Université Libre de Bruxelles

Faculté des Sciences Appliquées

Institut de Recherches Interdisciplinaires

et de Développements en Intelligence Artificielle

ULB

Algorithms for the analysis of gene expression data

David Venet

Thèse présentée en vue

de l'obtention du grade de

Docteur en Sciences Appliquées

Promoteur : Pr. H. Bersini

Université Libre de Bruxelles

Faculté des Sciences Appliquées

Institut de Recherches Interdisciplinaires

et de Développements en Intelligence Artificielle

ULB

Algorithms for the analysis of gene expression data

David Venet

Thèse présentée en vue

de l'obtention du grade de

Docteur en Sciences Appliquées

Promoteur : Pr. H. Bersini

Acknowledgements

First of all, I would like to thank those who have given me the opportunity to make this work and believed in me, that is Prof. Hugues Bersini of the IRIDIA and Profs. Jacques Emile Dumont and Carine Maenhaut of the IRIBHM. Thank you for your support and your help. Thanks you also for having listened patiently to my ideas, and for your suggestions.

I would also like to thank the experimentalists I have been working with. In chronological order: Frédéric Pécasse, Hortensia Mirescu, Agnès Burniat, Sandrine Wattel, Laurent Delys and Wilma van Staeveren. The continual flood of new data was an important motivation to develop and improve analysis methods. Constructive discussions with people having a very different point of view were often mind openers.

I would thank Vincent Detours for its careful reviewing of this thesis. To have such a colleague correct my style and simplify my thinking was an important help. His following of the recent literature also proved very useful.

I could not forget the help given to me by various members, past and present, at the IRIDIA. Marco Sarens for its encyclopedic knowledge on data mining. Patrice Latine for its help on statistics and bootstrapping. Frank Vanden Berghen for always finding the latest version of the software I needed.

This thesis was supported by the Région Wallonne and UCB Pharma. I would especially thank Pierre Chatelain of the UCB for financing this project.

Introduction

This work stands somewhere at the fringe between biology and computer science, in this new field called bioinformatics. This means it is an exciting new domain of research, but also that it is difficult to write to be understood by both biologists and computer scientists at the same time. Or even separately. For this reason, an introductory background chapter presenting the prerequisite for both biologists and computer scientists has been written. The notions presented in this chapter shall be considered as known for the rest of this work.

A few years ago, as a by-product of the sequencing effort and of the automatization of the biological sciences, it became possible to measure the level of expression of thousands of genes in parallel. This gave rise to very high hopes – it was suddenly possible to perform in one afternoon the experiments which would have taken months with classical means. It seemed to be possible to uncover slight differences between diseases in a few months, while it would have taken centuries of man-years, or to infer parts of the regulatory mechanisms which govern our cells. The numbers of publications exploded in the field, a trend which is not likely to subside anytime soon (figure 1).

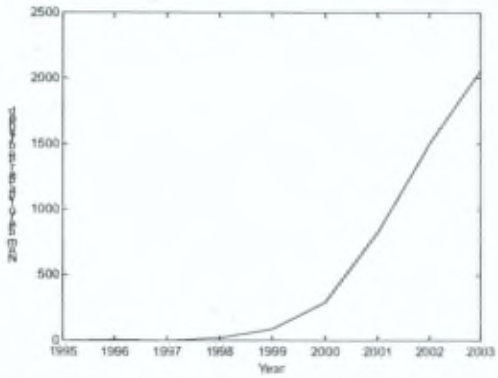


Figure 1. Number of microarray papers published by year, 1995 to present. Number for 2003 is the results of the first six months times two.

It appeared however that the data were not always what they seemed to be. Microarrays were plagued by reproducibility issues, the results from one laboratory did not seem to fit the results from another and in some cases the identity of a large proportion of the genes proved to be incorrect (Knight 2001). This gave rise to some discomfort, as it was not clear to which extent the data were able to fulfill their promises.

But as time went by, the technology improved and some of the boldest claims were withdrawn, or at least amended. Some of the technology improvement came from the wet side of the laboratory, and some came from the data analysis. As the data proved more complex than previously thought, new, more powerful, means of extracting the information present were designed.

This thesis is a small part of this whole story. As such, it treats of the issues which must be tackled by anyone trying to work with those new data.

The first chapter treats of a seemingly simple point: if two groups of samples are compared, how to determine which genes are differentially expressed, and to give a significance level to those differential expression. The problem proved to be more complex than anticipated however, for two reasons. The first is the fact that the microarray experiments are an extreme case of multiple-testing. To apply the classical solution for multi-testing proved too conservative, so another approach had to be designed. The second is the

fact that there are information as to the confidence that could be given to the result for a gene – basically, a gene with a high expression tends to be more reproducible than a gene with a low expression. The question is then, how to introduce that information in the significance analysis, without simply throwing away a large part of the data.

The second chapter treats of the problem of the correction of the data. Microarray data are plagued with errors. The good news is that some of these errors are systematic, and so can be removed. To decide which correction to apply, and how, and to assess whether the correction really improves the data quality is not trivial. A general method of assessing the data quality, based on the significance analysis developed in the first part, was created. Using this method, many different options to improve the data were tested, and a general normalization scheme was designed. The tools developed in this part of the work were made public, as a Matlab toolbox.

The third chapter treats of the data storage and retrieval. It consisted in the creation of a database, with a Web front-end. The main problem however proved to come not from the database creation, but from the data curation – to make sure that the data is clean, fits in the database format and is correctly annotated. This proved to be a very time-consuming and bothersome work. As public databases seemed to be in the pipeline, from which they emerged recently, the project was essentially abandoned. There were also a few design choices which might have proven to be inefficient in the long run.

The fourth chapter treats of clustering. Clustering is a natural choice for gene expression data sets, because those data sets have an unknown structure which would be very useful to uncover. Almost all gene expression papers use clustering for their data analysis. The precise point treated in this part is the finding of the different clustering of the samples present in the data. For instance, the samples could be clustered in function of their inflammation level, or the sex of the patient, or its age,... The hypothesis is that those clusterings are independent, and so could be uncovered separately. This chapter offers an algorithm to determine those overlapping clustering. This algorithm can also be useful for other types of data, which is demonstrated on a census data set where the presence of overlapping clustering is also expected.

The fifth chapter treats of complex samples. When gene expression experiments are performed on solid samples, comprising many cell types, the results depend on the composition of those samples. Depending on the way the samples are extracted, the concentration of the different cell types can vary widely. If the samples are clustered, then the main source of variability is this variation in cell type concentration – a very real effect, but usually unhelpful for biological understanding. A technique to correct for this effect is presented in this chapter.

The sixth chapter treats of the most ambitious hope about high-throughput gene expression data – the hope that it might some day allow the determination of the regulatory program inside the cells. There are many difficulties on that road, one of which being that genes do not directly control each other, but that there are many intermediates. The search for direct regulators of gene expression among the expression of other genes might be doomed to failure. An alternative approach is proposed here. If the number of regulators is relatively small, it might be possible to determine both the value of the regulators in each condition and the way they regulate the genes. The way the regulators themselves are regulated can then be treated separately. The effectiveness of this approach is shown using a binary model of gene regulation, and is successfully applied on two real data sets.

1 Reference

Knight, J., (2001) "When the chips are down", Nature, 410, 860-861.

0 - Background

The biological and technical backgrounds needed to understand this thesis are given in this chapter. The first part describes the relevant biology. The second part describes the experimental protocols used to generate the data analyzed in this thesis. The first two parts are geared towards computer scientists. The third part describes some classical analysis techniques commonly used in the field, and is geared towards biologists. The fourth part describes a data set which has been created in the IRIBHM laboratory and was used for the validation of the techniques developed in the first and second chapter. The analysis techniques described are demonstrated on this data set.

1 A short introduction to the relevant biology

The biological background is presented here. It is only a sketchy presentation of the current biological knowledge, based on the book "Molecular Cell Biology" (Darnell *et al.* 1990).

1.1 DNA holds genetic information

Living organisms are composed of cells. Some organisms, like yeast or bacteria, consists of only one cell while others (metazoans), like humans, consist of many cells. In metazoans, cells can perform a variety of functions. They can for instance act as neurons, muscles or lymphocytes.

The cells could be viewed as chemical factories, using energy and other resources in the environment to produce other cells or to perform a function useful for the organism. The program of the factory is coded in DNA. DNA is a long polymer. It consists of a sugar phosphate backbone on which any of four different bases can be attached: adenine, guanine, thymine or cytosine. DNA is made of two complementary strands, with the bases paired: adenine with thymine and guanine with cytosine (figure 1). The program of the cells is coded in the succession of those bases.

Each cell possesses one or more of those polymers made of DNA, which are called chromosomes. The chromosomes encode all the information the cell needs to survive.

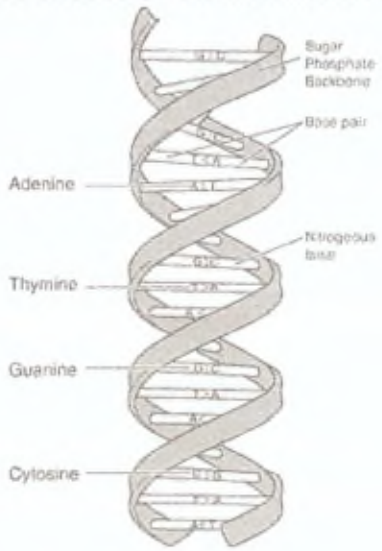


Figure 1. The DNA structure

1.2 The translation of DNA into protein

Proteins are the basic building blocks of the cells. They perform most of the functions needed for the cells. Proteins can, for instance, catalyze chemical reactions, be used as

structural elements or turn chemical energy into mechanical energy. Proteins are polymers, with the mers made of any of 20 amino acids. The amino acid sequence of proteins is encoded in DNA.

The synthesis of proteins is a two-step process (figure 2). Firstly, the part of DNA containing the sequence of the protein is copied into RNA. This first step is called transcription. RNA is a polymer similar to DNA, the main differences being that it contains only one strand and that the base thymine is replaced by the base uracile. Those differences make RNA a less stable molecule than DNA. The RNA containing the copy of the plan of the protein is called a messenger RNA (mRNA). This mRNA is used as a plan for the protein synthesis.

The information concerning the protein sequence is coded in the mRNA. As each mer in a protein is one of 20 amino acids, 3 bases are needed to code the information. Hence, the sequence of an mRNA can be divided in 3-bases long blocks called *codons*. Each codon encode for one amino acid. As 3 bases give $4^3 = 64$ possibilities, the coding from codon to amino acid is redundant: one amino acid can be associated with more than one codon. Some codon can also have specific meaning, like start or stop transcription.

The second step of protein synthesis is translation from mRNA to protein, or simply translation. It is performed by a specialized molecular complex called the ribosome. An mRNA is continuously translated in protein until it is degraded. This means that only a very small amount of mRNA is needed, as one mRNA can give rise to hundreds of proteins.

Only a small part of the genome encodes proteins. Some of the rest of the genome is used to control when or where the genes should be expressed, that is for instance when they should be transcribed into mRNA. The usefulness of most of the genome remains however a mystery.

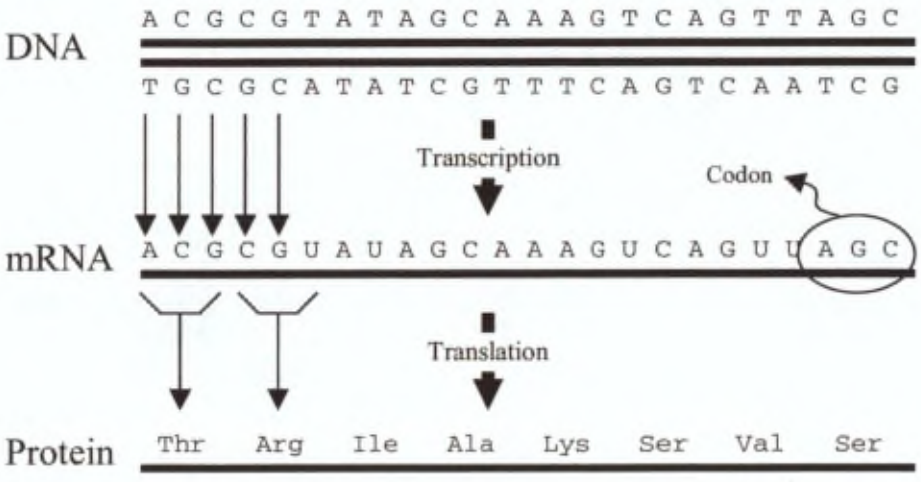


Figure 2. The protein synthesis.

1.3 Control of transcription

In a multicellular organism, all cells have the same genetic code, but can perform different tasks. The behavior of a neuron is very different from the behavior of a muscle cell. Unicellular organisms can have different behavior, mating, feeding or moving, depending on the environment and their internal state. A large part of these variations in behavior are due to variations in the expression level (concentration) of the proteins.

Cells have different ways of controlling the expression of the proteins. A large part of this control is done at the transcription level, that is when DNA is copied into mRNA. This control can for instance be made by proteins, which bind to a specific DNA sequence and activate the transcription of a nearby sequence. The DNA sequence corresponding to a

protein plus all its regulatory sequences is called a gene. For instance, if a cell senses a signal, it can activate a protein which, by binding to the genome, can activate or inhibit the transcription of some genes. The proteins coded by some of those genes can themselves alter the expression of other genes (or of themselves).

Since the behavior of a cell is largely determined by the concentration of the various proteins in the cell, and the concentration of those proteins is largely controlled by the concentration of the corresponding mRNAs, measuring the level of expression of the different mRNAs (an expression profile) gives a relatively comprehensive view of the molecular state of the cell. The advantage of working with mRNA concentrations is that those are much easier to detect and quantify than protein concentrations, as will be explained in the next section.

2 The microarray technology

In this section the microarray technology which was used to generate the data analyzed in this thesis is described. This technology allows a simultaneous quantification of the concentration of thousands of mRNAs.

2.1 Gene expression data

All gene profiling technologies essentially produce the same type of data. Most of the techniques presented in this thesis can be applied to data generated by any technology.

Gene profiling is the high-throughput quantification of mRNAs in a biological sample. A sample consists of mRNAs which have been extracted from cells. Those cells can be taken from a living organism (*e.g.* after a tumor has been surgically removed) or from cells grown in laboratory. The resulting data can be organized as an array, with as many lines as there are genes (mRNAs) quantified and as many columns as there are samples (see figure 3 for an example). The expression of the different genes for a sample is called an expression profile. Those profiles give a relatively accurate measure of the molecular state of the cells. They can be used for many different purposes.

Firstly, it is possible to find molecular differences between groups of samples. For instance, one group could consist of cancerous samples and the other of normal samples. This can be used to classify new samples as either normal or cancerous. This can also be used to help the understanding of the molecular basis of the disease.

Secondly, it is possible to discover previously unknown subtypes in a disease. For instance, measures on a set of cancer samples can lead to the discovery that only two patterns really exists in the set, each sample being identified with one pattern. Clustering of the data is the classical mean to uncover the existence of such groups.

Thirdly, genes which show a similar evolution through the samples (co-regulated genes) can be clustered. Co-regulated genes often share a similar function (Eisen *et al.*, 1998). Thus the discovery of co-regulated genes allows the inference of the function of unknown genes.

Ideally, it should be possible to ignore the details of the data generating processes and still make a meaningful data analysis. Practically however, it has been found that high-throughput technologies lead to high-throughput errors. Those errors are often a function of the technology used, and can sometimes be corrected. Also, differences between the scaling of the values between the techniques can lead to non-trivial effects (see chapter 5 for an illustration). For those reasons, some details about the experimental protocols used to create the data are presented in the next few sections.

	Sample 1	Sample 2	Sample 3	Sample 4	...
Gene 1	10	50	12.5	32	...
Gene 2	1	.3	1.2	3.7	...
Gene 3	957	360	1456	1057	...
...	...	...	...	...	...

Figure 3. A small part of a typical high-throughput gene expression data set. Each line corresponds to one gene measured, each column to one sample. The measures are the mRNA expression levels of gene A in sample B.

2.2 The hybridization property of DNA

DNA in the chromosomes is present as a double stranded DNA. However, single stranded DNA can also exist. If a single stranded DNA is put in presence of another single stranded DNA of complementary sequence (with A replaced by T and C replaced by G), then the two single stranded DNA will hybridize to form a double stranded DNA. This property of DNA is at the base of many experimental protocols in molecular biology.

There are some enzymes (*i.e.* proteins) which are able to reverse-transcribe mRNA into the complementary DNA (cDNA). Those enzymes were isolated in RNA viruses. In those viruses, the genetic information is coded in RNA, and must be transcribed into DNA in the host cells to use the host cell machinery to replicate.

It is possible during the reverse-transcription to modify the building blocks (the bases) of cDNA. Typically, a fluorescent label is attached on one of those bases. The resulting DNA is then traceable.

This labeled cDNA can then be hybridized with some DNA of known complementary sequence immobilized on a surface. After washing, only the cDNA whose sequence is complementary to the one of the immobilized DNA remains. The amount of labels present in the cDNA immobilized on the surface is then proportional to the amount of cDNA present in the solution, and hence gives a measure of the concentration of a particular mRNA in the original sample.

Many different protocols have been designed which make use of those properties of genetic material. Two protocols, microarrays and Affymetrix oligonucleotide array, are essentially a high-throughput version of the experiment outlined here. They are succinctly described in the next sections.

2.3 The microarray protocol

The microarray technology (Schena *et al.*, 1995) compares mRNAs from two different samples. Firstly, mRNA is extracted from the samples. It is then reverse-transcribed into cDNA. During the reverse-transcription process, a label is incorporated into the cDNA. Two different labels are used for the two samples, so that each can be detected independently. Schematically, one label is detected in red and the other in green. Both labeled cDNA are mixed and hybridized simultaneously on the array.

Microarrays are microscope slides on which thousands of spots are laid down. The DNA corresponding to one gene is present in each of these spots. After hybridization and washing, the fluorescence from each spot can be measured using a specially designed scanner. The ratio of the fluorescence intensities of the two labels gives the ratio of the abundance between the two samples, for the gene corresponding to the spot. Figure 4 shows an example of the result of a microarray experiment. The intensity from one channel is represented in green, and from the other channel in red. If the intensity is the same in both channels, the resulting spot is yellow.

There are a few advantages in comparing two samples, instead of measuring just one sample:

1. It should work at any mRNA or immobilized DNA concentration. If only one sample is measured, the amount of labeled cDNA bound to the spot is proportional to the amount in the solution only inasmuch as there is no saturation. By comparing two samples, the

binding chemistry should be independent of the labeling, and so the ratio of intensity with one label to the intensity with the other label is independent of the chemical details.

2. Many types of errors are correlated between the two labels, and should disappear when the ratio is taken. For instance, the intensity measured is proportional to the size of the spot. So if only one label is used, bigger spots would lead to bigger intensity values, independently of the mRNA concentration. As the current microarray spotting technology is far from perfect, this effect could ruin the data. However, taking two different labels and focusing on the ratio cancel the spot size effect.

The main inconvenient in using two samples is that it does not always fit with the experimental protocol. For instance, if cancer samples are compared, is it not clear what to use as the second sample. The usual solution is to use a reference sample, that is an artificial sample used only for comparison purpose. In this case, the scaling of the values for the genes is very dependent on the reference sample used, and a normalization of the genes values may prove necessary.

As the protocol used in the IRIBHM is based on the microarray technique, some parts of the work presented in this thesis are geared towards microarrays. Most of the latter parts, however, can be applied on microarray, Affymetrix and other gene expression data.

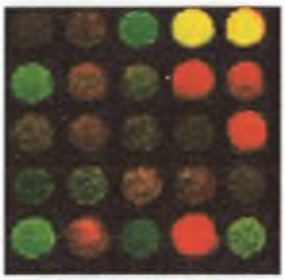


Figure 4. Example of a small part of a microarray slide.

2.4 The Affymetrix protocol

As a large part of the examples are based on data generated using the Affymetrix protocol (Lockhart *et al.*, 1996), it is described succinctly here.

The Affymetrix protocol is quite similar to the microarray protocol. One of the main differences lies in the slides. In Affymetrix slides, contrary to microarrays, the DNA in the spots is directly synthesized *in situ*, using a process reminiscent of the one used for semi-conductors. This process allows the synthesis, at precise positions and concentrations, of small spots containing oligonucleotides, that is stretches of DNA about 20 base-pairs long.

An advantage of this fabrication process is that the spots are much more reproducible. For this reason, it is possible in this protocol to use only one label, instead of the two used in the microarray protocol. The resulting image, after hybridization and washing of the sample, is quite similar to the one generated with microarrays (figure 5).

A disadvantage is that the oligonucleotides are short. This could lead to specificity problems, that is an oligonucleotide might hybridize with a cDNA different from the one expected. To address this issue, 11 oligonucleotides are used for each gene, and 11 mismatch oligonucleotides, having just a one base difference with the main oligonucleotides, are also present to measure the amount of non-specific hybridization. Those 22 values are integrated to give a unique intensity for each gene.

In conclusion, the Affymetrix protocol leads to good quality gene expression data. The measures made are absolute, contrary to microarrays where two samples are compared.

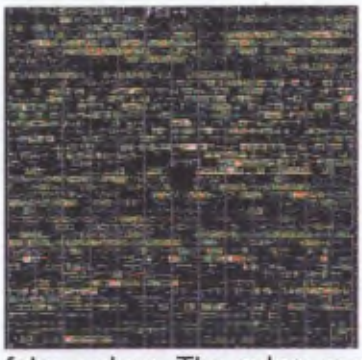


Figure 5. An Affymetrix slide, in false colors. The colors represent the intensity of the spots. The side-by-side layout of the oligonucleotides concerning the same gene explains the presence of horizontal lines.

3 An example from the IRIBHM laboratory

A microarray experiment is described in this chapter. The aim of the study was to define gene expression profile in different thyroid tumors: autonomous adenomas (benign hyperfunctioning tumors) and papillary cancers (malignant tumors). The data produced by this experiment will be used in the first and second chapter to illustrate and validate the methods proposed. The experiments were performed in the IRIBHM laboratory, by Frédéric Pécasse, Agnès Burniat, Carine Maenhaut and Sandrine Wattel.

3.1 The thyroid and some of its malfunctions – a very short summary

3.1.1 The normal thyroid

The thyroid is a gland located in the neck (figure 10.A). Its influence is both far-reaching and critical to normal body function. It affects heart rate, cholesterol level, body weight, energy level, muscle strength, skin condition, vision, menstrual regularity, mental state and a host of other conditions.

The thyroid gland operates as part of a feedback mechanism involving the hypothalamus and the pituitary gland. First, the hypothalamus sends a signal to the pituitary gland through a hormone called TRH (thyrotropin releasing hormone). When the pituitary gland receives this signal, it releases TSH (thyroid stimulating hormone) to the thyroid gland. Upon receiving TSH, the thyroid responds by releasing two of its own hormones, T4 and T3, which then enter the bloodstream and affect the metabolism of the heart, liver, muscle and other organs. T4 is the main hormone released by the thyroid. T3 is made in the tissue after T4 to T3 conversion. Finally, the pituitary "monitors" the level of thyroid hormone in the blood and increases or decreases the amount of TSH released, which then changes the amount of thyroid hormone in the blood.

The thyroid tissues is mainly composed of thyroid follicular cells, the thyrocytes (70%), and of their supporting tissue and cells. The thyrocytes are arranged in follicles. A follicle (figure 10.B.) consists in thyrocytes organized as a sphere, with an interior cavity, the lumen, filled with colloid. Upon stimulation by TSH, the thyrocytes ingest the colloid, digest it and release the products of this digestion. These products are the T3 and T4 hormones.

The main cells of interest in the thyroid are the thyrocytes.

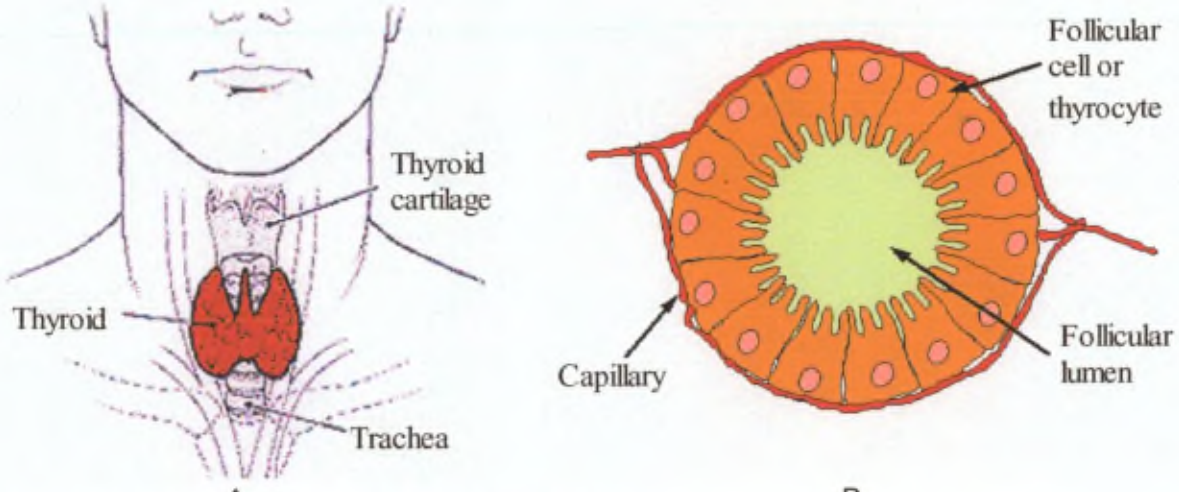


Figure 10. **A.** The thyroid gland. **B.** A thyroid follicle.

3.1.2 Autonomous adenomas

Autonomous adenomas (Deleu *et al.*, 2000) are benign encapsulated tumors that grow and secrete thyroid hormones independently of the normal TSH control. They result from a constitutive activation of their TSH signaling cascade. This means that the thyrocytes behave as if there is large amount of TSH in the environment, independently of the actual TSH concentration. Studies have found activating mutations in the TSH receptor in about 80% of the autonomous adenomas in Europe.

If untreated, autonomous adenomas cause hyperthyroidism and therefore represent an important medical problem.

3.1.3 Papillary thyroid carcinomas

There are different types of thyroid cancers. Papillary thyroid carcinomas (PTC) are the most common (70-80%) and can occur at any age. Those are malignant tumors, showing papillary and follicular architecture. This cancer is the most common form of solid cancer associated with radiation exposure. This is due to the assimilation of radioactive iodine, which accumulates in the thyroid.

Two different sources of PTC were used in the laboratory: Chernobyl PTC, for which the cause and timing of the cancer are known, and sporadic PTC, which come from western Europe patients for which no precise cause can be pinpointed. It is not clear whether those two groups show an identical pathology. One of the goals of this study was to uncover a difference between those two groups, or to demonstrate that they are actually identical.

3.2 The experiments

Samples of pathological and the paired normal tissue were obtained. The mRNA was extracted from those samples, and reverse-transcribed in cDNA. The cDNA were marked using a modified labeling protocol which allows using a much smaller amount of starting material (1µg RNA instead of 50-80µg – TSA methodology, Perkin-Elmer). On each slide, the pathological tissue was marked in one color and the normal tissue in the other color. Hence, two sets of numbers are obtained: one set for the normal sample and one set for the pathological sample. The ratio of those numbers should be proportional to the difference between normal and pathological samples.

The rationale in comparing each pathological sample with the corresponding normal sample is that there is a large variation in the gene expression between normal samples, and that this variation is carried over in pathological samples. However, the modifications which turned the normal tissue into a pathological tissue should be similar from sample to sample. So, it should be more efficient to compare the differences between paired normal and

pathological samples, than to compare a group of normal samples to a group of pathological samples.

The goals of the experiments were to:

1. Find genes differentially regulated in autonomous adenomas (AA) and in papillary thyroid carcinomas (PTC).
2. Determine to which extent AA are different from PTC.
3. Determine if AA and PTC are homogenous groups, that is if they could be divided in logical sub-groups.
4. Determine if there is a difference between post-Chernobyl and sporadic PTC.

The first and fourth points are treated in the chapter 1 of this thesis. The second and third points can be answered using the clustering techniques described.

4 A primer on clustering

The microarray technology generates a lot of data, since typically the expression of thousands of genes are measured in tens of conditions, that is there are hundreds of thousands of measurements. It is unrealistic, and inefficient, to consider those data as simply as many individual experiments. Statistical techniques must be used to uncover the information present in those data.

Often, it is not known beforehand that the samples could be separated in groups. As microarrays are research tools, the objective of the experiments is often to distinguish possibly heterogeneous group of samples. In this case, the structure of the variations between the samples might be of the utmost interest. For this, so called non-supervised classification methods must be used. Those methods search a structure in the data, like for instance the presence of groups of similar samples.

Unsupervised techniques produce results with any data – even random data. Most algorithms do not give a satisfactory measure of the significance of the results. This means that the output of the algorithms must always be taken cautiously. They must be considered more as hypotheses generating algorithms than as truth givers. Their results have to be checked, using external knowledge or additional experiments.

Data has to be visualized as to highlight its structure. For this reason, it is necessary to use some sort of unsupervised organization technique. Those techniques are usually called clusterings. The most common two of those techniques, the hierarchical clustering and the K-means clustering, are presented next. A more thorough description of those algorithms can be found in the book of Jain and Dubes (1988).

4.1 Hierarchical clustering

Hierarchical clustering techniques organize the data as a tree, like a phylogenetic tree. This organization can be performed on the genes, on the samples or on both.

Figure 6 shows an example of the output of a hierarchical clustering algorithm. In this example, the objects A, B, C, D, E, F, G and H have been clustered. In gene expression analysis, those objects could be genes or samples. The places at the bottom of the tree, where the object names are written, are called leaves. The junctions are called nodes. The distance between two objects is given by the height of the first node which links these two objects. For instance, B and H are very close. The distance between G and D is very large, since the lowest node linking the two is the top of the tree. The fact that they are next one to the other on the bottom of the tree is not relevant. The distance between G and D is the same than the distance between any of (B,G,H) and any of (D,C,F,E,A).

It is possible to use a hierarchical clustering algorithm to find groups in the data, by cutting the tree at a certain height. For instance, it might be considered than on the example there are two groups, (H,B,G) and (D,C,F,E,A). Or three groups, (H,B,G), (D,C) and (F,E,A). Or eight groups, each containing only one leaf. The number of groups is a choice from the user.

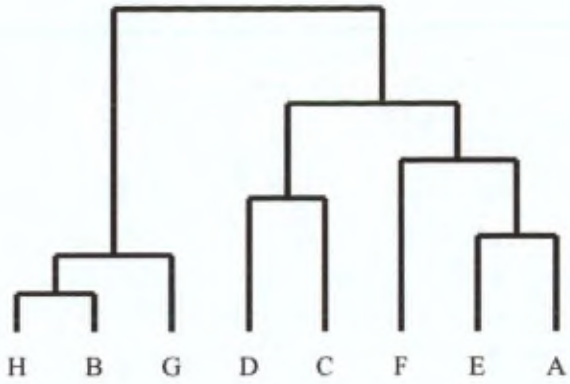


Figure 6. An example of hierarchical clustering.

There are a many different hierarchical clustering algorithms. The most common ones are described here. Those algorithms are bottom up, in that they start by making small groups and then try to merge those small groups to form bigger groups. They only use pair-wise differences (or similarities) between objects.

The input of the algorithm is a matrix of pair-wise dissimilarities between the objects (see figure 7 for an example). Those dissimilarities might be calculated using any function, like for instance Euclidean distance. A similarity measure, like correlation, might also be used simply by multiplying all values by –1 to turn it into a dissimilarity measure.

An example of the algorithm in motion is given figure 6. At each step, the closest two objects are merged. For the first step, those are the objects B and C, with a distance of only 1. A tree with the two objects linked with a node of height 1 is created. The distance matrix is then updated, because distances between the objects and this new node must be created. There are different ways to calculate those distances:

Single linkage: the distance between two nodes is the distance between the closest leaves from those two nodes. With single linkage, the distance between A and the group (B,C) would be 2.

