences in the DNA,

2 - Measuring the abundance of RNAs,

3 - Studing the diversity of Proteins.

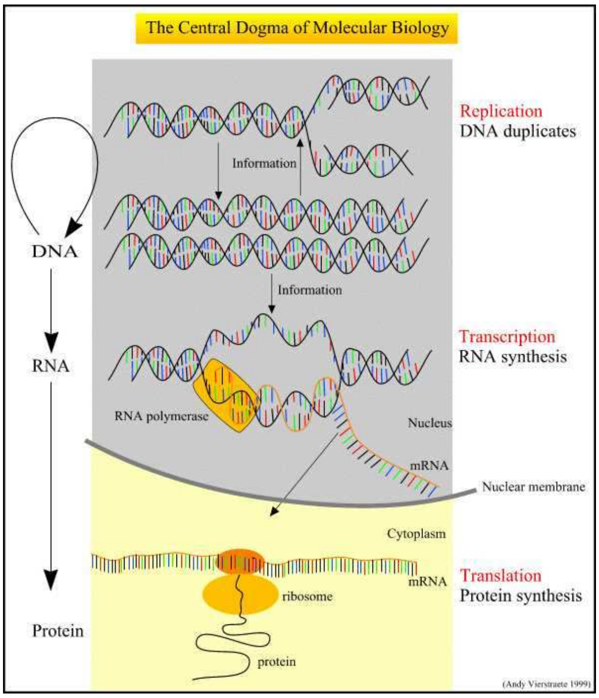


Figure 1.1: The central dogma of molecular biology

1.1.2 The success of sequencing projects

In the past decades, considerable effort has been made in the collection and in the dissemination of DNA sequences informations, through initiatives such as the Human Genome Project [1]. The explosion of sequence based informations is illustrated by the sequencing of the genome of more than 800 organisms, that represents more than 3.5 million genetic sequences deposited in international repositories (Butte (2002)). The aim of this first phase of the genomic area consisted in the elucidation of the exact sequence of the nucleotides in the DNA code, that has allowed the search for coding sequences diluted all along the genomes, via automatic annotation. Nevertheless there is no strict correspondance between the information contained in the DNA and the effective biological activity of proteins. In a more general point of view genotype and phenotype do not correspond strictly, due to the physical specificity of genomes which has a dynamic structure (Pollack and Iyer (2003)), and also due to environmental influences. This explains why there is now a considerable desequilibrium between the number of identified sequences, and the understanding of their biological functions, that remain unknown for most of the genes. The next logical step is then to discover the underlying biological informations contained in the succession of nucleotides that has been read through sequencing projects. Attention has now focused on functional genomics, that aims at determining the functions of the thousands of genes previously sequenced.

[1] http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

1.1.3 Aims of functional genomics

Assessing the function of genes can be tackled by different approaches. It can be predicted through homology to genes with functions that are better known, possibly from other organisms. This is the purpose of comparative genomics. An other way to determine the function of genes is through repeated measurements of their RNA transcripts. Investigators now want to know which genes are responsible for important healthy functions and which, when damaged, contribute to diseases. Accordingly, the new field of functional genomics focuses on the expression of DNA. To that extend, functional genomics has been divided into two major fields : transcriptomics and proteomics.

1.2 A new technology for transcriptome studies

The study of the transcriptome requires the measurement of the quantity of the messenger RNAs of thousands of genes simultaneously. As sequencing projects needed a new technology for "*en masse*" sequencing, the field of transcriptomics has explosed with the progress made in the development of technologies that merge inventions from the semiconductor and computer industry with laser engineering (Duggan *et al.* (1999)). Various techniques have been developped to exploit the growing number of sequence based data, like Serial Analysis of Gene Expression (SAGE) for instance (Boheler and Stern (2003)), and microarrays have become the standard tool for the understanding of gene functions, regulations and interactions.

1.2.1 The potential of transcriptome studies

More than the direct interest of transcriptome studies in fundamental biology, high throughput functional genomic technologies now provide new potentialities in areas as diverse as pharmacogenomics and target selectivity, pronostic and biomarkers determination, and disease subclass discovery. In the first case, gene expression profiles can be used to characterize the genomic effects of an exposure of an organism to different doses of drugs, and to classify therapeutic targets according to the gene expression patterns they provoke. Then gene expression profiling can be used to find genes that distinguish a disease from an other, and that correlate and predict the disease progression (Golub *et al.* (1999b)). In the latter situation, the classical classification of diseases based on morphological and histological characteristics could be refined using genetic profile classification (Alizadeh *et al.* (2000)). Since the cost of microarrays continues to drop, their potentialities could be widely used in personnalized medicine, in order to adapt treatments to the genetics of individual patients.

1.2.2 The basis of microarray experiments

The basics of microarray experiments take advantage of the physical and chemical properties of the DNA molecules. A DNA molecule is composed of two complementary strands.

Each strand can bind with its template molecule, but not with templates whose sequences are very different from its own. Since the sequences of thousands of different genes are known and stored in public data bases, they will be used as template, or probes, and fixed on a support. The DNA spots adhere on a slide, each spot being either a cloned DNA sequence with known function or genes with unknown function. In parallel, RNAs are extracted from biological samples, converted into complementary DNAs (cDNAs), amplified and labelled with fluorescent dyes (called Cy3 and Cy5) or with radioactivity. This mixture of transcripts, or targets, is hybridized on the chip, and cDNAs can bind their complementary template. Since probes are uniquelly localized on the slide, the quantification of the fluorescence signals on the chip will define a measurement of the abundance of thousands of transcripts in a cell in a given condition. See Duggan *et al.* (1999) and references therein for details concerning the construction of microarrays.

1.2.3 Different types of microarrays

Selecting the arrayed probes is then the first step in any microarray assay : it is crucial to start with a well characterized and annotated set of hybridization probes. The direct amplification of genomic gene specific probes can be accomplished for prokaryotes and simple eukaryotes, but remains impossible for most of eukaryotic genomes, since the large number of genes, the existence of introns, and the lack of a complete genome sequence makes direct amplification impracticable. For these species, EST data can be viewed as a representation of the transcribed portion of the genome, and the cDNA clones from which the ESTs are derived have become the primary reagents for expression analysis. For other array based assays, such as Affimetrix Genechips assays, little information is provided concerning the probe set, and the researcher is dependent on the annotation given by the manufacturer. Nevertheless, probes are designed to be theoretically similar with regard to hybridization temperature and binding affinity, that makes possible the absolute quantification of transcript quantities, and the direct comparison of results between laboratories (this is also the case for membrane experiments). On the contrary, for cDNA microarrays, each probe has its own hybridization characteristic, that hampers the absolute quantification of transcripts quantity. To that extend cDNA microarray assays will necessarily require two biological samples, referred as the test and the reference sample, that will be differentially labelled by fluorescent dyes, and competively hybridized on the chip to provide a relative measurement of the transcripts quantity. The comparison between different microarray technologies is given in table 1.1.

1.2.4 Data collection

After biological experiments and hybridizations are performed, the fluorescence intensities have to be measured with a scanner. This image acquisition and data collection step can be divided into four parts (Leung and Cavalieri (2003)). The first step is the image acquisition by scanners, independently for the two conditions present on the slide. The

	oligo-arrays	cDNA arrays	nylon membrane
support of the probes	glass slide	glass slide	nylon membrane
density of the probes ($/cm^2$)	~ 1000	~ 1000	~ 10
type of probes	oligonucleotides	cDNAs	cDNAs
labelling	fluorescence	fluorescence	radioactivity
number of condition on the slide	1	2	1

Table 1.1: Comparison of different types of arrays. The ratio of densities between membranes and slides is 1/100 but the ratio of the number of genes is rather 1/10 since nylon membranes are bigger in size.

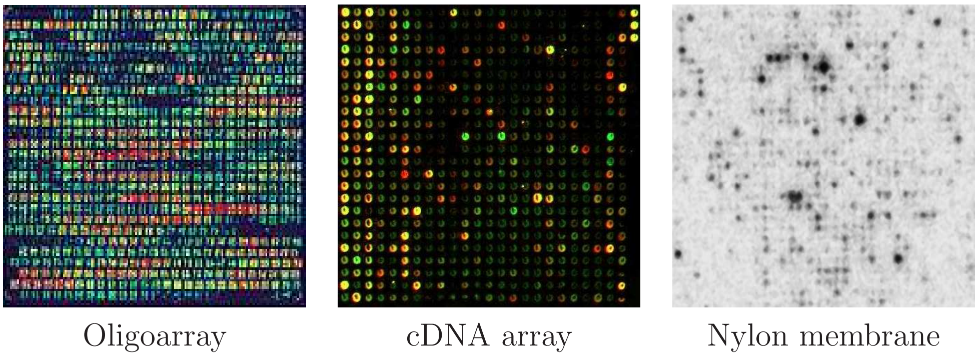


Figure 1.2: Comparison of acquired images for different arrays

quality of the slide is essential in this step, since once an array has been imaged, all data, high or poor quality are essentially fixed. The second step consists in the spot recognition or gridding. Automatic procedures are used to localize the spots on the image, but a manual adjustment is often needed to the recognition of low quality spots, that are flagged and often eliminated. Then the image is segmented to differentiate the foreground pixels in a spot grid from the background pixels. The quality of the image is crucial in this step, since poor quality images will result in various spot morphologies. After the spots have been segmented, the pixel intensities within the foreground and background masks are averaged separately to give the foreground and background intensities. After the image processing is done, the raw intensity data have been extracted from the slide, indenpendently for the test and the reference, and the data for each gene are typically reported as an intensity ratios that measure the relative abundance of the transcripts in the test condition compared to the reference condition.

1.3 Upstream intervention of statistical concepts

Once biological experiments are done and images are acquired, the researcher disposes of the mesurement of relative expression of thousands of genes simultaneously. The aim is then to extract biological significance from the data, in order to validate an hypothesis.

The need for statistics has become striking soon after the apparition of the technology, since the abundance of the data required rigorous procedures for analysis. It is important to notice that the intervention of statistical concepts occurs far before the analysis of the data *stricto sensu*. Looking for an appropriate method to analyze the data, whereas no experimental design has been planed, or no normalization procedure has been applied, is unrealistic. This explains why the first two chapters of this review will detail the construction of an appropriate experimental design, and the choice of normalization procedures.

1.3.1 The variability of microarray data and the need for normalization

Even if the microarray technology provides new potentialities for the analysis of the transcriptome, as every new technology, several problems arise in the execution of a microarray experiment, that can make two independent experiments on the same biological material differ completely, because of the high variability of microarray data. Let's go back to the experimental procedure detailed above : every single step is a potential source of technical variability. For instance the RNA extraction and the retro-transcription efficiency are not precisely controlled, that can lead to various amounts of biological material analyzed *in fine*. Despite the control of hybridization conditions (temperature, humidity), the efficiency of the binding on the slide is not known precisely. As for the image acquisition, many defaults on the slide can lead to bad quality images that hampers any reliable interpretation. This is considered "conditionnaly" to the fact that many experimentators can perform microarray experiments, on the same biological sample, in the same laboratory or in different place, but with the objective to put their work in common.

1.3.2 Experimental design

Despite the vast sources of variabilities, some errors can be controlled and some can not, leading to a typology of errors : systematic errors and random errors. The first type of errors can be viewed as a bias that can be controlled using strict experimental procedures. For instance, assays can be performed by the same researcher all along the experiment. The second type of errors constitutes a noise that leads to a lack of power for statistical analysis. Normalization procedures will be crucial for its identification and correction. The first need for a biologist is then to consider an appropriate experimental design. This will allow not only some control quality for the experimental procedure, but also the optimization of the downstream statistical analysis. Chapter 2 will explain why a precise knowledge of the analysis that is to be performed is required when designing an experiment.

1.3.3 Normalization

Even if some variability can be controlled using appropriate experimental design and procedures, other sources of errors can not be controlled, but still need to be corrected. The most famous of these source of variability is the dye bias for cDNA microarray experiments : the efficiency, heat and light sensitivities differ for Cy3 and Cy5, resulting in a systematically lower signal for Cy3. Furthermore, this signal can present an heterogeneous spatial repartition on the slide, due to micro physical properties of the hybridization mix on the slide. Normalization allows the adjustment for differences in labelling and for the detection efficiencies for the fluorescent labels, and for differences in the quantity of initial RNA from the two samples examined in the assay.

1.4 Downstream need for appropriate statistical tools

For many biologists, the need for statistical tools is new and can constitute a complete change in the way of thinking an experiment and its analysis. Although it is advisable for a biologist to collaborate with statisticians, it is crucial to understand the fundamental concepts underlying any statistical analysis. The problem is then to be confronted to various methods and concepts, and to choose among the appropriate ones. To that extend, it is crucial, from the statistician point of view, to diffuse statistical methods and concepts, to provide biologists as many informations as possible for them to be autonomous regarding the analysis needed to be performed. The role of softwares is central for microarray data analysis, but this review will rather be focused on statistical methods. Description of softwares dedicated to microarrays can be found in Parmigiani *et al.* (2003). Other informations can be found about general aspects of microarray data analysis in Quackenbush (2001), Leung and Cavalieri (2003), Butte (2002), Nadon and Shoemaker (2002) (this list is of course not exhaustive).

1.4.1 Class Discovery

The first step in the analysis of microarray data can be to perform a first study, without any *a priori* knowledge in the underlying biological process. The considerable amount of data requires automatic grouping techniques that aim at finding genes with similar behavior, or patients with similar expression profiles. In other words, the question can be to find an internal structure or relationships in the data set, trying to establish expression profiles. The purpose of unsupervised classifications is to find a partition of the data according to some criteria, that can be geometrical for instance. These techniques are widely used in the microarray community, but it is necessary to recall some fundamentals about clustering techniques: the statistical method will find a structure in the data because it is dedicated to it, even if no structure exist in the data set. This to illustrate that clustering will define groups based on statistical considerations, whereas biologists will want to interpret these groups in terms of biological function. The use and definition of appropriate clustering methods is detailed in chapter 4.

1.4.2 Class Comparison

Then the second question can be to compare the expression values of genes from a condition to another, or to many others. To know which genes are differentially expressed between conditions is of crucial importance for any biological interpretation. The aim of differential analysis is to assess a significance threshold above which a gene will be declared differentially expressed. Statistical tests consitute the core tool for such analysis. They require the definition of appropriate statistics and the control of the level of the tests. Chapter 5 show how the statistic has to be adapted to the special case of microarrays, and how the considerable amount of hypothesis tested leads to new definitions of control for statistical procedures.

1.4.3 Class Prediction

An other application to microarray data analysis is to use gene expression profiles as a way to predict the status of patients. In classification studies, both expression profiles and status are known for individuals of a data set. This allows to built a classification rule that is learned according to this training set. Then the objective is to be able to predict the status of new undiagnosed patients according to their expression profile. Since the number of studied genes is considerable in microarray experiments, another issue will be to select the genes that will be the most relevant for the status assignement. These problems of classification are detailed in chapter 6.

Chapter 2

Experimental designs

2.1 Aim of designing experiments

The statistical approach does not start once the results of an experiment have been obtained, but at the very first step of the conception of the experiment. To make the analysis really efficient, the way data are collected must be consistent with the statistical tools that will be used to analyze them. Our general message to biologists in this section is '*Do not wait till you get your data to go and discuss with a statistician.*'

The goal of experimental designs is to organize the biological analysis in order to get the most precise information from a limited number of experiments. Therefore, the design of experiments can be viewed as an optimization problem under constraints. The quantity to optimize is typically the precision of some estimate, which can be measured by the inverse of its standard deviation. A wide range of constraints (time, money, etc.) can occur. In this section, they will be summarized by the limitation in terms of number of experiments, i.e. by the number of slides.

What is a replicate? A basic principle of experimental designs is the need of replicates. In this section, most results will depend on the number R of replicates made under each condition. However, the definition of a replicate has to be precised. A set of R replicates can be constituted either by R samples coming from a same patients, or by R samples each coming from a different patient. In the former case, the variability between the results will be mostly due to technological irreproducibility, while in the latter it will be due to biological heterogeneity. The former are called *technological* replicates, and the latter *biological* replicates (see Yang and Speed (2002)).

The statistical approach presented in this section can be applied in the same way to both kinds of replicates. A *significant difference* between 2 conditions may be detected with technological replicates, but not with biological ones, because the biological variability is higher than the technological ones. Therefore, the significance is always defined with respect to a specific type of variability (technological or biological).

However, the biological conclusions will be completely different depending on the kind of replicates. In most cases, the aim of the experiment is to derive conclusions that are valid for a population, from which the individuals under study come from. In this purpose, only biological replicates are valid, since they take into account the variability between individuals. Effects observed on technological replicates can only be interpreted as *in vitro* phenomena: technological replicates are only useful to evaluate or correct technological biases

Contrasts and model. This chapter does not present a general overview of experimental designs for microarray experiments (that can be found in Draghici (2003)). Our purpose is to focus on the connection between the two following elements:

1. The kind of information one wants to get: we will mainly consider comparative experiments, the results of which are summarized in *contrasts*;
2. The model with which data will be analyzed: we will use the general framework of the analysis of variance (*anova*) model, proposed for example by Kerr and Churchill (2001) for microarray data analysis.

Paired and unpaired data. Of course, the experimental design strongly depends on the technological framework in which the biological analyses are performed. From a statistical point of view, there are two main type of microarray technology that respectively produce *unpaired* and *paired* data.

Unpaired data are obtained with technologies that provide measures under only one condition per slide, that is Affymetrix chip or nylon membrane. In this case, the different measures obtained for a given gene may be considered as independent from one chip (or membrane) to the other.

Paired data are produced by technologies where two different conditions (labeled with different dyes) are hybridized on the same slide. The values of the red and green signals measured for a same gene on a same slide can not be considered as independent, whereas the difference between them can be considered as independent from one slide to the other.

2.2 Two conditions comparison

The specific case of the comparison between 2 treatments will be intensively studied in chapter 5. We introduce here the general modeling and discuss some hypotheses, without going any further into testing procedure and detection of differentially expressed genes.

In such experiments, for a given gene, we may want to estimate

- its mean expression level μ_t in condition t ($t = 1, 2$),
- or its differential expression level $\delta = \mu_1 - \mu_2$.

2.2.1 Unpaired data

Statistical model

Assume that R independent slides are made under each condition ($t = 1, 2$), and denote X_{tr} the expression level of the gene under study, in condition t and replicate r (that is chip or membrane r). The basic statistical model assumes that the observed signal X_{tr} is the sum of a 'theoretical' expression level μ_t under condition t and a random noise E_{tr}, and that the residual terms $\{E_{tr}\}$ are independent, with mean 0 and common variance σ^2:

$$X_{tr} = \mu_t + E_{tr}, \qquad \{E_{tr}\} \text{ independent}, \quad \mathbb{E}(E_{tr}) = 0, \quad \mathbb{V}(E_{tr}) = \sigma^2. \tag{2.1}$$

Independence of the data. The model (2.1) assumes the independence of the data and all the results presented in this section regarding variances rely on this assumption. Independence is guaranteed by the way data are collected. Suppose the data set is constituted of measurements made on P different patients, with R replicates for each of them. The data set can not be naively considered as a set of PR independent measures, since data coming from a same patient are correlated. The analysis of such an experiment requires a specific statistical modeling, such as random effects or mixed model, which is not presented here.