Complete linkage: the distance between two nodes is the distance between the furthest leaves from those two nodes. With complete linkage, the distance between A and the group (B,C) would be 3.

Average linkage: the distance between two nodes is the mean of the distances between the leaves from those two nodes. With average linkage, the distance between A and the group (B,C) would be 2.5.

In this case, average linkage is chosen. This leads to an updated distance matrix shown in the step 2. The two closest objects are again merged, this time it is the group (B,C) and the object A, with a distance of 2.5. The distance matrix is again updated: for instance, the distance between the group (A,B,C) and the object D is (5.5*2 + 7)/3 = 6. This continues until all objects are linked.

The different matrix updating schemes usually lead to different results. The single linkage tends to give long elongated clusters, the complete linkage small compact clusters and the average linkage something in-between. There are also other, more complex, updating schemes which are outside the scope of this primer. Most of the time, average linkage proves to be the soundest choice.

	A	B	C	D	E
A		2	3	5	7
B	2		1	4	8
C	3	1		6	3
D	5	4	6		6
E	7	8	3	6	

	A	B C	D	E
A		2.5	5	7
B	3		5	6
C				
D	5	5		6
E	7	5.5	6	

	A B C	D	E
A			
B		5	6
C			
D	5		6
E	6	6	

	A B C D	E
A		
B		6
C		
D		
E	6	

Step 1 Step 2 Step 3 Step 4

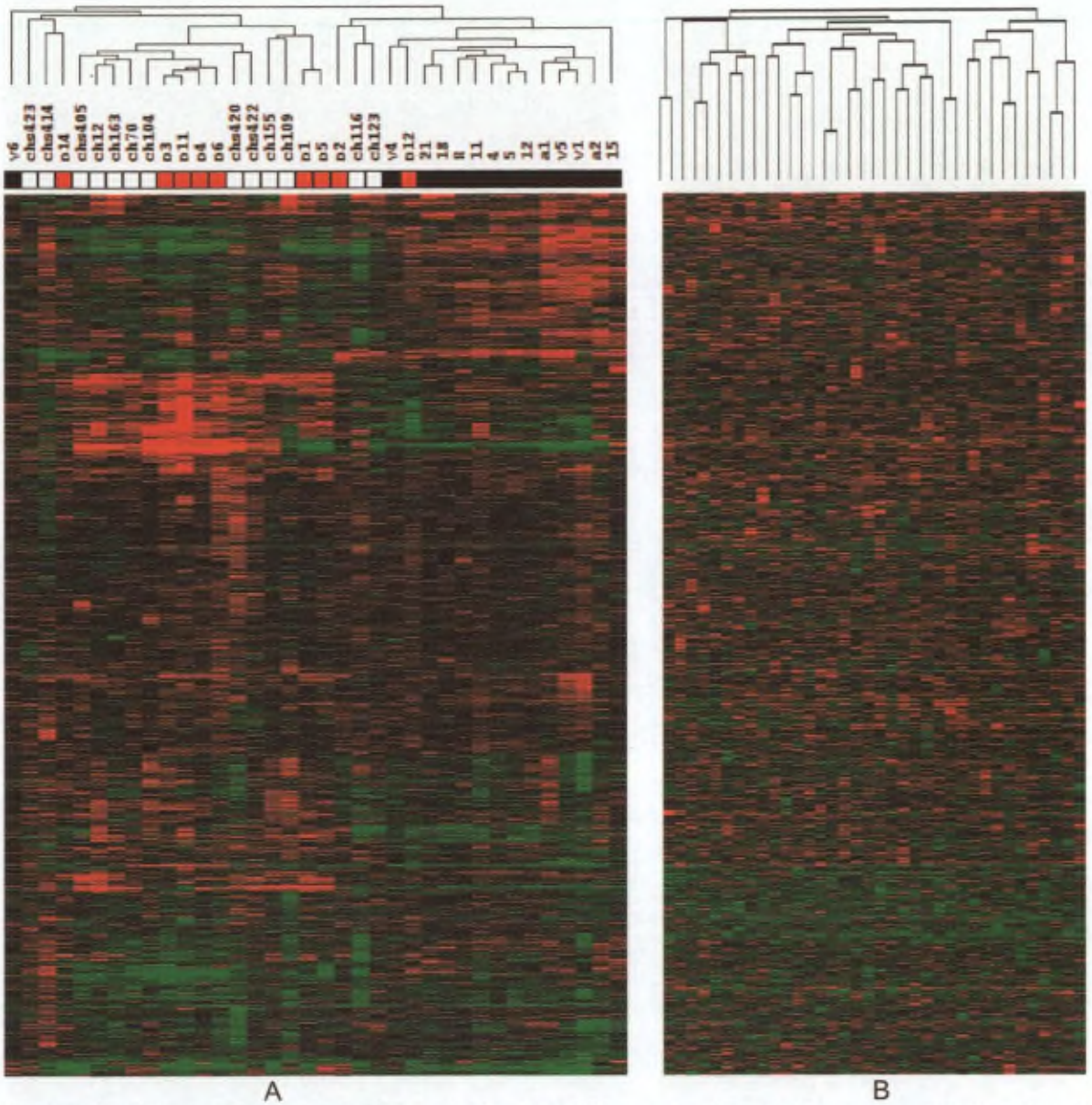
Figure 7. Iteration of a hierarchical clustering algorithm, using the average linkage algorithm. Upper row: distance matrices. The values have been rounded if needed to fit into the boxes. Lower row: the growing tree.

Hierarchical clustering organizes the data. This fact has been used by Michael Eisen (1998) to create a visualization method which proved very popular (figure 8). The idea is to perform a hierarchical clustering on the genes and most of the time on the samples. The data is represented with the gene clustering on the side, the sample clustering on the top, and the reordered data in the middle using a color code, usually black for values close to one, red for values higher than one and green for values lower than one. As the gene expression data is very redundant, there is a sense of continuity in the resulting image which makes it understandable by human beings. This representation method makes the following immediately clear:

1. There is a large difference between the autonomous adenomas (AA) and the papillary thyroid carcinomas (PTC), so that they cluster in two different groups.
2. The AA group is quite homogenous. There are non-microarray-related reasons to believe that the two samples which do not fit the group very well, 15 and v6, have a different histology.
3. There does not seem to be a difference between the Chernobyl and the sporadic PTC, although this should be verified using more precise techniques.
4. The PTC group is less homogenous than the nodule group. It might be that the PTC could be further divided in subgroups. The number of samples is too small to draw any firm conclusion.

This illustrates the usefulness of the clustering and visualization technique to gain a broad qualitative understanding of the data. Groups with a striking signature are readily detected, as shown by the PTC vs. AA difference.

On figure 8B the clustering of a randomized version of the data of figure 8A is presented. The lack of organization is striking. This highlights the fact that the gene expression data has indeed a non-trivial organization, and that it can be valuable to uncover it using non-supervised techniques.



A B

Figure 8. Hierarchical clustering of the thyroid expression data represented using the technique of M. Eisen. The tree representing the gene hierarchical clustering has been omitted for clarity. The samples were labeled as: black - autonomous adenomas; red - sporadic PTC; white - Chernobyl PTC. **A**. With real gene expression data. **B**. With randomized data (for each gene, the sample labels have been permuted randomly).

4.2 K-means

Another classical clustering algorithm is K-means. A neural network version of K-means, self organizing maps (SOM), was used in some of the first microarray papers (*e.g.* Tamayo et al. 1999). The version described here is the basic K-means. It is possible to give a precise statistical meaning to the algorithm, however for the sake of simplicity the algorithm will simply be described and some of its properties highlighted.

The K-means algorithm searches for a certain number, *K*, of groups in the data. *K* must be chosen beforehand. For instance, it could be used to find two groups of samples in cancer data. The assumption underlying the algorithm is that in each of the groups, the objects are identical except for some random variations. Thus each object could be viewed as the sum of a prototype, which depends on the group, and some random noise.

If the group memberships were known, the prototypes could be estimated as the mean of the objects in the groups. If the prototypes were known, it would be possible to assign each object to the group whose prototype is closest. These two ideas form the basis of the K-means algorithm, which works as follow:

1. Each object is assigned to a random group.
2. Each group prototype is calculated as the mean of the objects in the group.
3. The objects are moved to the group whose prototype is closest.
4. If no convergence, back to 2.

An example of the K-means algorithm in action is presented in figure 9. The upper line represents one run of the algorithm. It starts from a certain random initialization. The prototypes are then calculated. They do not seem to be very different. However, when the group memberships are updated, a clear separation between the four objects on the left and the four objects on the right appears. The algorithm has converged, and the solution does seem reasonable.

The lower line of figure 9 shows what happens with a different initialization. In this case, there are more objects from the black group in the top rank. This has an influence on the prototypes, with the black prototype being much higher than the white prototype. After the group memberships are updated, another separation appears, with in one group the four objects of the top and in the other the four objects of the bottom. The algorithm has then converged. This shows that the K-means algorithm can be very dependent on its initialization. In this case, both clusterings are valid. The first one only seems better because the distances on the horizontal axis are larger than the distances on the vertical axis. If the two axes were put on a similar scale, it would be hard to decide which clustering to choose. This also highlights the importance of the normalization and re-scaling of data on the outcome of a clustering.

The K-means algorithm can be applied to discover groups of samples in the thyroid data. Searching for two groups leads to one group (N=11) containing nothing but papillary thyroid carcinomas (PTC), and the other (N=25) containing all autonomous adenomas (AA) plus 11 PTC. The first group corresponds to the group of tightly correlated PTC in the hierarchical clustering (figure 8). Using three groups lead to a one group containing all the AA but v6 plus the PTC p12, one group corresponding to the tight cluster of PTC and one group containing the remaining PTCs plus the AA v6. This fits the results obtained with the hierarchical clustering algorithm.

Hence both algorithms gave similar results, presented differently.

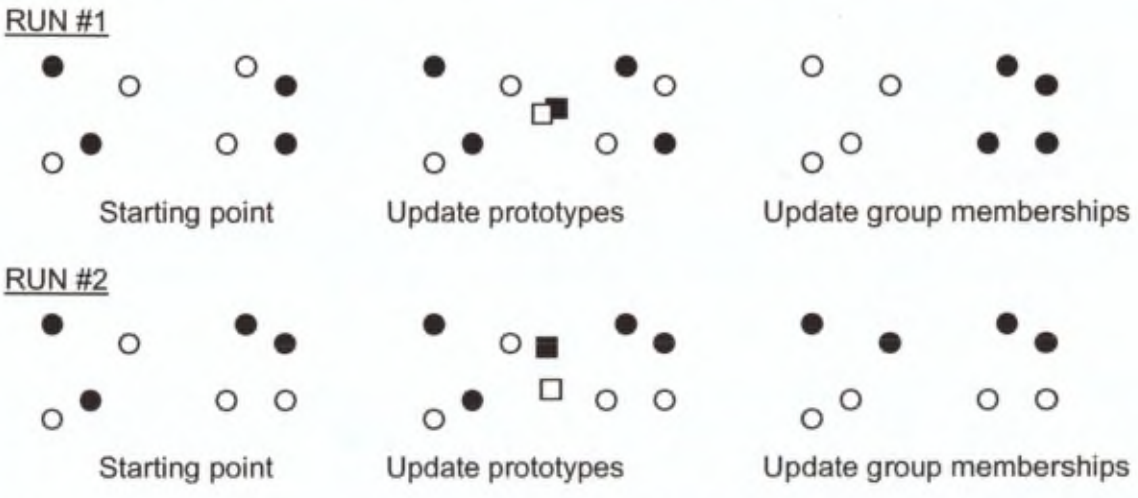


Figure 9. Example of a K-means algorithm in action. The black circles belong to one group, the white circles to the other. The squares are the group prototypes. Each line is a different run of the algorithm, starting from a different initialization.

5 Conclusion

The different high-throughput gene expression quantifying technologies essentially all generate the same type of data (figure 3). The resulting product is an array, with as many lines as there are genes and as many columns as there are samples measured. Typically, there are thousands to tens of thousands of genes measured, and only a dozen to a hundred samples.

Each value represents, often in some arbitrary unit, the expression of a gene in a sample. As the behavior of a cell is largely determined by the level of expression of the different genes in the cell, this gives a relatively comprehensive picture of what is happening inside the cell.

This picture can be used for many purposes. For instance, it could be used to determine the molecular difference, if any, between two samples. It could also be used to assess if a group of samples is homogeneous or not, *i.e.* it might lead to the identification of new subgroups in a known disease. It can also be used to identify co-regulated genes, such genes having often similar biological roles.

The amount of gene expression data for even a simple experiment is such that automated means of analysis must be created. This is what justifies this work.

6 References

Darnell, J., Lodish, H. and Baltimore, D. (1990) Molecular Cell Biology, 2nd edition. Scientific American Books.

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* **95**, 14863-8.

Jain, A.K., and Dubes, R.C. (1988) Algorithms for Clustering Data, Prentice Hall.

Lockhart, D.J., Dong, H., Byrne, M.H., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H. and Brown, E.L. 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology* **14,** 1675—1680.

Schena, M., Shalon, D., W. Davis, R.W. and Brown, P.O. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270,** 467—470.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S. and Golub, T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* **96**, 2907–2912.

Deleu, S., Allory, Y., Radulescu, A., Pirson, I., Carrasco, N., Corvilain, B., Salmon, I., Franc, B., Dumont, J.E., Van Sande, J. and Maenhaut, C. 2000. Characterization of autonomous thyroid adenoma: Metabolism, gene expression, and pathology. *Thyroid* **10**, 131-140.

1 - Determination of differentially expressed genes

1 Introduction

Microarrays are primarily used to screen genes in a high-throughput fashion. This is done by comparing groups of samples and assessing which genes are differentially expressed, that is have a different expression level from group to group. In the thyroid data set presented in the background chapter, this can correspond to the finding of genes having a different expression level in normal thyroids compared to autonomous adenomas, or in autonomous adenomas compared to papillary thyroid carcinomas.

Two different experimental setups are commonly used for microarray studies, which lead to different questions.

In the first setup, two samples belonging to two different groups are hybridized on each slide, with different labels. The results obtained are the ratios of the gene expressions in the two samples. In the thyroid data set, this corresponds to the comparison between normal and pathological tissue: on each slide, a pathological sample and the corresponding normal sample are compared. The goal is to assess whether those ratios are significantly different from unity.

In the second setup, each sample is hybridized on a different slide. If there are two groups of samples, then for each gene two sets of numbers are obtained. The goal is to assess whether the expression values differ significantly between the two sets. On the thyroid data set, this corresponds to the comparison between autonomous adenomas and papillary thyroid carcinomas.

The main impetus in our laboratory is on the comparison between pathological tissues and the corresponding normal tissues, as done with two-colors microarrays. This corresponds to the first setup. For this reason, most of the analyses in this chapter concern this case. However, the techniques presented can be readily generalized to the two-groups case, as is done at the end of this chapter. As the ratio distributions are skewed, the log of the ratios is taken. So the test is to assess if the log-ratios are significantly different from zero.

The discovery of differentially expressed genes has always been an important part of the microarrray technology, and was performed from the first experiments. In the first papers, genes were selected using a simple fold-change requirement. For instance, all genes whose average ratio was at least 2 were considered as differentially expressed. It is not clear with this method if the genes obtained are really differentially expressed or only the result of random variations in the data. Furthermore, it is unclear how to decide which fold-change is sufficient.

It would be useful to design a statistically sounder method to select differentially expressed genes. In statistics, a significance level is assigned to each result. This significance, or p-value, is the probability that the measures could be obtained by random variations of a null-hypothesis variable. For microarrays, the null-hypothesis is that the mean of the log-ratios measured for a gene is 0. The p-value is the probability that values drawn from such distribution could reproduce the measures. A low p-value (say < 1%) means that it is unlikely that the differential expression is due to chance alone.

Techniques have been developed for a century to calculate such significance levels (Gosset 1904, Gosset 1908), so this may seem to be a straightforward task. However, two difficulties complicate the matter: non-verified hypotheses of parametric tests, and multi-testing.

Classical parametric tests (*e.g.* Student t-test) can be used to assign significance levels. However, those tests are based on hypotheses, the most common being the normality

of the distribution. The importance of this hypothesis decreases with the number of replicates. Alas, in microarrays there are typically only a handful of experiments. A solution is to use non-parametric tests. Those tests however typically lack power, that is the ability to discern differences with a small number of replicates. Permutation techniques have been proposed as a mean to increase power while remaining non-parametric. In those techniques, a scoring function is used to judge the differential expression of the genes, like for instance the average fold-change. The null-hypothesis distribution of this scoring function is estimated by permutation of the samples, that is the group membership of some samples is changed. In our setup, this consists in the inversion of some of the ratios measured. These modifications have no influence on null-hypothesis genes. Hence, by repeatedly inverting different samples, an estimate of the null-hypothesis gene distribution can be obtained. The significance of the result is measured by counting the proportion of the permutations for which higher scores are obtained.

The other difficulty is that the p-values obtained must be corrected for multi-testing. If for instance 1000 null-hypothesis genes were assessed simultaneously, by definition on average 10 would have a significance level of 0.01 or better. Hence, the significances obtained using any statistical test must be modified. This classically is achieved using the Bonferroni correction, which multiplies the p-values by the number of tests performed. This correction is usually too stringent for gene expression data: if 10000 genes are tested, an original p-value of 10^{-6} is needed to reach a corrected p-value of 0.01. Also, it is very dependent on the number of tests done simultaneously. This can lead to non-desirable results. If 10% of the genes are measured with more precision than the remaining 90%, then the significances could be improved 10-fold simply by keeping only those 10%. However, that would mean that 90% of the results are discarded, even though they may contain useful information. A lot of valuable data could be discarded just to get more significant results on the rest. A more gradual technique, more stringent on the badly quantified spots and less stringent on the well-quantified spots should be more effective.

Those two difficulties are addressed in this chapter, and solutions are proposed to limit their influence.

Firstly, a permutation technique is presented which is used to estimate the null-hypothesis distribution. The p-values are then directly derived from this distribution. The limits and the advantages of the approach, compared to similar works in the literature, are highlighted. Different scoring functions are presented and compared.

Secondly, in agreement with recent literature, a different definition of the multi-testing corrected p-values is proposed: the false discovery rate (FDR). For a gene, this rate is the proportion of null-hypothesis genes which would be selected if this gene and all better genes were selected. It is shown that this rate has the advantage of allowing a coherent merging of data of different quality. A mean to calculate a local FDR, that is the probability for a given gene to be a null-hypothesis gene, is also proposed.

2 Determination of the p-values

2.1 Techniques proposed in the literature

2.1.1 Classical statistics

The classical parametric technique to assess if a set of numbers is statistically different from 0 is to use the Student t-test (Cui and Churchill, 2003). This test is based on the fact that for measures drawn from a normal distribution of mean 0 and any standard deviation the value (called the Z-score)

$$Z = \frac{|\mu|}{\sigma} \tag{2.1}$$

where μ is the mean of the measures and σ their standard deviation behaves like a Student distribution with N-1 degrees of freedom, where N is the number of samples (up to a multiplicative constant). The significance level is the probability to obtain a Z-score equal or higher than the Z-score measured, if the measures were drawn from a normal distribution of mean 0. This probability can be calculated explicitly.

The t-test is correct if the data have a normal distribution. It remains asymptotically correct for any distribution if the number of replicates tends to infinity. In microarrays, the error distribution is not normal and the number of replicates is small, so the p-values obtained cannot be trusted. Other, non-parametric, ways to determine the p-values must be devised.

On table 2.1 an example data set is shown. Three genes are quantified in four samples. The p-values calculated using the t-test on those genes show that the first could be considered as significantly different from 0 (p<5%) while the others cannot.

	Values in the samples				Significance levels – p-values		
	S1	S2	S3	S4	T-test	Perm 1	Perm 2
Gene 1	1	2	3	4	3%	12.5%	4.2%
Gene 2	-1	1	0	2	49.5%	75%	54.1%
Gene 3	0	1	1	0	18.2%	50%	33.3%

Table 2.1 Example of p-values calculation. Perm 1 are the p-values with the first permutation implementation (Dudoit), perm 2 with the second (Efron). Z-scores were used as scoring functions for the permutations techniques.

2.1.2 Permutation techniques

The permutation techniques generate a null-hypothesis data set from the original data set by permuting sample labels at random. Those techniques make assumptions about the data, like normality, superfluous.

Different implementations have been proposed.

The first implementation was proposed by Dudoit *et al.* (2002). Each gene is treated independently. For each gene, a Z-score (eq. 2.1) is calculated. The null-hypothesis distribution of the Z-scores is estimated by permutation of the group labels of the samples. In a 2-color experiment, this means that the sign of the values are inverted for some samples. Ideally, all permutations are performed.

For instance, the first gene in table 2.1 has a Z-score of 1.93. A first permutation might invert the value of the first sample, leading to the measures (-1, 2, 3, 4), which have a Z-score of 0.93, below the original Z-score. All the possible sets of samples are so permutated, and the Z-scores obtained are compared to the original Z-score. In this case, the original Z-score is the highest possible, so the number of Z-scores higher or equal to the original Z-score is two: the original order and the complete permutation (-1, -2, -3, -4). The total number of permutations is 16, so the p-value is 2/22 = 0.125. This is the lowest possible p-value with this technique.

Hence, a hypothesis-free test is obtained for each gene. The number of permutations limits the power of this test. The power of the test is its ability to reject (correctly) the null-hypothesis, that is its ability to obtain low p-values if the values are different from the null-hypothesis. With 10 samples, 2^{10} permutations are possible. Because the Z-score is symmetric if all values are inverted, only $2^9 = 512$ different permutations are effectively available. This means that the lowest possible p-value is only about 0.002 with 10 replicates. Although this might seem reasonable, the massive multi-testing increases the p-values (see lower) so practically this is often not sufficient.

A different implementation was proposed (Efron *et al.*, 2001, Storey and Tibshirani, 2003 and Tusher *et al.*, 2001) which improves the power at the cost of the hypothesis that all genes have a similar distribution. In this case, the null-hypothesis distribution is inferred from permutations on **all** genes. So if 1000 genes on 10 experiments are analyzed, the lowest possible p-value is 0.002 / 1000 = 2.10^{-6}. This large gain in power may prove useful. For the first gene on table 2.1, this lead to a p-value three time smaller, as expected.

The hypothesis that all genes have the same distribution is however not correct. In particular, the distribution of differentially expressed genes after permutation is not identical to the distribution of null-hypothesis genes. This can lead to under-evaluated p-values. In order to limit the influence of those genes, only the balanced permutations are used, that is permutations in which half of the samples are inverted. Pan (2003) has proposed a modified Z-score to address this issue. This modified score essentially stabilizes the variance, so a lot of power is lost in the process. We argue however that this problem is not very important, because its influence is only noticeable when a large proportion of the genes show large differential expression. In this case precise p-value estimates are useless. This point is further developed lower.

Another improvement proposed is to take a different scoring function than the Z-score. Jain *et al.* (2003) have proposed that a median-based Z-score could improve the robustness.

A weakness of the Z-score is that is relies on an estimate of the standard deviation, which is not precise with a limited number of replicates. This has lead to the use of modified Z-scores, whose general form could be written as

$$Z^* = \frac{|\mu|}{\sqrt{(1-k)\sigma^2 + k\sigma_p^2}} \qquad (2.2)$$

where σ^2 is the variance estimated from the gene measures and σ_p^2 is a prior on the variance. This prior is the expected value of the variance, before the gene measures are taken into account. It is estimated from the variance of all genes, or of genes similar to the gene of interest, so that a different prior can be used for each gene. The constant *k* gives the relative weight to the prior and the measured variance.

Modified Z-score like (2.2) have been used in a large number of studies (Baldi 2001, Efron *et al.*, 2001 and others). Baldi has shown that the equation (2.2) can be understood in a Bayesian framework if some prior information about the variance is available.

In conclusion, there are many different possible implementations of the permutation technique. There is no theoretical best, as this depends on the characteristics of the data.

2.1.3 Other techniques

Some other techniques have been proposed to estimate the p-values.

Cole *et al.* (2003) proposed a pattern analysis algorithm, which learns the conditions under which the genes should be considered as differentially expressed. The advantage of the method is that genes consistently regulated but with a large variance could be kept without penalty.

Dozmorov *et al.* (2003) proposed a method in which genes that seems to follow a null hypothesis distribution are taken as a reference. Other genes are then compared to this reference.

Those techniques are much more complex than the permutation methods proposed, for no obvious gain in power. They are also based on assumptions, which can have unexpected effects and may not hold in practice. For these reasons the technique developed in this thesis is based on the permutation method framework.

2.2 The permutation technique proposed

The technique proposed is loosely based on the SAM technique presented in Tusher *et al.* (2001). It uses a *scoring function* for the genes, like for instance the Z-score (eq. 2.1). The higher the scoring function, the most likely the gene is to be regulated. A significance level is assigned to the score obtained for each gene. This significance level is the probability to obtain a higher score from the null-hypothesis distribution.

An approximate of a null-hypothesis distribution is created by inverting the ratios in half of the slides. This manipulation should have no effect on the distribution of the non-differentially expressed genes, while it should prevent the differentially expressed genes from having a large score. If most genes are not differentially expressed, the resulting data set is close to the ideal null-hypothesis one. Many different data sets can be created by repeatedly drawing random permutations. The p-values for a gene with a score Z is the fraction of the permutation scores which are higher or equal to Z.

The proposed treatment is original in the following:

1. Different scoring functions (*e.g.* eqs. (2.1) and (2.2)) have been proposed. Those functions were compared to assess their efficiencies, firstly on simulated data to underscore their differences and then on the thyroid data set.
2. The permutation of differentially expressed genes leads to distribution far from the null-hypothesis distribution. The possible importance of this effect is assessed.
3. The permutations are performed on intensity windows. This allows the comparison of genes with genes which have a similar distribution.

2.3 The permutation technique may underestimate the FDR

The score distribution obtained by permutation on the differentially expressed genes is different from the score distribution obtained on non-differentially expressed genes. The importance of this fact is estimated in this section. The conclusion is that in most cases it should lead to small under-estimate of the p-values, which should have little practical influence. In the case where most genes are differentially expressed, the influence can be much larger but in this case we argue that the p-values are useless. Hence we conclude that trying to correct for this effect is not necessary.

There are some systematic differences between a permuted data set and the null-hypothesis data set. For example, let a data set have null-hypothesis and differentially expressed genes. Both types of genes have the same distribution, but the differentially expressed genes have a mean different from one.

After permutation, the expected mean of all genes is zero, differentially and non-differentially expressed genes alike. However, the distribution of the differentially expressed genes after permutation is very different from the null-hypothesis distribution: for a differentially expressed gene with a mean m, the non-permuted samples are taken from a distribution with a mean of m and the permuted samples from a distribution with a mean of $-m$. The distribution of the mean is not affected by this fact, but the expected variance of those permuted genes is increased by m^2.

Hence the variance of the differentially expressed genes after permutation is higher, and possibly much higher, than it is under the null hypothesis.

The impact of this effect depends on the scoring function. If the scoring function uses only the mean, then it is irrelevant and the permutation technique offers a very good estimate of a null hypothesis data set. However, many reasonable scoring functions use an estimate of the spread of the distribution, like the variance. If the Z-score (eq. 2.1) is used then the scores on the permuted data sets will be underestimated compared to the null hypothesis distribution. This will lead to underestimated p-values.

In the worst case, the score of the differentially expressed genes would be zero in the permuted data set. In this case, the p- values inferred by permutation analysis would be underestimated by a factor $(1-N/M)$, where N is the number of differentially expressed genes

and *M* is the total number of genes. This means that the error is proportional to the fraction of differentially expressed genes (*N*/*M*). If only a small fraction of the genes are differentially expressed, then the error remains small. If this is not the case, then the p-values obtained can be significantly under-evaluated. However, this case can be easily detected (very low p-values for many genes). Moreover, the need for a precise estimate of the significance levels in this case is not obvious. Too many genes would be deemed to be significantly differentially expressed for follow-up analysis, so the most useful feat for the biologist would be to rank the genes by significance order, which is still possible.

In conclusion, the permutation technique works perfectly for the non-differentially expressed genes as long as their distribution is symmetric. It also works for the differentially expressed genes if the scoring function uses only the mean. If the scoring function uses a variance estimate, it does not reflect the true null-hypothesis distribution. The divergence is proportional to the number of differentially expressed genes. It is expected that in most cases the fraction of differentially expressed genes remains small, and so error should remain small. For this reason, no corrections for this effect are attempted. The remaining cases where a large fraction of the genes are differentially expressed can be easily detected. In those cases, a precise estimate of the significance levels is useless.

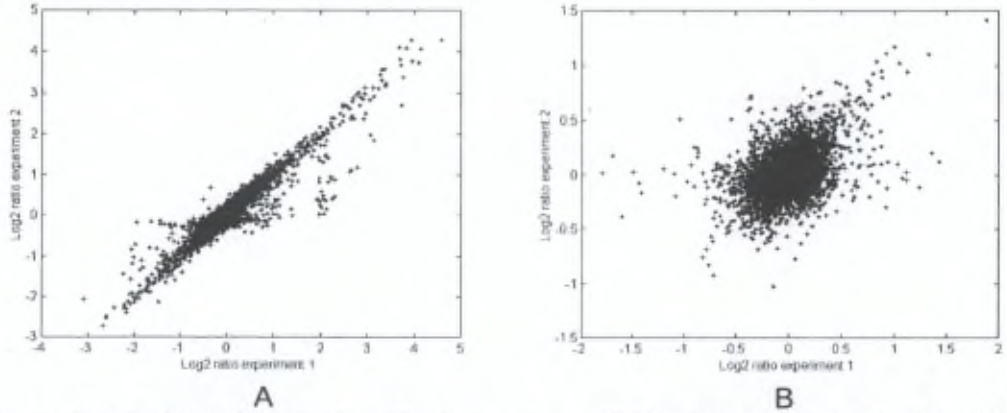
Furthermore, it will be shown in the second part of this chapter that the multi-testing issue introduces a similar overestimate of the p-values, so that the two effects can largely cancel each other.

2.4 Precision of expression measures is intensity-dependent

Precision of the measures is intensity-dependent: high intensity spots are more reproducible than low intensity spots (Yang *et al.*, 2002). Figures 2.1 A and B show an example of a correlation between replicates for high intensity and low intensity genes. The correlation is much lower for the low intensity genes (43% versus 94%).

To illustrate this differently, genes were sorted in function of their mean intensity across the samples, and the correlation was calculated on windows of intensity. For a given gene, the *N* genes with intensity just higher and the *N* genes with intensity just lower were kept. A correlation was calculated on those 2*N*+1 genes, giving a value for the gene of interest. The same was done for each gene. The results are shown figure 2.2. The correlation increases significantly with the gene intensities, and there is no clear cut-off. Low intensity genes are less reliable, but not to the point that they should be discarded. The decrease of the correlation for highly expressed genes is due to the saturation of the scanner.

To take into account this variation in function of the intensity, the permutation technique is applied on intensity windows.



A B

Figure 2.1 Scatter plot of the ratios from two replicated experiments. **A**. The 25% genes with the highest intensities. **B**. The 25% genes with the lowest intensities.

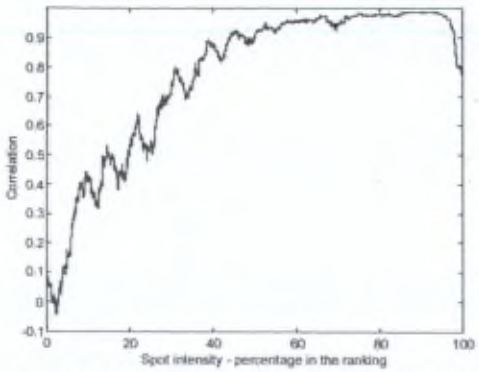


Figure 2.2 Correlation as a function of the gene intensities.

2.5 The scoring functions

The permutation technique compares the score of each gene with scores obtained from a null-hypothesis distribution. Different scoring functions based on different assumptions about the data are possible. The different scoring functions proposed are:

- Absolute mean value. This is a simple fold change requirement.
- Absolute median value. The theoretical advantage compared to the mean value is that it is less sensitive to outliers.
- Z-score (eq. 2.1). This is comparable to the Student t-test.
- Median divided by the standard deviation. This could be viewed as a more robust Z-score.
- Median divided by the average error. The average error is defined as $e = 1/N \sum_j |x_j - \mu|$. The average error is supposedly more robust to outliers than the variance.
- Bayesian corrected Z-score (eq. 2.2).

The relative value of those different tests can then be compared on a real data set using the framework presented.

Some care must be taken for the evaluation of the Bayesian corrected Z-score. In this test the variance of the whole set of genes is used in the scoring of each gene. As shown before, this variance is overestimated after permutation for differentially expressed genes. This means that all scores would be underestimated in the permutated data set compared to the real null-hypothesis. For this reason, the prior on the variance used in the permutation analysis is the original variance obtained on the non-permuted data.

2.6 A simulation check

A simulation was made to check the efficiency of the framework presented and to assess how it compares to parametric statistical methods. This simulation underscores the influence of the distribution of the values in the simulated data set on the effectiveness of the scoring functions.

Since the hypotheses taken for the creation of the simulated data set are crucial, three different sets were created. In the first two, the noise model was a normal distribution. In the first, each gene had the same variance while in the second the variance was allowed to vary from gene to gene. In the third data set, the noise model was a uniform distribution, to show that the permutation method still works in this case while the t-test loses its accuracy.

Each data set contains 10000 genes, 1000 differentially expressed (or positive) and 9000 null-hypothesis (or negative). 6 samples were created. For each test, all genes with $p < 0.05$ were kept. The number of differentially expressed genes selected determines the power of the test. The difference between the number of null-hypothesis genes selected and the expected number determines the accuracy of the test. A good test should be accurate and have a high power.

2.6.1 A data set where all genes have the same variance

For the first data set, the 9000 null-hypothesis genes were drawn from a normal distribution of mean 0 and standard deviation 1. The remaining 1000 positive genes were also drawn from normal distributions with standard deviation of 1, but their means were taken as random numbers between 0.5 and 1.5.

At a significance level of 5%, it would be expected that each test would select 9000 * 5% = 450 null-hypothesis genes.

Using the classical t-test, 953 genes were selected at the 5% level: 532 positives and 421 negatives. The t-test is, not surprisingly, accurate.

Using the permutation technique with the Z-score, 1015 genes were selected at the 5% level: 549 positives and 466 negatives. The permutation version of the t-test behave similarly to the parametric t-test, as expected. The permutation-based test selects a bit more genes than its classical counterpart because the positive genes do not have a null-hypothesis distribution after permutation, but the difference is lower than 10%, as expected. As all genes are used to estimate the null distribution, there is no noticeable loss of power. If each gene were treated separately, the lowest possible p-value would have been 0.03.

Using the permutation technique with the absolute mean scoring function, 1136 genes were selected: 659 positives and 477 negatives. Again, the accuracy is good. The power however is significantly improved compared to the t-test. This can be explained by the fact that all genes have the same variance, which is automatically taken into account by the permutation technique. The test behaved like a t-test where the variance is known beforehand.

In conclusion, both the classical t-test and the permutation-based tests were accurate in this case. The power of the different tests varied because some made better use of the characteristics of the data set.

2.6.2 A data set where the variance of the genes varies

To underline the difference between the t-test and the absolute mean scoring function, another data set was created, identical to the first except that the variance was different for each gene, varying between 0.1 and 2. In this case, the variance must be estimated independently for each gene so the t-test should perform better.

The classical t-test at p<5% selected 998 genes, 519 positives and 479 negatives. The permutation technique using the Z-score selected 1034 genes, 528 positives and 506 negatives. Again, the performances of both tests are similar.

In comparison, the permutation technique with the absolute mean scoring function selected only 540 genes, 100 positives and 440 negatives. The test remains accurate, but its power is much lower.

In conclusion, the permutation t-test behaves essentially like the classical t-test, even though it slightly underestimated the p-values (by less than 10%, as expected). The mean change scoring function had low power in this case.

2.6.3 A data set with a non-gaussian noise model

This third data set was similar to the first, except that the noise model was taken as a uniform distribution between –2 and 2.

The classical t-test selected 906 genes, 350 positives and 556 negatives. In this case, the p-values inferred from the classical test are incorrect, with a discrepancy of more than 20%. Using the permutation with the Z-score, 817 genes were selected, 315 positives and 502 negatives. The permutation-based test is more accurate than the parametric test.

In this case the parametric t-test did not work properly, while the permutation technique estimated correctly the number of false positives, up to the usual 10% correction.

2.6.4 Conclusion

In conclusion, the permutation technique gives results similar to the classical parametric techniques, if the assumptions on which the parametric techniques are based hold. The advantage of the permutation technique is that it works even if those assumptions are not fulfilled.

The t-test has less power than the simple mean fold-change difference if the variance is the same for each gene. In this case there is no point in calculating the variance from single genes, an estimate from the whole set is more precise. Hence, if for real data every gene has the same variance, then the permutation technique can be more powerful than the t-test.

The question is whether the variance of all genes is similar in microarray data. It is expected that the variance is the sum of a variance due to the technique and a variance due to biological variation. The variance due to the technique is probably somewhat constant for all genes at a similar intensity. The biological variance can probably vary a lot from gene to gene. Hence it could be that the use of the global variance information in addition to the local gene variation, like in the Bayesian corrected Z-score, could be the most effective setup. Those hypotheses must be confirmed on the actual data.

2.7 Comparison of different scoring functions on a real data set

There are many possible scoring functions. Those can be judged using two criteria: accuracy and power. As the permutation technique proved to be accurate with any scoring function when applied on simulated data, the power remains to decide which scoring function is the most appropriate for microarray data.

Two different ways to measure the power can be designed. Either the number of genes selected at a given p-value can be maximized, or, equivalently, the p-value for a given number of selected genes can be minimized. The later criterion is used in the following analysis. For each scoring function, a graph of the evolution of the p-values in function of the number of genes selected is obtained. The different curves can then be compared. The curve with the smallest p-values is the best one. If two curves intersected, the decision would be harder to make but this did not happen for the best scoring functions.

The results are finally compared to those obtained using a classical parametric method, to underline the advantage of the permutation method.

The data set used is the 14 samples comparison between autonomous adenomas and normal thyroids, taken from the thyroid data set. Autonomous adenomas are histologically quite similar to the normal thyroid, so the number of differentially expressed genes should remain low. 1000 permutations were used for the permutation analysis.

The distribution of the p-values as a function of the number of genes selected with the different tests is shown figure 2.3. The median divided by the average error is the worst performer. The mean is better than the median, whether it is divided by the standard deviation or not. The average error largely underperforms the standard deviation. The best scoring function is the Bayesian corrected Z-score. This fact supports the idea that the standard deviation of a gene can be partially inferred from other genes at a similar intensity.

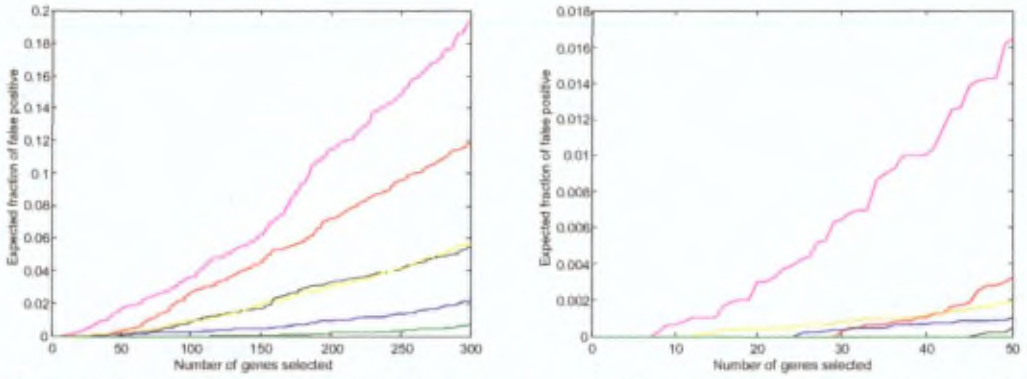


Figure 2.3 p-values estimated from the permutation test as a function of the number of genes selected, with different scoring functions. **A**. With 0-300 genes, **B**. Zoom of A, with 0-50 genes. The different colors correspond to the different tests: black: absolute mean value; red: absolute median value; blue: z-score; yellow: median divided by the standard deviation; magenta: median divided by the average error; green: Bayesian corrected Z-score (k=.5).

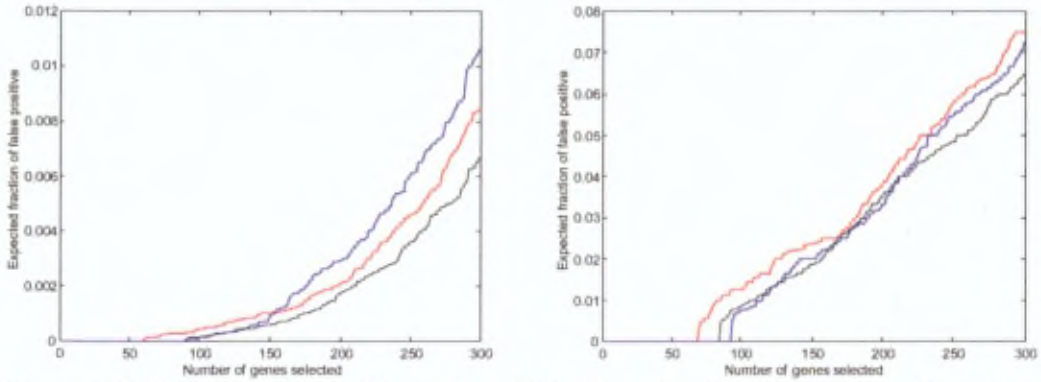


Figure 2.4 Comparison of different weights given to the prior in the Bayesian corrected Z-score. Red: k=0.25; black: k=0.5; blue: k=0.75. **A**. Using all 14 samples; **B**. Using only 6 samples.

Different prior weight, as defined by the parameter k in the Bayesian corrected Z-score (eq. 2.2) were compared (figure 2.4 A). The quality does not change much with the prior weight, and k=0.5 seems reasonnable. With that k, the weight given to the estimate using the gene values is identical to the weight given to the prior. The same comparison was performed using only 6 samples. With a lower number of samples, it would be expected that the estimate of the variance from the gene itself would worsen, hence a higher k should give better result. As can be seen figure 2.4 B, the results with a higher k (blue curve) are similar to those with k=0.5 (black curve), while they were notably worse in the previous case. However, the different weights given to the prior had little influence on the results within a large range, so that k=0.5 is a good trade-off in general.

In conclusion, the best scoring function with this data set is the Bayesian corrected Z-score. The power of the function is relatively constant for a wide range of the parameter k. Trying to make the scoring function more robust by using the median instead of the mean leads to disastrous results.

The results were then compared with the significance levels obtained using a parametric t-test (figure 2.5). The t-test gave lower estimates of the p-values compared to the permutation technique using the Z-score. This should be viewed in light of the results on simulated data sets, in which the permutation technique and the t-test gave similar results if the values where normally distributed. Hence the discrepancy between the two should be attributed to the non-normality of the log ratio distribution. This justifies the use of the permutation method instead of classical parametric techniques.

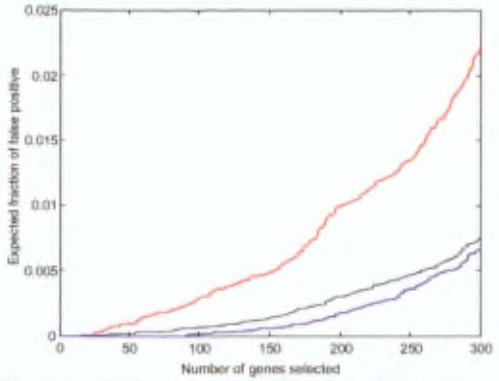


Figure 2.5 Comparison between parametric t-test and permutation technique. Red: permutation, Z-score; Black: parametric t-test; Blue: permutation, Bayesian corrected Z-score.

3 Correction for multi-testing

3.1 Introduction

The p-values as determined previously are designed for the case where only one gene is tested at a time. But microarrays are a case of massive multi-testing. This makes it likely that some genes have a small p-value just by chance. For instance, significance levels of 1% are obtained one time every 100 null-hypothesis genes on average. Genes with such significance levels should not necessarily be considered as differentially expressed. There are two main paths to correct the p-values for multi-testing.

The first is to keep the classical meaning of significance, and to correct the significance levels. The corrected p-values are the probabilities to select at least one null-hypothesis gene. Those p-values are called the family-wise error rate, or FWER (Dudoit *et al.*, 2002). Different corrections exist to control this FWER. The most well know is the Bonferroni correction, which multiplies every significance values by the number of tests. So if 1000 genes are tested, an original significance level of 10^{-5} is needed to reach a corrected significance level of 0.01. The Bonferroni correction is overly conservative, although the difference with more powerful schemes is not very large. With this correction the significance levels are a direct function of the number of genes tested. This is especially relevant if it is possible to know in advance that some genes are less reproducible than others. If low-quality genes are removed without testing, the significance level of the other genes improves. As high intensity genes are more reproducible than low intensity genes, an efficient way to improve the significance levels is to discard the data on the low intensity genes.

The second path to address the multi-testing issue is to redefine the significance. The most used redefined significance is the false discovery rate (Cui and Churchill, 2003, Efron *et al.*, 2001, Dudoit *et al.*, 2002, Storey and Tibshirani, 2003). For a gene with a given non-corrected p-value p, the false discovery rate (FDR) is

$$FDR(p) = \frac{Mp}{\sum_i P_i < p} \tag{3.1}$$

that is the expected number of null-hypothesis genes having a lower p-value, that is the product of the number of tests (M) by the significance level (p), divided by the effective number of genes having a lower p-value, that is the number of calculated p-values (P_i) lower than the significance level (p).

More intuitively, if a threshold is set on the p-values so that all genes with a p-value below this threshold are selected as differentially expressed, the FDR is the expected number of null hypothesis genes selected (false positive) divided by the total number of genes selected.

The FDR is more adapted to microarray data, since there are usually a large number of differentially expressed genes, which means that it is possible to have reasonable FDRs even though the Bonferroni-corrected significance levels are too large.

Another important advantage of the FDR is that it allows to set different criteria on badly and well quantified genes in a coherent manner. With FWER, lower quality data raises the p-values on the higher quality data, because the correction is applied on all genes simultaneously. If FDRs are calculated on windows of intensity, low intensity genes are not directly compared with high intensity genes. Hence independent FDRs are obtained on the different intensity windows, which can then be merged. This part of the work is original.

3.2 Correction to the real number of null-hypothesis genes

The definition of the FDR by the equation (3.1) is not accurate. The expected number of false positive is the product of the number of null-hypothesis genes (and not the total number of genes) by the significance level:

$$FDR^*(p) = \frac{(M-N)p}{\sum_i P_i < p} = \frac{(M-N)}{M} FDR \qquad (3.2)$$

where FDR^* is the true FDR and N is the number of null-hypothesis genes. The FDR calculated using (3.1) are overestimated by a factor $(M-N)/M$.

Storey and Tibshirani (2003) have proposed to estimate the number of null-hypothesis genes from the p-values distribution. This distribution is to sum of the distribution of differentially expressed genes, whose shape is arbitrary but should be concentrated on low p-values, and the distribution of null-hypothesis genes, which should be flat. Hence, the distribution of the highest p-values can be used to determine the number of null-hypothesis genes.

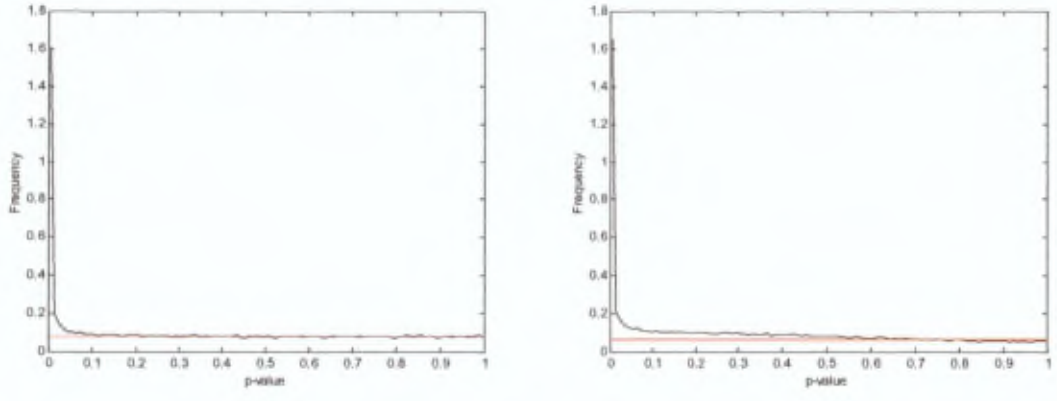
Figure 3.1 A shows an example on an artificial data set with 80% null-hypothesis genes. The p-values were calculated using the parametric t-test, which is exact on this data set. The distribution of the p-values is as expected: some genes with low p-values (the differentially expressed genes) on top of a background of null-hypothesis genes. The mean of the frequency for the p-values from 0.5 to 1 is 0.802, so the proportion of null-hypothesis genes is estimated at 80.2%, which is very close to the correct value.

This method is based on the availability of accurate p-values, especially at low significance. However, it has been shown before in this chapter (section 2.3) that the permutation method can lead to incorrect p-values, because differentially expressed genes have a very low score after permutation. This had a limited effect on genes having a low p-value, but can have a more dramatic effect on the genes having a high p-value. The score of those genes, even if very low by null-hypothesis gene standard, can still be large compared to the score of permuted differentially expressed genes. So large p-values are less frequent than expected.

Figure 3.1 B shows the distribution of the p-values on the same data set as figure 3.1 A, except that the p-values were calculated using the Z-score permutation method. The distribution at low p-values remains similar, but for higher p-values instead of being flat, the frequency decreases gradually. The number of null-hypothesis genes is estimated at 67% in this case, which is largely incorrect. Estimates of the number of null-hypothesis genes based on the shape of the distribution are unreliable if the permutation method is used to calculate the p-values.

A saving grace is that the permutation method tends to underestimate the p-values, by a factor of at most $(M-N)/M$, as shown before (section 2.3). Hence, if no correction is applied the two effects will to a certain extent cancel each other. The end result is that the non-corrected FDRs are higher than the real FDRs by a factor strictly inferior to $M/(M-N)$, which should remain small and on the conservative side.

In conclusion, the effect of the two inaccuracies which are difficult to correct – the estimate of the number of null-hypothesis genes and the permutation of differentially expressed genes – are close to cancel each other. For this reason, no attempt is made to correct for any of them.



A B

Figure 3.1 Histogram of the p-value distribution of an artificial exemple. The frequency scale has been chosen to have a mean of 1. In red is the mean of the proportion for the p-values from 0.5 to 1, used to estimate the number of null-hypothesis genes. **A.** Using the parametric t-test. **B.** Using the permutation method, with the Z-score.

3.3 FDR on intensity windows

3.3.1 Introduction

As stated before, the reproducibility (or quality) of the genes is function of their intensity. If the FDRs are calculated on all genes at the same time, the low quality genes will have an effect on the values obtained for the high-quality genes, increasing their apparent FDRs. The usual solution is to discard a significant proportion of the data, in order to increase the significance on the rest. However, those data can still contain interesting information. A more subtle and gradual approach would be preferable. The FDR allows such an approach in a natural and efficient manner.

The solution is to estimate FDR on intensity windows. That is, for each gene the N genes with the closest intensities are kept, and the analysis is performed on those genes. This way, the FDRs at high intensity are independent of the FDRs at low intensity. This leads to different stringencies at different gene qualities, as desired. The only difficulty is to merge the FDRs obtained at different intensities in a coherent fashion.

3.3.2 Merging FDRs on different windows

Results obtained on different windows must be compared. This is not straightforward, different criteria could be used. The most natural is to maximize the power of the test, that is to have the lowest possible FDR for a given number of genes selected (positives).

To illustrate the point, say the genes are separated in two different groups. Let the number of false positive (null-hypothesis genes selected) as a function of the number of positive be $FP_1(P_1)$ in the first group, $FP_2(P_2)$ in the second group. The total number of positive is $P=P_1+P_2$ and the total number of false positive is $FP=FP_1+FP_2$. The number of positive in both groups should be selected as to minimize the FDR, for a given number of positive. Hence, the function

$$FDR = \frac{FP_1 + FP_2}{P_1 + P_2}$$

should be minimized under the constraint that $P=P_1+P_2$. Using the Lagrange multipliers technique, the function

$$\frac{FP_1 + FP_2}{P_1 + P_2} + \lambda(P_1 + P_2)$$

must be minimized with respect to P_1, P_2 and λ. After some developments, the following relation is obtained:

$$\frac{dFP_1}{dP_1} = \frac{dFP_2}{dP_2} \tag{3.3}$$

The number of positive should be chosen so that the derivative of the number of false positive by the number of positive is the same in both groups. However, estimating those derivatives is a difficult task and to use such noisy values to choose thresholds is tricky.

It is possible to sidestep the problem by noting that the number of false positive is the product of the number of positive by the false discovery rate:

$$FP(P) = P \cdot FDR(P)$$

If the false discovery rate could be considered as varying slowly with P, the derivative $\frac{dFDR}{dP}$ could be neglected and eq. (3.3) simplifies to

$$FDR_1 = FDR_2 \tag{3.4}$$

so the condition would be to have identical FDRs in both groups, which is easy to ensure.

A more reasonable hypothesis is that the FDR is proportional to the number of positive genes, P. If this is the case, then $\frac{dFDR}{dP}$ can be considered as a constant. So, at a level of positive P, we have

$$FDR = P\frac{dFDR}{dP}$$

Putting that equation in equation (3.3)

$$\frac{dFP_1}{dP_1} = \frac{d(P_1 \cdot FDR_1(P_1))}{dP_1} = FDR_1 + P_1\frac{dFDR_1}{dP_1} = FDR_1 + P_1\frac{FDR_1}{P_1} = FDR_2 + P_2\frac{FDR_2}{P_2}$$

which also lead to eq. (3.4).

For those reasons, and for simplicity, the thresholds are chosen so that the FDRs are identical in all windows, even though this might not be optimal. With this criterion, the global FDR is the same than the local FDRs. In practice, the dependence between the FDR and the number of positive can always be linearized, but with the more complex equation

$$FDR = aP + b$$

With this form of dependence, eq. (3.4) does not guarantee an optimal choice anymore.

The criterion (3.4) only works if FDRs can be calculated on each window. If at a certain FDR positive are only found is some windows, the effect of the other windows has to be estimated. As different criteria are applied on the different windows, this cannot be done exactly. A conservative estimate must be calculated.

The first possibility is to estimate that as many false positive genes are selected in the other windows than in the windows of interest, but that no true positive are selected. This effectively amount to divide the FDR by the fraction of windows which have no FDR higher than the FDR of interest.

The second possibility is to take the best gene in each window for which normally no gene would be taken, and to modify the FDR accordingly.

As an example, say there are two intensity windows. The best gene in the first window has a FDR of 1%, and the best gene in the second window a FDR of 10%. Using the first possibility, the FDR would be corrected to 0.01 x 2 = 0.02. Using the second possibility, the FDR would be corrected to (0.01+0.1) / (1+1) = 0.55.

As both methods are conservative, the one giving the lowest estimate is used.

3.3.3 Estimate of the improvement

An estimate of the expected improvement is given here. In order to make this estimate, some assumptions must be made concerning the dependence between the FDR and the number of positive. In line with the previous section, two different windows are taken. In each window, the FDR is supposed to be proportional to the number of positive genes:

$$FDR_i(P) = a_i P_i \tag{3.5}$$

where a_i (i=1,2 is the window membership) is a constant and P_i the number of positive genes. The FDR can also be calculated directly from the definition

$$FDR_i(P_i) = \frac{p_i(P_i)N}{P_i} \tag{3.6}$$

where $p_i(P)$ is the significance level of the gene P and N is the number of null-hypothesis genes. This leads to a direct dependence between p_i and P_i

$$p_i = \frac{P_i^2 a_i}{N} \tag{3.7}$$

If the FDRs are calculated independently and then merged, the FDRs on the two groups are identical, so with eq. (3.4):

$$a_1 P_1 = a_2 P_2$$

So for a total number of positive $P=P_1+P_2$, the FDR using windows (FDR_w) is

$$FDR_w(P) = \frac{a_1 a_2 P}{a_1 + a_2} \tag{3.8}$$

If the FDR is calculated on the whole set of genes, a limit is set on the significance level of all genes simultaneously. Hence the significance level of the less significant gene in both group is the same, so using (3.7) the following equality can be written:

$$\frac{P_1^2 a_1}{N} = \frac{P_2^2 a_2}{N}$$

Introducing the total number of positive, the following second order equation in P_1 is obtained:

$$P_1^2(a_1 - a_2) + 2a_2 P P_1 - P^2 a_2 = 0$$

Solving the equation for P_1 leads to

$$P_1 = \frac{P\sqrt{a_2}}{\sqrt{a_1} + \sqrt{a_2}} \tag{3.9}$$

The FDR using all genes (FDR_a) can be calculated using its definition and (3.6), (3.9)

$$FDR_a(P) = \frac{(p_1 + p_2)N}{P} = \frac{2Pa_1 a_2}{\left(\sqrt{a_1} + \sqrt{a_2}\right)^2} \tag{3.10}$$

The gain (g) in FDR is the ratio of the two FDR, that is (3.10) divided by (3.8)

$$g = 2\frac{a_1 + a_2}{\left(\sqrt{a_1} + \sqrt{a_2}\right)^2} \tag{3.11}$$

As this is difficult to understand, say one of the group has a reproducibility ε as good as the other, that is $a_1 = a_2\varepsilon$. In this case, (3.11) simplifies to

$$g = 2\frac{1+\varepsilon}{\left(1+\sqrt{\varepsilon}\right)^2} \tag{3.12}$$

This function is plotted figure 3.2. With the assumptions taken, the merging of different windows has a large effect only if the difference of quality between them is quite large. In practice, this is usually the case for the most differentially expressed genes, but the

difference becomes tamer when the number of selected genes is increased. This means that the technique of the intensity windows can have a large effect, but only on a small subset of genes.

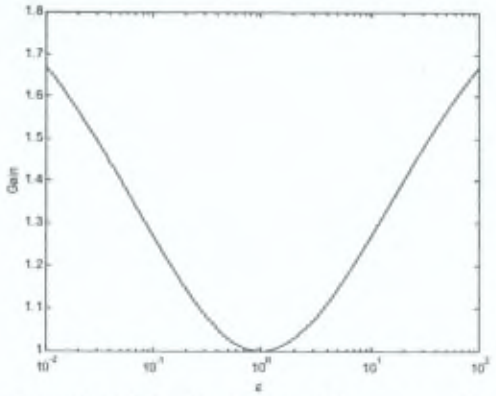


Figure 3.2 Effect of the difference of quality between the two windows (ε) on the gain in FDR obtained when using the intensity windows.

If the dependence between the FDR and the number of positive is set to the more realistic linear dependence $FDR(P)=a+bP$ (with $a<0$ and $b>0$), the calculation is much more complicated. For a given total false discovery rate f, the number of genes selected using the windowed method is

$$P_w = \frac{a_1 N}{f - b_1 N} + \frac{a_2 N}{f - b_2 N}$$

while the number of genes selected using all genes at once is (both the numerator and the denominator are negative)

$$P_t = \frac{2N(b_1 a_2 + b_2 a_1)}{(b_1 + b_2) f - 2N b_1 b_2}$$

The difference between the two is

$$\Delta P = \frac{f(b_1 - b_2)(f(a_2 - a_1) - N(b_2 a_1 - b_1 a_2))}{K > 0}$$

which is not a very readable expression. It is necessary to make a further approximation to make it understandable. In general, a_i seems to be roughly proportional to b_i: $a_i = -kb_i$. Using this approximation, the equation can be rewritten

$$\Delta P = \frac{f^2 k(a_1 - a_2)^2}{K > 0}$$

So there is also an improvement in this case. It is of course possible to find specific values of the parameters for which this does not hold, but in general the windowed method improves, or at least does not deteriorate, the power of the test.

3.3.4 An illustration of the importance of intensity windows

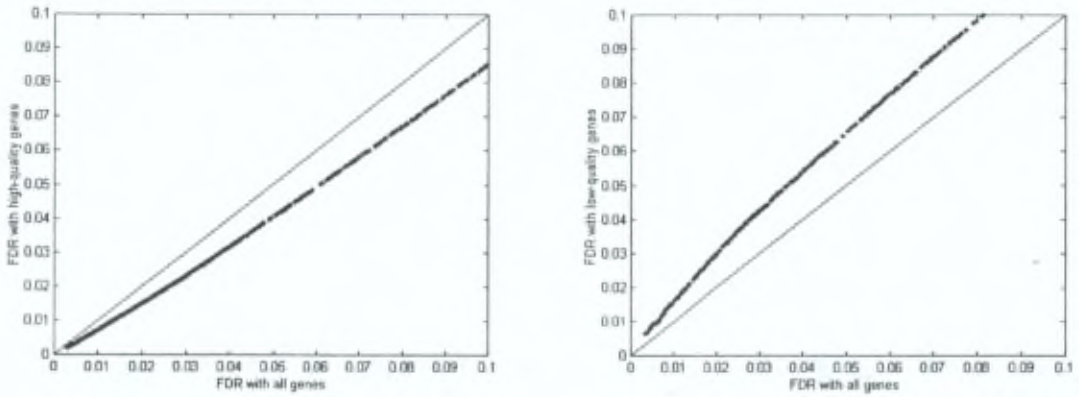
The influence of intensity windows on the FDR is illustrated on an artificial data set. This artificial data set consists of 100000 good quality genes, drawn from a normal distribution of standard deviation 0.8 and 100000 bad quality genes, drawn from a normal distribution of unity standard deviation. 10% of the genes in both groups were taken as differentially expressed, and a value randomly drawn between 0.5 and 1.5 was added to their values. 10 samples were simulated. Such a large number of genes were simulated to get stable estimates. The p-values were calculated using the parametric t-test, which is applicable in this case. Two windows were used, one containing only the good quality genes and the other only the bad quality genes.

On figure 3.3, the FDRs calculated using all genes or only genes in quality windows are compared. On figure 3.3 A, the FDRs for the differentially expressed good quality genes are represented. When those FDRs are calculated using only the good quality genes and not all genes, an improvement is noted: the FDRs are smaller. So without intensity windows an efficient way to improve the FDR is to discard bad quality data. On the bad quality genes (figure 3.3 B), the discarding of the good quality genes has the opposite effect of raising the FDR. This highlights the fact that calculating the FDR on intensity windows imposes different stringencies on the genes depending on their intrinsic reproducibility.

If the FDRs are calculated separately and then merged, as proposed, the power of the test is increased. It is possible to select more genes at a similar FDR or the same number of genes with a lower FDR. For instance, at a FDR of 1%, 1663 genes (1151 good quality, 512 bad quality) were selected with FDRs calculated on all genes at once, compared to 1850 genes (1654 good quality, 196 bad quality) with FDRs calculated on the two groups separately. In this case there is an improvement, although it is not very large.

Using a more stringent FDR raises the difference. At a FDR of 0.005, the difference is between 630 and 420 genes, a 50% improvement. At a FDR of 0.003, the difference is between 237 (among which 2 bad quality genes) and 15 genes, a more than ten-fold improvement. Using less stringent FDR lowers the differences. At a FDR of 0.05 the two methods behave similarly, the version on all genes selecting 2% more genes.

In this case, the difference in quality between the two windows was quite small, which explains the relatively small improvement.



A B

Figure 3.3 Comparison of the FDR calculated using all genes or only the genes from the group on **A** the 10000 differentially expressed good quality genes or **B** the 10000 differentially expressed bad quality genes (cropped).