Variance homogeneity. The model (2.1) also assumes that the variance of the noisy variable E_{tg} is constant. Most of the statistical methods we present are robust to moderate departure from this hypothesis. However, a strong heterogeneity can have dramatic consequences, even on the estimation of a mean. This motivates the *systematic use of the log-expression level,* for the log-transform is the most common transform to stabilize the variance. In this chapter, expression levels will always refer to log-expression levels.
It must be reminded that the common variance σ^2 can describe either a technological, or a biological variability, depending on the kind of replicates.

Parameter estimate

The estimation of the parameters of the model (2.1) is straightforward. The following table gives these estimates (denoting $X_{t\bullet} = \sum_r X_{tr}/R$, the mean expression level in condition t[1]) and there variances. We define the precision as the inverse of the standard deviation:

parameter	estimate	variance	precision
μ_t	$\hat{\mu}_t = X_{t\bullet}$	$\mathbb{V}(\hat{\mu}_t) = \sigma^2/R$	$\sqrt{R}/\sigma$
$\delta = \mu_1 - \mu_2$	$\hat{\delta} = X_{1\bullet} - X_{2\bullet}$	$\mathbb{V}(\hat{\delta}) = 2\sigma^2/R$	$\sqrt{R}/(\sigma\sqrt{2})$

The first observation is that the precision of the estimate is directly proportional to $1/\sigma$: the greater the variability, the worst the precision. This result reminds a fairly general

[1] In all this chapter, the symbol '$\bullet$' in place of an index means that the data are averaged over this index. For example, $X_{\bullet j \bullet} = \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ijk}/(IK)$.

order of magnitude in statistics: the precision of the estimates *increases at rate* $\sqrt{R}$. The number of experiments must be multiplied by 4 to get twice as precise estimates, and by 100 to get 10 times more precise estimates. It will be shown in chapter 5 that the power of the tests in differential analysis evolves in the same way.

2.2.2 Paired data

Slide effect

As explained in the introduction, the glass slide technology produces paired data. Due to heterogeneity between slides, a correlation between the red and green signals obtained on a same slide exists. Formally, the slide effect can be introduced in model (2.1) as follows:

$$X_{tr} = \mu_t + \beta_r + \varepsilon_{tr} \tag{2.2}$$

where β_r is the effect of slide r that can be either fixed or random. When two treatments are compared on the same slide r, β_r vanishes in the difference:

$$\begin{array}{ccccc} \underbrace{X_{1r} - X_{2r}} & = & \underbrace{\mu_1 - \mu_2} & + & \underbrace{\varepsilon_{1r} - \varepsilon_{2r}} \\ Y_r & = & \delta & + & E_r. \end{array}$$

This explains why most statistical analyses of glass slide experiments only deal with differences Y_r, generally referred to as *log-ratio* because of the log-transform previously applied to the data. Differences Y_r can be considered as independent, since they are obtained on different slides.

Labeling effect

The slide effect introduced in model (2.2) is not the only technological effect influencing the signal. It is well known that the two fluorophores Cy3 and Cy5 have not the same efficiency in terms of labeling, so there is a systematic difference between the signal measured in the two channels. Using index c (for 'color') to denote the labeling, the expression X_{tcr} of the gene in condition t, labeled with dye c on slide r can be modeled as:

$$X_{tcr} = \mu_t + \alpha_c + \beta_r + E_{tcr}. \tag{2.3}$$

Since there are only two dyes and conditions, indexes t and c are redundant given r. Treatment t can be deduced from the slide r and dye c indexes, or, conversely, dye c from slide r and treatment t. However, we need here to use both t and c to distinguish the biological effect we are interested in (μ_t) from the technological 'bias' (α_c).

Risk of aliasing. The redundancy described above may have strong consequences on parameter estimates. Suppose treatment $t = 1$ is labeled with dye $c = 1$ (and treatment $t = 2$ with dye $c = 2$) *on all slides.* Then, the treatment effect μ_t can not be estimated independently from the dye effect α_c since the mean expression level in condition 1 ($X_{1\bullet\bullet}$)

equals the mean expression level with dye 1 ($X_{\bullet 1 \bullet}$) and $X_{2\bullet\bullet} = X_{\bullet 2 \bullet}$ for the same reason. When each treatment is systematically labeled with the same dye, it is impossible to separate the true treatment effect from the labeling bias. This motivates the use of the 'swap' design.

Swap experiment

Design. The goal of the swap design is to correct the bias due to cDNA labeling by inverting the labeling from one slide to the other. This design involves two slides:

dye c		condition t	
		1	2
slide r	1	1	2
	2	2	1

Such a design is known as a *latin square* design.

Contrast. When comparing condition 1 and 2, the contrast δ is estimated by

$$\hat{\delta} = X_{1\bullet\bullet} - X_{2\bullet\bullet} = (X_{111} + X_{122})/2 - (X_{221} + X_{212})/2.$$

According to the model (2.3), the expectation of $\hat{\delta}$ is $\mathbb{E}(\hat{\delta}) = \mu_1 - \mu_2$, so the labeling and the slide effects are removed, simply because of the structure of the design. Hence, the swap design can be considered as a normalizing design.

Aliasing. The model (2.3) does not involve interaction terms, whereas they may exist. A general property of latin square design is that the interaction effects are confounded with the principal effects. For example the dye*slide interaction is confounded with the condition effect. This is because, in a swap design, the condition remains the same when both the labeling and the slide change.

When analyzing several genes at the same time, the aliasing mentioned above implies that the gene*treatment interaction is confounded with the gene*dye*slide interaction. The gene*treatment interaction is of great interest, since its reveals genes which expression differs between conditions 1 and 2.

Consequences of the tuning of the lasers. The tuning of the lasers is a way to get a nice signal on a slide. In many laboratories, a specific tuning of the lasers is applied to each slide, depending on the mean intensity of the signal. This specific tuning induces a dye*slide interaction, which often implies a gene*dye*slide interaction since the efficiency of the labeling differs from one gene to another.

Hence, the slide-specific tuning of the lasers implies a noisy effect (the gene*dye*slide interaction) that is confounded with the interesting effect (the gene*treatment interaction), due to the properties of the swap design. Any procedure (such as the loess regression, presented in chapter 3) aiming at eliminating the gene*dye*slide interaction will also reduce the gene*treatment effect. Therefore, *it is strongly advised to abandon slide-specific tuning*, and to keep the laser intensity fixed, at least for all the slides involved in a given experiment.

2.3 Comparison between T conditions

Many microarray experiments aim at comparing T conditions, denoted $t = 1, \dots, T$. We use here the term ‘condition’ in a very large sense. Conditions may be different times in a time course experiment, different patients in a biomedical assay, or different mutants of a same variety. In some cases, a reference condition (denoted 0) can also be considered, which may be the initial time of a kinetics, or the wild type of the variety.

In such experiments we may want to estimate the mean expression level μ_t in condition t of a given gene, or its differential expression level $\delta_{tt'} = \mu_t - \mu_{t'}$ between conditions t and t', with the particular case of $\delta_{t0} = \mu_t - \mu_0$ where t is compared to the reference.

Unpaired data. The results given in section 2.2 for unpaired data are still valid here. The estimates of μ_t and $\delta_{tt'}$, their variances and their precisions are the same.

2.3.1 Designs for paired data

When designing an experiment that aims at comparing T treatments, the central question is to choose which pairs of treatments must be hybridized on the same slide. This choice will of course have a major influence on the precision of the estimates of the contrast $\delta_{tt'}$. Figure 2.1 displays two of the most popular design to compare T treatements with paired data: the *star* and *loop* designs.

Two preliminary remarks can be made about these designs:

1. In both of them, the conditions are all connected to each other. This is a crucial condition to allow comparisons.

2. These 2 designs involve the same number of slides: TR (if each comparison is replicated R times); differences between them are due to the arrangement of the slides

“Star” design

In this first design, each of the T conditions is hybridized with a common reference. We assume here that each hybridization is replicated R times, and denote $Y_{tt'r}$ the logratio between condition t and t' on slide number r. In this setup, the estimates of the contrast

Figure 2.1: Design for comparing conditions $(0), 1, \ldots, T$ in paired experiments. Left: star design, right: loop design. Arrow '$\leftrightarrow$' means that the 2 conditions are hybridized on the same slide.

$\delta_{tt'}$ and their variances are the following.

contrast	estimate	variance	precision
δ_{t0}	$Y_{t0\bullet}$	σ^2/R	$\sqrt{R}/\sigma$
$\delta_{tt'}$	$Y_{t0\bullet} - Y_{t'0\bullet}$	$2\sigma^2/R$	$\sqrt{R}/(\sigma\sqrt{2})$

We see here that the precision of $\hat{\delta}_{t0}$ is better than the precision of $\hat{\delta}_{tt'}$. The weak precision of $\hat{\delta}_{tt'}$ is due to the absence of direct comparison between t and t' on a same slide.

In this design, half of the measures (one per slide) are made in the reference condition which means that half of the information regards the reference conditions. If the aim of the design is to compare, for example, a set of mutants to a wild type, it seems relevant to accumulate information on the wild type, which plays a central role. In this case, the star design is advisable. On the contrary, if the reference condition is arbitrary and has no biological interest, and if the main purpose is to compare conditions between them, then the star design is not very efficient in terms of precision of the contrasts of interest.

"Loop" design

In this design, conditions $1, \ldots, T$ are supposed to be ordered and condition t is hybridized with its two neighbor conditions $(t-1)$ and $(t+1)$ (Churchill (2002)). This design is especially relevant for time course experiments where the ordering of the conditions (times) is natural, and where the contrast between time t and the next time $t+1$ is of great biological interest.

Using the same notations as for the star design, the estimates of the contrasts, their variances and precisions are:

parameter	estimate	variance	precision
$\delta_{t(t+1)}$	$Y_{t(t+1)\bullet}$	σ^2/R	$\sqrt{R}/\sigma$
$\delta_{t(t+d)}$	$Y_{t(t+1)\bullet} + \cdots + Y_{(t+d-1)(t+d)\bullet}$	$d\sigma^2/R$	$\sqrt{R}/(\sigma\sqrt{d})$

The main result is that, *with the same number of slide* as in the star design, the precision of $\hat{\delta}_{t(t+1)}$ is twice better. Of course, the precision of the contrasts decreases as conditions t and $t+d$ are more distant in the loop: the variance increases linearly with d.

Loop designs are particularly interesting for time course analysis since they provide precise informations on the comparisons between successive times. They essentially rely on some ordering of the conditions. This ordering is natural when conditions correspond to times or doses but may be difficult to establish in other situations. In this last case, the ordering can be guided by the statistical properties described above: conditions that must be compared with a high accuracy must be hybridized on the same slide, or at least be close in the loop.

Normalization problem. The comparison between treatment 1 and T may induce some troubles in the normalization step. We remind that some normalization procedures are based on the assumption that most genes have the same expression level in the two conditions hybridized on the same slide (see 3) . If treatments are times or doses, this assumption probably holds when comparing condition t and $(t+1)$, but may be completely wrong for the comparison between conditions 1 and T.

Reducing the variance of the contrasts. Because the design forms a loop, there are always two paths from one condition to another. Because the variance of the estimated contrast $\hat{\delta}_{tt'}$ is proportional to the number of steps, it is better to take the shortest path, rather than the longest one, to get the most precise estimate. Suppose we have $T = 8$ conditions, the shortest path from condition 1 to condition 6 has only 3 steps: $1 \rightarrow 8 \rightarrow 7 \rightarrow 6$, so the variance of $\hat{\delta}_{16} = Y_{1,8\bullet} + Y_{87\bullet} + Y_{76\bullet}$ is $3\sigma^2/R$. The longest path leads to the estimate $\hat{\delta}'_{16} = Y_{12\bullet} + \cdots + Y_{56\bullet}$, the variance of which is $5\sigma^2/R$.

A new estimate $\tilde{\delta}_{tt'}$ can be obtained averaging the two estimates: $\tilde{\delta}_{tt'} = w\hat{\delta}_{tt'} + (1 - w)\hat{\delta}'_{tt'}$. The weight w has to be chosen in order to minimize the variance of $\tilde{\delta}_{tt'}$. If $\hat{\delta}_{tt'}$ is based on a path of length d (and $\hat{\delta}'_{tt'}$ on a path of length $T-d$), the optimal value of w is d/T. The variance of $\tilde{\delta}_{tt'}$ is then $d(T-d)\sigma^2/(TR)$. In particular, the variance of $\tilde{\delta}_{t(t+1)}$ is $(T-1)\sigma^2/(TR)$, which is smaller than the variance of $\hat{\delta}_{t(t+1)}$ (which is σ^2/R). Even in this very simple case, the estimate is improved by considering both the shortest and the longest path.

Chapter 3

Data normalization

Microarray data show a high level of variability. Some of this variability is relevant since it corresponds to the differential expression of genes. But, unfortunately, a large portion results from undesirable biases introduced during the many technical steps of the experimental procedure. Thus, microarray data must be corrected at first to obtain reliable intensities corresponding to the relative expression level of the genes. This is the aim of the normalization step, which is a tricky step of the data process. We present in 3.1 exploratory tools to detect experimental artifacts. Section 3.2 reviews the main statistical methods used to correct the detected biases, and Section 3.3.2 discusses the ability for biologists to reduce experimental variability and facilitate the normalization step in microarray experiments.

3.1 Detection of technical biases

Most technical biases can be detected with very simple methods. We recommend as many authors the systematic use of graphical representations of the slide and other diagnostic plots presented in the following. We distinguish here exploratory methods that look for no particular artifact, from methods that diagnose the presence of a specific artifact.

3.1.1 Exploratory methods

A simple way to observe experimental artifacts is to represent the spatial distribution of raw data along the slide, as in Figure 3.1. Cy3 or Cy5 log-intensities, background, log-ratios $M = logR - logG$ or mean intensity $A = (logR + logG)/2$ can be plotted this way as an alternative to the classical scanned microarray output images. These representations are very useful to detect unexpected systematic patterns, gradients or strong dissimilarities between different areas of the slide. As an example, we present here a simple case where a single *Arabidopsis* slide was hybridized with Cy3 and Cy5 labeled cDNA samples to analyse the differences in gene expression when *Arabidopsis* is grown either on environment A or B. The spotting was performed with a robot whose printing head consisted of 48 (4×12) print-tips, each of them spotting in duplicate all the cDNA sequences of an entire rectangular area of the glass slide, defining a block. In this experiment, we are interested by the impact of the treatments and of some possible technical artifacts.

Figure 3.1 (left) represents the distribution of M along the slide. It shows particular areas with high level signals that could correspond to cDNA sequences spotted with faulty print-tips: for instance, if the opening of these print-tips is longer than those of other ones, the amount of material they deposit could be systematically more extensive for sequences deposited by these print-tips.

3.1.2 Detection of specific artifacts

Graphical analysis: Once an artifact is suspected, plots that reveal its presence can be performed. Typical specific representations include *boxplots*. For a given dataset, the boxplot (Fig. 3.1, right) represents the middle half of the data (first to third quartiles) by a rectangle with the median marked within, with *whiskers* extending from the ends of the box to the extremes of the data or to one and a half times the interquartile range of the data, whichever is closer . To compare the distribution between different groups, side-by-side per group boxplots can be performed. Figure 3.1 (right) shows per print-tip boxplots for the *Arabidopsis* slide, and confirm a differential effect of print-tip 6(shown) and 32, 35, 36 (not shown).

At last, a convenient way to compare variables distribution of different slides from a same experiment is to use a *Quantile-Quantile plot* (QQplot). A QQ plot plots empirical quantiles from the signal distribution on a slide against the ones of an other slide. If the resultant plot appears linear, then the signal distributions on both slides are similar.

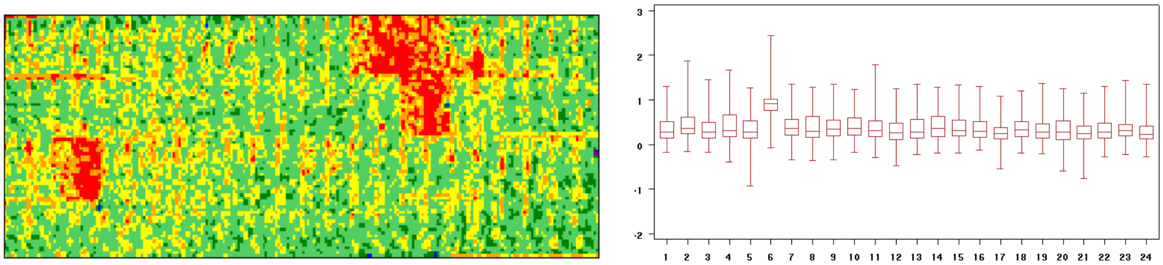


Figure 3.1: **Left:** Spatial distribution of the signal on the slide. Each pixel represents the uncorrected log-ratio of the median Cy5 (635 nm) and Cy3 (532 nm) channel fluorescence measurements, associated to a printed DNA feature. Background is not represented. Red squares correspond to print-tip effect. **Right:** Box plots per print-tip for the first 24 blocks of the previous slide. Print-tip 6 corresponds to the red square on the left of the slide.

Analysis of variance: An alternative to graphical displays is the use of the *Analysis of Variance* (ANOVA). The ANOVA is a powerful statistical tool used to determine which factors explain the data variability. To this end, sums of squares are used to quantify the effect of each factor, and tests can be performed to state their significance. The use of the

ANOVA to analyse microarray data was first proposed by Kerr *et al.* (2000). We present here the ANOVA analysis performed for the *Arabidopsis* slide.

The effect of four factors is studied: growth in the presence of treatment A or B, Cy3 and Cy5 intensity dependent effect, print-tips artifacts, and genes. Interactions between factors are also considered. We denote X_{gtfp} the measured signal of gene g whose RNA was extracted from cells of *Arabidopsis* grown in presence of treatment t, labeled with fluorochrome f, and spotted with print-tip p. The complete ANOVA model is:

$$\begin{array}{lll} X_{gtfp} & = \mu + \alpha_g + \beta_t + \gamma_f + \delta_p & \text{main effects} \\ & +(\alpha\beta)_{gt} + (\alpha\gamma)_{gf} + (\alpha\delta)_{gp} & \text{interactions of order 2 with gene effect} \\ & +(\beta\gamma)_{tf} + (\beta\delta)_{tp} + (\gamma\delta)_{fp} & \text{other interactions of order 2} \\ & +(\alpha\beta\gamma)_{gtf} + ... & \text{interactions of order 3} \\ & E_{gtfp} & \text{residual} \end{array} \tag{3.1}$$

where residuals E_{gtfp} are supposed to be independent with common variance and 0-centered random variables, that represent the measurement error and the biological variability altogether. In practice, most of the interaction are neglected or confounded with other effects, leading to simpler models (see Kerr *et al.* (2000)). Notice that in our example, the *Treatment* effect is confounded with the *Dye* effect. In this case the model sums up to:

$$X_{gfp} = \mu + \alpha_g + \gamma_f + \delta_p + (\alpha\gamma)_{gf} + (\gamma\delta)_{fp} + E_{gfp} \tag{3.2}$$

were γ_f is the confounded effect of both fluorochrome and treatment.