3.3.5 Conclusion

The use of intensity windows FDR leads to a sizeable increase in the power of the tests for the most differentially expressed genes. With this technique, it is not necessary to discard low-quality data to improve inference on the high quality data anymore. The results obtained on data of different qualities are merged in a coherent manner. Other quality measures than the crude intensity could be used, like the one proposed in Wang *et al.* (2003). This improvement is another argument for the use of the FDR instead of the FWER for microarray data, as no similar correction could be applied with the FWER.

3.4 Going from global to local

The FDR of a gene gives an estimate of the fraction of null-hypothesis genes in the group comprising this gene and all better genes. However, it is not in general the probability for the gene to be a null-hypothesis gene. As by definition the other genes in the group are better than the gene of interest, this probability is higher than the average of the group, and so is higher than the FDR. For this reason, it would be useful to assign a probability of being a null-hypothesis gene (a local FDR) to each gene, instead of using aggregate values. This

point was raised in Efron *et al.* (2001), in which they used a binning technique. We propose here a continuous implementation of the same idea.

Let $FP(P)$ be the estimated total number of null-hypothesis genes selected (false positive) as a function of the total number of genes selected (positive). The probability $f(P)$ for a gene to be a false positive must be derived from this number. For the most significant gene the local FDR is the same as the total number of false positive, that is

$$f(1)=FP(1)$$

For the following genes, the local FDR is the incremental increase in the aggregate number of false positive, that is

$$f(i) = FP(i) - FP(i-1) \tag{3.13}$$

An artificial data set was created to illustrate the local FDR. This data set had 10000 genes and 6 samples. Each gene was drawn from a normal distribution of standard deviation of 1. 8000 genes had a mean of zero (null-hypothesis genes) and 2000 had a mean randomly drawn between 0.5 and 1.5 (differentially expressed gene). The scoring function was the absolute mean. The result of the calculation of $f(P)$ is shown figure 3.4 A, black curve. As can be seen, the differences $FP(i)$-$FP(i$-1) in eq. (3.13) lead to a very noisy estimate of $f(P)$, which cannot be used directly. A smoothing with a window of size 101 lead to a more reasonable-looking function, as shown figure 3.4 A, red curve. The local FDR is close to zero for the first genes, then increases steadily to finally settle at a constant level for the last 7000 genes. This level is 1.25 and not 1 because the values should be multiplied by the fraction of null-hypothesis genes (80%).

As an artificial data set was used, the real number of false positive is known, and can be compared with the estimate obtained. The local number of false positive for a gene was estimated by counting the number of false positive present in a window of 101 genes centered on the gene of interest. The results are shown on figure 3.4 B. The estimated local FDR multiplied by the fraction of null-hypothesis genes (blue curve) fits the measured number of false positive (red curve). Without the correction, the local FDR (black curve) is overestimated by a factor 1.25, as expected. On this example, the framework presented is effective, up to the correction for the fraction of null-hypothesis genes.

As stated before, if the permutation method does not give a perfect image of the null-hypothesis data set because the scoring function uses the variance, the correction for the fraction of null-hypothesis genes can be almost automatic. To illustrate what happens in this case, a similar artificial data set was used, except that the mean of the differentially expressed genes was taken between 0.5 and 5. The Z-score was used as the scoring function. The results are shown figure 3.5. As can be seen, for the first 2500 genes or so the estimate is correct: the two sources of inaccuracies cancel each other. For the following genes, this is not the case and the probability rises above one. This is of no concern because the p-values on those genes are useless. Hence, the two effects of the limited number of null-hypothesis genes effectively cancel each other if the Z-score is used.

To finally illustrate the method, it was applied on the thyroid autonomous adenomas data set, using the Bayesian corrected Z-score method. The results are shown figure 3.6. The first 245 genes could be chosen as positive with a very high confidence (local FDR<1%), but that for about 1000 genes there is a difference between the autonomous adenomas and the normal thyroids, although with a low confidence. This underlines the fact that given a sufficient number of replicates, many genes can be demonstrated as being differentially expressed. For the last 1200 genes or so, the probabilities are much higher than 1, probably because of the effect of the differentially expressed genes on the permutation.

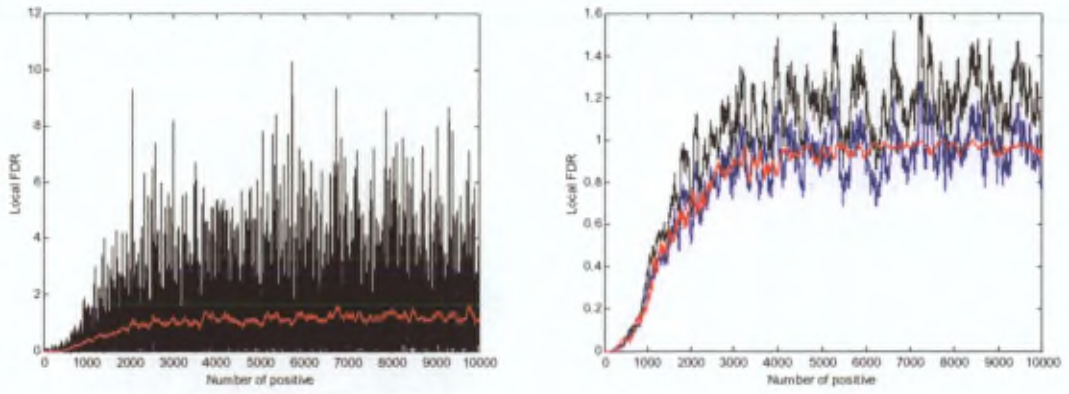


Figure 3.4 Estimate of the local FDR on an example artificial data set. **A** Comparison between a non-smoothed (black curve) and a smoothed (red curve) estimate. **B** Comparison between the estimated (black curve) and the measured (red curve) local FDR. Blue curve: estimated local FDR multiplied by the fraction of null-hypothesis genes.

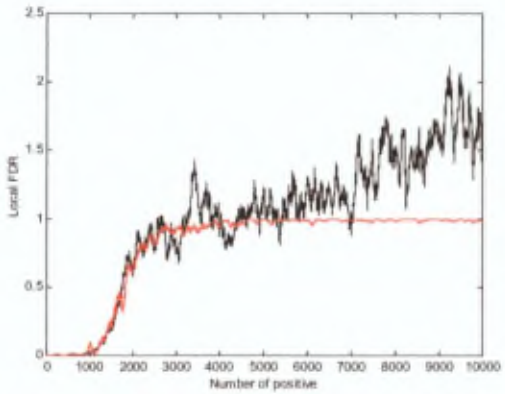


Figure 3.5 Illustration of the compensation of the two sources of inacurracies in the calculation of the local FDR, when using the Z-score permutation method. The estimated (black curve) and the measured (red curve) local FDR are compared.

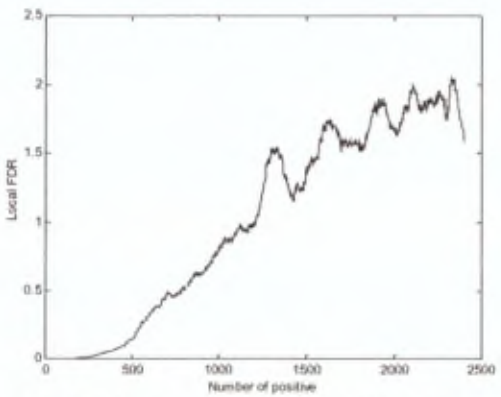


Figure 3.6 Estimated local FDR for the autonomous adenoma data set.

4 Application to the thyroid data set

The statistic framework described was applied on the thyroid data set. As this data set contains different groups, the following type of differentially expressed genes can be searched for:

1. Autonomous adenomas (AA) vs. normal thyroids.

2. Papillary thyroid carcinomas (PTC) vs. normal thyroids.
3. Autonomous adenomas vs. papillary thyroid carcinomas.
4. Sporadic PTC vs. Chernobyl PTC.

For the first two questions, the comparisons were made directly in the experiments, so the measured ratios should be compared to 1. This corresponds to what was presented so far in this chapter. For the last two questions, ratios obtained from two different experiments must be compared, so a different scoring function must be used which uses two groups instead of one. A modification of the technique to take this into account is presented below.

4.1 Single-group comparisons

The framework is applied as described previously on the AA and on the PTC data sets, to identify differentially expressed genes.

There are 2400 genes on the slides. 1000 permutations were randomly drawn.

The first comparison is between autonomous adenomas and the corresponding normal thyroids. 342 genes had a FDR below 1%, which is more than 10% of the genes. 170 of those had a FDR below 10^{-3}. The number of differentially expressed genes is large, probably too large to allow a follow-up analysis of each significant result. Autonomous adenomas are very different from normal thyroids, and the differences are reproducible.

The second comparison is between papillary thyroid carcinomas and the corresponding normal thyroids. 214 genes had a FDR below 1%, and 61 genes a FDR below 10^{-3}. The number of differentially expressed genes is lower than in the AA case, which is quite surprising as histologically PTCs are more different from normal thyroid than AAs are. The FDRs are higher because the PTCs are a less homogeneous group than the AAs. There are large differences between normal thyroids and PTCs, but those variations are less reproducible than for the AAs.

In both cases, the number of differentially expressed genes obtained is too large to be of any real use: any meaningful mean to select interesting genes, like for instance a simple fold-change requirement, would select significantly differentially expressed genes. The advantage of the statistical analysis is to validate the results, but it might make more sense to select interesting genes for follow-up analysis based on biologically motivated reasons than on the significance levels.

4.2 Two-group comparisons

The results from two groups of experiments are compared. Two such comparisons are performed: the first between AA and PTC, and the second between sporadic PTC and Chernobyl PTC.

For two-group comparisons, a two-group scoring function must be used. A natural choice is the value used in the calculation of the two-sample t-test with equal variance:

$$Z = \frac{|\mu_1 - \mu_2|}{\sqrt{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}}$$

where μ_1 is the mean of the gene if the first group, n_1 is the number of samples in the first group, σ_1^2 is the variance of the gene calculated in the first group and similarly for the second group.

The estimate of the variance can be improved with a prior, as in the one-group case:

$$Z = \frac{|\mu_1 - \mu_2|}{\sqrt{k\left((n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2\right) + (1 - k)\sigma_p^2}}$$

where σ_p^2 is the prior on the variance and k gives the relative importance of the prior and the estimate on the gene itself.

For the comparison between AAs and PTCs, 139 genes had a FDR below 1%, and 52 below 10^{-3}. This confirms that AAs and PTCs are different diseases.

The comparison between sporadic and Chernobyl PTCs lead to a different result. Using the prior on the variance, the lowest FDR found was 31%. So with this function, no difference between the two types of PTCs was found. Without the prior on the variance, the lowest FDR found was 2%, which is significant. Only one other gene had a FDR below 10%, at 7%. The other genes had a FDR over 20%. In this case, the prior hindered the discovery of differentially expressed genes, as the genes found had a very small variance. The finding of those two differentially expressed genes suggests that there is a difference between Chernobyl and sporadic PTC, albeit tenuous. In this case, the statistical analysis lead to significant results which would have been difficult to trust otherwise.

Two different biologists have performed the experiments on the PTCs. Applying the same statistical analysis to the grouping in function of this criterion, nine genes with a FDR below 10% were found, the lowest being 2%. This means that the choice of experimentalist had more influence on the results than the origin of the tumor. The fact that different persons dissected the two types of tumors might be the main reason for the differences observed between Chernobyl and sporadic PTCs. This means that no definitive biological conclusion could be drawn, except that the difference, if any, between the two types of PTCs is very faint. This however underlines the ability of the technique presented to detect and statistically validate small differences.

5 Conclusion

A framework for the determination of the significance level of the genes in a replicated microarray experiment has been presented. This framework allows for the determination of the false discovery rate in a robust fashion.

This work highlighted the advantages and disadvantages of different scoring functions. The choice of function depends largely on the distribution of the data. However, in the framework presented, accuracy of the p-values is kept even with an inferior scoring function, although power might be limited. The best overall function on the thyroid data set was the Z-score with a prior on the variance. It performed better than the other functions, and proved to be resilient to changes in the weight given to the prior within a large range.

It was shown that one of the advantages of working with the FDR instead of the family-wise error rate is that it is possible to use different stringencies for different genes. This is important for microarray data, as usually the reproducibility of the genes can be estimated using their intensity or a more complex quality criterion. Without the variation in stringency, the lower quality genes must be discarded in order to improve the p-values on the others. With the technique presented in this work, this is no more necessary. The lower quality genes can be incorporated without hurting the results on the better genes.

The main issue with this work is that the significance level is usually not what the biologists are interested in. The fact that a powerful algorithm can select 200 genes while another can only select 100 is usually not relevant. The number of differentially expressed genes is often too large to be treated effectively by biologists on a gene-by-gene basis. Also, biologists tend to prefer a gene with a large mean, even if it has a large variance, to a gene with a smaller mean with a low variance. The rationale is that a gene with a large differential expression is more likely to produce biologically interesting effects. This means that it might make sense to use the absolute mean instead of Z-score in order to select the genes most likely to interest the biologists.

The significance levels are also largely meaningless: the microarray technique is only a rough estimate of the reality. Many technical issues make it possible that what is measured is not what is expected. This means that even a large FDR (say 5%) can remain small

compared to the chance that the measures do not correspond to the gene. This is highlighted by the low level of confirmation obtained on microarray data. The quoted level of confirmation in the literature is usually 90% at most, and the non-confirmed genes cannot be explained by statistical variability alone. As long as the microarray remains what it is, elaborate statistical analysis will remain a largely useless feat.

The main interest in this statistical analysis is as a criterion to assess the efficiency of the data cleansing algorithms. Corrections of the data could be considered as positive if they improve the false positive rate. This fact is used in the next chapter to compare and assess different normalization algorithms.

6 References

Baldi, P., and Long, D.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.

Cole, S.W., Galic, Z. and Zack, J.A. (2003) Controlling false-negative errors in microarray differential expression analysis: a PRIM approach. *Bioinformatics*, **19**, 1808–1816.

Cui X, Churchill GA (2003) Statistical tests for differential expression in cDNA microarray experiments. *Genome Biol.*, **4**, 210.

Dozmorov, I. and Centola, M. (2003) An associative analysis of gene expression array data. *Bioinformatics*, **19**, 204-11.

Dudoit, S., Yang, Y.H., Callow, M.J. and Speed, T.P. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarrays experiments. *Statistica Sinica*, **12**, 111-139.

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001) Empirical Bayes Analysis of a Microarray Experiment. *J. Am. Stat. Assoc.*, **96**, 1151-60.

Gosset, W.S. (1904) The application of the law of error to the work of the Brewery. nota interna presso Guinness.

Gosset, W.S. (1908) The probable error of a mean. *Biometrika*, **6**, 1-25.

Jain, N., Thatte, J., Braciale, T., Ley, K., O'Connell, M. and Lee1, J.K. (2003) Local-pooled-error test for identifying differentially expressed genes with a small number of replicated microarrays. *Bioinformatics*, **19**, 1945-51.

Pan, W. (2002) A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics*, **18**, 546-54.

Pan, W. (2003) On the use of permutation in and the performance of a class of nonparametric methods to detect differential gene expression. *Bioinformatics*, **19**, 1333-40.

Storey, J.D. and Tibshirani, R. (2003) Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA*, **100**, 9440-5.

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.

Wang, X., Hessner, M.J., Wu, Y., Pati, N. and Ghosh, S. (2003) Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction. *Bioinformatics*, **19**, 1341-47.

Yang, I.V., Chen, E., Hasseman, J.P., Liang, W., Frank, B.C., Wang, S., Sharov, V., Saeed, A.I., White, J., Li, J., Lee, N.H., Yeatman, T.J. and Quackenbush, J. (2002) Within the fold: assessing differential expression measures and reproducibility in microarray assays. *Genome Biol.*, **3**, research0062.

2 - Data quality improvement

1 Introduction

Gene expression data can be improved by making the right technical choices in the data generation process and by removing specific biases.

Different choices can be made for the experiments and the quantification of the resulting images. There are many parameters which can be tuned in the quantification program, and a mean to choose the optimal set is needed.

Also, microarrray data must be corrected after quantification, as has been acknowledged from the first microarray papers (*e.g.* DeRisi, 1997). This is because the magnitude of the measurements is proportional to the amount of biological material and to the scanning gain. If for instance the volume of the sample is doubled, then roughly every value measured is multiplied by two. As those effects are artifacts of the technique and not an interesting signal, they are removed through a correction. For instance, the values of each experiment were divided by a constant, usually the mean or the median of the values of the experiments. More complex schemes to choose those constants, based on the fitting of the gene distributions, were also proposed (Zien *et al.*, 2001). Those types of corrections are called normalizations in the field.

Later, it appeared that more complex biases existed (Yang *et al.*, 2002), and methods were proposed to correct for them. The modifications made seemed to improve the quality of the data, as judged visually, but no quantitative measure of the improvements was made. The need for a criterion to judge the effectiveness of a normalization was only acknowledged recently (He *et al.*, 2003, Tsodikov *et al.*, 2002, Wang *et al.*, 2003), but no definitive quality criterion was found. Ideally, the criterion should give just one number, which would represent the quality of the data. However, such criteria are often biased by the fact that the quality of a spot is correlated with its intensity: brighter spots give better results. This means that a normalization procedure which amount to discard low intensity spots can seem very good for some data quality measures.

Anyway, the optimal choices are clearly platform-dependent. Hence, the different possibilities must be tested on the data generated in our laboratory in order to decide how to treat them.

This chapter focuses on this data quality improvement. It firstly describes two measures of data quality, which are more complex than those usually proposed but give a more faithful impression of what the normalizations achieve. The use of two different quality measures is important, as each of those can lead to biases. A normalization which improves only one measure should be regarded with suspicion. For the normalizations tested, only one lead to a discrepancy between the two measures. This discrepancy is explained by a shortcoming in one of those.

Secondly, different technical choices and normalizations were tested using the quality measures described. This testing was done on the autonomous adenomas part of the thyroid data set generated locally in our laboratory and described previously. This allowed the creation of a data correction protocol which could be applied systematically to all our data.

2 Assessing the effect of a modification

In order to compare different treatments of the data, a measure of the data quality should be defined. Ideally, such measure should give a single number, which could be used directly. This is the approach in the literature. However, we argue that such a criterion leads to biases which can, and does, lead to aberrant normalization choices.

The problems stem from the dependence between the quality of microarray data and the intensity of the spots: the higher the intensity, the better the quantification. If a global quality criterion is taken, like for instance the correlation calculated on the ratios, then an effective mean to improve the correlation is to lessen the importance of the low-intensity genes. This can be done for instance by adding a constant k to each value. With such modification, the ratios calculated on low-quality spots are close to unity and have little influence on the correlation. The quality measure is improved, although the accuracy of the data (*i.e.* how close it is to reality) is decreased. This example seems to be overly simplistic, however similar problems appear in the literature.

The quality of the data was assessed in the first papers using fairly general principles. Tseng *et al.* (2001) used only a biologically motivated criterion. Yang *et al.* (2002) assessed the quality of the data by seeing how it behaves in plots, and by analysis its distribution properties. They also verified the ability of the data to give the expected biological results. Those two criteria are not quantitative, and cannot be used for a systematic analysis of normalization procedures.

Different measures of data quality were proposed in later papers. Tsodikov *et al.* (2002) proposed to use a statistical test to rank the genes, and to measure how many of the N best genes (supposedly known) are among the N genes ranked first. This test cannot be used on real data, were the real ranking of the genes is not known. As the effectiveness of normalization is very dependent on the properties of the data, a test which can only be used on simulated data is of little value.

He *et al.* (2003) proposed two different quality measures, based on two different uses of microarrays. The first measure is based on gene screening: microarrays should be able to detect differential expression. So the criterion used is the number of false positive detected in an experiment divided by the total number of positive. The number of false positives is estimated by performing null-experiments. This criterion is global, which as explained before can lead to biases, and the use of real null-experiments can be costly. It must also be ensured that null-experiments are performed exactly like the real experiments. The second criterion is based on clustering: the discrepancies between the known clustering and the clustering obtained are recorded. This criterion is again based on the existence of a very peculiar data set. In most cases, it proved too insensitive to variation in the data to be used in practice.

Wang *et al.* (2003) proposed two different criteria, based on differentially expressed genes (DEGs). DEGs are defined as the 2.5% most extreme genes. The first criterion is the correlation coefficient between the DEGs on the different slides. The second is the concordance rate, which is the number of DEGs in common between two slides. Those two methods are again global. In this paper, this lead to a biased result. One of the normalization methods divides the log ratios by a function of the spot quality, so that the ratios of low quality spots are made closer to unity. This normalization leads to the best values of the two criteria. This however is to be expected, as the results from the worse spots are effectively discarded or at least largely tamed. Hence, those two criteria cannot be used to obtain unbiased data quality estimate.

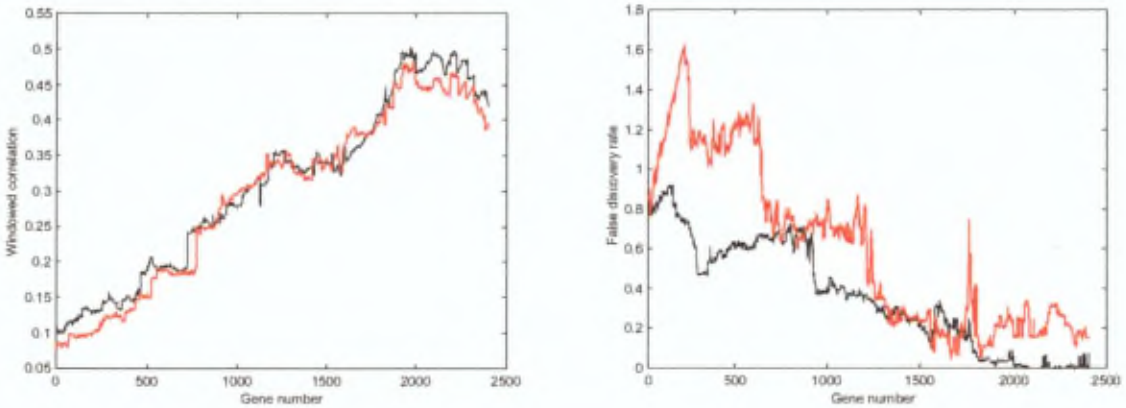
The data quality estimators proposed in the literature being insufficient, new criteria had to be designed. Two different schemes are proposed. Both give an evaluation of the data quality in function of the data intensity. Hence, a curve is obtained instead of a single number. This way, a more comprehensive picture of the data quality is obtained at the cost of having a quality curve and not just a number. A normalization which improves the data at only some intensities should be considered with suspicion.

The first criterion is based on correlation. The second is based on the number of genes selected as differentially expressed at a certain level (as defined in the previous chapter). The use of two unrelated method ensures that no bias is introduced. A good normalization method should produce improvements in both methods. One of the

normalization proposed leads to improvements with only one of the two criteria. This discrepancy is explained because of limitations in one of the criteria.

2.1 The windowed correlation method

Correlation can be used to quantify the reproducibility, but it should only compare genes of similar quality. For this reason, a windowed version is used. In this method, the genes are sorted in function of their intensity. For each gene, a window of 401 genes (the size is arbitrary), comprising the 200 genes with an intensity just higher and the 200 genes with an intensity just lower than the gene of interest are kept. The correlation of the logarithm of these ratios is then calculated. The end result is a curve which shows the correlation as a function of gene intensity (figure 2.1 A). The higher intensity genes (higher gene number) have a larger average correlation, of about 45% in this case. The correlation remains positive even for genes with a very low intensity, showing that information in present at all levels. In this case, the two normalizations shown seem to have similar quality, with only a slight advantage for the one represented in black.



A B

Figure 2.1. Example of (**A**) the windowed correlation method and (**B**) the false discovery rate method. Two different normalization are shown, in red and black. Genes with a higher gene number have a higher intensity.

2.2 The false-positive method

The technique designed in the previous chapter in order to determine the false discovery rate can be used to assess the reproducibility of the data. A good treatment of the data should lower the false discovery rate for a given number of positive, or should increase the number of positive at a given false discovery rate.

In the following, the ratio of positive is kept constant, at 10%. The false discovery rate is then calculated, using the permutation method described in the previous chapter. The scoring function used is the Bayesian correction Z-score, with k=0.5 and 100 permutations. The false discovery rate is calculated on intensity windows of 201 genes (the size is arbitrary). The result is a graph showing the false discover rate if the top 10% of the genes are considered as positive in function of the intensity.

As the false discovery rate is (as was shown before) an effective mean to merge information at different intensity levels, it would be possible obtain just one number describing the quality using this technique. However, to highlight the possible presence of biases and to visualize the data quality the intensity dependence is displayed.

This leads to curves like those shown figure 2.1 B. The false discovery rate is much lower at high intensity (high gene number) than at low intensity where it is close to 1. In this case, the normalization shown in red is of a much better quality than the one presented in black.

3 Merging of scans of different gains

3.1 Introduction

Genes are expressed at widely different levels in the sample studied. It is usually admitted that the most abundant genes are expressed at levels of about 10,000 mRNA copies per cell, while the least expressed genes are present at only a few copies per cell. Since the intensity of a spot is roughly proportional to the amount of mRNA in the sample, the intensities may vary by four orders of magnitude on a microarray. This is the same order as the dynamic range of all common scanners. Therefore, either the brighter spots saturate the detector, or the fainter spots are not quantified accurately.

Most scanners permit to choose their gain. This gain, the photomultiplicator voltage in the case of the Affymetrix 418 array scanner, controls their sensitivity. When this gain is too high, the fainter spots are visible but the brighter spots are saturated. When it is too low, the brighter spots are correctly quantified, but the fainter spots are invisible. The usual advise is to make sure that the brighter spots are expressed slightly below the saturation limit in order to avoid saturation while keeping the dynamic range as high as possible (Axon and Genome Systems sales representatives). This solution is cumbersome and error-prone - thousands of spots must be checked on each slide - and cannot be easily automated. Furthermore, with this scanning gain the dimmer spots are usually too weak to be accurately quantified.

The issue of the saturation in microarrays has been addressed in several publications. Different types of solutions have been proposed. The first methods proposed amounted to the detection of the saturated spots and their removal from the analysis (Wang *et al.* 2001, Hsiao *et al.* 2002). This was done either at an early stage or by analysis of their effect on data mining algorithms. However, while those methods can be effective in the removal of saturated spots, they do so at the expense of throwing away some information. They tried to correct a problem which should be avoided altogether.

We propose here a method to address the issue of saturation by using a combination of scans done with various gains. This combination allows a precise quantification of both the brighter and the fainter spots, by choosing for each spot the best scan available. The proposed method simply consists in the addition of a final step after the scanning and quantification. Furthermore, the technique is easy to automate, and could be incorporated directly in the scanning and quantification programs, making the process invisible to the user.

A similar method has been proposed recently (Dudley *et al.*, 2002). However, our method has several advantages which makes it at the same time mathematically sounder and more resilient to certain types of noise in the data.

3.2 Effect of the photomultiplicator gain

A slide was scanned twice, using two different photomultiplicator gains and the resulting images were quantified. As shown on fig. 3.1A, the values obtained with a high scanning gain were proportional to the values obtained with a lower gain as long as the scanner was not saturated. For saturated spots, the values at the highest gain remained around 60,000 intensity units, close to the maximum achievable by the scanner (65,535 i.u. - 16 bits). This means that, as expected, the quantified value of a spot is proportional to the amount of fluorescent material present in the spot, unless this spot is saturated. Hence, the value of the non-saturated measurement M_{ij} of the spot i in the scan j can be expressed as

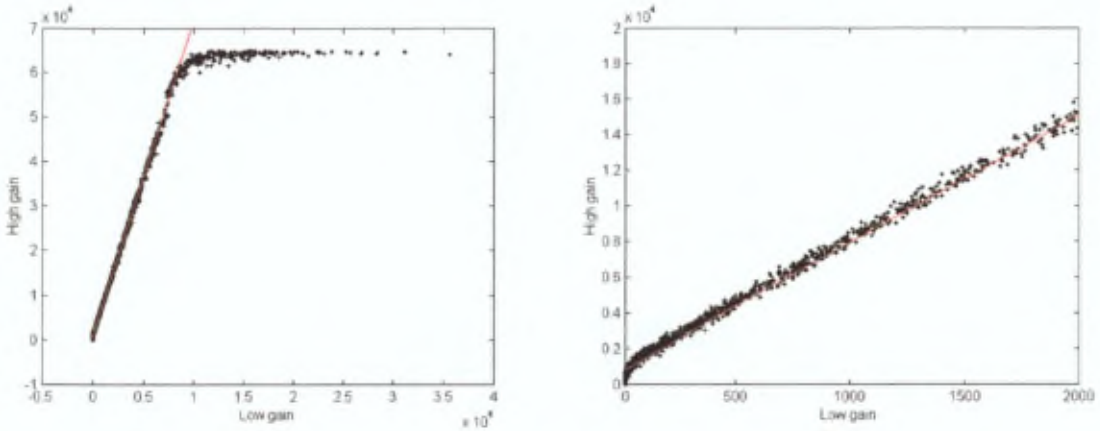
$$M_{ij} = v_i a_j \tag{3.1}$$

where v_i is the amount of material bound to the spot i and a_j the gain of the scan j.

This is valid for spots at relatively high intensities. However, experiments have shown that the results at one gain are not directly proportional to the results at another gain, but that a certain constant must be added (fig. 3.1B). Hence, equation (3.1) must be rewritten as

$$M_{ij} = v_i a_j + b_j \tag{3.2}$$

where b_j is the additive constant for the scan j. The origin of this constant is not clear. It is probably a result of the scanning process, as it seems to exist with the GMS scanner but not with the more recently acquired Axon scanner. This additive constant is problematic, as it suggests that additive constants should be used for the normalization, while most algorithms only take into account multiplicative constants. Practically, equation (3.2) is used. Most of the time, the constant b_j proved to be relatively small, of the order of a few hundreds on a scale of 64,000.



A B

Figure 3.1. A. Effect of the photomultiplicator gain on the quantified values. Each point represents a spot quantified with two different gains. The red line is the linear regression on the linear part. **B**. is a zoom on the lower left part.

3.3 Merging the scans

In order to precisely quantify a spot, the gain of the photomultiplicator should be chosen such that the intensity of the spot lies at a reasonable level, *i.e.* a level as high as possible but below saturation. This optimal gain is different for each spot, making the simultaneous quantification of all spots on a microarray in one scan problematic. For any gain chosen, some spots are far from their optimal condition. The goal of the technique presented here is to quantify most spots at gains close to their optimum.

The method can be described as follows: A slide is scanned a few times using different gains. The gains should vary from very low (to ensure that no spot is saturated) to very high (to clearly see the fainter spots). As demonstrated, the measured values are a function of the amount of fluorescent material bound to the spot as long as there is no saturation (eq. 3.2). The actual value of each spot is the amount of material bound v_i. These real values, along with the scanning gains a_j and the biases b_j, can be evaluated from the measurements M_{ij}. This is done by minimizing a least squares error criterion on the set of non-saturated spots *NS*:

$$\sum_{i,j \in NS} \left(M_{ij} - v_i a_j - b_j \right)^2 \tag{3.3}$$

There is an indetermination in this minimization, the values of the gains being given relatively to some unknown base state. In order to lift this indetermination, the value of a_1 is set to 1 and the value of b_1 is set to 0. This means that the values of v_i have the same magnitude as the values of the first scan. This is of course arbitrary, the measures being only defined up to a multiplicative and an additive constant. The data must be normalized, for instance by dividing them by the mean of the values, or by using some other normalizing scheme as those described later.

The main difficulty is to determine which spots should be considered as saturated. Taking ideas from robust regression, non-saturated the spots are those for which

$$M_{ij} > v_i a_j + b_j - 8e_j \tag{3.4}$$

where e_j is the median error for the scan j:

$$e_j = \underset{i,j \in NS}{\text{median}}\left(\left|v_i a_j + b_j - M_{ij}\right|\right) \tag{3.5}$$

The idea is that values which are at least 8 times the median error lower than the predicted value are statistically unlikely to be due to chance. This gives a test to determine which spots are saturated.

A problem with the equation (3.4) is that it is non-symmetric, that is only outliers on one side are considered. This can lead to biases, especially for the determination of **v**. In order to avoid such biases, two different sets of non-saturated spots are defined, *NS1* and *NS2*. A spot belongs to *NS1* if (2.4) holds. A spot belongs to *NS2* if

$$\left|M_{ij} - v_i a_j - b_j\right| < 6e_j \tag{3.6}$$

This set *NS2* is used for the robust determination of **v**. As precision is needed and a lot of points are available compared to the determination of **a** and **b**, the criterion (3.6) can be made more stringent than (3.4), hence the difference in the factor.

The determination of **a**, **b**, **v**, *NS1* and *NS2* is done iteratively:

1. *NS1* and *NS2* are initialized as the whole set of spots, *i.e.* all spots are considered as non-saturated. v_i is initialized to the scan value, that is $v_i = M_{i1}$.
2. (3.3) is minimized for ***a*** and **b** using the current ***v*** and *NS1*.
3. (3.3) is minimized for ***v*** using the current ***a*** and **b** *NS2*.
4. If no convergence on **a**, **b** and **v**, back to step 2.
5. ***e*** is calculated using (3.5).
6. *NS1* and *NS2* are the sets of spots for which (3.5) or (3.6) holds, respectively.
7. If no convergence, or too many iterations, back to step 2.

As for other robust evaluation scheme, the algorithm can cycle if some spots are on the verge of saturation. Hence, the algorithm must be stopped after a certain number of iterations even if convergence is not obtained. We found that the fact that the algorithm cannot determine if some borderline spots are saturated has very little influence on the results.

3.4 Influence of the method on reproducibility

In order to assess the effectiveness of the method, a set of two duplicated experiments were taken from the thyroid data set. Only two experiments were taken in order to highlight the effect of the technique. The slides corresponding to those experiments where scanned at three gains: one as would have been chosen normally (the medium gain), one much lower and one much higher. The data quality was assessed at those gains using the two measures described before. It was then compared to the data obtained by merging the scans using the technique described.

The influence of the method on the windowed correlation between two duplicated experiments is shown figure 3.2A. As expected, for the high intensity genes, the correlation is better with a low scanning gain (red curve), which avoids saturation. The last 700 genes show decreased correlations in the case of the highest scanning gain, and the last 200 in the case of the medium scanning gain. For the 1500 genes with the lowest intensities, the correlation is lower when a low scanning gain is chosen. For some unknown reason, the best choice here seems to be an intermediate scanning gain, but the results with the two higher gains are quite similar. The results after merging the three gains are shown on figure 2.2A as the black curve. This curve behaves like the low gain curve (red) for high intensity genes, like the average gain curve (green) for average intensity genes and like the high gain curve (blue) for low intensity genes. This means that every spot is quantified like if it has been scanned at a gain close to its optimum.

The influence of the method on the false discovery rate on the same data is shown figure 3.2B. The plot has been smoothed using the Matlab (Matworks inc., MA) function

`idfilt` for visualization purpose. With the lowest gain (red curve), the 1000 genes with the lowest expression have much higher false discovery rate than what could be obtained with a better gain. Would this scan be effectively used, 300 genes would not even be measured above background and would have to be thrown away. For a large part of those low intensity genes, the false discovery rate can be reduced to around 25% by using the highest scanning gain. For highly expressed genes, the higher scanning gains lead to saturation on the last 600 genes in the case of the highest gain, and on the last 200 genes in the case of the average gain. With a false discovery rate of about 10%, at least 10% of the data should be thrown away because of saturation. With the merging of the three scans (black line), the quantification is done with the highest gain for the dim spots, and the lowest gain for the bright spots. This leads to a precise quantification on the whole range of intensity, usually with a quality close to the best available. The false discovery rate is reduced to less than 25% for most of the spectrum, which means that usable information is available at almost all intensity levels.

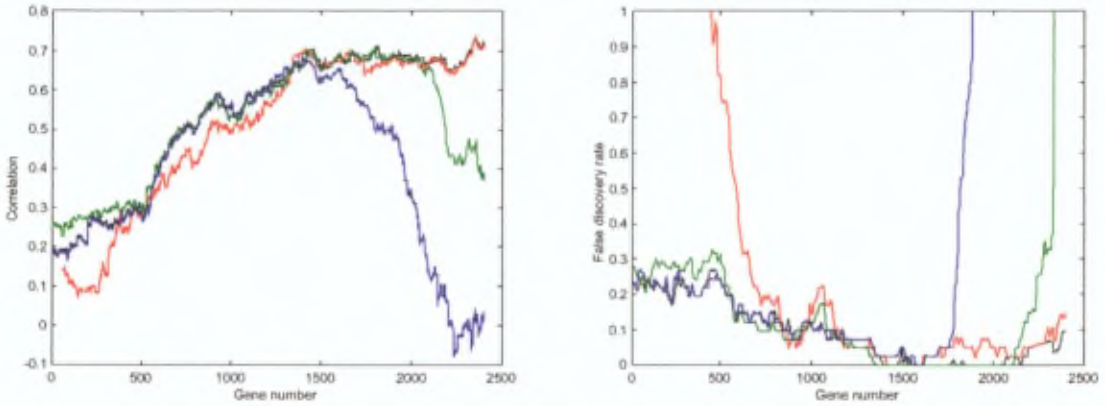


Figure 3.2. (**A**) Correlation and (**B**) false discovery rate in function of the relative gene intensity. Genes with a higher gene number have a higher intensity. Red: low gain. Green: medium gain. Blue: high gain. Black: merged results.

3.5 Comparison with the "masliner" method

Dudley *et al.* (2002) have proposed a comparable method, "masliner", to merge the values obtained by scans at different gains. However, our method is different from theirs in the following aspects:

1. The masliner method proper uses two scans. With three or more scans, it is proposed to merge the two lowest scans, and then to merge that result with the next lowest scan and so on. This is numerically less efficient than the least squares criterion (3.3) that we use.
2. The spots are considered in the masliner method as being saturated if their intensity is higher than a threshold, 40,000 being the suggested value. However, our experience have shown that many spots with intensity higher than such threshold are not saturated, so that valuable information would be discarded, and that spots with low intensities can be saturated. This can happen if only a small part of the spot is saturated (when a spot is very inhomogeneous) or if the spot is very saturated. In that case, the spot may contaminate the background, whose intensity can get quite high. Subtracting the high intensity of the background from the high intensity of the spot can lead to a small resulting intensity. On the web site with the supplementary information for Dudley *et al.* (http:/arep.med.harvard.edu/masliner/supplement.htm) that problem is acknowledged, although no solutions are given.

For those reasons, we believe our technique is more robust and numerically sounder.

In practice, the results given by both methods are similar, except if some genes are saturated at a low intensity. In that case, the error on those genes can be much larger with the masliner method than with ours.

3.6 Conclusion

A method which dramatically improves the dynamical range of a scanner was presented. This method permits precise quantification of all spots on a microarray slide with scanning gains close to optimum. This effectively solves the problem of saturation in the quantification of DNA microarrays as long as saturation is due to the scanner. When saturation is caused by something else (*e.g.* chemical saturation, quenching...) the method proposed is of course useless, and another solution should be found.

The precise numerical improvements given are specific to the particular experiments performed here. The correlation and the false discovery rate obtained are functions of the experimental setup, as are the importance of saturation and of weak spots. Nevertheless, the general trend remains constant: with a high scanning gain, the brighter spots are saturated. With a low scanning gain, the fainter spots are not quantified accurately. Our technique usually offers results close to the best possible choice of scanning gains for every spot on the array.

The algorithm has been implemented as a stand-alone application. This application, a MatLab (Matworks inc, MA) script, must be run after the quantification of the scans in order to merge the data.

The method proposed can be integrated seamlessly into the actual software packages, with minimal effort. The scans of a slide with multiple gains could be run automatically, and saved as a multi-layer tiff image by the scanning software. Such composite image can be quantified in a manner similar to regular tiff images by the quantification software. The same software can then do the fusion of the data immediately. This would improve the results in a way totally transparent to the user, except for longer scanning and quantification times.

4 Effect of color-flip

In the experiments performed in our laboratory a control sample (*e.g.* a normal thyroid) is labeled with one fluorophore (*i.e.* Cy3) while a non-control sample (*e.g.* an autonomous adenoma) is labeled with another fluorophore (*i.e.* Cy5), both being then hybridized on the same slide. The value of interest is the relative expression of the genes in both samples. It is useful to determine which genes are differentially expressed in the non-control sample compared to the corresponding control sample. A possibility however is that in addition to the difference in the samples, the difference in the fluorophore could have a substantial effect. This could be due for instance to structural differences in the chemical structure (steric hindrance). For instance, a gene could have a systematically higher fluorescence with one fluorophore than with the other. This means that the result of an experiment with the control sample labeled with Cy3 and the non-control sample labeled with Cy5 could be different from an experiment where the two fluorophores are inverted (which is called a color-flip experiment).

From the autonomous adenomas (AA) part of the thyroid data set, there are 39 slides. Those slides can be divided in two groups: a first group of 19 slides with the normal tissue labeled with Cy3 and the AA with Cy5, and a second group of 20 slides with the normal tissue labeled with Cy5 and the AA with Cy3. The correlations between those experiments were calculated.

The average correlation (the results are summarized table 4.1) between the 19 slides in the first group was 39% and 27% for the 20 slides in the second group. The correlation between the slides of the first and the second group was only 22% on average. However, the correlation can vary widely from slide to slide so this difference is very significant but not as significant as the difference between the correlation in the first and the second group.

In order to correct for the slide-to-slide variability, the correlation matrix was modified as follow. Firstly, the mean correlation was subtracted from each value. Then, the elements of the matrix were modified as follow:

$$cc^{*}_{ij} = cc_{ij} - \sum_{k} cc_{kj} - \sum_{k} cc_{ik}$$

With this modification, the mean of the corrected correlations for each slide is zero. This way, the corrected correlations are the comparisons between the correlation expected from the estimate of the slide quality and the actual correlation.

Using those modified correlation, the average correlation between the slides of the first group was 3.3%, between the slide of the second group 4.5% and the average correlation between the slides of the first and the second group was –7%. The difference between the first intra-group correlations is not significant (P=0.34), while the difference between the intra-group correlations and the cross-group correlations was extremely significant ($P<10^{-16}$).

In conclusion, there is a bias due to the fluorophore. For this reason, each experiment is performed at least twice, once with the control labeled with one fluorophore and once with the control labeled with the other fluorophore. The ratio of the two is the correct ratio squared.

	Group 1	Group 2
Group 1	39%	22%
Group 2		27%

Before correction

	Group 1	Group 2
Group 1	3.3%	-7%
Group 2		4.5%

After correction

Table 4.1 Average correlation between the slides intra-group and inter-group. Group 1: normal labeled with Cy3; group 2: normal labeled with Cy5.

5 Background correction

The spots are quantified using a program which detects them and measure their average intensity (mean or median). It has appeared however that some background signal was present everywhere on the slide, even where there are no spot. It has been proposed that this background part of the signal is an additive error, which can be inferred from the signal measured outside of the spot. It is not clear, however, if this hypothesis is correct. The chemical properties of the slide are very different from the chemical properties of the spots. Hence it is not possible to decide *a priori* if the subtraction of the background values from the spot value improves the quality of the measurements.

The only way to decide if the background should be subtracted from the spot intensity measurements is to check the results on real data. The conclusions are of course limited to the data produced locally, with the local protocol, scanner and quantification program.

A first empirical reason seems to point to the usefulness of the background removal, at least with an Axon scanner. The same slide was scanned twice with this scanner, at two different scanning gains. The scatter plots of the quantification at low gain versus the quantification at high gain are shown figure 5.1. On figure 5.1A, no background correction is applied. The dimmer spots have a positive value. The best-fit line, shown in red, does not cross the origin, so a constant dependent on the scanning gain is added to the values. On figure 5.1B, the background is subtracted. The dimmer spots have now an intensity of about zero. Moreover, the best-fit line passes almost exactly through the origin. Hence the scanning gain acts only as a multiplicative effect in this case, as it should. So the behavior of the scanner seems more reasonable with background subtraction.

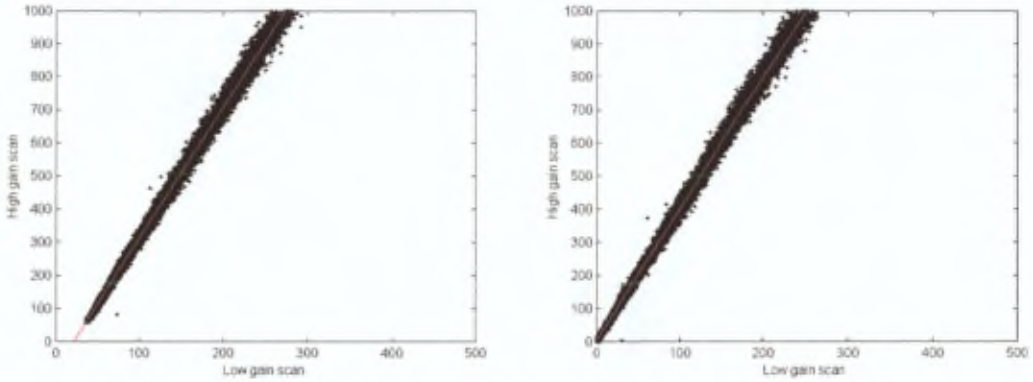
The background subtraction is a typical case in which the criterion used to assess the quality of the modification is extremely important. The subtraction of the background decreases the measured values. Hence for low intensity genes, the variability on the ratios tends to be increased. If a global correlation is used as the quality criterion, then this increase in variability would be the most important effect, and the background subtraction would seem to decrease the quality of the data.

The effect of background subtraction was assessed using the two criteria presented before. The results for the windowed correlation are shown figure 5.2A. The correlation does not change much with the background subtraction. However, if anything holds then the subtraction of the background improves the correlation. Among the 45 possible pair-wise correlation, in 31 cases the median correlation improved while in 14 cases it decreased. This

is significant at the 1% level. Hence correlation-wise, the subtraction of the background seems beneficiary.

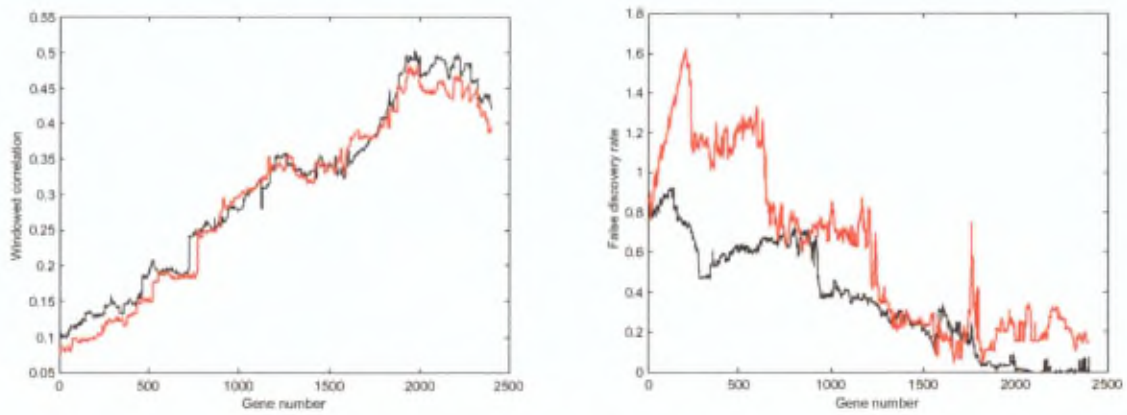
The results for the false discovery rate are shown figure 5.2B. The differences are much more dramatic using this quality measure. In this case, at about any intensity level, the background subtraction notably improves the reproducibility.

In conclusion, the subtraction of the background has a positive effect on our data, in both the correlation and the false discovery rate tests. The background is subtracted in all subsequent analyses.



A B

Figure 5.1. Scatter plot of low photomultiplicator gain scans versus high gain scans, using an Axon scanner. The parts shown are zooms on the low-intensity values. **A**. Without background correction. **B**. With background correction.



A B

Figure 5.2. A. Windowed correlation and **B**. false positive rate in function of the gene intensity. Red: without background correction; black: with background correction. Genes with a higher gene number have a higher intensity.

6 Normalization

The microarray technology leads to measurements which have certain biases, or systematic errors. The removal of such errors is called normalization in the field. In the following, techniques for the removal of two different types of biases are presented.

6.1 Spatial dependence

It can happen that the ratios are not evenly distributed on the slide. For instance, the signal can be higher in the green channel than in the red channel on the left side of the slide, and the other way around on the other part of the slide. See figure 6.1A. for an example.

The main normalization proposed for this effect is the one from Yang *et al.* (2002). They normalize separately in each sub-array. Because their arrays are divided in many smaller arrays this can be used as an approximation of spatial dependence. In our case, the arrays are only divided in 4 quarters and the effect does not seem to follow the limit of sub-arrays. So, a new technique had to be designed.

Different means could be used to remove this dependence. The technique which was implemented is to calculate the median ratio in a neighborhood around each spot. The intensity of one of the channel is then multiplied by the square root of this median and the intensity in the other channel is divided by the square root of this median.

The shape and the size of the neighborhood are arbitrary. A circular neighborhood with a radius of 7 pixels was chosen because it seem to be small enough to pick the geometrical variations but big enough to average the spot to spot variations.

Figure 6.1B shows the correction factor calculated using the algorithm. Figure 6.1C shows the resulting image after correction. On most of the image, the results are satisfying. On the lower left corner, the results are worse because the red signals are very faint. This is a case were there is an interplay between a dependence of the ratios on the intensity and on the geometry. The same effect appears, although in a much tamer way, on the right part of the slide. In general, the resulting image seems to be more reasonable than the original image. On most slides, the effects are less pronounced and easier to correct.

To validate the method, the influence of the spatial normalization on the windowed correlation and on the false discovery rate was measured (figure 6.2). The non-linear normalization, which is described in the next chapter, was applied. Both the correlation and the false discovery rate are largely improved by the spatial normalisation, at all intensity levels. This normalization is an important improvement on our data and is applied in all the following analyses.

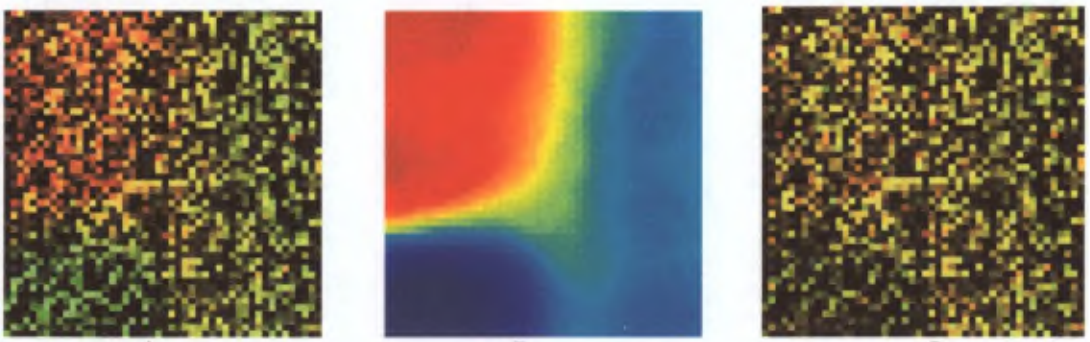
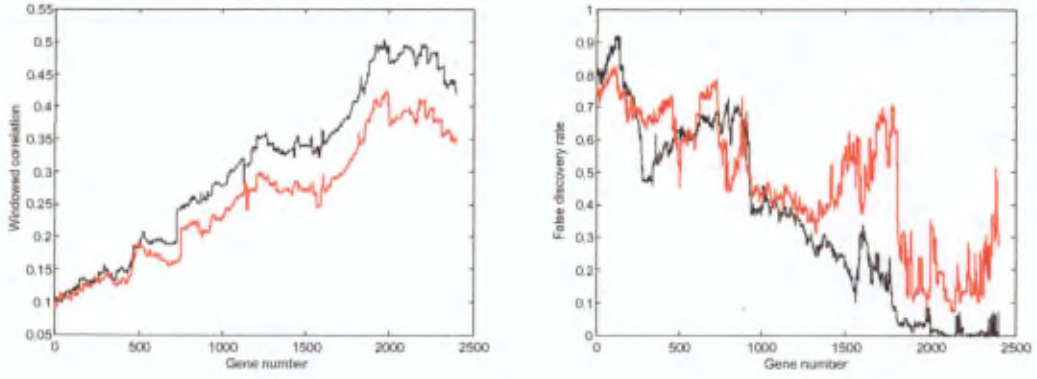


Figure 6.1. A. The ratios on a slide with a pronounced geometrical dependence. **B**. The calculated correction factor in false colors. **C**. The corrected ratios.



A B

Figure 6.2. Effect of the spatial normalization on **A**. the windowed average correlation and **B**. the false discovery rate. Red: without spatial normalization. Black: with spatial normalization. Genes with a higher gene number have a higher intensity.

6.2 Intensity dependence

Yang *et al.* (2002) have discovered that the ratios measured by the microarrays showed an unexpected intensity dependence. For instance for highly expressed genes the intensity could be higher in one channel than in the other (*e.g.* the signal in green is systematically higher than in red) while the opposite holds for less expressed genes.

This can be seen by plotting, for a given slide, the log ratio (R/G) in function of the log intensity (R^*G) (figure 6.3A). If no intensity dependence existed, the genes should be disposed symmetrically around the horizontal axis. In this case, there is an intensity dependence and the genes are disposed in a typical "banana" shape.

There are no biological reasons for the most expressed genes to behave differently than the least expressed genes: the differential expression (the ratio) should be independent on the intensity. This means that this dependence is a systematic error, which should be corrected. Of course, it could happen that this dependence is a genuine signal, but this possibility seems unlikely in general.

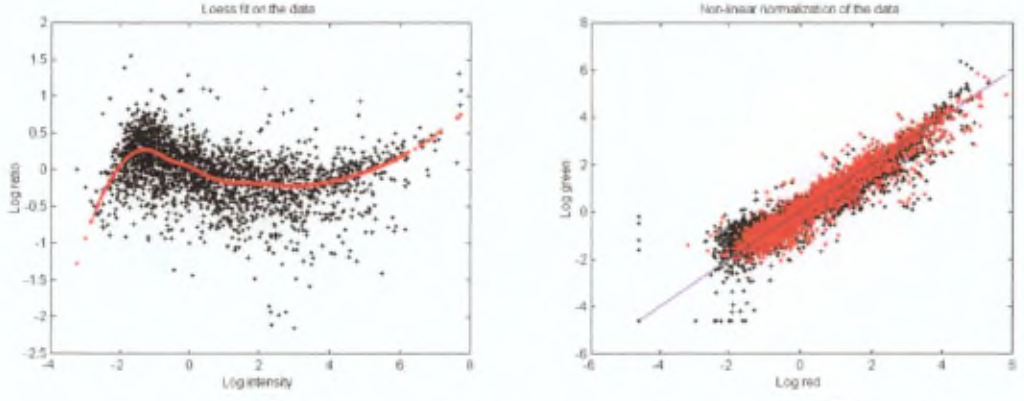
The proposed mean to normalize the data is to fit a curve through the cloud of points, using a robust algorithm. Loess (Cleveland and Grosse, 1991) has been proposed in the literature (Yang *et al.*, 2002), and is the one implemented and used here (figure 6.3A). The results of the loess fit must be checked by eye, as it can happen that it overfit or underfit the data, although this is rare in practice. In the case shown, there seems to be an overfitting on the low intensity part of the graph, which is not too problematic because those data are badly quantified anyway. At each intensity, the value of the loess fit is subtracted from the ratio value. The intensity/ratio pair of values are then turned back into a red/green pair of values. On figure 6.3B, the corrected values are superimposed on the original values to show the effect of the intensity-dependent normalization. The corrected data are much closer to the 45° line than the original data.

The improvement in reproducibility given by this normalization was assessed using the windowed correlation and the false discovery rate (figure 6.4). The correlation is marginally decreased by the normalization. This lack of improvement is due to the fact that the correlation is a pair-wise measure, performed on intensity windows. For genes of similar intensities, the corrections are similar. Hence, inside an intensity window, all genes are modified in a similar fashion. This means that the cloud of point is only translated in a log-log graph, which has little effect on the correlation (*e.g.* figure 6.5). On the example shown, the median ratio value is much closer to zero for the intensity-dependent normalized values (-0.03 and 0.02) than for the non-normalized values (-0.15 and 0.03). This difference however has no effect on the correlation measure.

Testing for the false discovery rate gives a completely different picture. The FDR is dramatically decreased by the non-linear normalization, proving the usefulness of this normalization.

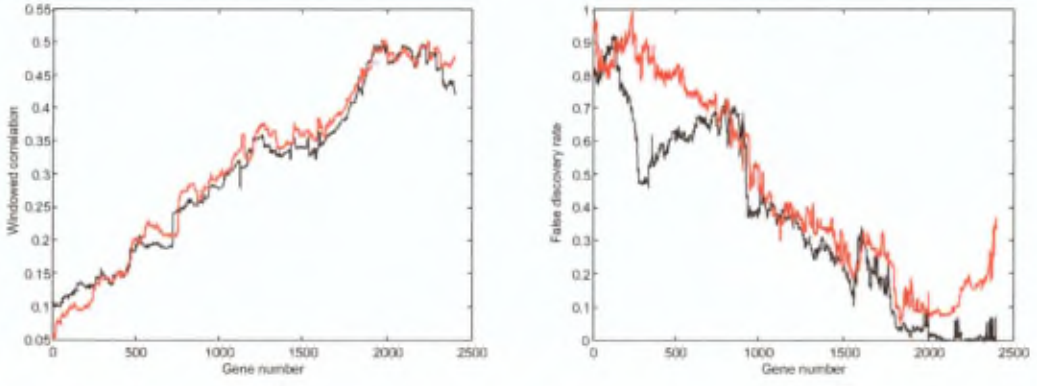
This shows that the choice of the quality criterion to determine the effectiveness of a normalization is paramount. It is important to be aware of biases which they might have. The use of two very different criteria is useful to highlight those biases and to insure that the normalization improves the data and is not simply optimizing the data with regard to any bias the quality criteria might have.

In conclusion, the intensity-dependent normalization improves the reproducibility of the data and is used in all following analyses.



A B

Figure 6.3. A. Ratio vs intensity plot and loess fit of the data (red). **B**. Plot of the red versus the green channel values for each spot. Original data shown in black, corrected data in red.



A B

Figure 6.4. Effect of the intensity-dependent normalization on **A.** the windowed average correlation and **B.** the false discovery rate. Red: without intensity-dependent normalization. Black: with intensity-dependent normalization. Genes with a higher gene number have a higher intensity.

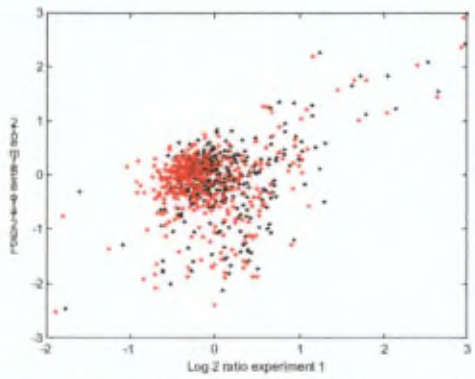


Figure 6.5. Effect of the intensity-dependent normalization on the measure in a window of intensity. Black: with intensity-dependent normalization; red: without intensity-dependent normalization.

6.3 Normalization order

The two normalization steps, spatial and non-linear, are performed independently. As the origin of the biases those normalizations eliminate are not clear, there are no theoretical reasons to believe that the normalization order is important. The testing framework presented was used to compare the reproducibility obtained with the two normalization orders (figure 6.6). The results are not strikingly different, but if anything the order with the spatial normalization first seems a bit better. This seems to show that the two normalizations act in a largely independent way.

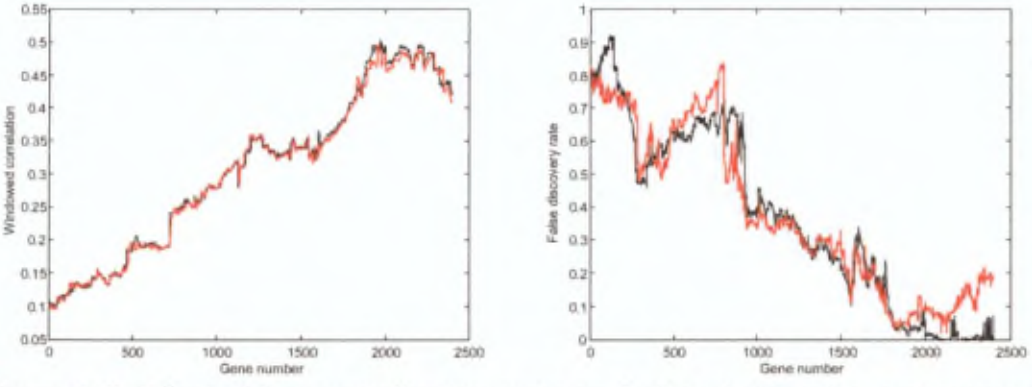


Figure 6.6. Effect of the normalization order on **A**. the windowed average correlation and **B**. the false discovery rate. Red: non-linear first. Black: spatial first. Genes with a higher gene number have a higher intensity.

7 Conclusion

When dealing with microarray data, there is a wide array of technical choices possible, and many different data correction procedures can be applied. Those are data dependent, the same choice could improve the data on one platform but not on another. This means that the different normalizations proposed in the literature had to be tested on the data generated in our laboratory to be validated.

The design of a sound data quality measure is complicated by the difference in quality from spot to spot: the quality of the quantification is correlated with the spot intensity. This explains the shortcomings of the data quality measures proposed in the literature, which made them unfit for our purpose. Indeed, it is possible to improve those quality measures by making trivial modifications of the data, like for instance by adding a constant to each value. This decreases the importance of the low-intensity genes, which improves the data quality

measure. For this reason, two new measures were defined: the windowed correlation and the false discovery rate. Those two methods display a measure of the data quality in function of the gene intensities. The advantage of this graphical display is that genes are only compared with genes of a similar intensity, and hence quality, and that the effect of the improvement is better visualized. Two different measures are used with the expectation that if by accident one would be unfit to properly detect the effect of a data modification, the other would. The discrepancy between the two measures would prompt attention. The importance of this was made clear with the non-linear normalization, on which the windowed correlation measure was unable to correctly quantify the improvement, because it is based on pair-wise comparisons. In general, the false discovery rate method was more resilient and more sensitive to the data quality.

Different technical choices were compared. In the scan quantification process, the subtraction of the background improved the quality of the data. The labeling was shown to have a large effect, so that experiments have to be performed twice, with the labels inverted in the duplicate. The saturation of the scanner had a negative effect on the data quality, and a technique based on the merging of scans at different gains was designed to correct this effect.

Once the data are quantified, different normalizations can be applied to improve their quality. On our slides, two normalizations are paramount. The first removes the dependence between the intensity and the ratios. The second removes the dependence between the spatial localization of the spots and the ratios. These two normalizations were shown to dramatically improve the quality of the data.

This work has allowed the creation of a systematic data treatment program, which automatically corrects the data produced in our laboratory. Changes in slides or in protocol can make some of the choices non-optimal, so the effectiveness of the normalization procedure should be tested on any new platform.

8 Bibliography

Cleveland, W.S. and Grosse, E. (1991) Computational methods for local regression. *Statistics and Computing*, **1**, 47-62.

DeRisi, J.L., Iyer, V.R. and Brown, P.O. (1997) Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale, *Science*, **278**, 680-686.

Dudley, A.M., Aach, J., Steffen, M.A. and Church, G.M. (2002) Measuring absolute expression with microarrays with a calibrated reference sample and an extended signal intensity range. *Proc. Natl. Acad. Sci. USA*, **99**, 7554-7559.