The analysis of variance is summarized in Table 3.1. The *Dye* × *Gene* interaction appears to be the less important effect in this experiment. This can be worrisome, since due to aliasing this interaction also corresponds to the *Treatment* × *Gene* interaction of interest. It seems then that the differential effect of the treatments on genes in negligible compared to the experimental effects. But these low MS are partly due to the huge degree of freedom of the interaction, that makes the detection of a differential effect more difficult: indeed we look for the differential effect of at least one gene among 10080, whereas for the print-tip effect for instance we look for the differential effect of at least one print-tip among 48 (explicit formulas of expected sums of squares can be found in Draghici (2003), Chap. 8). We will see in Section 3.2.2 that with a simpler modelling, the *Dye* × *Gene* effect appears to be strong.

Table 3.1 shows that the $Print-tip$ effect is one of the main experimental artifacts of this experiment, confirming the results of the exploratory analysis of the previous section. Normalization will then be crucial step of the data analysis. Moreover, the quantification of effects is a precious knowledge for the experimenter, who will carefully control the print-tips in following experiments.

The application of the presented descriptive tools already enabled the discovery of several sources of experimental noise , such as dye or fluorophore, and print-tips (Yang *et al.* (2002), Schuchhardt *et al.* (2000)). Even if exploratory methods seem to be more appropriate for the identification of new experimental artifacts, it should be clear that

Effect	d.f.	M.S.
Print-tip	47	131.17
Dye	1	1647.19
Gene	10032	4.24
Dye×Print-tip	47	4.60
Dye×Gene	10032	0.08

Table 3.1: Analysis of variance (d.f.=degrees of freedom, M.S.=Mean Squares)

the detection of experimental sources of noise is mostly based on an accurate knowledge and analysis of the experimental process that will help to propose adapted tools for the normalization.

Once these experimental effects are detected, one needs procedures to correct them. The following section presents the main tools that are used in common normalization procedures.

3.2 Correction of technical artifacts

Most experimental artifacts alter the signal mean, i.e. the mean value of the log-ratios of genes. The main function of normalization methods is then to quantify the effect of a given experimental bias on a gene, and second to subtract this quantity from the observed gene log-ratio value. The tricky part of the normalization is obviously the estimation of the effect contribution. One has to distinguish between systematic biases, that do not depend on gene and can be easily corrected with simple methods, and gene dependent biases, that generally request a more sophisticated modelling to be corrected. These two kinds of biases and their associated normalization procedures are described in the two following sections.

Alternatively, some artifacts can alter the signal variance. Methods that have been proposed for variance correction are presented in Section 3.2.3.

3.2.1 Systematic biases

Since most experimental sources of noise can be considered as systematic, the effect they have will be identical for all the genes they affect. For instance, we saw that the print-tip effect alter all gene log-ratios of an block. A possible modelling of the print-tip effect is to assume that the bias is constant within each block. The log-ratios are corrected by subtracting a constant c_i to log-ratios of block i, where c_i is estimated from the log-ratio mean of block i. This normalization can be performed with the previous ANOVA model by just adding a print-tip effet in model (3.1). A more robust estimation of systematic effects can be made replacing the mean by the median (Yang *et al.* (2002)), which is the method usually implemented on normalization softwares. Figure 3.2 shows the boxplots after per print-tip median normalisation: the bias observed for print-tip 6 is now corrected. Other systematic biases that can be considered as systematic and similarly corrected include slide and plate effects (this list is not exhaustive).

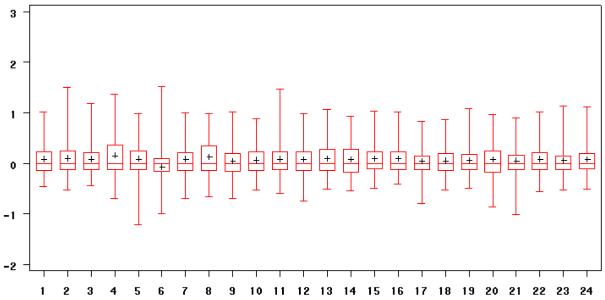


Figure 3.2: Box plots per print-tip for the first 24 blocks of the *Arabidopsis* slide, after print-tip normalization.

3.2.2 Gene dependent biases

All biases cannot be modeled as systematic effects, because their impact is gene dependent. We present the case of the dye or fluorochrome effect for cDNA microarrays.

To perform a comparison between two conditions labelled with Cy3 and Cy5, respectively, one needs to state that the differential labelling will not corrupt the log-ratio values. Yet, it is well known that a dye effect exists, that can have two different causes:

- optical : the higher the mean intensity of the gene is, the more the green label prevails over the red one when the slide is scanned.
- biological : some specific genes are systematically badly labeled by Cy3 or Cy5. For instance, Cy3 can be preferentially incorporated into some sequences, relative to Cy5.

The dye effect is then clearly gene dependant. To correct it, one can estimate each $Dye \times Gene$ interaction in model (3.2), and subtract it from log-ratios per gene. But this requests as many estimations as G. Most of them will be very imprecise, and the resulting normalized log-ratios could be noisier than the raw log-ratios. The estimation problem can be avoided by proposing a simpler modelling of the $Dye \times Gene$ interaction. For instance, we can assume that the dye effect depends on gene only through its mean intensity A. This assumption allows a convenient graphical observation of the dye effect, the M-A plot, proposed by Yang *et al.* (2002), along with a more robust estimation of the effect. In figure 3.3 (left) we observe the differential effect of the two dyes: M values increase with A values, confirming that Cy5 signal prevails for high mean expression genes. Moreover, it is clear that the shape of the data cloud is neither constant nor linear, meaning that a constant or linear modelling will not adequately correct the dye effect. In this case, one needs to perform non linear normalization methods.

The Loess procedure (Cleveland (1979)) was the first non linear method proposed to correct the dye effect (Yang *et al.* (2002)). The Loess is a robust locally weighted regression based on the following model:

$$M = c(A) + E \tag{3.3}$$

where c is an unknown function and E is a symmetric centered random variable with constant variance. The aim of the Loess procedure is to locally approximate c with a polynomial function of order d, and to estimate the polynomial parameters by weighted least square minimization from the neighbor points (A_i, M_i). Weights depend on the distance between point (A_i, M_i) and the neighborhood center: the lower the distance, the higher the weight. The size of the neighborhood is fG, where f is a proportion parameter that ranges from 0 to 1. If f is close to 1, the neighborhood will contain almost all the sample points and the estimated function will be very smooth. Conversely, if f is close to 0, the function will be very adaptive to the data cloud. The correction will be more specific but the risk for overfitting will increase. In figure 3.3 (left) the Loess estimation of the data cloud trend appears in grey. As for systematic biases, once the trend is estimated it is substracted from the log-ratio to obtain a centered data cloud.

As described above, the Loess function request the tuning of many parameters, mainly the weight function, the order of the polynomial function, and the size of the neighborhood. In dedicated softwares, all these parameters are fixed to a by default value. Yet, it is worth mentioning that the efficiency of the normalization can be highly dependent on the choice of these parameters. Alternative non linear methods have been proposed to correct intensity dependent biases: for instance, Workman *et al.* (2002) proposed the use of cubic splines instead of Loess. But the Loess has become the reference method implemented in most softwares. Common normalization procedures also include by-print tip Loess normalization.

One has to know whether the Loess procedure completely corrects the dye effect, i.e. if the assumption that the dye effect is gene dependent only through A is satisfied. In Martin *et al.* (2004), it is shown that the incorporation bias can be important, and is not corrected by the Loess procedure. This is the reason why it is recommended to make swap experiments (see 2.2.2), even if the Loess or any other intensity dependent procedure is performed during the normalization step.

3.2.3 Variance normalization

Besides, most of the statistical methods that are used to normalize and analyse the data assume that all observations have the same variance. To ensure this hypothesis, data are systematically log-transformed at first in order to stabilize the variance (see 2.2.1). Although most sources of experimental variability mainly affect the level of log-ratios, the variance of the observation can also be affected by artifacts. In this case one has to

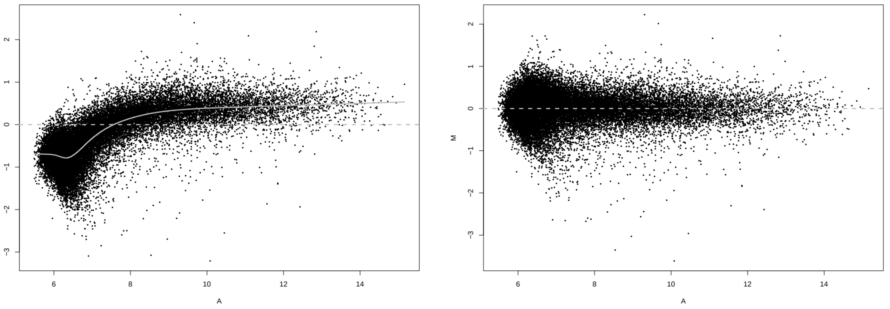


Figure 3.3: **Left:** M-A graph on raw data. The gray line is the loess estimation of function c, the dotted line represents the abscissa axis **Right:** M-A graph after Loess normalization

normalize the variance. For instance, boxplots on figure 3.2 show that log-ratio variances slightly differ from one print-tip to another after a per print-tip median correction.

As for bias normalization, the distinction between systematic and gene dependent artifacts exists, with the same consequences. We only deal here with systematic heteroscedasticity through the print-tip example. Genes that were spotted by the same print-tip are assumed to have the same variance, that can be estimated from the empirical standard deviation. The log-ratios are divided by their respective empirical SD to be normalized. As for mean effect normalization, robust methods of estimation exist for the standard error: in Yang *et al.* (2002), the authors propose the use of MAD (Median Absolute Deviation) estimation.

3.3 Conditions for normalization

Considering the previous section, it is clear that some fundamental hypotheses have to be verified to perform any normalisation procedure. At the same time, normalization can also be simplified by a sharp control of the data quality and an adapted experimental design. The first following section discusses the three main points to be checked before normalization and the second one proposes some guidelines to enhance the data normalization.

3.3.1 Three hypotheses

Normalization procedures are based on the three following hypotheses:

- Most of genes that are used to estimate the artifact contribution to signal are supposed not to be differentially expressed,

- The artifacts that are corrected are not confounded with a biological effect,

- The technical variability of the artifact estimator is small compared to the biological variability.

The first hypothesis is stated to be sure that genes used for estimation have a constant expression w.r.t. the biological problem, and therefore only reflect bias effects (Ball *et al.* (2003)). The use of housekeeping genes whose expression is supposed to be constant has been proposed, but such genes are difficult to identify. This is the reason why in many cases all genes are used for the normalization, implying that only a minority of them are expected to be differentially expressed. Notice that for some specific experiments this last hypothesis cannot hold: dedicated experiments where only a few but relevant genes are spotted on the slide, or loop designed kinetics experiments where the last time point is compared to the first time point on a same slide are typical examples of departure to the hypothesis.

The second hypothesis is also important since normalization aims at reducing the experimental variability of the data without altering the biological information contained in the data. It is then important to determine the conditions in which the correction of an experimental effect is appropriate. In Section 2.2.2, we already saw that if a given treatment is always treated with the same fluorochrome, it will be impossible to distinguish the dye effect from the treatment effect. The same problem exists with other biases correction, for example in by-plate normalization (Mary-Huard *et al.* (2004)). It is worth mentioning that no correction can be performed when confusion occurs, meaning that the experimental effect remains, and can considerably affect the biological conclusions of experiments (Balazsi *et al.* (2003)).

The last hypothesis amounts to state that the normalization step does correct data rather than adds noise. We already observed in the previous section that the estimation of the $Dye \times Gene$ interaction is based on very few observations, leading to a estimator possibly noisy enough to alter the data. This can be generalized to other normalization procedures, such as background correction for example. In background correction, the background measurement is subtracted to the signal at each spot. Such correction is reasonable only to the condition that the background is a sharp indicator of the local quality of the slide. In practice, the background measurement can be as imprecise as the signal measurement, therefore the background corrected signal will be unreliable. To ensure the normalization quality, one can increase the number of technical replicates, in order to have an accurate estimation of the technical variance to compare to the biological variance. Alternatively, it is important to verify that estimations of technical artifacts are based on a large enough number of observations to be robust.

3.3.2 Enhancement of the normalization

As pointed out by Quackenbush (2002), "the single most important data-analysis technique is the collection of the highest-quality data possible". It is clear that no normalization procedure can compensate for poor quality data: it is thus important to control carefully the wet laboratory microarray process. We consider here guidelines that can help to design and perform an efficient normalization procedure.

The previous section and chapter 2 already pointed out that the normalization process and its efficiency intimately depend on the experimental design. Optimizing the design will lead to accurate estimations of the log-ratios, and will help the quantification and the correction of experimental biases. A good experimental design will also avoid confusion between biological and experimental effects when possible. Therefore a particular care must be given to the experimental design.

We already considered the fact that any normalization procedure is susceptible of altering the data, so every effort must be made to avoid intensive data transformation. The data normalization process should be as reduced and as specific to the platform as possible. For instance, it is clear that the dye effect is detectable in most experiments, along with block effects. Nonetheless the use of per-block loess normalization should not be systematical, since the number of genes spotted on a block vary from less than a hundred to more than four hundred. In the former case, the use of a local regression can lead to an overfitted adjustment. Therefore, depending on platform, the experimenter will have to choose either to tune parameter f appropriately, or to perform a global loess and a per block median normalization.

Due to the now intensive microarray production, it is unrealistic to question the normalization procedure at each microarray analysis. But the elaboration of an effective and platform-tailored normalization procedure can be eased by the use of self-hybridized microarray experiments. Self-hybridization experiments have proved to be efficient in detecting systematic biases (Ball *et al.* (2003)) and provide simple means to test normalization procedures. They can be used by platforms as test data to calibrate the normalization process, but also as quality control experiments that can be regularly performed to adapt the normalization with time.

Chapter 4

Gene clustering

Aim of clustering

Summarizing information. Clustering analysis is probably the most widely used statistical tool in microarray data analysis. Because of the size of the data sets provided by microarray experiments, the information needs to be summarized in some way for any synthetic interpretation. Clustering techniques are of great help in this task, since they reduce the size of the data sets by gathering genes (or tissues) into a reduced number of groups. In many cases, clustering analysis are only considered as a convenient way to display the information present in the data set. One purpose of this chapter is to show that the choice of the clustering algorithm has a strong influence on the final result, so this result can never be considered as an objective representation of the information.

Defining biologically relevant groups. From a biological point of view, a more ambitious task is often assigned to clustering analysis. The understanding of gene functions and the discovery of 'co-regulated' genes are two typical goals of microarray experiments. A natural way to achieve them is to try to gather genes having similar expression profiles in a set of conditions, at different times or among different tissues into *clusters*. These clusters may then be interpreted as functional groups and the function of an unknown gene can be inferred on the basis of the function of one or several known genes belonging to the same cluster (cf. groups labeled A to E in Figure 4.2).

Data set

The basic data set is an array $\mathbf{X}$ with G rows and T columns, G being the number of genes and T the number of conditions (or times, or tissues). The element x_{gt} at row g and column t denotes the (log-)expression level of gene g in condition t.

All along this chapter, we will consider the problem of clustering genes according to their expression profiles among conditions or tissues. However, the clustering of tissues (according to the expression levels of the different genes) can also be relevant to discover particular subclasses of disease. In this case, the algorithm is simply applied to the transposed matrix **X**. An example of such a dual analysis can be found in Alizadeh *et al.* (2000) where the authors both define groups of patients and groups of genes.

Two approaches for a general problem

The aim of clustering technique is to build groups of items without any prior information about these groups: such algorithms perform an *unsupervised classification* of the data, or *class discovery*. Schaffer *et al.* (2001) presents a typical clustering analysis of gene expression data: genes are spread into 5 clusters, each characterized by an 'idealized pattern' that is a smoothed version of the mean expression profile of the cluster.

There are essentially two families of clustering methods: *distance-based* and *model-based methods*. The former only aim at gathering similar genes according to a dissimilarity measure given a priori. These methods are essentially geometric and do not assume much about the structure of the data. The latter are based on a statistical modeling that is supposed to reveal the underlying structure of the data. The aim of these methods is to discover this underlying structure, that is the potential belonging of each gene to the different cluster, as well as the general characteristics of these clusters.

Distance-based methods are the most popular in microarray data analysis, mainly because of their computational efficiency. However, these methods do not take the variability of the data into account, while model-based methods do, thanks to the statistical modeling. This is a major drawback of distance-based methods, because of the weak reproducibility of microarray data.

Moreover, most clustering techniques provide disjoint clusters, which means that they assign each gene to one single group. This property is not always biologically desirable: clusters are often interpreted as groups of co-regulated genes and, therefore, connected with regulation networks. A gene can be involved in several networks and should therefore be allowed to belong to more than one cluster. In contrast, model-based methods perform fuzzy affectation by assigning to each gene a probability of belonging to each of the clusters. Up to now, these methods have received very few attention in the microarray community, probably because of their computational complexity.

The first aim of this chapter is to present in detail the most popular distance-based algorithms, emphasizing the arbitrary choices that underly all of them, in particular the definition of the distance. Our second purpose is to introduce model-based methods and to show that, in some situations, they seem to be more adapted to the biological questions under study.

4.1 Distance-based methods

4.1.1 Dissimilarities and distances between genes

The dissimilarity $d(g, g')$ between gene g and g' is the basic element of the first type of clustering algorithms presented here. Many algorithms only require a *dissimilarity*, that is a function d satisfying the 3 following properties: (i) d is positive: $d(g, g') \geq 0$, (ii) symmetric: $d(g, g') = d(g', g)$, and (iii) null only between g and itself: $\{d(g, g') = 0\} \Leftrightarrow \{g = g'\}$. Some algorithms require a *distance*, that is a dissimilarity satisfying the triangular inequality:

$$\forall g, g', g'' : \quad d(g, g'') \leq d(g, g') + d(g', g'').$$

Euclidian distances. The most popular distances are the simple and standardized Euclidian distances. Denoting $x_{\bullet t}$ the mean expression level in condition t: $x_{\bullet t} = \sum_g x_{gt}/G$ and σ_t^2 the variance of these levels in condition t: $\sigma_t^2 = \sum_g (x_{gt} - x_{\bullet t})^2/G$, this distances are defined as

$$\begin{array}{lrcl} \text{simple Euclidian:} & d^2(g, g') & = & \sum_t (x_{gt} - x_{g't})^2, \\ \text{standardized Euclidian:} & d^2(g, g') & = & \sum_t (x_{gt} - x_{g't})^2/\sigma_t^2. \end{array}$$

The simple distance gives the same weight to all conditions t, while the standardized one penalized the conditions with high variance, presuming that a large difference $(x_{gt} - x_{g't})$ is more admissible in highly variant conditions than in very stable ones.