He, Y.D., Dai, H., Schadt, E.E., Cavet, G., Edwards, S.W., Stepaniants, S.B., Duenwald, S., Kleinhanz, R., Jones, A.R., Shoemaker, D.D. and Stoughton, R.B. (2003) Microarray standard data set and figures of merit for comparing data processing methods and experiment designs. *Bioinformatics*, **19**, 956-65.

Hsiao, L.L., Jensen, R.V., Yoshida, T., Clark, K.E., Blumenstock, J.E. and Gullans, S.R. (2002) Correcting for signal saturation errors in the analysis of microarray data. *Biotechniques*, **32**, 330-336.

Tseng, G.C., Oh, M.-K., Rohin, L., Liao, J.C. and Wong, W.H. (2001) Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variation and assessment of gene effects. *Nucleic Acids Res.*, **29**, 2549-2557.

Tsodikov, A., Szabo, A. and Jones, D. (2002) Adjustments and measures of differential expression for microarray data. *Bioinformatics*, **18**, 251-60.

Wang, X., Ghosh, S. and Guo, S.W. (2001) Quantitative quality control in microarray image processing and data acquisition. *Nucleic Acids Res.*, **29**, e75-e83.

Wang, X., Hessner, M.J., Wu, Y., Pati, N. and Ghosh, S. (2003) Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction. *Bioinformatics*, **19**, 1341-47.
Yang, H.Y., Dudoit, S., Luu, P., Lin, D.M., Peng, V., Ngai, J., and Speed, T.P. (2002) Normalization of cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, **30**, e15.
Zien, A., Aigner, T., Zimmer, R. and Lengauer, T. (2001) Centralization: a new method for the normalization of gene expression data. *Bioinformatics*, **17**, S323-S331.

3 - A database for microarray data

1 Introduction

High-throughput gene expression technologies generate a large amount of data. However, although many data sets are readily available on the Internet, it is hard even for specialists to collect, standardize and use them. For the bench biologist, these data are available in theory, but are not used in practice.

Publicly available data are scattered on different Web sites, and now databases. This system presents many inconveniences: the sites are not always as stable as they should be and the results are presented in a non-standard form (plain text, or HTML, or presented as a database,...), sometimes indexed by gene, sometimes by EST (Image ID or Genbank ID or...) and finally the numerical data are either non-normalized or normalized in a certain, and often unique, way. Because of this, extracting an information of interest in these data is a daunting task.

In order to make gene expression information intelligible, we, and other (Aach *et al.*, 2001) have developed databases, allowing to store and retrieve those data. Moreover, a biologist-friendly web-based interface allowing querying the data was created.

This database project was conducted at the beginning of this thesis. It proved very difficult and time consuming to clean and enter data in the database, for a non-obvious return. Moreover, it was expected that other databases would soon be created by much larger groups (EBI and NCBI, essentially) during the course of this work. Moreover, some design choices might have proven to be incorrect. For those reasons, no work has been performed on this database since the end of 2001. This explains some shortcomings, like for instance in the normalization options. It would not have been very difficult to improve the database, but it did not seem worthwhile.

2 One size fits it all

Gene expression data coming from different studies had to be standardized in order to be introduced in our database.

The main point in common between all gene expression studies, that justifies the creation of the database, is that they are measuring amounts of mRNA in different conditions. This puts the measurements at the center of our scheme. The database structure can be viewed as a generalized two-dimensional array comprising many empty cells, with the experiments on one ordinate and the genes on the other.

From there, all had left to do was to find standard ways to store the experimental conditions, to decide how to store the information describing the genes and finally to normalize the values obtained so that they could be compared to each other.

2.1 Storing of the experiments

We considered that experiments are a set of manipulations which share the same experimental protocol except for the parameter that is modulated in the experiment. Examples are measures of the response of a cell as a function of the time in certain condition (kinetics), e.g. after growth factor, hormone or drug treatment, comparisons of different lineages of cells or comparison between normal and diseased tissues.

There are different kinds of experiments, depending on the nature of the parameter of interest. One important characteristic we took into consideration is the fact that the parameter might be ordered (i.e. the order of the manipulation matters) or not. If an experiment is ordered, then the variable can be expressed as a number in a certain unit. This difference led us to design two groups of experiments, ordered and unordered, each comprising two sub-groups.

1. Ordered experiments

These are the experiments for which the parameter can be expressed as a number. Two sub-types of experiments belong to this group:

1.a. Kinetic studies, where the parameter is the time (e.g. measures taken at different intervals during the cell cycle of the yeast.)

1.b. Experiments where the parameter is a value expressed in a non-temporal unit (e.g. concentration/response curves.)

Of course, kinetic studies could be considered just as a particular case of type 1b. We could thus consider that we have only one kind of experiment, or we could as well have as many kinds of experiments as there are different ways to generate ordered manipulation. We chose to make a special case of kinetic studies firstly because it is a very common and important case and secondly because facilities are already built in the SQL standard regarding time variables, so reprogramming it would be a waste of time for a probably lower quality.

2. Unordered experiments

These are the experiments for which the changing variable cannot be considered as a numerical value. Again, two sub-kinds of experiments were defined:

2.a. Comparison of cellular types (e.g. measures of normal versus cancer tissues, comparisons between different cell lineages.)

2.b. Everything else. This would include the experiments that could not be ascribed to any of the previous classes.

The rationale behind the type 2a is to use a standardized way to store sample information. The type 2b is used when only a text description would make sense.

Each experiment was stored with all the relevant information except for the independent parameter, which is stored in the description of the manipulations. This leads to a concise and ordered presentation of the experiments, with no data duplication.

There is a great deal of information that may be relevant to describe the experiments, and until now there was not consensus as to what is important and what is not. We chose to keep as little as we thought we could. The idea is not to bury the important data beneath a vast amount of unnecessary information. An interested person can always look in the publication if he needs more details.

We chose to store only the following data to describe the experiments:

1. The origin of the experiments, i.e. the publication.
2. The technology used (mostly its type (oligonucleotide arrays or microarray or SAGE) and its manufacturer, *e.g.* oligochip mu6500 from Affymetrix.
3. The strains, cells or samples used.
4. A text description describing the experiment performed.

Everyone will probably not consider this information as sufficient, but we think it should suffice in order to make sense of the data. We deliberately chose not to tackle this issue more deeply than was necessary, since there is an international working group devoted to doing so, the Microarray Gene Expression Database Group.

2.2 Standardization of the values

The expression studies vary a lot depending on the technologies used, but nevertheless some characteristics remain identical. By stressing these, it is possible to reduce the data to a common standardized form to be used in our database. The raw numerical values must be normalized in order to make them comparable from study to study.

Three technologies were considered: microarrays, oligonucleotide chips (Affymetrix) and serial analysis of gene expression (SAGE). We had to define a normalization for each of these technologies.

2.2.1 Microarrays

In this technology (Schena *et al.*, 1995 and 1996), the sample of interest is reverse transcribed and marked with a fluorescent dye. A control target is made in parallel with another dye. The two labeled targets are hybridized simultaneously to an array of spotted cDNA (typically corresponding to an EST). The fluorescence of both dye are quantified. In each experiment, two measures are taken for each spot: one for the control sample and one for the sample of interest.

In most microarrays, the precision of the spotting is not good enough to permit uses of these two absolute values in a reliable way. Moreover, the different lengths of the spotted cDNA, as well as their different composition and therefore hybridization introduce other sources of variability. By taking the ratio of the two measures, the systematic errors cancel each other and the precision of the technique is raised. Hence, for each measured gene there are two values, of which only the ratio is meaningful.

Each channel in each experiment was normalized by dividing its measures by the mean of the values for each gene measured in the experiment, effectively putting the mean to 1. This insured that all the experiment values were in a comparable range. More complex normalization schemes have been described since this database was finished, but were not implemented.

2.2.2 Oligonucleotide chips

This technology (Lockhart *et al.*, 1996), developed by Affymetrix (Santa Clara, CA), uses slides with oligonucleotides being directly synthesized by using a combination of photolithography and oligonucleotide chemistry.

Each gene is represented by several 20 to 25 base oligonucleotides. Next to each of these oligonucleotides lies another one with one mismatch in the center, serving to determine the background hybridization. The arrays are hybridized with labeled antisense mRNA, synthesized *in vitro* from cDNA reverse transcribed from the sample mRNA. The fluorescence is then quantified on each pixel of the arrays. The amount of a particular mRNA can be measured by taking the average of the difference in the fluorescence of the perfect match probe to the single base mismatch probe.

This technique leads to a direct estimate of the amount of a particular mRNA present in the target. Therefore values should have a meaning by themselves; no comparison has to be made like in the microarrays. Nevertheless, the absolute value is generally meaningless, depending on the efficiency of the labeling, on the amount of mRNA present... In each experiment, the values can be multiplied (or divided) by a constant without changing their meaning.

A natural way to normalize such values would be to turn them into concentrations by dividing them by an estimation of the total amount of mRNA present, i.e. the sum of the value of every gene. But only certain genes are quantified, so if we used the sum of the measured genes as an estimation of the amount of mRNA, we would have had different normalization depending of the number of genes assessed. To address this issue, we chose to normalize each experiment by dividing each of its measures by the mean (not the sum) of all the values, effectively equating this mean to 1. Each resulting value is a pseudo-concentration, a value of 1 corresponding to an average concentration.

2.2.3 SAGE

This technique (Velculescu *et al.*, 1995) works by sequencing small "tags" of a dozen nucleotides extracted at specific points in the mRNA. These short tags are usually long enough to be assigned to a unique gene. Tens of thousands of these tags are so sequenced. By counting the number of tags corresponding to each gene their concentration can be inferred. The values obtained are a direct estimate of the concentration of each mRNA in the sample. Therefore, even though the experimental protocol is very different from oligonucleotide chips, the type of result is similar, hence we have used the same kind of normalization for both.

2.2.4 The final format

As previously seen, the SAGE and the oligonucleotide chips give one value for each measure, while the microarrays give two values. These two values are usually only meaningful when being compared to each other. Because of this difference of design, the data were stored differently depending of the technology: for SAGE and oligonucleotide chips, normalized values were stored, while for microarrays both measured values alongside with the ratio were stored.

It should be noted that the difference between the two kinds of storage is deeply grounded, because it reflects an important difference between microarray and SAGE/Affymetrix data. The formers are always a comparison, while the latter are absolute values. This has an important impact as to the ways to query the data, since microarray data are directly meaningful (typically meaning an up or down regulation in an experimental condition) and can be queried directly. This raises questions like "Which measures are up-regulated by a factor 2.5 in this experiment?". In contrast, SAGE/Affymetrix data is usually only useful when comparing one experiment with another. This would raise questions like "Which genes are 2.5 times more expressed in condition *a* than in condition *b*?" Note that even though it may be possible to transform every SAGE/Affymetrix experiment to a comparison of two conditions (by taking one condition as the control), this is often not recommended. As an example, we could take the data from Golub *et al.* who were measuring bone marrow of patients with ALL or AML leukemia. The comparison here would be between the set of measures of ALL with the set of measures of AML, but there is no such thing as a pair-wise comparison permitting the storage of the data in a format similar to the microarrays.

As for the normalization, the rule is to store the normalized data. The normalization factors are stored along with the manipulation descriptions. The rationale is to use them as requested by the user, simply by changing the query parameters and the printed values. Other kinds of normalization could be added in the same way, would the need eventually arise.

2.3 The genes

As stated before, the expression studies consist in measuring the level of expression of thousands of genes in diverse conditions. We had already categorized the condition and standardized the level of expression, so we reach now the last part of the question: what was meant when we said we were measuring the level of expression of a gene?

The answer to this question depends on the technology, but nevertheless one thing remains: something is measured and there is a meaning attached to it, which is obtained by linking with information coming from another source.

In a typical microarray experiment, DNA corresponding to an EST is spotted, and so the corresponding mRNA is quantified. By extension, we call EST whatever is measured, being a real EST or something else. The meaning we give to this EST is the gene to which it corresponds. In the human case, this could be readily understood by visualizing the EST as the EST, and the gene as the UniGene cluster comprising this EST. The link between gene and EST may change as our understanding of the genome deepens, but the identity of the measured EST will not change.

In the case of the yeast, most of the data is taken by open reading frame (ORF, presumably coding for a protein), assigning a value to each of these ORF. So we would say an ORF is an EST, and also a gene. But sometimes measures are taken not only by ORF, but also by exon inside each ORF. In this case, the gene is still the ORF, but the EST is the exon measured.

We have thus defined an EST as the measured object. The meaning we assign to this EST is the gene, and it may change. A gene may comprise more than one EST, and an EST may belong to more than one gene (such things happen in UniGene). The link between the EST and the gene is obtained with a database that clusters the EST by genes and gives

a meaning (i.e. a description) to each gene. Such databases were statically linked with ours. The chosen ones were UniGene for the human and SGD for the yeast. This information will have to be updated on a regular basis from the source databases.

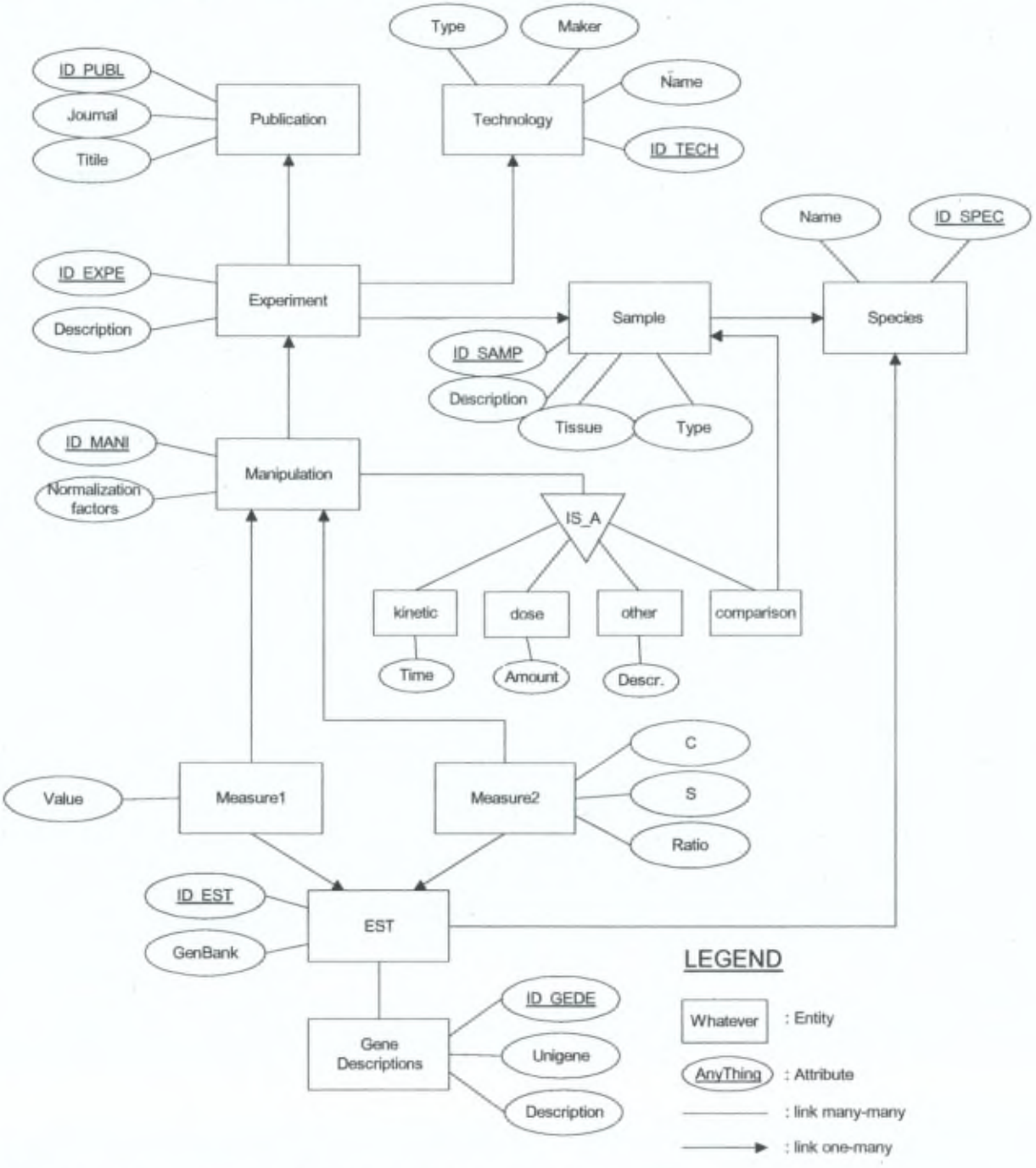


Figure 1. Diagram of the database, slightly simplified for clarity.

2.4 The final diagram

The complete diagram of the database, reflecting the discussion of the last few paragraphs, was finally drawn (fig. 1).

Some parts were not implemented as shown for efficiency purpose. For instance, the IS_A relationship between the different kinds of experiments (see figure) leads to too many queries to the database server just to know the type of experiment, so it had to be changed.

Nevertheless, this diagram captures the philosophy behind the database.

2.5 Critics on the scheme used

After the implementation was performed, it was obvious that the performance of the database was barely satisfying for typical queries.

One type of typical query would be to find the level of expression of a certain gene in a certain experiment. With the database scheme as it is, this implies a search in the entity Measure1 or Measure 2 on two different fields – ID_EST and ID_MANI. Double indexes are not very efficient, and the indexing of only one of the two proved to be no better.

Another type of typical query would be to find all genes in a certain experiments which have certain values. If a double indexing was created it would be useless in this case. Even if a single index on ID_MANI was created, the search of all the measurements proved to take quite a long time (a second or so when everything is in cache).

It might be than those performance issues were caused by the database system used. It could be possible that a system which is more geared towards queries and less towards data integrity like mySQL could prove more efficient than postgres. If I had to do it all again however, I believe I would have created a completely different framework. The coercion of the microarray data into a 3rd normal form might be something counter-productive. As the data are effectively presented as arrays, which can be queried extremely efficiently, it might be preferable to keep to main numerical data that way, and to only store the gene and the experiments description in a database format. This would mean that more bookkeeping should be performed and that an API should be designed to manage the database, but it might prove to be effectively the best option.

3 The database

Since the database was meant to be publicly accessible, we implemented it on a server and created a Web interface to query it. For reasons of cost, quality and availability, we implemented the database using only free software. We used embedded SQL/C on a gnu-Linux computer, with postgres (www.postreSQL.com) as the relational database and Apache (www.apache.org) as the Web server. Graphic display was implemented using GnuPlot. The CGI interface was implemented using three free libraries, fastCGI, yacgi and cgicc.

The query interface of the database was based as much as possible on the KIS principle (keep it simple). One big difficulty was to define natural ways to query the data, the questions asked being often imprecise and ill defined. As a parallel we could see the research of these natural ways to be somewhat like the creation of BLAST for the EST databases. Without such query tools, it would be much harder to make sense of the available information. What we offer presently in our database should only be considered as a first glimpse of what we think any forthcoming gene expression database should offer.

3.1 The queries

When a user connects to the database, the Web interface shows a form offering many different search criteria (fig. 2). Depending on what the user chooses, two different kinds of queries can be performed.

If the user selects one or more criteria relative to the genes, the database is searched for genes matching them. If some genes are found, the measures for these genes are shown, for every experiment in which they appear if the user was not specific about the experiments, or only for some experiments if the user added some search criteria regarding the experiments.

If no criterion was entered about the genes, the experiments matching the search criteria for the experiments are shown (fig. 3). It is then possible for the user to de-select the experiments he is not interested in, and more importantly to enter any new search criterion based on the measured values in the experiments.

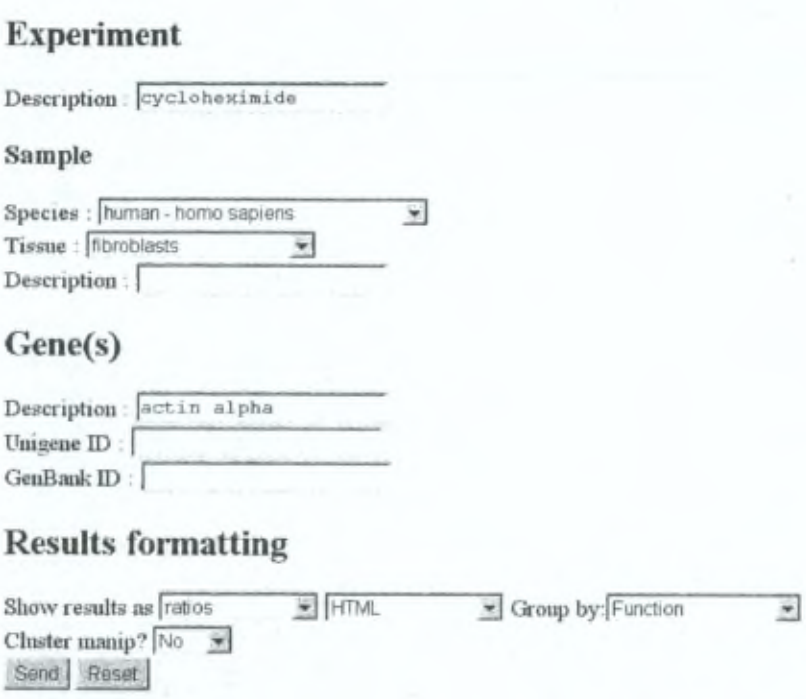


Figure 2. The first search screen. The example query shown is "Show the values for every gene having "actin" and "alpha" in his description, in experiments involving homo sapiens fibroblasts where "cycloheximide" appears in the description of the experiment". The result of this query is shown on fig. 4.

This is done in two different ways, depending on whether the experiments are microarrays (hence comparisons) or SAGE/Affymetrix (hence absolute values). In the former case, a simple threshold might be meaningful, for example asking for every ratio over 2 in a certain experiment will give every gene which is up-regulated by a factor of two or more in this experiment. In the latter, using a simple threshold would make much less sense, as it will only give the least or most expressed genes in a certain condition. The interface offers the possibility to compare any two values, asking for an n-fold difference between them.

All these search criteria may be combined on a per EST basis or on a per gene basis. This is especially useful when comparing results from different laboratories, where a given gene is generally represented by different EST.

3.2 Presentation of the results

The results obtained can be shown either as an HTML document or as a tab delimited text document for easy exporting to another application.

When presented in HTML, the gene descriptions act as hyperlinks to the Unigene or the YPD database, while the EST act as hyperlinks to GenBank. The one-line descriptions of the experiments are hyperlinks too, offering access to a more complete description. For the microarray experiments, the numbers are colored according to their level of up or down regulation: red for the most up-regulated ones, green for the most down-regulated, and everything in between for the modulations in between.

In the case of ordered experiments, it is possible to select some genes and to ask for a graphical representation of the experiments.

3.3 Data mining options

The results of the queries, even if they are much smaller than the whole set of data, can still be large, often tens of genes in tens of conditions. To make the visualization of such data easier, an implementation of a hierarchical clustering algorithm (as found in Eisen *et al.*, 1998) is offered.

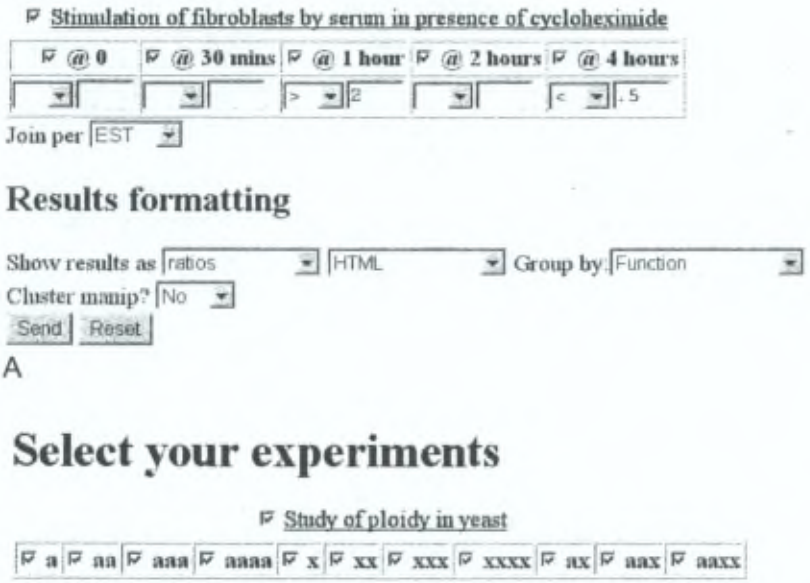


Figure 3. Search screen permitting to search for certain expression patterns (**A**.) for microarray data (the example query shown is "Show the genes which are upregulated by a factor of 2 at 1 hour and downregulated by a factor of 2 at 4 hours") (**B**.) for SAGE/Affymetrix data (the example query shown is "Show the genes which are 3 times more expressed in condition aa than in condition a, and four times less expressed in condition x than in condition xx").

The idea behind this algorithm is to reorder the data in such a way that genes whose expression signatures are close are shown next to each other. This orders the genes in a logical order if such an order exists. If there were clusters in the data, the algorithm would hopefully bring together the genes belonging to the same cluster. The end result is that the genes are shown in a more logical order, making it easier to find a particular pattern of expression.

Clustering can also be performed on the experimental conditions. This can be useful to see if some conditions cluster together, e.g. if the difference between cancer and normal tissue appears naturally from the data, or if different kinds of cancers would cluster together.

		Stimulation of fibroblasts by serum in presence of cycloheximide				
Description	Genbank ID	@ 0	@ 30 mins	@ 1 hour	@ 2 hours	@ 4 hours
actin, alpha, cardiac muscle	AA039938	0.765	2.09	1.48	2.02	1.68
actinin, alpha 1	AA043737	1.48	1.46	1.36	1.54	1.47
actin, alpha 1, skeletal muscle	AA026608	1.11	1.94	1.74	2.43	1.77
ARP1 (actin-related protein 1, yeast) homolog A (centractin alpha)	AA032015	0.813	0.905	1.17	0.837	0.799
actinin, alpha 4	AA009817	1.65	0.941	0.967	0.922	0.87
protein tyrosine phosphatase, receptor type, f polypeptide (PTPRF), interacting protein (liprin), alpha 1	N53430	0.942	0.983	1.35	1.81	0.911
capping protein (actin filament) muscle Z-line, alpha 1	AA053172	0.958	0.555	1.24	1.57	0.89
filamin A, alpha (actin-binding protein-280)	AA046721	1.41	3.43	2.71	3.95	4.32
actin, alpha 2, smooth muscle, aorta	AA040169	1.12	0.895	0.807	0.654	0.647
synuclein, alpha interacting protein (synphilin)	N21998	1.01	0.547	1.19	1.44	0.945
capping protein (actin filament) muscle Z-line, alpha 2	AA056767	0.989	0.863	1.14	1.21	0.958
PAK-interacting exchange factor alpha	W96027	0.734	1	0.735	0.787	0.804
actinin, alpha 2	W94733	0.898	0.622	0.867	1.89	0.627

A

		Stimulation of fibroblasts by serum in presence of cycloheximide				
Description	Genbank ID	@ 0	@ 30 mins	@ 1 hour	@ 2 hours	@ 4 hours
actinin, alpha 2	W94733	0.898	0.622	0.867	1.89	0.627
capping protein (actin filament) muscle Z-line, alpha 1	AA053172	0.958	0.555	1.24	1.57	0.89
synuclein, alpha interacting protein (synphilin)	N21998	1.01	0.547	1.19	1.44	0.945
capping protein (actin filament) muscle Z-line, alpha 2	AA056767	0.989	0.863	1.14	1.21	0.958
protein tyrosine phosphatase, receptor type, f polypeptide (PTPRF), interacting protein (liprin), alpha 1	N53430	0.942	0.983	1.35	1.81	0.911
ARP1 (actin-related protein 1, yeast) homolog A (centractin alpha)	AA032015	0.813	0.905	1.17	0.837	0.799
PAK-interacting exchange factor alpha	W96027	0.734	1	0.735	0.787	0.804
actin, alpha 2, smooth muscle, aorta	AA040169	1.12	0.895	0.807	0.654	0.647
actinin, alpha 4	AA009817	1.65	0.941	0.967	0.922	0.87
actinin, alpha 1	AA043737	1.48	1.46	1.36	1.54	1.47
actin, alpha 1, skeletal muscle	AA026608	1.11	1.94	1.74	2.43	1.77
actin, alpha, cardiac muscle	AA039938	0.765	2.09	1.48	2.02	1.68
filamin A, alpha (actin-binding protein-280)	AA046721	1.41	3.43	2.71	3.95	4.32

B

Figure 4. Result of the search shown on fig. 1, presented with the gene ordered (**A.**) by UniGene cluster ID or (**B.**) by clustering.

The opportunity to find the genes whose expression patterns are similar to a gene of interest is also offered, permitting firstly to find other genes of interest, and secondly to assign a tentative biological function to the gene if it is unknown.

These few data mining tools should be considered as a first draft, to be completed as the state of the database as well as the state-of-the-art of gene expression analysis improve.

4 Discussion

A database centralizing information from many expression studies, performed on different species with different technologies has been presented. To obtain this result, it was necessary to standardize the data. This standardization had to be performed not only on the numerical values, but also on the experimental protocols. To determine what was important and what was not in every experiment was not an easy task, especially when the expertise in the field was lacking. To make expression data globally available in their most pristine condition, a standard for the description of the experimental protocol should be created. This would make the curation of any expression database infinitely easier, and avoid the problem of the biological competence of the curator in the precise domain of the experiment.

Many options should still be implemented to make this database really complete. For instance, it would be useful to be able not only to search for genes differing from one experiment to another, but also to search for genes that discriminate best between two groups of experiments, for instance between cancerous and normal tissue (e.g. Student's t test). This would give a much more "averaged out" criterion to select the genes of interest, probably leading to more interesting findings than to compare systematically each pair of experiments. It could also be useful to include other clustering algorithms (e.g. K-means, self-organizing maps) which might be more adapted to the problem than the phylogenetic tree reconstruction algorithm actually used.

5 Bibliography

Aach, J., Rindone, W. and Church, G.M. (2001) Systematic Management and Analysis of Yeast Gene Expression Data, *Genome Research*, **10**, 431-45.

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*, **95**, 14863-14868.

Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, M., Kobayashi, M., Horton, H. and Brown, H.L. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays, *Nature Biotechnoly*, **14**, 1675-1680.

Schena, M., Shalon, D., Davis, R.W. and Brown, P.O. (1995) Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, **270**, 467-470.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P.O. and Davis, R.W. (1996) Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA*, **93**, 10614-10619.

Velculescu, V.E., Zhang, L., Vogelstein, B. and Kinzler, K.W. (1995) Serial Analysis of Gene Expression, *Science*, **270**, 484-487.

EBI : http://www.ebi.ac.uk/arrayexpress
NIH : http://www.ncbi.nlm.nih.gov/geo
NCGR : http://www.ncgr.org/research/genex/
ExpressDB : http://twod.med.harvard.edu/ExpressDB
MGED : http://www.ebi.ac.uk/microarray/MGED/index.html
YPD : http://www.proteome.com

4 - Discovery of overlapping clustering

1 Introduction

One of the most useful statistical analysis technique for gene expression data is clustering. The clustering of the samples groups them together on the basis of their expression profile. It reveals differences between groups of samples, like for instance between previously unidentified sub-types of cancer.

There can be more than one meaningful way to cluster the samples. For instance, in a study about tumors, samples could be clustered according to their pathological status or their inflammation level. If the complete data set, with all genes, is clustered, the result could be any of those clustering, or a chimerical clustering where some samples are grouped based on one concept and others based on some other concept. Getz *et al.* (2000) have shown that by clustering on only a subset of genes, different clusterings of the samples appear, each with its own biological interpretation.

The problem is to get the appropriate clustering for the application at hand. A first approach is to keep all genes. In this case, the clustering which appears is the one supported by the largest number of genes. This might be a reasonable strategy if it is expected that most of the variation in the data is due to the phenomenon of interest. However, if the appropriate clustering is not the most prominent, it is likely to be missed.

A related approach is to select a set of genes on which a well defined clustering can be found (Dash *et al.* 1997, Devney and Ram 1997, Xing and Carp 2000). Those techniques are based on the idea that some genes are irrelevant, that is, consist of noise. But as Getz *et al.* have shown, the problem is not so much the presence of noise than the existence of different organizations of the data. The genes which are irrelevant for one clustering can be relevant to another. The methods working by feature selection are not well adapted to such case.

Another approach is to select the genes based on some knowledge of the data (Alizadeh *et al.* 2000). Alizadeh *et al.* selected a cluster of genes using some biological insights (they selected genes which defines germinal centre B-cell signature), and performed their analysis using only this subset of genes. This might be an effective method for data where the genes are clearly defined, so that the choice of genes determines the question asked. For instance, a clustering made on genes expressed only in lymphocytes could separate the samples according to their inflammation level. In high-throughput gene expression experiments however, the genes are usually only succinctly annotated. This means that this approach can bias the results, as the choice of genes might be made to obtain the clustering expected by the investigator. It can also hide important, unexpected, features of the data.

Getz *et al.* (2000) proposed to search for tight groups of highly correlated genes, and to cluster the samples on those groups. This approach has two weaknesses: firstly, the sample clustering is done on only a handful of genes which have essentially all the same values. Secondly, it only works for correlated genes, hence showing a linear dependence, while relationships that are more complex can exist. For instance, some genes oscillate with time during the cell cycle, hence they show a similar profile. However, two oscillating genes with a 90° phase difference are not correlated.

Von Heydebreck *et al.* (2001) proposed to search for sets of genes for which a clear binary separation of the samples can be found. This technique searches only binary separations. The algorithm must be extended to find more complex clustering.

We present here a completely different framework for the determination of the different clusterings present in the data, which is based on two main hypothesis about the structure of the data. The first hypothesis is that there are a limited number of elementary clusterings, that is clusterings that cannot be simplified. For instance, a clustering between two cancer types cannot be simplified. Complex clusterings can be created using two or

more elementary clusterings, for instance a four-groups clustering could be created as the combination of two two-groups clustering. The second hypothesis is that each feature is relevant only to one elementary clustering. With those hypotheses, it is natural to try to group the genes, so that each group of genes is relevant only to one of the elementary clusterings. Doing this is akin to try to cluster the genes according to the clustering they are relevant to, hence the name MetaClustering.

It is possible to view results of the MetaClustering algorithm differently. A clustering could be understood as the assignation of a certain value to each sample. This value can be a discrete number, as in a k-means, or a more complex structure, as in a hierarchical clustering. A gene fitting a clustering must be a function of those values. Since all genes fitting a clustering are functions of the same values, they must depend on each other in some complex, possibly not univocal, way. This means that the determination of the different possible clusterings is a way to find groups of non-linearly related genes.

2 Methods

In order to clarify the presentation, we define in the following *clustering* as being a partition of the samples (observation) and *grouping* as being a partition of the genes (features). The goal of the method is to find a *grouping* (on the genes), each *group* (of genes) defining a *clustering* (on the samples).

2.1 Outline

The goal of the MetaClustering algorithm is to find groups of genes, so that each gene fits as well as possible the clustering of the samples calculated on its group. This is done as follow:

1. Start with some random group membership for all genes.
2. Calculate a clustering of the samples on each group.
3. Calculate the "fit" of each gene to each clustering, using some fitness function to be defined.
4. Move each gene to the group whose clustering it fits best.
5. Repeat steps 2-4 until convergence occurs.

The number of groups is a user-defined input of the algorithm; it sets the trade-off between variance and bias, as in the k-means algorithm. If the number of groups is too small, then a group of genes might support more than one elementary clustering. if the number of groups is too large, there is a risk of overfitting the data, giving small groups of genes which are determined mostly by the noise.

Different clustering algorithms and different fitness functions can be used in this framework. In this thesis, a version based on the average linkage hierarchical clustering algorithm and a version based on the neural network k-means are shown.

2.2 The hierarchical clustering version

This first version of the algorithm uses the average linkage hierarchical clustering.

2.2.1 Quantifying the fitness of a gene to a clustering

The algorithm goal is to group together genes relevant to the same sample clustering, so a choice of clustering algorithm must be made. Those clusterings are used only to represent their group structure, they do not have to be biologically interpretable. They only have to organize the data in a completely unsupervised fashion. As they are used intensively, they also have to be computationally efficient. We opted for average linkage hierarchical clustering (Jain and Dubes 1988) because it fulfills this criterion and is commonly used in gene expression analysis. It can also be efficient if properly implemented, taking a couple of seconds for the clustering of 2000 genes.

A measure of the fit of a gene expression pattern to a hierarchical clustering on the samples, *i.e.* a measure of the fit between a variable and a clustering, must be defined. Only a couple of such measures exist in the literature. However, those were made to compare various clusterings on the same data, not to estimate the relevance of a variable with respect to a given clustering. We chose to define a new measure of fit in parallel with the average linkage hierarchical clustering algorithm.

In that algorithm, at each step the two closest nodes are merged to form a new node. The distance $d(L,R)$ between two nodes L and R is defined as the mean of the distances between the leaves of each of those two nodes:

$$d(L,R) = \sum_{i}^{Ng} \frac{1}{s(L)s(R)} \sum_{k \in S(L)} \sum_{l \in S(R)} (x_{ik} - x_{il})^2 \qquad (1)$$

where Ng is the number of genes, $S(L)$ is the set of samples at the leaves of the node L, $s(L)$ is the cardinal of $S(L)$ and x_{ik} is the value of the gene i in the sample k. This means that at each junction a certain criterion is minimized in a greedy fashion. This criterion can be used to assess the fit of a gene to a clustering.

Practically, the fitness $F(i,c)$ between a gene i and a clustering c is defined as

$$F(i,c) = -\sum_{j=1}^{Nn} \frac{1}{s(L(j))s(R(j))} \sum_{k \in S(L(j))} \sum_{l \in S(R(j))} (x_{ik} - x_{il})^2 \qquad (2)$$

where Nn is the number of nodes in the clustering and $L(n)$ and $R(n)$ are the left and right children of node n.

The fitness (2) is a weighted sum of every sample differences. The weight for a difference between two samples can be understood as follow. The lowest node which contains both samples is taken. Somewhere in his left child tree is one of the sample, while the other is somewhere in his right child tree. The weight is the product of the number of leaves in those two child trees. The number of leaves joined by a node is higher if the node is higher in the tree. Hence, the weights are larger for the differences between samples joined by a node lower in the tree than for samples joined by a higher node. Since the elements linked by the lower nodes should be very close, while the higher nodes may link very different groups of observations, the function indeed quantifies the fitness of a gene to a clustering.

2.2.2 The quality function

The quality of the solutions obtained must assessed. A good method should group together genes which fit the same clustering. This is quantified using the quality function Q_0:

$$Q_0 = \sum_{i}^{Ng} F(i, C(G(i))) \qquad (3)$$

where $C(G(i))$ is the clustering calculated on the group $G(i)$ to which the gene i belongs. The algorithm should maximize Q_0 with respect to $G(i)$. An issue with the quality function (3) is that the fitness between a gene and the clustering calculated on its group can be influenced more by the gene itself than by the rest of the group. This effect is more pronounced in small groups. In order to really assess if a gene fits its group, a modified quality function is used:

$$Q = \sum_{i} F(i, C(G^*(i))) \qquad (4)$$

where $G^*(i)$ is the set of genes of the group $G(i)$, with the gene i excluded. With this modification, the quality function is computationally heavier but more meaningful. This also means that, for the calculation of the quality function, there are as many clusterings in each group as there are genes in the group. In practice, a few genes are excluded together in order to speed up the calculations.

The direct maximization of (4) does not lead to satisfying results, firstly because it is very heavy to calculate and secondly because of the presence of numerous local maxima.

Thus, the quality function is not directly maximized. A stochastic version is used instead. In that version, the following is done for each gene:

1. The clustering for the group to which the gene belong is re-calculated after the removal of the gene.
2. The fitnesses between the gene and the clusterings of each group are calculated.
3. The gene is moved to the group whose clustering it fits best.

See Figure 1 for an illustration. This version neglects the effect of the switching of a gene from one group to another on the quality of the other genes. This allows for a much faster calculation and permits to avoid many local maxima.

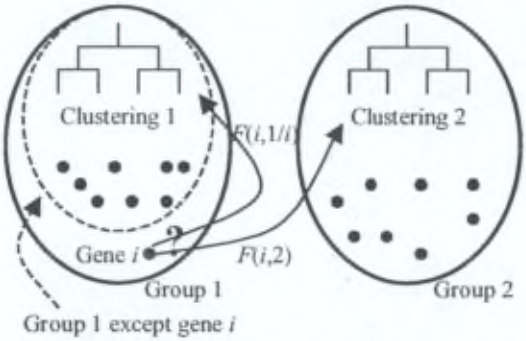


Figure 1. Scheme of the algorithm. Each point is a gene, the ovals are the groups and the trees are the clusterings. The gene *i* belongs to group 1. A clustering is calculated for the group 1 without the gene *i*. The fitness of the gene to this clustering is compared to its fitness to the clusterings of the other groups, and the gene *i* is moved to the group it fits best.

Since the algorithm is not maximizing a global criterion, convergence is not guaranteed. However, because the algorithm is deterministic, markovian and the search space is finite, it has to converge either to a fixed solution or to a cycle. If the number of genes or the number of groups of genes is small, then the cycles might be short enough to be detected. Otherwise, the algorithm can either be stopped after a determined number of iterations, or the quality (4) of the solutions can be monitored, and the algorithm can be stopped when no improvement is noted for a sufficient number of iterations. In our implementation, we stopped the algorithm after convergence, after detection of a cycle, or after a hundred iterations, whichever occurred first.

2.2.3 Complexity and algorithmic improvements

A first improvement is based on the fact that the fitness needed is not between a gene and a clustering, but between a gene and a group of genes on which a clustering is calculated. Hence, it is possible to reduce the variance of the calculation of the fitness by creating a few slightly perturbed clusterings. This is done using k-fold cross-validation: $(K-1)$ K^{th} of the genes of the group are selected K times for the calculation of the fitness. The results are then averaged. This leads to a better and more stable estimation of the fitness of a feature to a group of genes.

The complexity of the algorithm is quite high. Let the number of genes be M, the number of samples N, the number of groups of genes G and K-fold cross-validation be used. Typical values for those constants could be M=2000, N=50, G=5 and K=5.

During an iteration, each gene is taken in turn. For each gene, a clustering is calculated on the samples with all the genes in its group except itself (complexity: $O(MN^2/G)$ for the distance matrix and $O(N^3)$ for the clustering). The fitness between the gene and the clusterings must then be calculated (complexity: $O(GN^2)$). This is done K times for the cross-validation. If the gene is moved to another group a clustering must be calculated for the original and the new group (complexity: $O(MN^2/G) + O(N^3)$). So the complete complexity of an iteration is $O(KM^2N^2/G) + O(KMN^3) + O(KGMN^2)$. Since usually $M/G>>N$, the effective complexity is $O(KM^2N^2/G)$. The dominant term is the calculation of the distance matrices.

It is possible to reduce the complexity of the cross-validation. The distance matrix of each K^{th} of the genes can be calculated separately. The distance matrices for the cross-validation are sums of those matrices. This way, the complexity of the calculation of the distance matrices is reduced to $O(KMN^2/GK) + O(K^2N^2)$, that is $O(MN^2/G)$ if $K^2<<M/G$. Also, the fitness of a gene to a clustering is simply a weighted average of the values of the distance matrix of the gene. The weights are identical for each gene, so they can be calculated only once. The complexity of the calculation of the fitness of X genes can be reduced to $O(KN^2)$ for the weights plus $O(XN^2)$ for the distances, that is $O(XN^2)$ if $X>>K$. With those tricks the cross-validation comes at a low cost.

It is possible to make "cheaper" iterations by neglecting the effect of the removal of a gene from its group on its fitness, that is by using (2) as the quality function instead of (3). Those iterations have a reduced complexity of $O(GMN^2) + O(KGN^3)$. A couple of those iterations are performed before each complete iteration to fasten convergence.

Since the number of genes in any group is large, the removal of some genes should not have a large effect on the fitness of the others. Practically, the genes of each group are separated in F parts (*e.g.* 10). All the genes of each part are considered simultaneously (see Figure 2B). This way only GF distances and clusterings have to be calculated instead of M. Those improvements reduce the complexity of an iteration to $O(FMN^2) + O(GMN^2) + O(KFGN^3)$. This is comparable to the complexity of an average linkage hierarchical clustering (*i.e.* $O(MN^2) + O(N^3)$), but of course the constant in front of the complexity is much higher and many iterations must be performed. Altogether, the running time of the algorithm is reasonable, being about 10 seconds per full iteration on an Athlon 650 workstation with 2000 genes, 72 samples and 5 groups.

2.3 Neural networks version

The two-level clustering presented can easily be written in neural network form. This is done by taking a neural network clustering algorithm – a K-means in this case – and by adding a first layer which dispatch the features to one clustering or another.

An example of such architecture is presented in figure 2. Three K-means, with the neurons W, are on the right part of the figure. Each of those K-means clusters the observation in two groups. The Z neurons on the left of the figure are used to give a weight to each feature for each clustering.

The neurons Z_k and W_{jk} have M weights, where M is the number of features, noted Z_{ik} and W_{ijk}. With those notations, the equations of the network can be written:

$$y_{ik} = x_i Z_{ik} \tag{5}$$

$$o_{jk} = \sum_i y_{ik} W_{ijk} \tag{6}$$

As the clustering algorithm is a K-means, the supposition that only one neuron fires in each clustering is added. That means that for each k, the highest o_{jk} is set to 1 and the others to 0.

The quality function for this network can be written as

$$Q = \sum_{ikm} \left(x_{im} Z_{ik} - W_{ib(m,k)k} \right)^2 \tag{7}$$

where $b(m,k)$ is the neuron which fires in the map k when the observation m is presented. In order to avoid trivial solutions, some normalization constraints must be added:

$$Z_{ik} \in [0,1] \tag{8}$$

$$\sum_k Z_{ik} = 1 \tag{9}$$

A quick calculation leads to the stochastic gradient:

$$\frac{\partial Q}{\partial W_{ib(m,k)k}} = 2\left(W_{ib(m,k)k} - Z_{ik} x_{im} \right) \tag{10}$$

$$\frac{\partial Q}{\partial Z_{ik}} = 2\left(Z_{ik}x_{im} - W_{ib(m,k)k}\right)x_{im} - \frac{2}{K}\sum_{l}\left(Z_{il}x_{im} - W_{ib(m,l)l}\right)x_{im} \qquad (11)$$

where K is the number of self-organizing maps. This gradient is used for learning the weights.

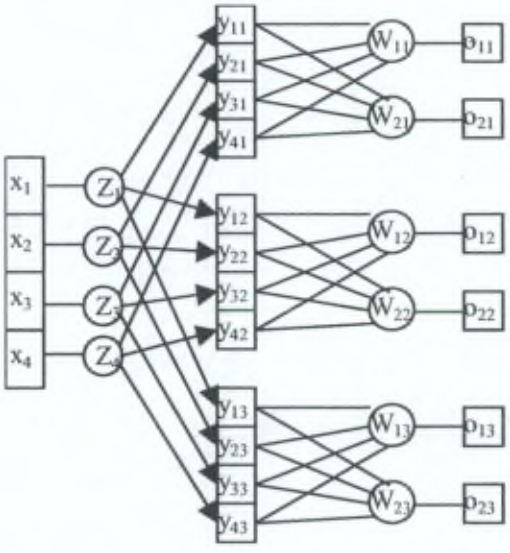


Figure 2. Scheme of the neural network version of the MetaClustering algorithm, with four features separated in three groups. The observation clusterings are two-groups K-means. The circles are the neurons and the boxes are the input/output values. x_i are the input, Z the neurons which perform the features grouping, y_{ij} the input values for each clustering, W the neurons which perform the K-means observation clustering and o_{ij} the outputs.

3 Results

3.1 Simulated data

An artificial data set including non-linearly linked genes has been created in order to show the power of the MetaClustering compared to more usual methods.

The data set consists of 20 genes and 50 samples. The 20 genes are organized in 4 groups of 5 genes. In the first three groups, the genes are linked together in a similar fashion. The last five genes are simply random noise.

For each of the first three groups, a random permutation ***s*** of the numbers 1 to 50 is drawn. This permutation gives a value from 1 to 50 to each sample. The genes F1-F5 are related to ***s*** as follow: F1 = ***s*** ; F2 = $(\boldsymbol{s}\text{-}25)^2$; F3 = sin(9***s*** / 50) ; F4 = sin(12***s*** / 50) ; F5 = -***s***. The permutation is different for each of the first three groups. Each gene is centered and normalized, then gaussian noise with a standard deviation of .3 is added and the resulting genes are centered and normalized again. The random genes are drawn from a gaussian distribution of unity standard deviation. This data set is constructed so that although the genes are linked, the correlation between some of them remains small. In particular, the second and third genes of each group (F2, F3) are hardly correlated to the other genes (F1, F4 and F5) of their group.

As shown in figure 3, some genes can be correctly grouped by a single linkage hierarchical clustering algorithm using an Euclidean distance, especially those which are linearly related (*e.g.* genes 1 and 5). However, only some small parts of the grouping are correct and the dendrogram does not give an accurate representation of the gene structure. Similar or worse results are obtained with other classical hierarchical clustering algorithms and K-means.

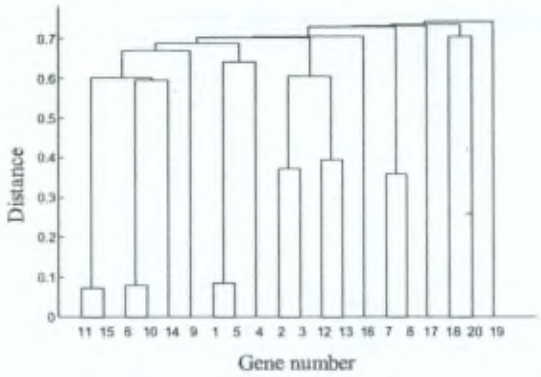


Figure 3. Dendrogram of a single linkage hierarchical clustering of the genes in the simulation data set, the distance being one minus the absolute correlation. The genes 1-5, 6-10 and 11-15 should be grouped together. The genes 16-20 are random.

A classical way to group together non-linearly related genes is to use a non-linear metric, mutual information being the most common choice. Mutual information can only be calculated on discrete data, so the continuous values must be discretized first. The values for each gene were discretized to 3, 4, 5 or 6 levels. The mutual information was then calculated between all genes, and the genes were clustered using one of the three classical hierarchical clustering algorithms: single linkage, average linkage or complete linkage. We then checked if the clusterings had captured the right structure. A clustering was considered as correct if it was possible to find 3 nodes such that the leaves in each of these nodes contained all the genes of one of the non-random group and no genes of any of the other non-random group. 100 simulated data set were drawn. The results are summarized on table 1. The hierarchical clustering using the mutual information was often able to pick the right structure, but this was not always the case. Even in the best case scenario (discretization in 4 levels and average linkage), only 70% of the data sets were correctly clustered. So the mutual information is not able to robustly recover the known structure. Moreover, it is not trivial to decide how many discrete levels are optimal for a particular data set.

The MetaClustering algorithm was then used to group the genes. A hundred data sets were randomly drawn along the scheme given. The algorithm was run twenty times for each data set, with different random initialization. The number of discrepancies between the known grouping and the obtained grouping were recorded, as well as the quality of the results as measured by (4). It is not expected for each run to give the right solution, as the algorithm is sensitive to its initialization. However, the solution with the highest quality, in the sense of (4), should be the correct one.

The hierarchical clustering version converged in 65% of the runs to the right solution (see table 2). The run with the best quality (4) was consistently the right solution, showing the effectiveness of the algorithm.

The results of the neural network version were less convincing, as the right solution was found less often and it happened that the run with the highest quality (7) was not the right one. However, the results could be considered as promising, as a few runs were still usually enough to find the right solution. The K-means algorithm used is the simplest, and more complex type of clustering like self-organizing maps could improve the result. The number of clusters in the K-means part of the network has a certain effect on the result: when this number is too small, the K-means is too coarse and the network is unable to uncover the real structure, when this number is too large the algorithm tends to have some trouble converging. However, the algorithm remained much more effective than the hierarchical clustering using a mutual information metric.

The last five genes, which are purely random, were split between the groups since the MetaClustering algorithm always keeps all features. Their presence did not compromise the ability of the algorithm to correctly group the other genes.

In conclusion, the MetaClustering algorithm proved to be able to robustly group genes which are non-linearly related, contrary to more classical approaches. The hierarchical clustering version seems more effective. It leads more consistently to good solutions and is computationally more efficient. The hierarchical version is systematically used in the next applications.

Number of discrete levels	3	4	5	6
Single linkage	1%	64%	31%	20%
Average linkage	8%	70%	43%	38%
Complete linkage	2%	2%	10%	8%

Table 1. Percentage of the simulated data sets in which hierarchical clustering using mutual information metric was able to find the expected structure, in function of the number of levels used for the discretization.

	Hierarchical	Neural network version, K-means with					
		2 groups	3 groups	4 groups	5 groups	6 groups	7 groups
Runs with right solution	65%	0%	14%	31%	29%	23%	18%
Data sets with at least 1 right solution	100%	0%	70%	100%	99%	99%	98%
Data sets with an error in the best solution	0%	100%	32%	4%	3%	2%	7%

Table 2. MetaClustering of the simulated data sets.

3.2 Leukemia data

Golub *et al.* (1999) have studied with oligochips the differences between two types of leukemias: *acute myeloid leukemia* (AML) and *acute lymphoblastic leukemia* (ALL). This last type of leukemia can further be separated into T-lineage ALL and B-lineage ALL. They showed that the distinction between ALL and AML could be inferred directly from the data with a clustering technique. We have discovered that this was indeed true, but depended on the normalization and filtering scheme used and on the initialization of the clustering algorithm.

The normalized data (see appendix for details) were clustered using an average linkage hierarchical clustering algorithm (Figure 4A). The leaves were ordered, as for all hierarchical clusterings shown, using the technique of Bar-Joseph *et al.* (2001). The images displayed were obtained using TreeView (Eisen *et al.* 1998). Some parts of this clustering were close to the ALL/AML separation, but other parts seemed unrelated. Since hierarchical clustering is based on local similarities, some samples were merged because they were close on one set of genes, others because of another set of genes.

We tried to cluster the samples in two groups using a k-means algorithm (Jain and Dubes 1988) with random initialization. This was done 1000 times, leading to 324 different solutions. We define δ as being the number of differences between a clustering and the ALL/AML labels. As shown Figure 5A, solutions close to the ALL/AML separation were obtained in only 4% of the runs. The sum of the square distances between the samples and their cluster center is a measure of the quality of a k-means clustering. As shown in Figure 5B, this criterion does not point to the solution closest to the ALL/AML separation. So the

ALL/AML separation might be obtained through k-means, but if it was not known beforehand, it could be missed.

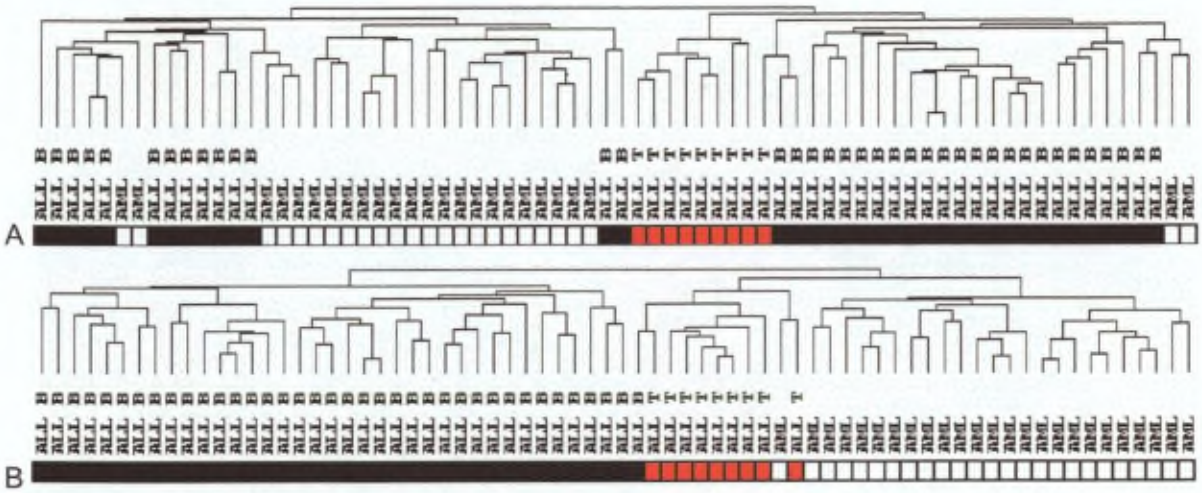


Figure 4. Clusterings obtained on the ALL/AML data set. **A.** With all the genes. **B.** With the genes from a group determined by MetaClustering in three groups. The samples are color-coded: white for AML, black for ALL-B and red for ALL-T.

MetaClustering allows the determination of the different elementary sample clusterings. Since the ALL/AML separation is expected to be one of those clusterings, it should be prominent on one of the group of genes determined by the algorithm. Indeed, this was often the case after MetaClustering in three groups (*e.g.* Figure 5C). Furthermore, the k-means error criterion did often point to the ALL/AML separation (*e.g.* Figure 5D). In the following analysis, the k-means algorithm was always run 100 times, and the solution with the lowest k-means error was kept.

Since different initialization of the MetaClustering can lead to different results, the algorithm was run 200 times with random initialization, with two or three groups. Among the runs with two groups, 61% had a δ below 6. With three groups, this fraction raised to 77%. It is possible to decide which of the MetaClustering run should be chosen by using the quality function (4). The solution with the highest quality (4) had indeed a low δ (Figure 5E and 5F), showing that it is possible to use (4) to decide which MetaClustering to keep. With the highest quality MetaClustering, δ was 1 with two groups and 2 with three groups. On the 200 runs, the algorithm converged to the best two groups solution 4 times and to the best three groups solution 3 times. This means that, as with k-means, it may be necessary to perform a hundred or so runs with random initialization to pick the best one.

Our results compare favorably with the technique of Xing and Karp (2001), which used a feature selection algorithm and obtained a δ of 3. Getz *et al.* (2000) have determined a cluster of 60 correlated genes on which they claim that a clustering close to the ALL/AML separation appears. Using a k-means on that group of genes, the result was disappointing as δ was 6. This shows the importance of using more than one pattern of genes for the clustering.

In the original paper, a test was created to determine the most informative genes for the ALL/AML separation. We picked 50 genes using the same test. The group on which the ALL/AML separation appears indeed concentrated many of those genes, with on average 74% in the two groups MetaClustering and 70% with three groups. The grouping with the highest quality concentrated a larger number of those genes than average, that is 86% with two groups and 72% with three groups, showing again that (4) is an effective criterion to judge the quality of the MetaClusterings.

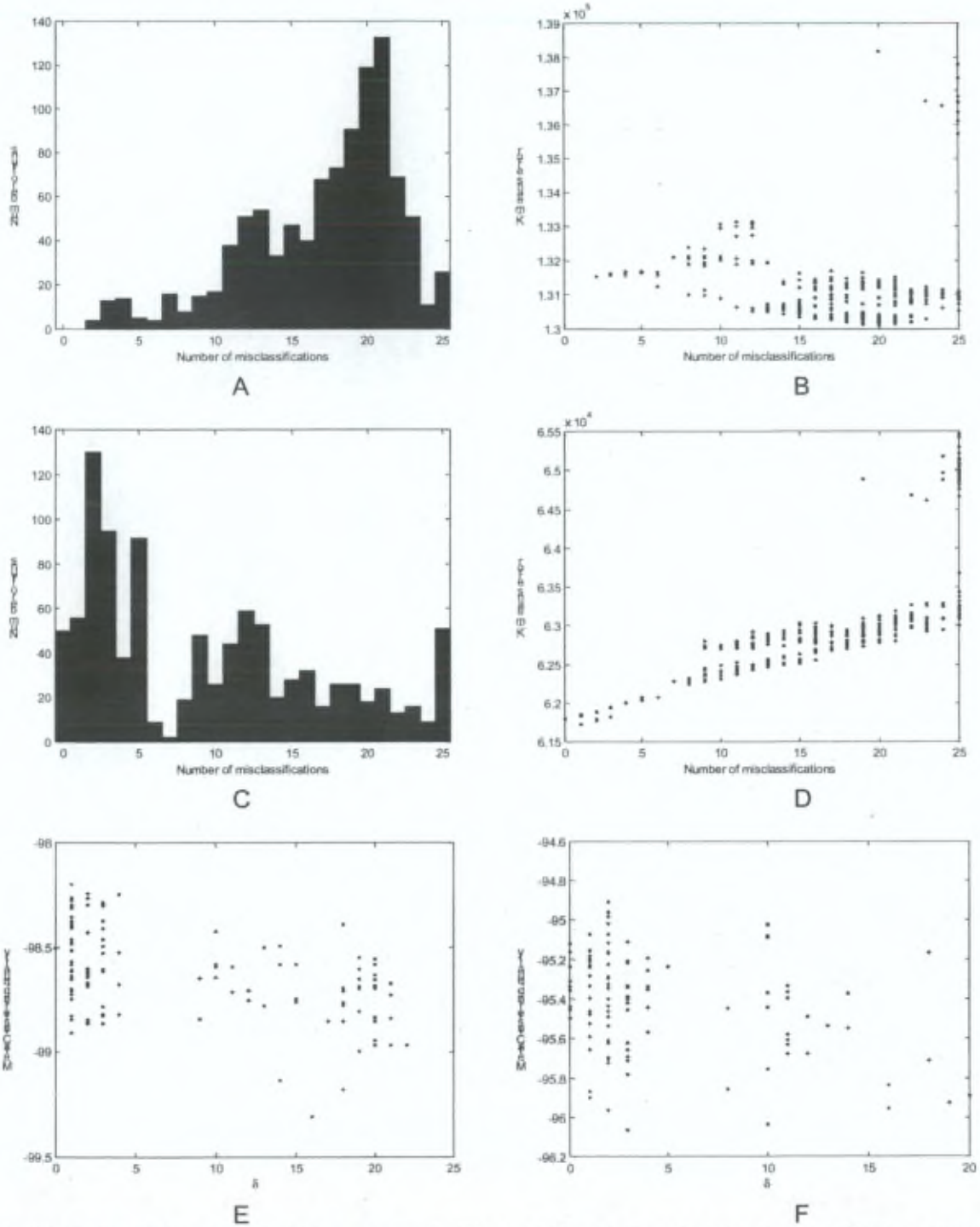


Figure 5. **A.** Histogram of the number of differences between the known ALL/AML separation and the k-means clustering (δ), using different random initializations. **B.** Mean square error of those clusterings. The best clusterings error-wise do not have the lowest δ. **C.** and **D.** are similar results obtained on one of the groups after MetaClustering in two groups. Results with low δ appear much more often and correspond to the minimum error of the k-means. **E.** and **F.** Number of differences between the known ALL/AML separation and the best k-means clustering (δ), as a function of the MetaClustering quality. The MetaClustering was done with two (E) or three (F) groups.

The separation of the samples in AML, T-ALL and B-ALL should also be found by clustering of the samples using one of the groups of genes obtained by MetaClustering. This

was assessed by k-means clustering in 3 clusters. We define δ as being the number of differences between a clustering and the T-ALL/B-ALL/AML labels. Again, clustering the whole data set lead to a large δ, 14. However, after MetaClustering, the results were closer to the expected separation. For the MetaClustering with the highest quality, δ was 10 using two groups of genes and 3 using three groups. In this case, three groups seemed necessary to recover the known structure.

The genes of the right group in the 3 groups MetaClustering with the highest quality were clustered using a hierarchical clustering algorithm (Figure 4B). This clustering is much closer to the known separation than the one obtained on the whole set of genes (Figure 4A). There were three main clusters: AML, T-ALL and B-ALL. This explains the presence of good quality MetaClusterings in three groups which have a δ with the ALL/AML separation of 10 (Figure 4F): the k-means algorithm being biased towards equivalent-sized groups, a solution with one tight B-ALL cluster and one loose T-ALL and AML cluster may compare favorably to a solution with one tight AML cluster and one not-so-tight B-ALL and T-ALL cluster.