Correlation coefficient. In their seminal paper on clustering technique for microarray data (and in the related free software), Eisen *et al.* (1998) proposed to use dissimilarity based on the correlation coefficient. Denoting $x_{g\bullet}$ the mean expression level of gene g : $x_{g\bullet} = \sum_t x_{gt}/T$, the (centered) coefficient is defined as

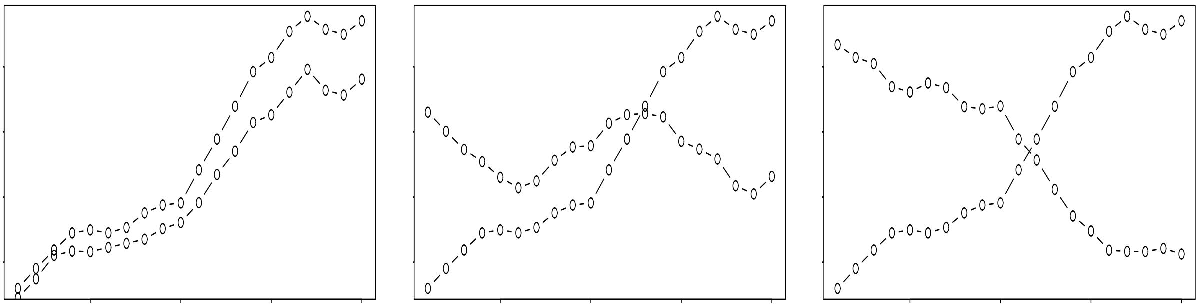
$$r(g, g') = \sum_t (x_{gt} - x_{g\bullet})(x_{g't} - x_{g'\bullet}) \Bigg/ \sqrt{\sum_t (x_{gt} - x_{g\bullet})^2 \sum_t (x_{g't} - x_{g'\bullet})^2} .$$

When the data are normalized (that is when the mean expression level of each gene $x_{g\bullet} = \sum_t x_{gt}/T$ is set to 0 and its variance $s_g^2 = \sum_g (x_{gt} - x_{g\bullet})^2/T$ is set to 1), $r(g, g')$ is related the simple Euclidian distance $d^2(g, g')$: $r(g, g') = 1 - d^2(g, g')/(2T)$.

Choice of the dissimilarity. A general discussion about the crucial point of the choice of a 'good' dissimilarity can not be given here. We only illustrate the influence of this choice on a simple example. The correlation coefficient must be transformed to take positive values, in order to get a dissimilarity. Two dissimilarities can be derived from $r(g, g')$:

$$d_1(g, g') = [1 - r(g, g')]/2, \quad \text{or} \quad d_2(g, g') = 1 - [r(g, g')]^2 .$$

Both d_1 and d_2 will be small for positively correlated genes ($r \simeq 1$), but d_1 will be high for negatively correlated genes ($r \simeq -1$), while d_2 will be small (see Figure 4.1). Using d_1, genes having opposite profiles will belong to different clusters, while, using d_2, they will be gathered in the same one. If clusters are to be interpreted as sets of genes involved in a same regulatory network, it seems that d_2 is more relevant since opposite profiles are often observed in a same pathway. The choice between d_1 and d_2 is a matter of definition of similar or 'co-regulated' genes, which is a biological question, and not a statistical one.



$r = 0.9$
$d_1 = 0.05 \quad d_2 = 0.81$

$r = 0.0$
$d_1 = 0.50 \quad d_2 = 0.95$

$r = -0.9$
$d_1 = 0.95 \quad d_2 = 0.81$

Figure 4.1: Fictitious time courses. r is the correlation coefficient between the two courses, $d_1 = (1 - r)/2$, $d_2 = 1 - r^2$.

Time course experiments. Clustering algorithms are often applied to time-course experiments in which conditions $1, \dots T$ are ordered times. The distances and dissimilarities presented here can be used for such data but it must be noted that they do not account for the ordering structure of the times. The columns of the data set can be randomly permuted without affecting the distances between genes. However, time-course data can be analyzed from a more dynamic point of view by considering variations $\delta_{g,t} = x_{g,t} - x_{g,t-1}$ instead of levels $x_{g,t}$. A specific modeling of time-course data will be presented in Section 4.2.2.

4.1.2 Combinatorial complexity and heuristics

A clustering is satisfying when groups are

(i) homogeneous (with low within-group variability),

(ii) well separated (with high between-group variability).

Given some criterion measuring the quality of the clustering (such as the within-group inertia defined in section 4.1.4), one may search for the *best* clustering, that is the best partition of a set of G genes into K groups. The number of such partitions is given by the Bell number: $\sum_{k=1}^{K}(-1)^k(K-k)^G/[k!(K-k)!]$. There are approximately 10^{47} possible partitions of $G = 100$ genes into $K = 3$ groups, and 10^{68} for $K = 5$ groups. This shows that, no matter the power of the available computers, there is no way to explore all possible partitions.

This complexity motivates the use of heuristics. We will now introduce two of the most popular clustering algorithms (see Anderberg (1973)):

- hierarchical clustering, that works with an unknown number of groups K
- and K means for which K has to be known.

4.1.3 Hierarchical clustering

The principle of hierarchical algorithms is to build the clusters by joining iteratively the two 'closest' genes or groups of genes. This is clearly a heuristic approach that aims at minimizing the within-group variability. The result is generally displayed as a *tree* (dendrogram), as shown in Figure 4.2.

It has to be noted that the tree structure is the result of the clustering history, but does not reveal some presupposed underlying structure. This makes a major difference with, for example, phylogenetic trees that are obtained in the framework of an evolutionary models that involves a tree structure. Hierarchical algorithms *always* provide a tree, even if the data are *not structured* according to a tree. Even though dendrograms are considered as simple visualization tools, it must be stressed that it is a very particular representation of the data, that can be completely irrelevant. This is a major drawback of these 'algorithmic' approaches: because of the lack of statistical modeling, the fit of the representation to the data is difficult to assess.

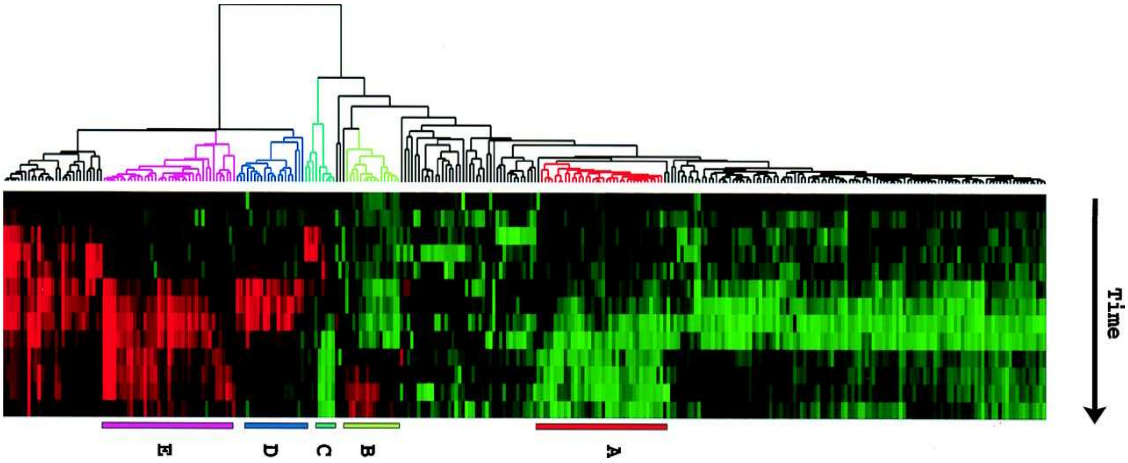


Figure 4.2: Hierarchical clustering of gene expression data, from Eisen *et al.* (1998). Groups A to E are putative functional groups, containing few genes with known function: A = cholesterol biosynthesis, B = cell cycle, etc.

Hierarchical algorithm

The general hierarchical clustering algorithm is the following:

Initialization. Calculate the $G \times G$ matrix $\mathbf{D}$ containing the dissimilarities between all the couples of genes (called the *dissimilarity matrix*);
Set the y-value of each gene in the dendrogram to 0.

Iteration: Proceed steps 1 to 4.

1. Find the smallest dissimilarity in $\mathbf{D}$ and the corresponding couple of genes or groups of genes (g_1, g_2);
2. merge g_1 and g_2 into a new group of genes g_{12}; set the y-value of g_{12} in the dendrogram to $d(g_1, g_2)$[1];
3. calculate the dissimilarity between the new group g_{12} and all the other genes or groups of genes g $(g \neq g_1, g_2)$;
4. remove rows and columns corresponding to g_1 and g_2 from matrix $\mathbf{D}$ and add one row and column corresponding to g_{12} and go back to step 1.

Distance between groups

The first steps of the algorithms generally result in the gathering of single genes into couples. Once genes have been merged into groups, we need a dissimilarity $d(\mathcal{C}, \mathcal{C}')$ between groups of genes to let the process go on. This second dissimilarity is sometimes called *aggregation criterion*, and traditionally gives the name of the general clustering algorithm ('Single linkage algorithm', 'Ward algorithm', etc). We present here some of the most popular.

Single linkage. The dissimilarity between groups $\mathcal{C}$ and $\mathcal{C}'$ is defined as the smallest distances between their elements: $d(\mathcal{C}, \mathcal{C}') = \min_{g \in \mathcal{C}, g' \in \mathcal{C}'} d(g, g')$. This criterion is often considered as parsimonious since it assumes that two groups are close to each other if some of their elements are close. It is known to give very unbalanced groups, the groups of big size absorbing isolated elements one by one.

Average linkage. The dissimilarity is the mean dissimilarity between elements of $\mathcal{C}$ and $\mathcal{C}'$: $d(\mathcal{C}, \mathcal{C}') = \sum_{g \in \mathcal{C}} \sum_{g' \in \mathcal{C}'} d(g, g') / (|\mathcal{C}||\mathcal{C}'|)$, where $|\mathcal{C}|$ denotes the number of genes in group $\mathcal{C}$.

Complete linkage. This criterion follows the opposite principle of the single linkage: $d(\mathcal{C}, \mathcal{C}') = \max_{g \in \mathcal{C}, g' \in \mathcal{C}'} d(g, g')$ and strongly penalizes large groups.

Centroid. The centroid dissimilarity only accounts for the centers $\overline{g}$ and $\overline{g}'$ of the groups, no matter of their size: $d(\mathcal{C}, \mathcal{C}') = d(\overline{g}, \overline{g}')$.

Ward. The Ward criterion is interesting because it is consistent with principle component analysis (PCA, see Anderson (2003) for a general presentation, or Alter *et al.* (2000) for an application to microarray). At each step, two elements (genes or groups) are gathered to form a new element. Ward defines the loss of information due to this gathering as the within inertia (defined in equation (4.1)) of these two elements and uses it as a dissimilarity (that is actually a distance). The resulting criterion is $d^2(\mathcal{C}, \mathcal{C}') = |\mathcal{C}||\mathcal{C}'| d^2(\overline{g}, \overline{g}') / (|\mathcal{C}| + |\mathcal{C}'|)$.

[1] Another representation can be obtained by cumulating the dissimilarities of all the past steps.

A huge number of different criterions have been proposed in the literature. It can be noted that one of the oldest references is Sokal and Sneath (1963) in which are defined the Unweighted/Weighted Pairwise Group Method Average/Centroid (UPGMA, UPGMC, etc.)

Stopping rule

From a theoretical point of view, the clustering is only achieved when groups are completely defined. Letting the aggregating process go on will lead from a classification where each gene is a class to another classification where all the genes belong to the same class. Of course, none of these two classifications is biologically relevant.

The general idea to choose the number of groups in hierarchical algorithms is to cut the tree at a given height d^*. Depending on the definition of the y-axis of the tree, we get two different stopping rules.

Local criterion. The y-axis of the tree is defined as the distance between the two elements being merged. Cutting this tree at a level d^* means that the aggregating process stops as soon as the distance between the two closest elements exceeds d^*.

Global criterion. The y-axis is the sum of all the distances between the elements that have been merged since the first step of the algorithm. In the case of the Ward algorithm, this sum is exactly the information (defined as the inertia) lost since the beginning of the process. Cutting the tree at height d^* means that the algorithm stops when the loss of information exceeds d^*.

In practice many users do not use any stopping rule and define the clusters simply by looking at the dendrogram. In Figure 4.2, we see that groups A to E correspond to very different heights in the tree: they have defined according to exogenous information regarding genes with known functions.

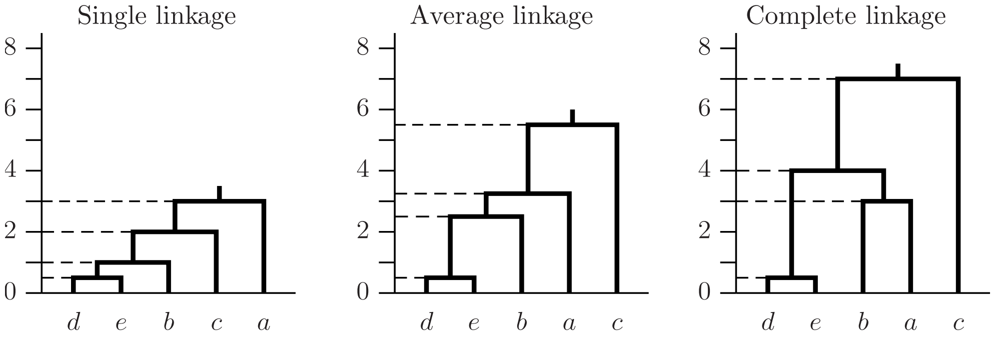
Comparison of trees

The upper part of Figure 4.3 presents a comparison of the first step of 3 methods for an artificial data set with $G = 5$ individuals. The middle part of the same figure displays the dendrograms obtained with 3 different methods. The y-axis is given by the distance between the two elements to be gathered. This comparison shows that these methods lead to different clusters. For example, for $K = 3$ groups, single and average linkage give $\{d, e, b\}$, $\{c\}$ and $\{a\}$ while complete linkage gives $\{d, e\}$, $\{b, a\}$, $\{c\}$.

One of the great difficulties in clustering is the validation of the method. Since the purpose is to discover unknown groups, there is generally no validation data (such as the validation set in *supervised classification*, see Chapter 6).

The quality of a clustering can be measured by comparing the distances between the elements in the dendrogram to the original dissimilarities. The distance between two elements in a tree is defined as the y-value of the highest node in the path from one

	original					single	average	complete
	distances					linkage	linkage	linkage
$d(g, g')$	a	b	c	d	e	$\{d, e\}$	$\{d, e\}$	$\{d, e\}$
a		3	7	3	4	3	3.5	4
b			4	4	1	1	2.5	4
c				2	6	2	4	6
d					0.5	-	-	-
e						-	-	-



a	b	c	d	e
	3.0	3.0	3.0	3.0
		2.0	1.0	1.0
			2.0	2.0
				0.5

a	b	c	d	e
	3.25	5.5	3.25	3.25
		5.5	2.5	2.5
			5.5	5.5
				0.5

a	b	c	d	e
	3.0	7.0	4.0	4.0
		7.0	4.0	4.0
			7.0	7.0
				0.5

Figure 4.3: Top left: original dissimilarity matrix. The two closest elements are d and e: $d(d, e) = 0.5$. Top right: comparison of the distances between $\{d, e\}$ and the other elements for three algorithms. Middle: clustering trees obtained with the same algorithms. Bottom: distances in the trees. Based on an example from Bouroche and Saporta (1998).

element to the other along the edges of the tree. These distances are given at the bottom of Figure 4.3.

These distance matrices can be compared to the original one with the *cophenetic coefficient* (see Sokal and Sneath (1963)). Denoting d the original dissimilarity and $\hat{d}$ the distance in the tree, this coefficient is defined as the correlation between d and $\hat{d}$. When applied to the example of Figure 4.3, the cophenetic coefficient is 0.54 for the single linkage, 0.67 for the average and 0.65 for the complete. In this case, the best cophenetic coefficient is obtained with the average linkage algorithm, but the difference with the complete linkage is not big. Due to the absence of a proper statistical framework, neither the fit of the average linkage clustering, nor the significance of its difference with the complete linkage can be assessed.

4.1.4 *K* means

An alternative way to build homogenous clusters is to characterize each cluster by a central point (its mean), and to assign each gene to the closest cluster. In this case, the distance from a gene to a cluster is defined as the distance from the gene to the mean of the cluster. This again is a heuristic way to minimize the within-cluster variability. The wide popularity of the K means algorithm comes from its simplicity.

Algorithm

Each gene g is represented by a point $\mathbf{x}_g$ with coordinates $(x_{g1}, \dots, x_{gT})$ in a T-dimensional space. The mean of cluster $\mathcal{C}_k$ is denoted $\mathbf{m}_k = (m_{k1}, \dots, m_{kT})$ where $m_{kt} = \sum_{g \in \mathcal{C}_k} x_{gt}/|\mathcal{C}_k|$. The K means algorithm updates these mean value after each affectation step.

Initialization. Choose K points (generally at random among $\mathbf{x} \dots \mathbf{x}_G$) that become the initial means $\mathbf{m}_1^0 \dots \mathbf{m}_K^0$ of the K groups.

Iteration h. Proceed steps 1 and 2.

1. Assign each element g to the closest group $\mathcal{C}_k^h$ with mean $\mathbf{m}_k^h$ such as $d(\mathbf{x}_g, \mathbf{m}_k^h) = \min_{k'} d(\mathbf{x}_g, \mathbf{m}_{k'}^h)$;
2. Update the mean of each group: $\mathbf{m}_k^{h+1} = \sum_{g \in \mathcal{C}_k} \mathbf{x}_g/|\mathcal{C}_k|$ and go back to step 1.

Stop. If $\mathbf{m}_k^{h+1} = \mathbf{m}_k^h$ for all k.

Step 1 (affectation) and 2 (updating) are respectively connected with the E and M steps of the EM algorithm described in Section 4.2.2.

Properties

Within-group minimization and convergence. The within-group inertia at step h

$$I^h = \sum_{k=1,K} \sum_{g \in \mathcal{C}_k^h} d^2 \left(\mathbf{x}_g - \mathbf{m}_k^h\right)^2 \tag{4.1}$$

decreases at each iteration and the K means algorithm converges in a finite number of iterations.
Indeed, I^h decreases during the affectation by definition of the affectation rule. Moreover, I^h also decreases during the updating step since, for each group $\mathcal{C}_k^{h+1}$ we have

$$\sum_{g \in \mathcal{C}_k^{h+1}} d^2 \left(\mathbf{x}_g - \mathbf{m}_k^{h+1}\right)^2 \leq \sum_{g \in \mathcal{C}_k^{h+1}} d^2 \left(\mathbf{x}_g - \mathbf{m}_k^h\right)^2$$

because $\mathbf{m}_k^{h+1}$ is precisely the mean of group $\mathcal{C}_k^{h+1}$.
Hence I^h, which is always positive, decreases at each step, so it converges. Furthermore, the number of repartitions of the G into K groups being finite, the number of iterations is finite.

In practice, it appears that the K means algorithm converge surprisingly quickly. Even for large data sets, the number of iterations is often smaller than 10. It should be noted that some groups may be emptied at some step, so the final number of groups can be smaller than K.

Local maxima. The major drawback of the K means algorithm is its high sensitivity to the choice of the starting points $\mathbf{m}^0 \dots \mathbf{m}_K^0$. As explained above, it is only a heuristic method and we have no guarantee that the final clustering is optimal in any sense. The fact that the inertia I^h always decreases only insures that it converges to a *local minimum*, but not to the global one. This problem is encountered in many optimization algorithm, for which no general optimality properties can be shown.
Simple simulations (not presented here) show that using the K means algorithm on the same data with different starting points leads to very different final clustering, some groups being split, and some other being merged.

Practical use. Because of the instability of the results it provides, the K means algorithm has to be used carefully or in specific cases. The basic prudential rule is to try a large number of starting points to check the variability of the clusters. This, of course, increases the computation time and reduces the advantage of the K means over hierarchical methods.

An interesting way to use the K means algorithm is to take advantage of its drawbacks. Instead of being chosen at random, the starting points $\mathbf{m}^0 \dots \mathbf{m}_K^0$ can be chosen on purpose, on the basis of some biological information. Typically, $\mathbf{m}^0 \dots \mathbf{m}_K^0$ can be defined as K genes known to be related to K specific functions or pathways. In this case, K means will gather unknown genes around known ones, and the interpretation of the clusters will be natural.

K means can also be used as a post-processing of a hierarchical clustering to check the stability of the clusters and to allow few genes to go from one cluster to another. Analyzing these genes can help in giving a biological interpretation to the clusters.

4.2 Model-based methods

We finally introduce mixture models that constitute the general framework for clustering problems in a model-based approach. These models assume that the profiles $\mathbf{X}_g$ are random, and that their distribution depends on the group to which gene g belongs. The randomness of $\mathbf{X}_g$ is coherent with the observed variability of microarray data. Moreover, mixture models provide additional informations with respect to distance-based methods:

- estimates of the parameters (mean, variance, etc.) characterizing each group,
- probability for each gene to belong to each group (rather than a deterministic affectation),
- statistical criterions to choose the number of groups.

Mixture models constitute a very large class of statistical models (see McLachlan and Peel (2000) for a general presentation), with numerous applications. We focus here on their use for clustering analysis and on the use of the EM algorithm to estimate the parameters of the mixture. EM is not the only available algorithm, but it is the most widely used and has some interesting similarity with the K means algorithm.

4.2.1 Mixture model

The set of the G genes is supposed to be a mixture of K groups (or *populations*) $\mathcal{C}_1$, $\dots$, $\mathcal{C}_K$. Each gene has marginal probability π_k ($\sum_k \pi_k = 1$) to belong to group $\mathcal{C}_k$. Conditionally to the group it belongs, the expression profile $\mathbf{X}_g$ of gene g has distribution $\phi(\cdot, \theta_k)$:

$$(\mathbf{X}_g \mid g \in \mathcal{C}_k) \sim \phi(\cdot; \theta_k) \qquad \Leftrightarrow \qquad \mathbf{X}_g \sim \sum_k \pi_k \phi(\cdot; \theta_k),$$

the parameter θ_k being characteristic of group $\mathcal{C}_k$. The log-likelihood of the profiles $\mathbf{X}_g$ ($g = 1 \dots G$) is

$$\log \mathcal{L}\left(\{\mathbf{X}_g\}; \{\pi_k, \theta_k\}\right) = \sum_g \log \left[\sum_k \pi_k \phi(\mathbf{X}_g; \theta_k) \right]. \tag{4.2}$$

Prior and posterior probabilities

In terms of clustering, the most interesting information provided by mixture models is the probability for gene g to belong to group $\mathcal{C}_k$ given its expression profile $\mathbf{x}_g$. The (unknown) marginal probability $\pi_k = \Pr\{g \in \mathcal{C}_k\}$ does not take into account the expression profile $\mathbf{X}_g$. It is called the *prior* probability and does not provide any specific information about gene g. π_k only informs use about the size of population $\mathcal{C}_k$. The conditional probability $\tau_{gk} = \Pr\{g \in \mathcal{C}_k \mid \mathbf{x}_g\}$ can be viewed as a version of π_k updated according to the observed profile $\mathbf{x}_g$. This probability, called *posterior* probability is given by Bayes' formula:

$$\tau_{gk} = \pi_k \phi(\mathbf{x}_g; \theta_k) \Bigg/ \sum_\ell \pi_\ell \phi(\mathbf{x}_g; \theta_\ell) . \tag{4.3}$$

Hence mixture models provide by the posterior probability for a given gene to belong to each of the K groups, instead of assigning it to a particular group. This justifies the term of *fuzzy classification*.

Gaussian mixture

Gaussian mixtures are naturally the most popular. In this case, parameter $\theta_k = (\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$ where $\boldsymbol{\mu}_k$ is the mean vector and $\mathbf{\Sigma}_k$ the variance matrix of population $\mathcal{C}_k$. $\boldsymbol{\mu}_k$ represents the mean expression profile of the group, while $\mathbf{\Sigma}$ describes the within-group variability of the profiles (see Fraley and Raftery (1998) for an introduction to the modeling of $\mathbf{\Sigma}$).

Figure 4.4 presents the calculation of posterior probabilities in a mixture of univariate Gaussian densities. In this case, the expression profile of each gene is reduced to one value x_g. In this example, given the x_g's, gene 1 most probably belongs to group 1, gene 2 may belong to groups 1 and 2 with equal probabilities and it is almost certain that gene 3 belongs to group 3.

τ_{gk} (%)	$g=1$	$g=2$	$g=3$
$k=1$	65.8	0.7	0.0
$k=2$	34.2	47.8	0.0
$k=3$	0.0	51.5	1.0

Figure 4.4: Univariate Gaussian mixture. Top left: mixture of 3 Gaussian densities $\phi(\cdot, \theta_k)$. Top right: posterior probabilities τ_{gk} as a function of x_g. Bottom: posterior probabilities for 3 particular values of x_g.

4.2.2 Parameter estimation

The most difficult part in mixture models lies in the estimation of parameters π_k and θ_k. We present here the maximum likelihood approach using the EM algorithm, which is a general algorithm for maximum likelihood estimation when the data are incomplete.

Complete likelihood

'Complete' data. Clustering problems can be presented as an incomplete data problem. For each gene g, we observe the expression profile $\mathbf{X}_g$ but we miss the group to which it belongs. This last information can be represented by a binary variable $Z_{kg} = \mathbb{I}\{g \in \mathcal{C}_k\}$ (where $\mathbb{I}\{A\}$ equals 1 if A is true, and 0 otherwise). In an ideal world, we should observe for each gene the profile $\mathbf{X}_g$ and the vector of binary variables $\mathbf{Z}_g = (Z_{g1} \dots Z_{gK})$ with multinomial distribution

$$\mathbf{Z}_g \sim \mathcal{M}(1; \pi, \dots, \pi_K), \qquad (\mathbf{X}_g \mid Z_{gk} = 1) \sim \phi(\cdot; \theta_k)$$

If g belongs to $\mathcal{C}_\ell$, the joint (log-) distribution of $(\mathbf{Z}_g, \mathbf{X}_g)$ is

$$\log[\pi_\ell \phi(\mathbf{X}_g, \theta_\ell)] = \sum_k Z_{gk} \log[\pi_k \phi(\mathbf{X}_g, \theta_k)]$$

since only $Z_{g\ell}$ is 1, all others Z_{gk} being 0. So the likelihood of the complete data set is

$$\log \mathcal{L}\left(\{\mathbf{X}_g, \mathbf{Z}_k\}; \{\pi_k, \theta_k\}\right) = \sum_g \sum_k Z_{gk} \log\left[\pi_k \phi(\mathbf{X}_g; \theta_k)\right].$$

It is called *complete likelihood*, while the likelihood given in (4.2) is called the *incomplete likelihood*.

Estimation with the EM algorithm

The direct maximization of the incomplete likelihood turns out to be very difficult in most cases. The idea of the EM algorithm is to work on the complete likelihood, which is more convenient to handle. Iteration h of the algorithm is composed of two steps.

E (expectation) step. Missing data $\mathbf{Z}_g$ is replaced by its conditional expectation given the profile $\mathbf{X}_g$. The conditional expectation of Z_{gk} is actually the posterior probability τ_{gk}^{h-1} given in (4.3), calculated with the estimates $\{\hat{\pi}_k^{h-1}, \hat{\theta}_k^{h-1}\}$ at step $(h-1)$.

M (maximization) step. The expectation of the conditional likelihood

$$\mathbb{E}\left[\log \mathcal{L}(\{\mathbf{x}_g \mid Z_g = k\})\right] = \sum_k \sum_g \tau_{gk}^h \log\left[\hat{\pi}_k^h \phi(\mathbf{x}_g; \hat{\theta}_k^h)\right]$$

is maximized (separately for each group $\mathcal{C}_k$).

Univariate Gaussian mixture. In this case, we have

$$\phi(x; \theta_k) = \exp\left[-(x-\mu_k)^2/(2\sigma_k^2)\right] \Big/ (\sigma_k\sqrt{2\pi})$$

with $\theta_k = (\mu_k, \sigma_k^2)$. At each M step, the updated versions of $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ are weighted version of the usual estimates, with weights $p_g^{h-1}(k)$:

$$\hat{\mu}_k^h = \frac{1}{\sum_g \tau_{gk}^{h-1}} \sum_g \tau_{gk}^{h-1} x_g, \qquad \hat{\sigma_k^2}^h = \frac{1}{\sum_g \tau_{gk}^{h-1}} \sum_g \tau_{gk}^{h-1} \left(x_g - \mu_k^h\right)^2.$$

Other versions. Several variations around the basic EM algorithm have been proposed.

CEM. The simplest one replaces the E step by an affectation step where Z_{kg} is set to one for group k having the maximal posterior probability τ_{gk}. The likelihood then obtained in the M step is called the classifying (C) likelihood; this algorithm is actually a generalized version of the K means algorithm.

SEM. A stochastic version of EM is obtained by drawing $\mathbf{Z}_g$ at random with probabilities $\tau g1, \ldots \tau_{gK}$. This version may help in avoiding local maxima since it allows the likelihood to decrease at some steps (see Celeux *et al.* (1995)).

Properties of the EM algorithm

It can be shown (Dempster *et al.* (1977), McLachlan and Peel (2000)) that the incomplete likelihood (4.2) increases at each iteration, so EM algorithm always converges. However, as for the K means algorithm, we have no guarantee that it converges towards the absolute maximum, for the same reasons as the K means algorithm. Both EM and K means algorithm are therefore highly dependent on the initial values of the parameters. Stochastic versions like SEM tend to limit this important drawback, and are preferred to the basic EM for complex likelihoods.

Mixture models for time course experiments

Luan and Li (2003) proposed to use a mixture model for time-course experiment on gene expression data. Each gene is characterized by its profile $\mathbf{x}_g = (x_{g1}, \dots x_{gT})$. The model is a mixture of Gaussian distributions that takes the time structure into account. Each cluster $\mathcal{C}_k$ of genes is characterized by a 'mean' profile $\mu_k(t)$:

$$X_{gt} \mid g \in \mathcal{C}_k \sim \mathcal{N}\left(\mu_k(t), \sigma_k^2\right).$$

The interesting point is that, in this approach, the clusters and their characteristic profiles μ_k are estimated simultaneously. Functions $\mu_k(t)$ are allowed to have a fairly general form (polynomial, B-splines). This is possible because all the data associated with gene putatively belonging to group $\mathcal{C}_k$ are used to fit μ_k. A more traditional way would be to estimate a specific function $\mu_g(t)$ for each gene and then to apply some clustering technique, but this would lead to very unstable results because of the lack of precision of the estimated function $\hat{\mu}_g(t)$.

4.2.3 Choice of the number of groups

As for all clustering methods, the choice of the number of groups is a difficult part of mixture modeling. However, for model-based methods, this problem can be stated in a model selection framework for which several standard statistical tools exists.

It is first important to remark that the criterion to be optimized (within inertia for Ward hierarchical method or for the K means, likelihood for mixture models) improves when the number of groups increases. Therefore, this criterion can not be used directly to compare clusterings, since clusterings with larger number of groups will systematically be preferred. We present here two solutions for this model selection problem in mixture models.

Penalized likelihood criterion. The number of groups can be chosen using some penalized contrast criterion (see Burnham and Anderson (1998)). Denoting D the dimension of parameter θ, a mixture model with K groups involve $P = K(D + 1) - 1$ independent parameters. The most popular criterions are AIC $= -2 \log \mathcal{L} + 2P$ and BIC $= -2 \log \mathcal{L} + P \log G$. Empirical studies (Fraley and Raftery (1998), Biernacki and Govaert (1999)) showed that BIC provides satisfying results.

Monte Carlo Markov Chain (MCMC) methods. An interesting way to choose the number of groups is to consider that this number is itself a parameter of the model, that has to be estimated together with others. The reversible jump algorithm (Green (1995)), defined in a Bayesian framework, is based on this idea. This MCMC algorithm estimates the posterior distribution of K (given the data) that allows to select the most likely number of groups.

Chapter 5

Differential analysis

A classical question motivating microarray experiments is the impact of treatments on genes expression. These treatments can be seen as covariates that could be discrete (irradiated sample vs. non irradiated sample), or continuous (dose of a drug). The purpose of differential analysis is the identification of differentially expressed genes or genes whose expression level differ from a condition to another. Differential analysis experiments include single slide experiments, where two conditions are hybridized on the same slide and identified by fluorescent dyes, and multiple slide experiments where biological samples are hybridized on different slides.

The statistical context of such analysis is the comparison of two populations according to a variable of interest : the level of expression of a gene, and the associated methodology is based on statistical hypothesis testing. This analysis always requires three steps : the definition of a statistic that scores the difference of expression between the two conditions, the definition of a decision rule based on this score to declare a gene differentially expressed or not, and the control of the probability to take the wrong decision.

The definition of an appropriate statistic is not new, and the traditionnal t-test remains relevant. Nevertheless, crucial choices of modelization have to be made, in order to adapt the t-test to the special case of microarrays. The question of the variability of the gene expression is central in this regard, and we will show that a compromise between statistical requirements and biological knowledge is essential for this analysis.

Classical decision rules can be applied in differential analysis studies, but the main problem will lie in the control of the tests that are performed. This question is also classical in the context of single hypothesis testing, where the problem is to control the probability to declare a single gene differentially expressed whereas it is not. Nevertheless, the characteristics of microarray data lies in the number of tests that are performed : as many as genes present on the slide, meaning thousands of tests. The question of differential expression is then restated as a problem of multiple testing.

5.1 Classical concepts and tools for hypothesis testing

Definition of a differential score

The question underlying differential analysis could be summarized as follows : does the expression of a given gene differ from condition A to condition B ? The first step is then to define a quantity that could score the difference of expression of a gene between the two conditions. Let us note $\bar{X}_A$ and $\bar{X}_B$ the mean expression of a given gene, calculated on R_A and R_B replicates, and S_A^2 and S_B^2 their variance.

$$\bar{X_A} = \frac{1}{R_A} \sum_{i=1}^{R_A} X_{Ai} \quad \text{and} \quad S_A^2 = \frac{1}{R_A - 1} \sum_{i=1}^{R_A} (X_{Ai} - \bar{X_A})^2.$$

A natural score is then :

$$T = \frac{\bar{X}_A - \bar{X}_B}{S\sqrt{\frac{1}{R_A} + \frac{1}{R_B}}} \quad \text{where} \quad S = \frac{(R_A - 1)S_A^2 + (R_B - 1)S_B^2}{R_A + R_B - 2}. \tag{5.1}$$

The choice of this criterion is partly arbitrary, but is easy to interpret : it quantitizes the difference of the average expression of a given gene between two conditions, normalized by the variability of the expression of this gene. Remark that this definition assumes that the global difference of expression between condition A and B has been set to zero due to normalization procedures.

This score could also be defined as the average difference of expression of this gene, normalized by the variability of this difference of expression. For this purpose, let us note $D_i = X_{Ai} - X_{Bi}$, the difference of expression of a given gene between conditions A and B, measured on replication i ($i = 1 \dots R$), $\bar{D}$ the average difference of expression, and S_D^2 the variability of this difference. The score is then :

$$T = \frac{\bar{D}}{S_D^2}\sqrt{R} \quad \text{where} \quad S_D^2 = \sum_{i=1}^{R} (D_i - \bar{D})^2. \tag{5.2}$$

Since we aim at declaring a gene differentially expressed or not, a high value of the score will indicate that the expression of the gene is ”really” different from condition A to B.

Statistical Hypothesis

Now that the differential score has been defined, the problem is to take a decision : is the considered gene differentially expressed or not? Two hypothesis are considered : the null hypothesis $\mathcal{H}_0$ of no difference between the two conditions, and an alternative hypothesis $\mathcal{H}_1$. The problem is then to define a decision rule that would accept or reject $\mathcal{H}_0$ given $\mathcal{H}_1$. Nevertheless, when the decision is taken, it can be the wrong decision, meaning that a gene can be declared differentially expressed, whereas it is not, or can be declared not differentially expressed whereas it is. Four situations are possible, and summarized in table 5.1.

		Decision	
		accept $\mathcal{H}_0$	reject $\mathcal{H}_0$
reality	$\mathcal{H}_0$ true	$1-\alpha$	α
	$\mathcal{H}_0$ false	β	$1-\beta$

Table 5.1: Statistical hypotheses and associated risks.

Each situation is possible with a certain probability. α is the Type I error, or the probability to declare a gene differentially expressed whereas it is not. This gene will be a false positive. On the other hand, β is the Type II error, or the probability to declare a gene not differentially expressed, whereas it is. This gene will be a false negative. The aim of any decision rule is then to facilitate the decision making but also to control those probability of errors.

Nevertheless the simultaneous control of the type I and type II error rates is not possible: if α is very low, then the probability to reject $\mathcal{H}_0$ is very low, meaning that the decision rule is very strict and could lead to the conservation of $\mathcal{H}_0$, even in wrong situations : the type II error rate increases as the type I error rate decreases.

Traditional statistical procedures aim at controlling the type I error rate and the parametric approach offers a theoretical framework for this purpose.

Why controlling the type I error rate?

As discussed above, classical statistical procedures aim at controling the type I error rate. Nevertheless, the error committed while taking the decision to reject $\mathcal{H}_0$ can be either the type I error rate, or the type II error rate. The need for a specific control of the type I error rate is simple to understand in our context, where microarray experiments results have to be further checked using a different technique, such as PCR. It is clear that if the type I error rate is large, a lot of genes will be declared differentially expressed and will have to be checked, even if they are not differentially expressed. This is why, it seems reasonnable to control this first type error in a practical point of view. An other reason is that the control of the type II error rate would require some knowlegde about the distribution of the statistics under $\mathcal{H}_1$, whereas it is not available (cf section 5.2.1).

5.2 Presentation of the *t*-test

5.2.1 The *t*-test in the parametric context

In the parametric context, the measures of the gene expression are considered to be the realizations of random variables, noted X_A and X_B. Since measures are repeated, R_A

and R_B times respectively, let us note $x_{A1},\dots,x_{AR_A}$ and $x_{B1},\dots,x_{BR_B}$ the realizations of the random variables $X_{A1},\dots,X_{AR_A}$ and $X_{B1},\dots,X_{BR_B}$. A classical assumption is that the distribution of X_{Ai} and X_{Bj} is gaussian, with parameters μ_A and σ_A^2, and μ_B and σ_B^2 respectively. The estimators of the parameters μ_A and σ_A^2 are $\bar{X}_A$ and S_A^2.

The question of differential analysis is then reformulated in terms of an hypothesis on the parameters: "there is no difference of mean expression for the gene g between conditions A and B" :

$$\mathcal{H}_0 = \{\mu_A = \mu_B\} \text{ vs } \mathcal{H}_1 = \{\mu_A \neq \mu_B\}.$$

The interest of this parametric context, is that the distribution of the differential score (5.1), or t-statistic is known under $\mathcal{H}_0$, and is a Student distribution with $R_A + R_B - 2$ degrees of freedom. Since the quantiles of this distribution are perfectly known, the decision to accept or reject $\mathcal{H}_0$ will be taken comparing the observed value of the statistics to its theoretical quantiles.

Nevertheless, before assessing the special problem of the decision rule, let us remark that the t-test requires some hypothesis :

1 - the X_{Ai} must be mutually idenpendent,

2 - the X_{Bi} must be mutually idenpendent,

3 - X_{Ai} and X_{Bj} must be independent.

Hypothesis 1 and 2 are generally reasonnable. Nevertheless in the case of cDNA microarray experiments, the two conditions A and B are hybridized on the same support, and distinguished by fluorescent dyes. This is why populations A and B are clearly not independent in a statistical point of view. In this case, hypothesis 3 is not valid, and the model should rather concern the difference of expression between the two conditions. Using same notations as above, if D_i represents the difference of expression of a given gene between conditions A and B, the new model is $D \sim \mathcal{N}(\mu_D, \sigma_D^2)$, and $\mathcal{H}_0$ is reformulated: "the mean difference of expression between conditions A and B is null" :

$$\mathcal{H}_0 = \{\mu_D = 0\} \text{ vs } \mathcal{H}_1 = \{\mu_D \neq 0\}.$$

In this case, the new statistic is the score defined in (5.2) and has a Student distribution with $R - 1$ degrees of freedom under $\mathcal{H}_0$. This test is called a t-test on *paired data* .

Decision rule

Since we dispose of the probability distribution of the t-statistic, we can compare the value of the realization of T, noted t_{obs}, to the theoretical quantiles of the Student distribution:

$$\text{If } |t_{obs}| > t_{1-\frac{\alpha}{2}} \text{ then reject } \mathcal{H}_0.$$

This decision rule is equivalent to the definition of a rejection zone $\mathcal{R}_\alpha$, defined as the set of values of T that are unrealistic under $\mathcal{H}_0$. The probability to declare a gene differentially expressed whereas it is not is α if this procedure is used since:

$$Pr\{|T| > t_{1-\frac{\alpha}{2}}\} = \alpha.$$

This decision rule ensures the control of the first type error to α [1].

***p*-values**

Softwares and automatic statistical procedures do not express the result of a test with the comparison of the observed value of the t-statistic to theoretical quantiles, but the result is rather expressed in terms of p-values. A p-value is defined by:

$$P_v(t_{obs}) = \Pr(|t_{obs}| \geq t_{1-\alpha/2}) = \Pr(|T| \geq |t_{obs}|).$$

It has two interpretations. First, it is the probability to obtain the observed score if $\mathcal{H}_0$ was true. In our context, it is the probability to observe a large value of the t-statistic if the gene considered was not differentially expressed. If this probability is small "enough", the null hypothesis will be rejected.

The threshold to which the p-value should be compared is α, and we have the fundamental property :

$$\{P_v(t_{obs}) \leq \alpha\} \Leftrightarrow \{t_{obs} \in \mathcal{R}_\alpha\},$$

meaning that if $\mathcal{H}_0$ is rejected when the p-value is lower than α, then the type I error is controlled and equals α. This leads to a second interpretation of the p-value : it is the level of the test at which $\mathcal{H}_0$ would just be rejected.

5.2.2 The non parametric context

In some situations, the assumption that data are realizations of gaussian random variables is not suitable. In the non-parametric context, no assumption is made on the distribution of the differential score, and theoretical quantiles and p-values are not caculable in a close form. The alternative proposed by non-parametric approaches is to compute the empirical distribution of the t-statistic, using permutation methods.

Let us recall that the data can be summarized in the following form :

$$X_{A1}, \ldots, X_{AR_A}, X_{B1}, \ldots, X_{BR_B}.$$

Under the null hypothesis of no difference between the two conditions, the control and treatment status is independent of gene expression. Resampling methods (bootstrap or permutation) randomly assign the label treatement A and B to the data. This permutation is done L times ($L \geq \mathcal{C}_{R_A+R_B}^{R_B}$), and ℓ^{th} permuation provides a pseudo value for the t-statistic. The empirical distribution of the statistic T is then obtained via the values $(t_1, \ldots, t_L)$.

The p-value associated with T is estimated via the proportion of pseudo values t_ℓ exceeding T :

$$\hat{p_v} = \frac{1}{L} \sum_\ell \mathbb{I}\{|t_\ell| > |T|\}.$$

[1] Note that the definition of the rejection zone depends on the alternative hypothesis $\mathcal{H}_1$. The results shown are valid for bilateral tests.

5.2.3 Power of the t-test

The power of a test is its ability to detect true differences: it is the probability to reject $\mathcal{H}_0$ when it is false. It is noted π, and equals $1-\beta$. Since the type I and type II error rate are linked (cf section 5.1), and the type I error rate is fixed, classical procedures do not control the power of the tests. Thus an easy way to compare different tests procedures will be to compare their respective power.

The next question is then : how can the power be optimized? The key factor in the optimization of the power of a test procedure is the number of replicates. Let us consider the moments of a t-statistic with distribution $\mathcal{T}_R$ under $\mathcal{H}_0$:

$$\mathbb{E}(T) = 0 \quad \text{and} \quad \mathbb{V}(T) = \frac{R}{R-2} \quad \text{if} \quad R \geq 2.$$

It is clear that the more replicates will be available, the more the variance of the t-statistic will decrease. In the first situation, where only few replicates are available, $\mathcal{H}_0$ is accepted, but when more replicated are available, $\mathcal{H}_0$ is rejected. The first situation leads to an acceptation of $\mathcal{H}_0$ whereas it was false, or to a high type II error rate β. When the number of replicates increases (second situation), the variance of the t-statistic is decreased, leading to less spread tails of distribution. As a consequence, the null hypothesis is rejected. The increase in the number of replicates leads to a decrease in the type II error rate, thus to an increase in the power of the test.

The next logical step would be to calculate the number of replicates required to reach a given power, or to calculate the power of a test given the number of replicates. Nevertheless, this exact calculus is not possible since it requires the knowlegde of the probability distribution of the statistic under $\mathcal{H}_1$. This calculus can be achieved with the expected normalized difference noted δ_R, depending on the number of replicates R. In the particular case of a t-test, we have

$$T \underset{\mathcal{H}_1}{\sim} \mathcal{T}_{2R-2}(\delta_R) \quad \text{with} \quad \delta_R = \frac{\mu_A - \mu_B}{\sigma}\sqrt{2R-2}.$$

Then the power of the test can be calculated with the formula:

$$\begin{aligned} \pi(\delta_R) &= \Pr(|T| > t_{1-\alpha/2}) \\ &= 1 - F(t_{1-\alpha/2}; 2R-2; \delta_R) + F(-t_{1-\alpha/2}; 2R-2; \delta_R) \end{aligned}$$

where $F(\cdot; 2R-2; \delta_R)$ is the distribution function of a non-central Student variable with parameter of non-centrality δ_R.

This calculus means that the question of the power has to be reformulated to:

- "How much power can I achieve, if I have R replicates in my experiment to observe a normalized difference of expression of δ_R" ?

- "How many replicates do I need to achieve a given power for the observation of a normalized difference of expression of δ_R?"

A figure like 5.1 can be used to answer these questions. This graph shows the power of a t-test for the comparison of two independant populations, with level $\alpha = 0.05$, according to the number of replicates. If the number of replicates is equal to 2, the probability to detect a difference of $|\mu_A - \mu_B| = 5\sigma$ is equal to 0.70. It can be seen on this graph that this probability is lower than 50% for differences lower than 4σ. Let's compare the power of the test according to the number of replicates for a given difference of 3σ. The power for $R = 2$ or $R = 4$ is approximatively of 40% and 90% respectively. Then 4 replicates are needed for each treatment to be nearly sure to detect a difference of 3σ.

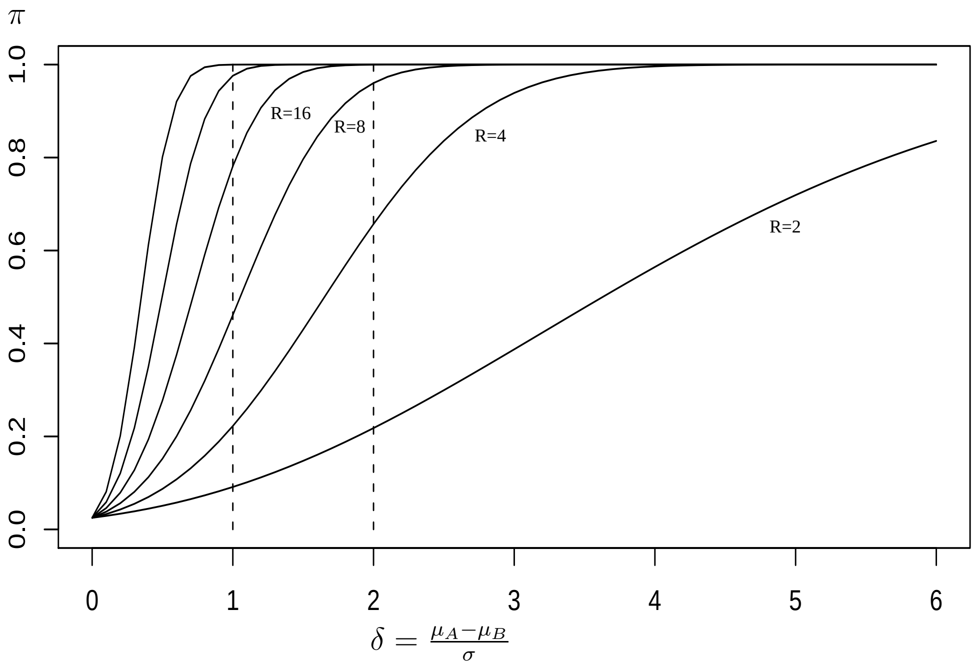


Figure 5.1: Power curves for different sample sizes (R = 2, 4, 8, 16, 32, 64), for a normalized difference δ. The level of the test is fixed and equals $\alpha = 0.05$.

Once more these considerations point out that the question of the power has to be asked before the experiment is performed, and thus is central in the design of any experiment that aims at comparing two populations, as explained in chapter 2.

5.3 Modeling the variance

As discussed before, the value of the t-statistic depends on two quantities : the mean difference of the expression values between two conditions, but also the variability of this difference of expression. Large values of this statistic can be obtained if the difference of expression is high, or if the variability of expression is low. The precision with which the variance is estimated becomes crucial, since small artefactual values of this variance can lead to an explosion of the statistic, thus to a decrease in power of the test. Modeling the variance is then of crucial interest for differential analysis.

5.3.1 A gene specific variance ?

Assuming that each gene presents a specific variability of expression between two conditions appears reasonnable in a biological point of view. Let us consider the case of a t-test on paired data. The estimator of the variance of expression of the gene g is :

$$S_g^2 = \frac{1}{R}\sum_{i=1}^{R}(D_{i,g} - \bar{D}_g)^2.$$

Notice that the number of replicates has a considerable influence on the estimation of the variance. In practical situations, only few replicates are available (R=3,4), leading to spurious small values of the variance due to errors of estimation, and thus to a lack of power. One simple solution to this problem is to add a constant s_0^2 to the variance estimator, and the t-statistic for gene g is then

$$T_g = \frac{\bar{D}}{\sqrt{S_g^2 + s_0^2}}\sqrt{R}.$$

This approach has been developped by Tusher *et al.* (2001), within a general framework for differential analysis of microarray data called SAM for Significant Analysis of Microarray data. The constant is chosen so that the median of the absolute deviation from the median of the test statistic is as uniform as possible over the standard error range. Other approaches have been developped in this context by several authors, see Efron *et al.* (2001) for a mixture model approach, Baldi and Long (2001) and Lönnstedt and Speed (2001) for a bayesian approach and Kerr *et al.* (2002), Rudemo *et al.* (2002) for intensity based approaches.

5.3.2 A common variance ?

An other possibility would be to suppose that all the genes present the same variability of expression between the two conditions. This solution, though simple in a biological point of view, presents some statistical advantages. In this case, the variance is estimated via :

$$S^2 = \frac{1}{m}\sum_{g=1}^{m} S_g^2.$$

This variance represents the average variability of expression for all the m genes and is fixed for all the t-statistics, leading to gaussian statistics, instead of t-statistics (this method does not allow missing data). This solution has the main advatage to calculate an estimator over a large number of data, leading to a robust estimation of the variance, and to a gain in the power of the test. Nevertheless, this modelization is very rigid in the biological point of view.

5.3.3 An intermediate solution

The two situations described above are biologically relevant and statistically of low power for the first one, and biologically simple and statistically robust for the second one. Delmar *et al.* (2003) thus propose and intermediate modeling, considering that groups of variability can be identified from the data. This model suggests that a gene can belong to a population defined by its variability of expression. For this reason, a variance mixture model is considered, where each gene has the variance of the group of genes it is found to belong to. If σ_k^2 is the "true" variability of expression of the group k, then the estimator S_g^2 of the variance for the gene g is supposed to follow a mixture of gamma distributions, such as:

$$RS_g^2 \sim \sum_{k=1}^{K} \pi_k \gamma_{\sigma_k^2, R}.$$

This parametrization allows the definition of exact testing procedures, and can reproduce complex patterns in the variance structure.

5.4 Multiple testing problems

The question of comparing to populations according to a variable of interest can be handled via classical statistical tools, such as the t-test, modulo some adaptations to the special case of microarray data. The procedures described above show how a difference of expression can be scored, and how the decision to declare a gene differentially expressed can be taken for one gene, controlling the type I error. Nevertheless, the reality of microarray data is much more complicated, since thousands of genes have to be studied simultaneously. Even if the same statistical score can be used for each gene, the question of differential analysis has to be restated in terms of error control.

False positives, false negatives, expected number of errors

Table 5.1 can be viewed not only in terms of probability of errors, but also in terms of number of errors, as shown in table 5.2.

A small example is used to illustrate the problem of multiple testing. Let's consider that all the genes are differentially expressed ($m_0 = m$), and that all the tests are performed with level α. Then the number of false positives is a random variable, with

		Decision	
		declared NDE	declared DE
reality	m_0 DE genes	TN	FP
	m_1 NDE genes	FN	TP
	$m = m_0 + m_1$ genes	$R = TN + FN$	$S = FP + TP$

Table 5.2: DE: Differentially expressed, NDE: non differentially expressed, TN: number of True Negatives, FN: number of False Negatives, R: number of Negatives, FP: number of False positives, TP: number of True Positives, S: number of Positives

Bernoulli probability distribution such as :

$$FP \sim \mathcal{B}(m, \alpha).$$

This simple modeling leads to the conclusion that the expected number of false positives, when m hypothesis are tested simultaneously is $\mathbb{E}(FP) = m\alpha$. Regarding the high number of tests performed in microarray experiments (for instance $m = 10000$), 500 genes will be declared differentially expressed whereas they are not, if the level of the tests is $\alpha = 0.05$. The purpose of multiple testing is then to control the global risk of the procedure.

Definition of global risks : the FWER and the FDR

Let us note $\mathcal{H}_0^j$ the null hypothesis concerning the individual gene j, and p_j the associated p-value. The multiple testing procedure requires the definition of a *complete null hypothesis*, noted $\mathcal{H}_0^c$: ”there is no difference of expression between condition A and B, for none of the genes”:

$$\mathcal{H}_0^c = \bigcap_{j=1}^{m} \mathcal{H}_0^j.$$

As classical procedures aim at controlling the individual risk associated with $\mathcal{H}_0^j$, multiple testing aims at controlling the global risk associated with the complete null hypothesis $\mathcal{H}_0^c$.

As many hypotheses are drawn simultaneously, the question of error could be reformulated : ”what is the kind of error that could be committed while testing the complete null hypothesis ?”. The natural first type of error is that among the m hypothesis tested, at least one decision taken is wrong. This defines he Family Wise Error Rate : the probability to have at least one False Positive.

5.4.1 Controlling the Family Wise Error Rate

The procedures of Sidak and Bonferroni are the most widely used to control the FWER, because of a simplicity of interpretation and implementation. Let us consider the following simplified situation. If the genes are assumed to be all differentially expressed ($m = m_0$), then the probability of having at least one false positive is equal to one minus the probability of making no error among the m hypothesis tested. In a first step, we can simplify the situation assuming the all the m tests performed are independent. Since the probability of no error for hypothesis $\mathcal{H}_0^j$ is $1 - \alpha$, the Family Wise Error Rate can be calculated directly :

$$FWER = \Pr\{FP > 0\} = 1 - \prod_{j=1}^{m}(1 - \alpha) = 1 - (1 - \alpha)^m$$

The principle of multiple testing is to recalculate the individual risk for each individual hypothesis $\mathcal{H}_0^j$, in order to control the global error. Then, performing each test at level $1-(1-\alpha)^{1/m}$ ensures the global control of the FWER at level α. This procedure is called the Sidak correction.

Even if the Sidak procedure offers the exact calculus of the Family Wise Error Rate, it requires oversimplifying assumptions. First, in real situations, the number of differentially expressed genes is rarely or never equal to the total number of genes. This number m_0 remains unknown, and could be estimated (see below). This leads to the definition of levels of controls : an exact control of the FWER requires the knowledge of the exact number of differentially expressed genes (m_0 known), a weak control if it is calculated under the assumption that all the genes are differentially expressed ($m_0 = m$), and strong if it is calculated over all the possible choices of sets of genes really non differentially expressed. In the case of microarrays, it is crucial to have an exact or a strong control of the FWER, since the assumption $m_0 = m$ is absolutely not verified.

An other criticism that could me made to the Sidak procedure is that is assumes that the tests are independent, whereas the gene expressions are obviously not. For this reason, an other procedure can be applied, the Bonferroni procedure. It is based on the inequality

$$\Pr\left\{\bigcup_j A_j\right\} \leq \sum_j \Pr(A_j).$$

This procedure does not provide an exact form for the FWER, but a majoration. This is the most famous of these procedures : performing each test at individual level α/m ensures the control of the FWER at level at most α.

5.4.2 Practical implementation of control procedures

As mentioned in 5.2.1, the practical use of statistical tests involves the use of p-values to declare a gene differentially expressed or not. Since the p-value p_j can be considered as the level of the test at which $\mathcal{H}_0^j$ would just be rejected, the adjusted p-value $\tilde{p}_j$ is defined as the global level of the procedure at which $\mathcal{H}_0^j$ would just be rejected. If interest is in controlling the FWER, the adjusted p-value for hypothesis $\mathcal{H}_0^j$ is

$$\tilde{p}_j = \inf\{\alpha \ : \mathcal{H}_0^j \text{ is rejected at } FWER = \alpha\}$$

and $\mathcal{H}_0^j$ is rejected at FWER α if $\tilde{p}_j \leq \alpha$.

Let us define two procedures of p-value adjustment for the Sidak and the Bonferroni methods.

Procedure	adjusted p-value	control
Sidak	$\tilde{p}_j = 1 - (1 - p_j)^m$	FWER=α
Bonferroni	$\tilde{p}_j = \min(1, m \times p_j)$	FWER $\leq \alpha$

Dudoit *et al.* (2003) provide a complete review of adjustment procedures for p-values.

5.4.3 Adaptative procedures for the control of the FWER

The previous procedures are called *single step* procedures, since they provide the same adjustment for all hypothesis, regardless of the ordering of the unadjusted p-values, meaning without consideration for the degree of significance of individual hypothesis. As a result, they lead to very conservative decisions, and thus to a decrease in the power of the procedure. Improvement of power, while preserving the control of the FWER may be achieved by considering step-down procedures which order p-values and make successively smaller adjustments.

Let denote $p_{(1)} < \ldots < p_{(m)}$ the sequence of ordered p-values, and apply the following correction :

Procedure	adjusted p-value
Adaptative Sidak	$\tilde{p}_j = \underset{j \leq g}{max} \left\{ min \left[1 - (1 - p_{(j)})m - j + 1, 1 \right] \right\}$
Adaptative Bonferroni	$\tilde{p}_j = \underset{j \leq g}{max} \left\{ min \left[(m - j + 1) p_{(j)}, 1 \right] \right\}$

The increase of power of these procedure lies in the fact that a particular hypothesis can be rejected provided all hypothesis with smaller unadjested p-values were rejected beforehand.

5.4.4 Dependency

Despite the increase in power provided by step down procedures, no method proposed above addresses the problem of dependency that could lie between the test statistics. In the special case of microarrays, since the expression of a gene is dependent on complex regulatory networks, the hypothesis of independence between the t-statistics can reasonnably be rejected. Westfall and Young (1993) proposed two alternative procedures to consider the dependency between statistics, based on permutations.

Procedure	adjusted p-value
Westfall and Young (1993) minP	$\tilde{p}_j = \frac{1}{S} \sum_S \mathbb{I} \left\{ \lvert p^s_{(j)} \rvert \leq \lvert p_j \rvert \right\}$
Westfall and Young (1993) maxT	$\tilde{p}_j = \frac{1}{S} \sum_S \mathbb{I} \left\{ \lvert T^s_{(j)} \rvert \geq \lvert T_j \rvert \right\}$

As discussed in 5.2.2, this procedures are very dependent on the number of permutations performed. When p-values are estimated via the minP procedure, more computations are needed, since since method requires the estimation of the unadjusted p-values before considering the distribution of their successive minima.

5.5 An other approach, the False Discovery Rate

An alternative approach to the control of the FWER has been proposed by Benjamini and Hochberg (1995), based on the principle that any reasercher is ready to tolerate some type I errors, provided their number is small regarding the number of rejected hypothesis. In comparison to the control of the FWER that often leads to conservative procedures, the control of the expected proportion of type I errors among the rejected hypothesis leads to less conservative results, thus to an increase in the power of the tests.

Let us define the False Discovery Rate: it is the expected proportion of false positives among the total number of positives

$$\begin{aligned} \text{FDR} &= \mathbb{E}\left[\frac{FP}{S}\right] \text{ if } S > 0 \\ &= 0 \text{ oherwise} \end{aligned}$$

The introduction of the False Discovery Rate is new compared to the traditionnal procedures to control the number of false positives. Two steps are thus important for the use of the FDR: its control and its estimation.

5.5.1 Controlling the False Discovery Rate

Before controlling the FDR, it is important to specify that the number of false positives and the total number of positive genes depends on a threshold fixed by the utilisator. It is noted t, and then we define:

$$FP(t) = \#\{null\ \ p_i \leq t, i = 1 \dots m\}$$

$$S(t) = \#\{p_i \leq t, i = 1 \dots m\}$$

and thus the False Discovery rate is also a function of this threshold. In our context, the threshold will be given by ordered p-values.

The control of the FDR can be performed via the separate calculus of $\mathbb{E}[FP(t)]$ and $\mathbb{E}[S(t)]$ since the number of hypothesis m is large. $\mathbb{E}[S(t)]$ can be replaced by $S(t)$, and if the procedure stops at threshold $p_{(g)}$, the observed number of positives is g

$$S(p_{(g)}) = g.$$

The problem is rather in the calculus of the expected number of false positives.

The central hypothesis is that the p-values are uniformly distributed under $\mathcal{H}_0$. Then the probability for a p-value to be lower than the threshold t equals t under $\mathcal{H}_0$:

$$\Pr\left\{p_{(g)} \leq t\right\} \underset{\mathcal{H}_0}{=} t.$$

The expectation of the number of false positives is then $\mathbb{E}[FP(t)] = m_0 t$ with m_0 being the number of true non differentially expressed genes, which is unknown. Then if the procedure threshold is $p_{(g)}$ the FDR equals:

$$FDR(p_{(g)}) = \frac{m_0 p_{(g)}}{g}.$$

Since the number of true positives m_0 is unknown, a classical strategy is to replace it by m that is known. If the aim of the procedure is to control the FDR at level α, then the stopping rule will be:

$$p_{(g)} < \frac{g\alpha}{m}.$$

5.5.2 Estimating the False Discovery Rate and the definition of q-values

The quality of this procedure can be improved if the number of true positive m_0 is not bounded, but estimated. This estimation is performed with respect to a tuning parameter λ:

$$\hat{m}_0(\lambda) = \frac{\{\#p_i > \lambda, i = 1 \dots m\}}{(1-\lambda)}.$$

Further details concerning the estimation procedure can be found in Storey and Tibshirani (2003) and Storey *et al.* (2004), who explain the choice of the tuning parameter λ. Then the False Discovery Rate at t is estimated such as:

$$\widehat{FDR}(p_{(g)}) = \frac{\hat{m}_0 \times p_{(g)}}{g}.$$

Storey and Tibshirani (2003) propose to define an equivalent of the p-value but dedicated to the case of the FDR. A q-value is defined such as:

$$\hat{q}(p_{(g)}) = \min_{t \geq p_{(g)}} \widehat{FDR}(t).$$

Contrary to p-values, q-values provide a measure of each significance taking into account the fact that thousands of genes are tested. If genes with q-values lower than 5% are called significantly differentially expressed, then there is a False Discovery Rate of 5% among the significant genes.

Chapter 6

Supervised classification

6.1 The aim of supervised classification

An other application of microarray experiment is the diagnosis. In the case of cancerous tumor one would like to predict the disease status sane (+) or tumorous (-) of a tissue sample t according to its gene expression profiles $x_t = (x_{1t}, ..., x_{Gt})$. To classify an undiagnosed tissue, a classifier - also called a classification rule - is constructed on the basis of a database of gene expression profiles from diagnosed tissues. The construction of the classifier is the goal of supervised classification or learning methods. The construction step is called the training phase, and the database employed to elaborate the classifier is called training data. In Section 6.2, we present learning methods that have been successfully employed by the microarray community.

What properties do we expect the classifier to have ? On one hand, we expect the constructed classifier to have a good generalization capacity, meaning that we do not want it to correctly classify the samples of the training data but to correctly predict the status of a new undiagnosed sample, or to err only when the expression profile of the tissue is ambiguous. The error rate, *i.e.* the probability for the classifier to err on a case, is then a natural indicator of the generalization of a classifier. Yet the real error rate of the constructed data is unknown and has to be estimated. Section 6.3 deals with the different methods to estimate this error rate. On the other hand, one would like the classifier to be easily interpretable, meaning that the way the classifier operates has to be clear and biologically relevant, and robust, i.e. not too dependant on the given sample used to construct it. These two goals can be achieved with the construction of a classifier based on only a few genes. This is one of the reasons why the variable (gene) selection is an important feature in supervised classification applied to microarray data. Section 6.4 is dedicated to this feature.

6.2 Supervised classification methods

Two important notions for the understanding of classifier construction and classifier performance are the Bayes classifier and the bayesian error rate. A well-known statistical result states that the best classification (the one that minimizes the error rate) is the Bayes classifier:

$$f_{Bayes}(s) = \begin{cases} + & \text{if } \mathbf{P}\{s \in \mathcal{C}_+ | X = x\} > \mathbf{P}\{s \in \mathcal{C}_- | X = x\} \\ \text{-} & \text{otherwise} \end{cases}$$

The decision is then based on the maximal posterior probability of belonging to class $\mathcal{C}_+$ or $\mathcal{C}_-$. The inequality between posteriori probabilities is equivalent to the following inequality :

$$\pi_+ \phi_+(X) > \pi_- \phi_-(X) \tag{6.1}$$

where ϕ_+, ϕ_-, are the conditional distributions of X in classes $\mathcal{C}_+$ and $\mathcal{C}_-$, and π_+, π_- are the prior probabilities to belong to class $\mathcal{C}_+$ and $\mathcal{C}_-$, respectively. In practice the posterior probabilities cannot be computed, since ϕ_+, ϕ_-, π_+, π_- are not known. Nevertheless having a good idea of the way the best classifier works will help for the construction of efficient classifiers. Besides, we now know that a good classification method should guarantee an error rate comparable to the bayesian error rate.

In this section we present three learning methods among the many powerful ones that exist. We start with a parametrical method, the Fisher Discriminant Analysis, to end with Support Vector Machines, that encapsulates the main concepts of recently developed non parametric learning methods. A very complete description of learning methods can be found in Hastie *et al.* (2001). Although a complete statistical analysis of each of the three methods is not possible here, we discuss some of the following interesting properties for each method:

- Interpretation facility: by construction, some classification methods provide insight in the biology of the data, or can be designed to explicitly include some prior knowledge about the data. One is then able to build a comprehensible classification rule that will be easier to interpret.

- Complexity of use: some classification methods require the choice of "tuning parameters", for instance the number of neighbors to consider in the kNN classifier. Although crucial for the classifier construction, one often lacks an efficient way to adjust these parameters.

- Universal consistency: since the bayesian error rate is the best we can hope for, it is interesting to know whether the considered methods produce classifiers whose error rate gets closer to the bayesian error rate as the sample size increases, whatever the distribution of the data is. This last statistical property is useful since the real distribution of the data is unknown.

6.2.1 Fisher Discriminant Analysis

In the previous section we saw that the Bayes classifier is based on unknown conditional distributions ϕ_+, ϕ_- and prior probabilities π_+, π_-. In Fisher Discriminant Analysis, we make the strong assumption that the conditional distributions are gaussian:

$$X_t \sim \mathcal{N}(\mu_+, \Sigma_+) \;\text{ if } t \in \mathcal{C}_+ \qquad\qquad X_t \sim \mathcal{N}(\mu_-, \Sigma_-) \;\text{ if } t \in \mathcal{C}_- \tag{6.2}$$

with unknown parameters μ_+, Σ_+, μ_- and Σ_-. In this parametric context, the training phase consists in estimating the unknown parameters of the gaussian distributions, along with the prior probabilities. Once the parameters estimated, a given sample t can be classified by *plugging-in* the estimates in inequality (6.2) as follows :

$$\widehat{f}_{FDA}(t) = \begin{cases} + & \text{if } \hat{\pi}_+\hat{\phi}_+(x_t) > \hat{\pi}_-\hat{\phi}_-(x_t) \\ \text{-} & \text{otherwise} \end{cases}$$

Thus in FDA the Bayes classifier is mimicked by estimating posterior probabilities with the help of the gaussian assumption.

Which are the critical points for which $\hat{\pi}_+\hat{\phi}_+(x) = \hat{\pi}_-\hat{\phi}_-(x)$, that define a frontier between classes (+) and (-) ? Solving the equality gives the expression :

$$2x\left(\widehat{\Sigma}_+^{-1}\widehat{\mu}'_+ - \widehat{\Sigma}_-^{-1}\widehat{\mu}'_-\right) + x\left(\widehat{\Sigma}_+^{-1} - \widehat{\Sigma}_-^{-1}\right)x' = Cst \tag{6.3}$$

where Cst is constant w.r.t. x. Thus the feature space, where all possible samples are represented and labeled according to the classifier, will be split by a quadratic function if the covariance matrices are different in each class (Fig.6.1). In the case where covariance matrices are supposed to be equal, the quadratic term vanishes in expression (6.3), and the frontier becomes linear.

The main interest of FDA is that parametric models make assumptions on the data explicit, and therefore facilitate the interpretation of the classification rule. For instance, in LDA, the only differences between classes are the mean expression of genes. This means that for a given problem, if the discrimination between classes lies in changes in a given gene regulation, *i.e.* in changes in covariance matrice between the two classes, then LDA will fail to take it into account and show poor performance, so QDA with fewer genes will be preferred. Besides, in the well-known gaussian framework stated in FDA, many results are available that can be directly applied to perform statistical testing procedures. Curiously, although the gaussian framework provides an explicit stepwise procedure for variable (gene) selection, described in Rao (1965), we found no application of this procedure in microarray data analyses.

A major drawback of parametric methods lies in over-parametrization: in the particular case of the FDA, if the number of samples is small compared with the number of genes, th covariance matrices will be singular. This means that a possibly important number of genes will be discarded to make the covariance matrices inversible, or that non interpretable generalized inverse matrices will be computed. Moreover, it is clear that no guarantee of universal convergence can be given for the FDA since one assumes normality for the conditional distributions. Yet, practical applications (Dudoit *et al.* (2002), Brown *et al.* (2000)) have shown that in many cases FDA or derivatives perform as well as other performant methods such as SVM or Neural Networks.

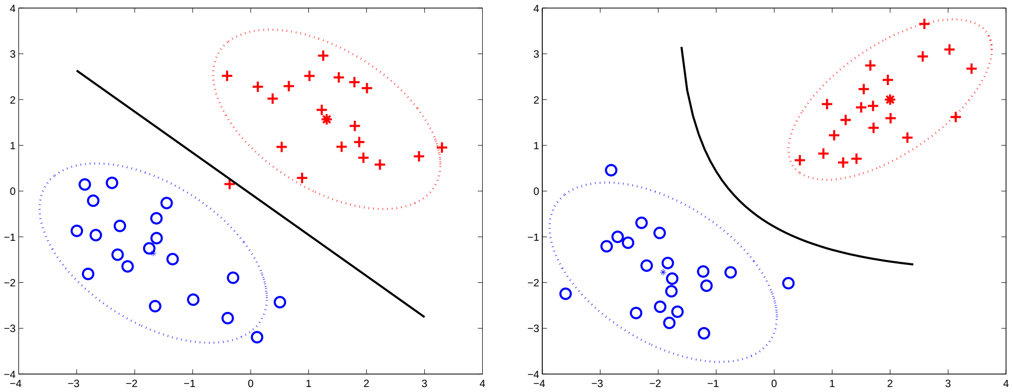


Figure 6.1: **Left:** Linear Discriminant Analysis. The frontier is a linear function of the gene expressions. The stars represent the empirical means of each class samples **Right:** Quadratic Discriminant Analysis. The frontier is a quadratic function of the gene expressions.

6.2.2 k-Nearest Neighbors

In its simplest form, the kNN can be understood as a pure geometrical conception of classification : an undiagnosed tissue sample t is diagnosed according to the most recurrent diagnostic among its k nearest neighbors. To class t, the algorithm as follows :

1. Find the k tissues in the training data whose gene expression profiles are the closest to the one of t

2. Count the number of "sane" and "tumorous" tissues among the k tissue samples selected

3. classify t as "sane" if most of the k neighbors are sane, "tumorous" otherwise.

The kNN decision is then based upon a majority vote. Compared with FDA, kNN method can be considered as a local estimation of the posterior probabilities: the probability to belong to class "+" is estimated by the proportion of "+"-samples in the neighborhood. The main advantage of kNN is to provide these estimations under no assumption on the conditional distributions. Such methods are called non-parametric

The only two parameters that have to be given are the number k of neighbors to be considered, and the distance d used to measure proximity between two tissues. An optimal k can be determined by comparing the error rate of each kNN classifier, $k = 1, 2, ...$, on test data (see 6.3). But this solution requires extra data, independent from the training ones, to choose k. Indeed, to estimate the error rate on the training samples would lead to the choice of low k. For instance, the 1NN classifier does not make any misclassification on the training data, but can err more than any other kNN classifier on new data, because it "consults" only one neighbor to classify a new tissue sample. Other rules to select k have been proposed (see the discussion on the topic in Devroye and Lugosi (1995)), but this remains a major difficulty of the method.

The choice of a distance reveals interesting possibilities to integrate some prior knowledge to the classifier. Consider for instance the euclidian distance that can be defined between two samples s_i, s_j as follows:

$$d^2(s_i, s_j) = \| X_i - X_j \|^2 = \sum_{g=1}^{G} w_g (X_i^g - X_j^g)^2 \text{ where } w_g = 1, \quad g = 1, ..., G$$

To the condition that gene expressions have been scaled to 1, the choice $w_g = 1$ means that genes equally contributes to the distance between tissues. A first solution to integrate prior information is to choose unequal weights. The use of unequal weights emphasizes the role of selected genes on the basis of biological considerations: irrelevant genes that are known to be unrelated to the classification problem can be weighted to 0. An alternative solution is to compute the distance between tissues according to their gene expression profiles along with information collected in previous experience for instance.

More refined applications of the kNN include weighted votes, where the influence of each voting neighbor to classify s_0 is proportional to its distance to s_0, and thresholding, where s_0 is classified only if the votes exceed a predetermined threshold, and is considered uncertain otherwise (see Golub *et al.* (1999a)).

At last, it is worth mentioning that despite its intuitive construction, the kNN method also has interesting statistical properties, such as universal convergence to the best classifier, that can be found in Devroye and Lugosi (1995).

6.2.3 Support Vector Machines

In Section 6.2.1, we noticed that classifying data with FDA (with equal covariance matrices hypothesis) amounts to split the sample in the g-dimensional expression space or "input space" by a hyperplane - linear function. This hyperplane is deduced from the estimated parameters of the conditional distributions. Based on this observation, the principle of Support Vector Machines (SVM) is also to find a separating hyperplane but that is not deduced from any distribution assumption. How to select the separating hyperplane then ? The SVM algorithm looks for the hyperplane that perfectly separate each class samples with a maximum margin, where the margin is defined as the distance from the hyperplane to the nearest point (see Fig.6.2). SVM are then a particular member of large margin classifier methods such as Boosting (Freund and Schapire (1996)), that have been proven to be efficient and robust in many applications. To choose the optimal classifier in the margin sense results in a better generalization of the trained classifier. This was first proved for SVM by the pioneering work of Vapnik (1998) and then by many successful practical applications of SVM to classification problems. To find the hyperplane with maximum margin, one needs to solve the following convex quadrating programming problem:

$$\max_{\alpha_t} \ \frac{1}{2}||w||^2 - \sum_{t=1}^{n} \alpha_t(y_t(\langle x_t, w\rangle + b) - 1) \ \text{ with } \alpha_t \geq 0 \tag{6.4}$$

where w and b are the normal vector and the constant that define the hyperplane, respectively, α_i are positive Lagrangian multipliers, and y_t is the label variable that takes value $+1$ if sample t belongs to class $(+)$, -1 otherwise. It is well known in the regularization theory that the solution is: $w = \sum_{t=1}^{n} \alpha_t y_t x_t$. Thus a new sample t_0 will be classified as follows :

$$\widehat{f}_{SVM}(t_0) = \begin{cases} + & \text{if } \langle w, t_0\rangle + b = \sum_{t=1}^{n} \alpha_t y_t \langle x_t, x_{t_0}\rangle + b > 0 \\ \text{-} & \text{otherwise} \end{cases} \tag{6.5}$$

In practice, only some of the α_t coefficient have a non-null value, meaning that the resulting classifier depends on a few samples that are called the support vector. In Fig. 6.2, the support vector are the closest ones to the frontier, i.e. those which define the margin (dotted lines). Thus the support vectors can be seen as the borderline cases of the training dataset.

The strength of SVM lies in the computational *kernel trick* (see Schölkopf and Smola (2002)). SVM look for a hyperplan that splits the dataset according to the sample labels. Such a linear separation does not always exist in the input space $\{X^1, ..., X^g\}$, and one would like to extend the search and find non-linear separations between classes, or equivalently to find an optimal hyperplan in a bigger *feature* space that includes $\{X^1, ..., X^g\}$ and some transformations of $X^1, ..., X^g$. This can be done by:

1. perform a data mapping φ: $x \mapsto \varphi(x)$
2. apply the SVM algorithm to the transformed data, *i.e.* in the feature space

According to the dimension of the feature representation, the mapping computation and the convex optimization resolution times become discouraging. For some particular transformations, this computational burden can be avoided by replacing the dot products $\langle ., .\rangle$ in (6.4) and (6.5) by a kernel function $k(.,.)$. In that case, the kernel function allows the display of the classifier f_{SVM} found in the feature space without the explicit computation of data transformation. For instance, the use of polynomial kernel with order 2, $k(x_i, x_j) = (1 + \langle x_i, x_j\rangle)^2$, allows the determination of quadratic frontiers (see Fig.6.2).

Thus the SVM algorithm can be generalized to non-linear separation by finding the maximal margin hyperplane in a very high (possibly infinite) dimension space without the computational difficulty of representing the feature points. While interesting from a computational point of view, the drawback of the kernel trick is that no conclusion about the predictive structure of the data can be obtained from the resulting classifier: roughly speaking, no representation of the separating hyperplane is available, so no interpretation can be made of it.

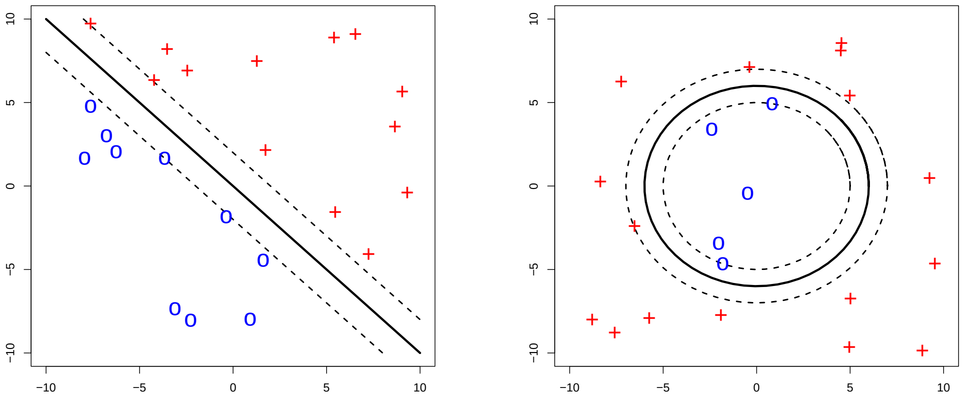


Figure 6.2: **a.** SVM classification: the two dotted lines delimit the margin. **b.** SVM classification with a quadratic kernel: in this situation, no linear classifier can correctly separate groups "0" and "+".

6.3 Error rate estimation

Having selected a classifier $\widehat{f}$ with a given classification method, one would like to known its performance in terms of error (misclassification) rate. Its intuitive estimator is the empirical error rate EER of $\hat{f}$ on the training set, defined as follows:

$$EER = \frac{1}{n} \sum_{i=0}^{n} I_{\{s_i \text{ is misclassified with } \widehat{f}\}} \tag{6.6}$$

where $I_{\{Cond\}}$ equals 1 if $Cond$ is verified, 0 otherwise. This results to an optimistically biased estimation, because $\hat{f}$ was selected in some sense to fit the training data. Moreover for sufficiently complex classifiers, for instance SVM classifier with high order polynomial kernels, the EER is known to be null whatever the minimum error rate achievable (i.e the bayesian rule error rate) is. Alternative estimation methods are:

- the estimation of the error rate of $\hat{f}$ on a test sample
- the r-cross validation (rCV) and in particular leave-one-out cross validation (LOOCV)

A test sample is a dataset that contains observations independent from the training dataset, but obtained under the same conditions. Estimating the error rate on a test sample gives a unbiased estimation of the true error rate, but means that a part of the data at hand will not be used to construct the classifier. This is an important drawback in microarray experiments where the sample size is usually small.

The cross validation method proceeds as follows: one withholds a tissue samples from the training dataset, builds a predictor based on the remaining samples, and predicts the class of the withheld sample. The process is repeated for each different sample, and finally the cumulative error rate is computed. The obtained estimation of the error rate has a small bias, but the difficulty here lies in the computation time: one needs to construct as many classifier as the number of samples. The *LOOCV* can be extended by withholding r samples at each iteration. The properties of the r-cross validation estimator has been studied by McLachlan (92), who shows that the estimator bias increases with r, whereas its variance diminishes. Yet, the disadvantage of r cross-validation is the computation time: as many classifiers as the number of combination of r samples among n, the total number of samples, have to be constructed.

6.4 Variable selection

The variable selection or feature selection aims to select a reduced subset of informative variables (genes) without loss in term of prediction. We suppose that no prior knowledge is available for the selection, which has to be done on the basis of the data. In microarray experiments, feature selection is an important step that fulfills different functions :

- From a statistical point of view, eliminating thousands of irrelevant variables will significantly reduce the complexity of the selected classifier, and will make results more robust.
- From a biological point of view, to select pertinent features that are strongly involved in the disease status will help to understand the mechanisms at work.
- From a practical point of view, the few genes are used to establish the diagnostic, the better.

While variable selection seems to be an important item, the specificity of microarray data makes selection a difficult task. First, gene expressions are highly correlated. This means that a given expression profile will correspond to many different genes. Choosing one among them will then be somehow arbitrary, and so can be the deduced biological interpretation. An other consequence of redundancy is that small changes in the training data result in a completely different gene selection. This problem will occur for instance when choosing a best subset by cross-validation: given the withheld sample, variations in the select subset can be strong.

Selection methods are usually classified in *filter* or *wrapper* methods. Filter methods consider the discriminative power of each gene separately. For instance, a score is computed for each gene on the basis of the correlation between the gene expression and the status, and genes with highest scores are selected. Many filter methods lead to the choice of genes that are strongly differentially expressed, and thus should be carefully considered: the subset of selected genes may be highly redundant, while genes with lower score but original information may be displayed.

In wrapper methods, subsets of genes are directly considered, and error rates of the resulting classifiers are used to compare them. The main difficulty is computational: one cannot test each possible subset of genes, so genes have to be sequentially selected. In forward sequential selection, genes are selected one by one according to the information they bring for discrimination that is not contained in already selected genes. In contrast, backward selection starts with the entire set and discards at each step the gene whose information is not relevant regarding the information of the remaining genes. Although more attractive than filter methods, wrapper methods can be very unstable, because the selection of an ith gene is highly dependent of the subset of genes that have been already selected. Moreover, due to the complexity of wrapper procedures, generally no guarantee for the resulting classifier error rate can be stated.

A gold rule pointed out by Ambroise and McLachlan (02) is to consider the variable selection as a part of the training phase. We showed in the previous section that estimating the error rate with the training set gives optimistically biased performances. Similarly, one should not estimate the error rate of a classifier based on a selected subset of genes with the same dataset that was used to perform the selection. The authors show that in some cases, the estimation of the error rate can be biased by more than 15% if the variable selection step is not taken into account. Practical consequences of this remark are the following: in a LOOCV procedure to estimate the error rate, the variable selection has to be performed once the sample is withheld of the training set, and will then be performed as many time as the number of samples. One can alternatively estimate the error rate one a test sample, after the feature selection and the training phase.

Methods for variable selection are numerous, a good review may be found in Krishnapuram *et al.*. Some articles have been dedicated to the comparison between classification methods applied to microarray data, one may consult Dudoit *et al.* (2002) or Brown *et al.* (2000).

Bibliography

Alizadeh, A., Eisen, M., Davis, R. E., Ma, C. A., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T. ., Yu, X., Powell, J., Yang, L., Marti, G., Moore, T., Hudson, J., Chan, W. C., Greiner, T. C., Weissenberger, D. D., Armitage, J. O., Levy, R., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature.* **403** 503–511.

Alter, O., Brown, P. and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA.* **97 (18)** 10101–10106.

Ambroise, C. and McLachlan, G. (02). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA.* **99 (10)** 6562–6566.

Anderberg, H. H. (1973). *Cluster Analysis for Applications.* Academic Press.

Anderson, T. (2003). *An introduction to multivariate statistical analysis.* Series in Probability and Statistics. Wiley, 3rd edition.

Balazsi, G., Kay, K., Barabasi, A. and Oltvai, Z. (2003). Spurious spatial periodicity of co-expression in microarray data due to printing design. *Nucl. Acids Res.* **31** 4425–4433.

Baldi, P. and Long, A. (2001). A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics.* **17** 509–519.

Ball, C., Chen, Y., Panavally, S., Sherlock, G., Speed, T., Spellman, P. and Yang, Y. (2003). *Section7: An introduction to microarray bioinformatics.* (D. Bowtell and J. Sambrook, ed.). In DNA Microarrays: A Molecular Cloning Manual. Cold Spring Harbor Press.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerfull approach to multiple testing. *J. R. Statist. Soc. B.* **57 (1)** 289–300.

Biernacki, C. and Govaert, G. (1999). Choosing models in model-based clustering and discriminant analysis. *J. Statist. Comput. and Simul.* **94 (1)** 49–71.

Boheler, K. R. and Stern, M. D. (2003). The new role of SAGE in gene discovery. *Trends in Biotechnology.* **21 (2)** 55–57.

Bouroche, J.-M. and Saporta, G. (1998). *L´analyse des données.* Number 1854 in Que sais-je ? PUF.

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

Burnham, K. P. and Anderson, R. A. (1998). *Model Selection and Inference: A Practical Information-Theoretic Approach.* Wiley: New-York.

Butte, A. (2002). The use and analysis of microarray data. *Nature Review.* **1** 951–960.

Celeux, G., Chauveau, D. and Dielbolt, J. (1995), On stochastic versions of the em algorithm. Technical Report RR-2514, Institut National de Recherche en Informatique et en Automatique.

Churchill, G. (2002). Fundamentals of experimental designs for cDNA microarray. *Nature Genetics.* **32** 490–495.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association.* **74** 829–836.

Delmar, P., Robin, S. and Daudin, J.-J. (2003). Mixture model on the variance for the differential analysis of gene expression. *to appear in J. R. Statist. Soc. C.*

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B.* **39** 1–38.

Devroye, L. and Lugosi, G. (1995). Lower bounds in pattern recognition and learning. *Pattern Recognition.* **28** 1011–1018.

Draghici, S. (2003). *Data Analysis Tools for DNA Microarrays.* Chapman & Hall.

Dudoit, S., Fridlyand, J. and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* **97** 77–87.

Dudoit, S., Shaffer, J. and Boldrick, J. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science.* **18 (1)** 71–103.

Duggan, D., Bittner, M., Chen, Y., Meltzer, P. and Trent, J. M. (1999). Expression profiling using cDNA microarrays. *Nature Genetics.* **21** 10–14.

Efron, B., Tibshirani, R., Storey, J. and Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *J. Amer. Statist. Assoc.* **96** 1151–1160.

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

Fraley, C. and Raftery, A. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *Comput. J.* **41 (8)** 578–588.

Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 148–156.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. and Lander, E. (1999a). Class prediction and discovery using gene expression data. *Science.* **286** 531–537.

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

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika.* **82 (4)** 1151–1160.

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

Kerr, M. K. and Churchill, G. (2001). Experimental design for gene expression microarrays. *Biostatistics.* **2** 183–201.

Kerr, M. K., Afshari, C. A., Bennett, L., Bushel, P., Martinez, J., Walker, N. J. and Churchill, G. A. (2002). Statistical analysis of a gene expression microarray experiment with replication. *Statistica Sinica.* **12** 203–218.

Kerr, M., Martin, M. and Churchill, G. (2000). Analysis of variance for gene expression microarray data. *J. Comp. Biol.* **7 (6)** 819–837.

Krishnapuram, B., Carin, L. and Hartemink, A. Gene expression analysis: Joint feature selection and classifier design. *To appear (2004).*

Leung, Y. F. *and* Cavalieri, D. *(2003). Fundamentals of cDNA microarray data analysis.* Trends in Genetics. **19 (11)** *649–659.*

LÖNNSTEDT, I. *and* SPEED, T. *(2001). Replicated microarray data.* Statistica Sinica. **12** *31–46.*

LUAN, Y. *and* LI, H. *(2003). Clustering of time-course gene expression data using a mixed-effects model with b-splines.* Bioinformatics. **19 (4)** *474–482.*

MARTIN, M. L., AUBERT, J., CABANNES, E. *and* DAUDIN, J. *(2004). An evaluation of the effect of labeling artefacts on the genes differential expression in cDNA microarray experiments. submitted.*

MARY-HUARD, T., ROBIN, S., DAUDIN, J., BITTON, F., CABANNES, E. *and* HILSON, P. *(2004). Spotting effect in microarray experiments.* BMC Bioinformatics. **5 (63)** *1–9.*

MCLACHLAN, G. *and* PEEL, D. *(2000).* Finite Mixture Models. *Wiley.*

MCLACHLAN, G. *(92).* Discriminant analysis and statistical pattern recognition. *Wiley.*

NADON, R. *and* SHOEMAKER, J. *(2002). Statistical issues with microarrays: processing and analysis.* Trends in Genetics. **18 (5)** *265–271.*

PARMIGIANI, G., GARRETT, E., IRIZARRY, R. *and* ZEGER, *editors. (2003).* The analysis of gene expression data: methods and software. *Springer.*

POLLACK, R. *and* IYER, V. *(2003). Characterizing the physical genome.* Nature Genetics. **32** *515–521.*

QUACKENBUSH, J. *(2001). Computational analysis of microarray data.* Nature Review Genetics. **2** *418–427.*

QUACKENBUSH, J. *(2002). Microarray data normalization and transformation.* Nature Genet. **32** *496–501.*

RAO, C. *(1965).* Linear statistical inference and its applications. *New York : John Wiley & Sons, Inc.*

RUDEMO, M., LOBOVKINA, T., MOSTAD, P., SCHEIDL, S., NILSSON, S. *and* LINDAHL, P. *(2002), Variance models for microarray data. Technical Report 6, Mathematical Statistics, Chalmers University of Thechnology.* http://www.math.chalmers.se/~rudemo/.

SCHAFFER, R., LANDGRAF, J., ACCERBI, M., SIMON, V. V., LARSON, M. *and* WISMAN, E. *(2001). Microarray analysis of diurnal and circadian-regulated genes in Arabidopsis.* Plant Cell. **13** *113–123.*

SCHÖLKOPF, B. *and* SMOLA, A. *(2002).* Learning with kernels. *MIT Press.*

Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H. *and* Herzel, H. *(2000). Normalization strategies for cDNA microarrays.* Nucl. Acids Res. **28** *e47.*

Sokal, R. R. *and* Sneath, P. H. A. *(1963).* Principles of numerical taxonomy. *Freeman.*

Storey, J. D., Taylor, J. E. *and* Siegmund, D. *(2004). Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: a unified approach.* J. R. Statist. Soc. B. **66** *187–205.*

Storey, J. D. *and* Tibshirani, R. *(2003). Statistical significance for genomewide studies.* Proc. Natl. Acad. Sci. USA. **100 (16)** *9440–9445.*

Tusher, V. G., Tibshirani, R. *and* Chu, G. *(2001). Significance analysis of microarrays applied to the ionizing radiation response.* Proc. Natl. Acad. Sci. USA. **98** *5116–5121.*

Vapnik, V. *(1998).* Statistical learning theory. *Wiley, NY.*

Westfall, P. *and* Young, S. *(1993).* Resampling-Based Multiple Testing: Examples and Methods for P-value Adjustment. *Wiley.*

Workman, C., Jensen, L., Jarmer, H., Berka, R., Gautier, L., Nielser, H., Saxild, H., Nielsen, C., Brunak, S. *and* Knudsen, S. *(2002). A new non-linear normalization method for reducing variability in DNA microarray experiments.* Genome Biol. **3 (9)** *1–16.*

Yang, Y., Dudoit, S., Luu, P. *and* Speed, T. *(2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.* Nucl. Acids Res. **30 (4)** *e15.*

Yang, Y. *and* Speed, T. *(2002). Design issues for cDNA microarray experiments.* Nature reviews. **3** *579–588.*