The meaning of the clustering obtained on the other groups of genes is harder to understand. On one of the remaining groups the samples coming from one of the sources (CALGB) were tightly clustered together, which corresponded to one of the clusters found by Getz *et al.* This shows that other clustering are indeed present in the data, and that would the determination of the source be the important parameter it could have been found by MetaClustering. The other clusters were not intelligible with available biological information.

3.3 Yeast cell cycle data

Spellman *et al.* analyzed cell cycle in yeast using microarrays in 1998. In those experiments, yeast cells were synchronized at a certain point in their cycle. They were then released and began to cycle while keeping their synchrony. The expression levels of many genes, the cell cycle regulated genes, showed a periodic behavior. However, other genes showed different profiles, like for instance steady increase or decrease with time. Spellman *et al.* used a method based on Fourier transform to identify cell cycle regulated genes. Since those genes are not all correlated, it is impossible to cluster them together using classical clustering algorithms like hierarchical clustering or k-means (results not shown). The MetaClustering method groups together genes which support the same organization of the data. Since cell cycle genes have a specific periodic organization, they should be grouped together.

The data from one of the cell cycle experiments were taken (see appendix for details). The genes were clustered in two groups using our algorithm. The first group (Figure 6A) contained genes which showed a periodical behavior, that is cell cycle genes. The second group (Figure 6B) was not as coherent, but its main feature seemed to be a large variation from one time point to the next. Clustering the samples using the genes of the second group lead to a suprising two clusters result, one cluster comprising the odd time points (70 mins, 90 mins...) and the other the even time points (80 mins, 100 mins...). We have no explanation for this fact. However, by visual inspection, the grouping of the cell-cycle regulated genes seems reasonable.

Spellman *et al.* have determined 800 genes to be cell cycle regulated. Among the 569 of those genes which survived our selection process, 458 (80%) were in the first group. The remaining 20% of the genes seem to show large fluctuations from one time point to the next, which is a characteristic of the second group (Figure 6C). On the other hand, 601 genes were grouped with the cell cycle regulated genes while they were not considered as such by Spellman *et al.* As judged visually (Figure 6D), a large part of those genes could indeed be considered as cell cycle regulated. Since every gene must be assigned one group, even if it does not fit any group well, the presence of a certain percentage of non cell cycle regulated genes in the cell cycle group was expected.

In conclusion, the MetaClustering algorithm was able to group together cell cycle regulated genes in an unsupervised fashion, leading to results similar of those of Spellman *et al.*, who used a specialized algorithm. Since cell cycle regulated genes are not all correlated, this is a result that could not have been obtained by clustering algorithms based on pair-wise similarity (like hierarchical clustering) nor by algorithms based on prototypes defined in the original space (like k-means).

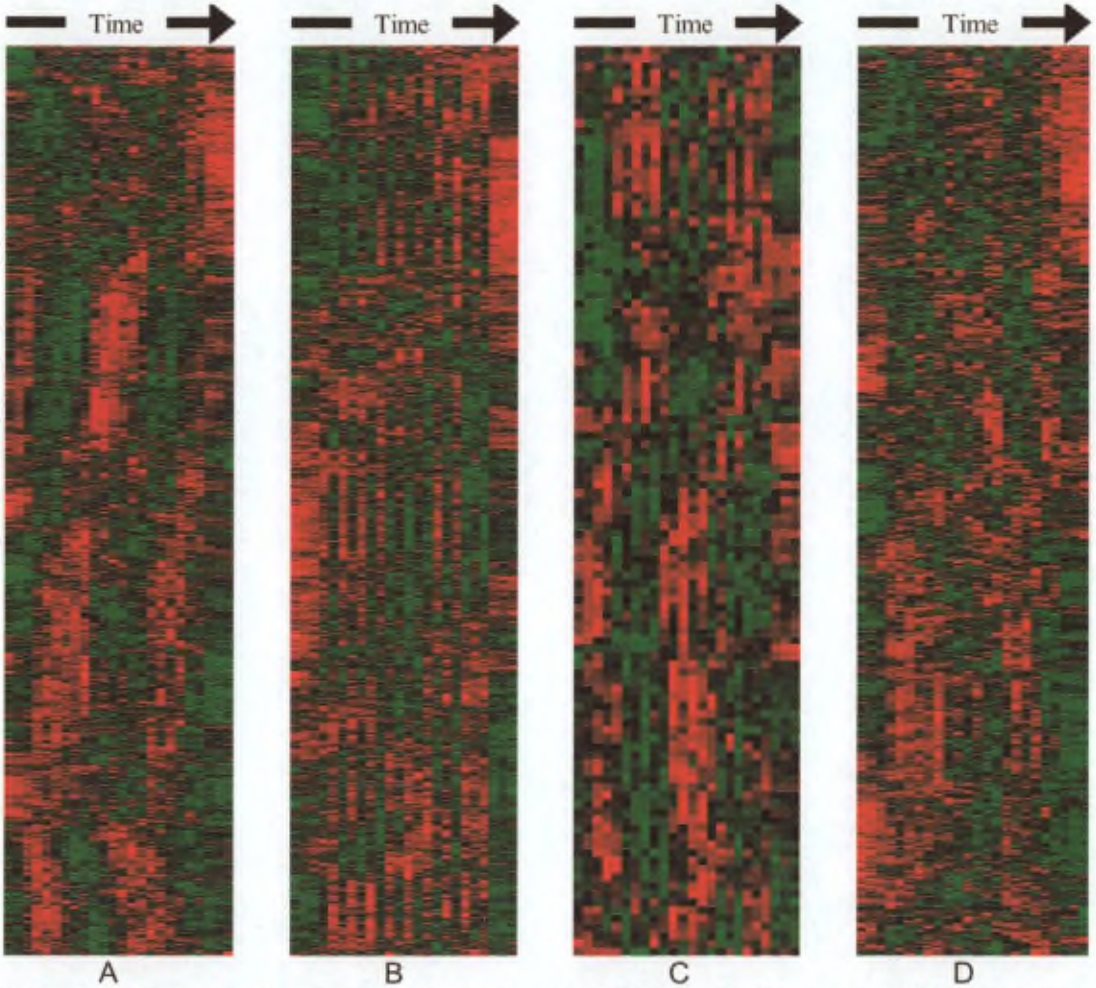


Figure 6. Groups of genes obtained by MetaClustering the "cell cycle" experiment. Genes were clustered using average linkage hierarchical clustering and ordered for the display. **A.** Cell-cycle genes group. **B.** Non cell-cycle genes group. **C.** Genes of group B considered by *Spellman et al.* to be cell-cycle regulated. **D.** Genes of group A not considered by Spellman *et al.* to be cell-cycle regulated.

3.4 IPUMS census data

To further demonstrate the power of the MetaClustering algorithm, it was applied on a completely different data set, in which the cohabitation of different sample clustering is likely. This data set is the IPUMS census data. The IPUMS data set is a subset of the American census data in the Los Angeles region, were the features have been standardized and some secondary features have been added. Each observation in the data set concerns an individual. The data set has many features, which can be relative to very different types of information – like for instance income and family status. The idea is to see if groups of features exist on which different individual clusterings can be found.

The data from the year 99 were taken. The house value and renting price were consolidated as one feature. The features which showed little variation were discarded. After that selection, there remained 40 features, 25 being continuous and 15 discrete. As the

algorithm makes a very intensive use of the hierarchical clustering algorithm, the number of observation must be limited. For that reason, only a random subset of 1000 observations was kept. The results do not seem to be very sensitive to the choice of subset.

For the hierarchical clustering algorithm to work, the distance between two observations must be defined. The difficulty is that observations have both discrete and continuous features. In order to render the relative contribution of each feature similar, the continuous features were normalized to a mean of zero and a standard deviation of one. The distance between two observations was then calculated as

$$d(i,j) = \sum_{k=1}^{MC} \left(xc_{ki} - xc_{kj}\right)^2 + \sum_{k=1}^{MD} w_k \left(xd_{ki} \neq xd_{kj}\right) \tag{12}$$

where *MC* (resp. *MD*) is the number of continuous (resp. discrete) features, xc_{ki} (resp. xd_{kj}) is the continuous (resp. discrete) feature *k* in the observation *i*, the value of $(a \neq b)$ is 1 if *a* is different from *b* and 0 otherwise and w_k is the weight for the feature *k*, calculated as

$$w_k = \frac{2}{1 - \sum_{i=1}^{N_k} f_{ki}^2} \tag{13}$$

where N_k is the number of discrete values in the feature *k* and f_{ki} is the frequency of the value *i* in the feature *k*. With those choices, the average contribution of each feature (discrete or continuous) to the distance between two randomly chosen observations is 2.

The features were separated in two groups using the hierarchical version of the MetaClustering algorithm (see table 3). The first group of features seems to be relative to the wealth of the individual. This group comprises all the income features (e.g. ftotinc, family total income) and other money or job related features. The second group of features seems to be related to the family status of the individual, with features like the number of mothers in the household (nmothers), the family size (famsize) and the marital status (marst). Hence the grouping found seems reasonable.

The quality of each feature in each group, calculated as the contribution of the feature to the quality function (4) would it be in the group, is also given in table 2. With the normalization used, this quality would be 2 if the feature does not fit the group and close to 0 if it fits perfectly. Those quality can be used to assess whether the hypothesis that the groups are independent holds or not. If the groups of features were really independent, the quality of the features in the group they are not part of should be around 2. For instance, the incbus feature (income from business) seems to fit only in group 1, as its quality in group 2 is quite close to 2. The raceg feature (race) however has qualities close to 2 in both groups, so it seems that neither wealth nor family status are very informative for the race, although the first group seems somewhat more appropriate. Some features, like age, can fit in both group: the wage income is informative for the age, as is the marital status or the number of children. Hence, the independence hypothesis does not completely hold in this case. However, this does not prevent the algorithm from giving a meaningful grouping.

Feature name	Group membership	Quality in group 1	Quality in group 2
Value/rent	1	1.4293	1.7302
Ftotinc	1	0.9227	1.7595
Incwage	1	0.3489	1.6265
Incbus	1	0.8609	1.9631
Incss	1	0.0295	0.7758
Incwelfr	1	0.0263	0.765
Nfams	2	1.8694	1.0953
Ncouples	2	1.7006	0.5037
Nmothers	2	1.6178	0.6472
Nfathers	2	1.5982	0.4594
Famsize	2	1.3495	0.5011
Nchild	2	1.6149	0.4885
Nchlt5	2	1.9512	1.3082
Famunit	2	1.9949	0.5626
Nsibs	2	1.5412	0.5971
Age	2	0.6553	0.5977
Chborn	2	1.4609	0.7348
Educrec	1	0.648	1.0658
Occscore	1	0.3697	1.2575
Sei	1	0.546	1.4513
Wkswork2	1	0.5091	1.3424
Hrswork2	1	0.3953	1.4128
Inctot	1	0.3528	1.56
Poverty	1	0.8789	1.5082
Movedin	2	1.3236	0.492
Ownershg	1	1.4978	1.6398
Momrule	2	1.1764	0.3113
Poprule	2	1.2488	0.2132
Sprule	2	1.5024	0.2607
Relateg	2	1.4282	0.4608
Sex	2	1.7274	1.0621
Raceg	1	1.834	1.9385
Marst	2	1.4184	0.6216
Bplg	1	1.6938	1.7702
Schltype	1	1.3408	1.3524
Empstatg	1	0.292	1.2584
Classwkg	1	0.4233	1.3896
Migplac5	2	1.711	1.5985
Vetstat	1	0.5408	0.9649
Tranwork	1	0.523	1.4776

Table 3. MetaClustering of the IPUMS data set.

4 Conclusion

A new framework allowing to uncover the overlapping clusterings of the samples has been presented. The algorithm works as well for discrete structures (*e.g.* cancer type, as in the leukemia data) than for continuous structures (*e.g.* cell cycle phase, as in the yeast data). The outputs of the algorithm are groups of genes which have a similar sample structure. This means that any clustering algorithm can then be used on those groups of genes, be it hierarchical clustering, k-means, or anything else. This flexibility makes the MetaClustering a powerful tool for the discovery of the different structures present in the data.

This work could also be viewed as a means to perform feature selection: the features are selected so that each group gives a tight clustering. The main idea which allows for this

feature selection is that all features are informative, but not to answer the same question. Hence, features can be selected according to the question asked, *i.e.* according to the clustering obtained on the samples. It could be possible to further select the features, by excluding those which do not really fit any group. We are investigating this possibility.

In conclusion, the algorithm presented here is able to find groups of linearly or non-linearly linked genes. Any clustering algorithm can then be used to extract the type of relationship between the samples on those groups of genes. We demonstrated that it uncovered the expected results when applied to an artificial and two real-world data sets.

5 Appendix – normalization procedures

For the leukemia data set, the complete data set (72 samples, 7129 genes) was taken from the original paper companion web site. The "present" calls were neglected. The values below zero were put to zero. The data were normalized so that the mean gene expression level was 1 in each sample. The genes having a normalized mean intensity across the samples lower than .2, having more than 25% of their values at zero or having a coefficient of variation (standard deviation divided by the mean) less than .5 were excluded. 1990 genes passed this filter. The remaining data were normalized to a mean of 0 and a standard deviation of 1 across each gene and sample.

For the cell-cycle genes, there are different experiments in the study, which differ in the method used to synchronize the cells. The one discussed is the growth arrest of a cdc15 temperature-sensitive mutant (24 time points), but similar results were obtained on other experiments. Since the data contain many very noisy measurements, genes with a low signal to noise ratio had to be discarded first. The data were log transformed and normalized so that each sample and each gene had an average log-intensity of zero. A smoothed version of the data was obtained using a Butterworth filter (Matlab Signal Processing Toolbox, MathWorks inc.). The 2067 genes for which at least 80% of the variation remained after smoothing and for which no more than 3 measures were missing were kept, in their non-smoothed form. The remaining genes were normalized to a mean of zero and to a standard deviation of one across the genes and across the samples.

6 Bibliography

Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson Jr, J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., 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.

Bar-Joseph, Z., Gifford, D.K. and Jaakkola, T.S. (2001) Fast optimal leaf ordering for hierarchical clustering. *Proc. 9th ISMB, Bioinformatics*, **17**, S22-S29.

Chen, Y. and Church, G.M. (2000) Biclustering of expression data. *Proc. 8th ISMB*, AAAI Press, 93-103.

Dash, M., Liu, H. and Yao, J. (1997) Dimensionality reduction for unsupervised data. *Proceedings of the 9th IEEE international conference on tools with AI*, 532-539.

Devney, M. and Ram, A. (1997) Efficient feature selection in conceptual clustering. *Proceedings of the 14th conference on machine learning*, 92-97.

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* **95,** 14863—14868.

Getz, G., Levine, E. and Domany, E. (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, **97**, 12079-12084.

von Heydebreck, A., Huber, W., Poutska, A. and Vingron, M. (2001) Identifying splits with clear separation: a new class discovery method for gene expression data. *Proc. 9th ISMB, Bioinformatics*, **17**, S107-S114.

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

Anil K. Jain, Richard C. Dubes. (1988). Algorithms for Clustering Data. Prentice-Hall.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell* 9, 3273-3297.

Xing, E.P. and Karp, R.M.. (2001) CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Proc. 9th ISMB, Bioinformatics*, **17**, S306-S315.

5 - Mathematical dissection of heterogeneous samples

1 Introduction

One difficulty with the analysis of gene expression data is the sample composition. The most convincing works have been conducted on pure samples, *i.e.* samples containing only one type of cells. When more than one cellular type is present, a situation which is encountered in solid tumors and tissues, it is much more difficult to draw any conclusion. Any cellular type present in the tissue contributes differently to the measured expression of a given gene. Two samples containing precisely the same cellular types (*e.g.* coming from the same tissue in the same patient) can have two different profiles of gene expression simply because the proportions of these cellular types are different.

Two methodological solutions are commonly used to address this issue: in situ hybridization, to check where a given gene is expressed; and micro-dissection, to isolate one particular cellular type before performing the experiment. Both of these methods are time consuming. Moreover, the first one cannot be realistically performed for every gene measured. We propose here a completely different approach. The idea is to start directly from the gene expression data obtained on the composite samples to determine mathematically the profile of expression of the cellular types present.

We will show in this chapter that, with certain assumptions, the problem of the identification of the cellular types is tractable. We will present an algorithm able to perform the separation and present briefly other approaches. The algorithm will then be applied to simulated results to show that it actually works. The problem of the correlation between cellular types will be addressed. We will finally apply the techniques developed to real-world data and will show that they permit to identify cellular types which seem mathematically and biologically meaningful.

2 General framework for the solution

2.1 Formulation of the problem

Each measured tissue sample is composed of a mix of cells of different types (*e.g.* fibroblasts, epithelial cells...). We are quantifying the total mRNA coming from this pool. Since the mRNA quantities from each cellular type (CT) simply add up (considering the measurement system as linear), the measures made are simply a mix (linear combination) of the measures we would have with each CT alone. The relative importance given to each CT is proportional to its concentration in the sample.

We will suppose that we have a set of measures concerning many samples (where many is much more that the number of CT) which contain the same CT in various concentrations. These variations in the concentrations will allow us to infer the signature (profile of gene expression) of the pure CT. We will also suppose, in all of the following, that we know the number of CT.

The problem is formalized using three concepts, expressed as three matrices (*e.g.* see table 1):

M : matrix of measures, containing as many lines as there are genes and as many columns as there are measures.

G : signature of the cellular types. It is a matrix with as many lines as there are genes, and as many columns as there are cellular types.

C : concentration matrix, with as many lines as there are cellular types and as many columns as there are measures.

Definition A cellular type (CT) is a column of the matrix G.

Table 1 Example data.

Example of G (2 cellular types, 3 genes):

	Cellular type 1	Cellular type 2
Gene 1	30	80
Gene 2	50	10
Gene 3	20	10

Example of C (2 cellular types, 4 samples):

	Sample 1	Sample 2	Sample 3	Sample 4
Cell. Type 1	50%	20%	30%	70%
Cell. Type 2	50%	80%	70%	30%

Example of M (4 samples, 3 genes):

	Sample 1	Sample 2	Sample 3	Sample 4
Gene 1	55	70	65	45
Gene 2	30	18	22	38
Gene 3	15	12	13	17

The measurements for each sample are only meaningful to a multiplicative constant. If for instance the amount of material measured varied, every result would be multiplied by a certain value. This multiplication is biologically meaningless. To remove this un-determination, we decided to normalize the measures such that the sum of the values for every sample is N, the number of genes:

$$\sum_{i=1}^{N} M_{ij} = N$$

We can write each measure as a function of the signature of the CT and of their concentrations in the samples:

$$M_{ij} = \sum_{k}^{Nct} G_{ik} C_{kj} \qquad (1)$$

where M_{ij} is the measure of the gene i in the sample j, G_{ik} is the expression of gene i in cell type k, C_{kj} is the concentration of the cellular type k in sample j and Nct is the number of CT.

This can also be written in matrix notation:

$$\mathbf{M} = \mathbf{GC}$$

The goal is to infer the matrices G and C from the matrix M. Stated as it is, this problem is under-determined. However, the matrices G and C have to obey certain physical constraints.

Constraints on G:

$$G_{ik} \geq 0$$

$$\forall k : \sum_{i=1}^{N} G_{ik} = N \qquad (2a), (2b)$$

(2a) states that the expression for each gene should be non-negative. (2b) states that a measure on the pure CT should have the same normalization as the matrix M.

Constraints on C:

$$C_{kj} \geq 0$$

$$\forall j : \sum_{k} C_{kj} = 1 \qquad (3a), (3b)$$

(3a) states that the concentrations should be non-negative. (3b) states that the sum of the concentrations of the cell types should be equal to 1 for every sample, i.e. that there is nothing in the samples but the cell types considered.

We add two assumption to this in order to obtain a better-defined problem. These assumptions should not be a problem in real cases:

1. The CT (i.e. columns of G) are linearly independent.
2. Their is a square sub-matrix of C of size equal to the number of CT which is invertible.

With these assumptions, we have a first lemma:

Lemma 1 Let there be a real solution, $\mathbf{M} = \mathbf{GC}$ respecting the two assumptions.
Then each calculated CT of any solution of (1) is a linear combination of the real CT.

Proof In this proof, we will only keep the samples corresponding to the invertible part of C. On this sub-part, we have G=M/C. Let G*, C* be another solution to the same problem. We have G=G* C*/C. The columns of G are a linear combination of the columns of G*. Since the columns of G are linearly independent, so must be the columns of G. So C*/C must be invertible. Hence G* = GC/C* = GT, which is to say that the CT of G* can be expressed as a linear combination of the CT of G.

2.2 Conditions to have unicity of the solution

It would be convenient to have only one solution to the problem. We will now show under which conditions this unicity is guaranteed.

Definition A marker is a gene which is present only in one CT. We suppose that each CT has at least one marker:

$$\forall j \exists i \mid G_{ij} \neq 0 \text{ and } G_{ik} = 0 \text{ if } k \neq j$$

Lemma 2 Let there be a real solution, $\mathbf{M} = \mathbf{GC}$
Let each CT have at least one marker.
Then each calculated cellular type of any solution of (1) is a linear combination with positive coefficient of the real CT.

Proof Using lemma 1, we can write any solution G* as a linear combination of G:

$$G^*_{ij} = \sum_{k=1}^{Nct} G_{ik} T_{kj}$$

Where Nct is the number of CT. Considering the marker for the CT m (say gene n):

$$G^*_{nj} = G_{nm} T_{mj}$$

Since $G_{nm} > 0$ and $G^*_{nj} \geq 0$, T_{mj} has to be non-negative for every m and j.

Theorem Let there be a real solution, $\mathbf{M} = \mathbf{GC}$
Let each CT have at least one marker.
Let the correlation between the CT of G be zero.
Then for any solution G* of (1) with non-correlated CT

$$G^*_{ij} = \sum_{k}^{Nct} G_{ik} T_{kj} \qquad (4)$$

the matrix T is a permutation of the identity.

Proof Applying (2b) to (4)

$$\sum_{i=1}^{N} G^*_{ij} = \sum_i \sum_k G_{ik} T_{kj} = \sum_k T_{kj} \sum_i G_{ik} = N \sum_k T_{kj} = N$$

so

$$\sum_{k=1}^{Nct} T_{kj} = 1 \qquad (5)$$

We supposed that the CT of G are not correlated:

$$\sum_k (G_{ki} - 1)(G_{kj} - 1) = 0 \qquad \text{if } i \neq j$$

hence

$$\sum_k \left(G_{ki}G_{kj} - G_{ki} - G_{kj} + 1\right) = \sum_k \left(G_{ki}G_{kj}\right) - N - N + N = 0$$

so

$$\sum_k \left(G_{ki}G_{kj}\right) = N \qquad \text{if } i \neq j \tag{6}$$

We can now write the no-correlation condition on G*:

$$\sum_k \left(G^*_{ki} - 1\right)\left(G^*_{kj} - 1\right) = 0 \qquad \text{if } i \neq j$$

Replacing with (4)

$$\sum_k \left(\sum_l G_{kl}T_{li} - 1 \right)\left(\sum_m G_{km}T_{mj} - 1 \right) = 0$$

$$\sum_k \left(\sum_{lm} G_{kl}T_{li}G_{km}T_{mj} - \sum_l G_{kl}T_{li} - \sum_m G_{km}T_{mj} + 1 \right) = 0 \tag{7}$$

We will calculate each term of (7) successively. First term of (7):

$$\sum_k \left(\sum_{lm} G_{kl}T_{li}G_{km}T_{mj} \right) = \sum_{lm} T_{li}T_{mj} \sum_k G_{kl}G_{km} = \sum_l T_{li}T_{lj} \sum_k G_{kl}^2 + \sum_{l \neq m} T_{li}T_{mj} \sum_k G_{kl}G_{km}$$

Using (6)

$$= \sum_l T_{li}T_{lj} \sum_k G_{kl}^2 + \sum_{l \neq m} T_{li}T_{mj}N = \sum_l T_{li}T_{lj} \sum_k G_{kl}^2 + \sum_{lm} T_{li}T_{mj}N - \sum_l T_{li}T_{lj}N$$

$$= \sum_l T_{li}T_{lj} \sum_k \left(G_{kl}^2 - 1\right) + N \sum_l T_{li} \sum_m T_{mj}$$

Applying (5)

$$= \sum_l T_{li}T_{lj}N\sigma_{Gl}^2 + N$$

where σ_{Gi}^2 is the variance of the cellular type i.

We can now calculate the second term of (7), using (2b) and (5):

$$\sum_k \sum_l G_{kl}T_{li} = \sum_l T_{li} \sum_k G_{kl} = \sum_l T_{li}N = N$$

The third term of (7) gives the same result as the second, so we can put everything together:

$$\sum_k \left(\sum_{lm} G_{kl}T_{li}G_{km}T_{mi} - \sum_l G_{kl}T_{li} - \sum_m G_{km}T_{mi} + 1 \right) = \sum_l T_{li}T_{lj}N\sigma_{Gl}^2 + N - N - N + N$$

$$= \sum_l T_{li}T_{lj}N\sigma_{Gl}^2 = 0 \qquad \text{if } i \neq j$$

By the lemma 2, every value in T has to be positive. By the lemma 1, T must be non-singular; hence, each line and column of T contains at least one non-null entry. Say $T_{nm} > 0$:

$$\sum_l T_{lm}T_{lj}N\sigma_{Gl}^2 = 0 \qquad \text{if } m \neq j$$

The only way to respect this is to have $T_{nj} = 0$ for $j \neq m$. So each line of T contains one and only one value. Since each column contains at least one value and T is a square matrix, each column contains one and only one value. With (5), the sum of the values of a column of T, *i.e.* the only value present, is 1. Hence, T is a permutation of the identity matrix.

3 Algorithms

Many different approaches can be used to solve our problem. We will present a direct method and discuss two other methods we have tried but seemed less satisfactory.

3.1 Direct solution

The idea is to solve the problem by using a least square criterion: we search two matrices G and C which minimize the norm of the reconstruction error

$$\|\mathbf{M}-\mathbf{GC}\|^2 \tag{8}$$

subject to the constraints (2a), (2b), (3a) and (3b).

The algorithm is simple:

1. Considering G as known, calculate C minimizing (8) subject to (3a) and (3b)
2. Considering C as known, calculate G minimizing (8) subject to (2a) and (2b)

These two steps are performed sequentially until convergence. This algorithm is guaranteed to converge to a minimum (local or global) because the error (8) decreases at each step and there is a lower limit for the value of the error.

We take into account the positivity constraints (2a) and (3a) by solving (8) with the Matlab function nnls, non-negative least squares. The normalization constraints (2b) and (3b) are applied by dividing each column of either G or C after each iteration. These normalizations could affect the convergence of the algorithm, but in practice their effects are small enough.

It remains to add the uncorrelation constraint. Since the uncorrelation is a strong unproven hypothesis, we introduce this constraint in a relatively soft fashion. After each calculation of G, we subtract from each CT a fraction of the values of every other CT to which it is correlated. This fraction is proportional to the correlation and to a constant alpha to be chosen:

$$\mathbf{G}' = \mathbf{G} - \alpha\mathbf{G}(corrcoef(\mathbf{G}) - \mathbf{I})$$

this will tend to de-correlate the calculated CT. Alpha is a key parameter as the following will show. The convergence is not guaranteed anymore with this new step since it usually lead to a raise in the error (8), but this is usually not a problem in practice when α is sufficiently small. The solutions obtained with this new step do not minimize (8) anymore, but hopefully are closer to the real ones.

3.2 Other possibilities

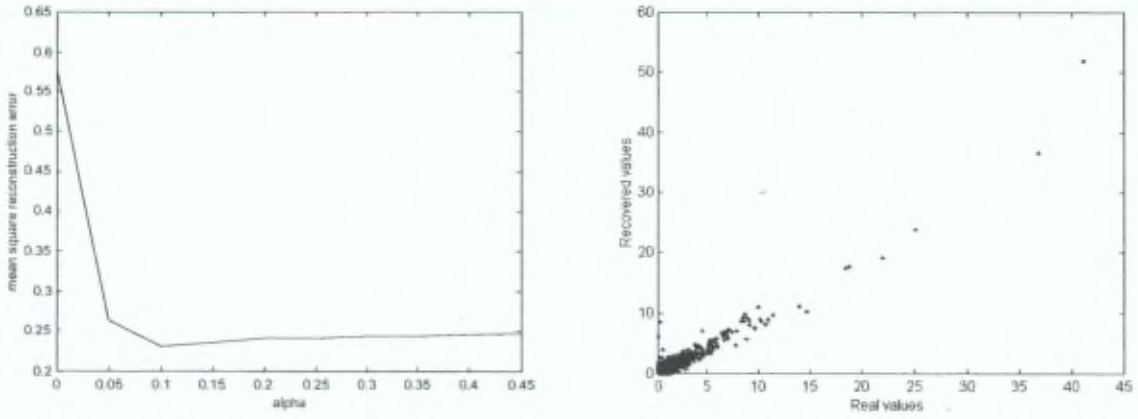
3.2.1 Principal component and factor analysis

We could view the separation in CT as a kind of dimensionality reduction: a certain number of samples are described as a linear combination of a smaller number of CT. Two main approaches exist in the literature for dimensionality reduction: principal component analysis and factor analysis. Both find a set of orthogonal vectors whose linear combinations fit optimally the data in a certain sense. The CT will then be the linear combination of these vectors which respects as well as possible the conditions (2a), (2b), (3a) and (3b).

Algorithms based on these techniques present the following shortcomings:

1. The positivity constraints (2a) and (3a) are not directly taken into account, while they are important for the unicity of the solution.
2. In real life, we expect other effects to be superimposed to the separation in CT – e.g. the fact that the samples are not the same or the presence of a cluster of heat shock proteins. These effects probably cannot be expressed as a CT with only positive coefficients, but could explain a relatively important proportion of the variance and hence be much more detrimental if the positivity constraints are not directly taken into account.
3. The assumption of orthogonality, which is central to these techniques, is probably most often incorrect.

On simulated results with uncorrelated CT, an algorithm based on principal component analysis gave results of a quality comparable to the first algorithm, although much faster. With very noisy data, or with correlated CT, the quality of the solutions degraded faster than with the direct method.



A. B.

Fig 1.**A**. Mean square difference between the known and the recovered CT. **B**. Plot of the recovered values against the real values. Each point represent a gene in a CT, the x-axis is the expression of the gene in the real CT while the y-axis is the recovered expression of the gene in the calculated CT.

3.2.2 Projected gradient

The idea here is to start with a first solution respecting (2a), (2b), (3a) and (3b) with uncorrelated CT and to try to minimize the error (8) while still respecting the conditions and the no-correlation.

We did not use this technique because, although it looks promising, it takes ages to converge and tends to get stuck in local minimum. In addition, the strict no-correlation hypothesis, without which this technique is similar to the direct method, is probably too strong.

We think this type of technique should be the most effective, would the technical problems be solved. In the mean time, we refrain from using it.

4 Simulation

The CT were taken from the data in (Perou 2000) concerning five breast cancer cell lines (T47D, RPMI-8226, 184A1, HUVEC, NB4+ATRA). These cell lines were chosen because they are relatively uncorrelated, with correlation ranging from –12.2% to +15%. 1000 genes out of the 8999 were kept. The values for each cell lines were normalized to a mean of 1. An artificial mixing matrix for these 5 CT was created. It was supposed that 40 samples were measured. The values for the mixing matrix were generated from a uniform distribution [0 1[. Each column of the matrix was then multiplied by a constant in order to satisfy (3b). The artificial measure matrix was generated as the product of these two matrices. Noise was then added as a sum of a multiplicative noise (for the biological variations, which we suppose are proportional to the values) and an additive noise (for the measurement errors, supposedly independent of the quantity measured):

$$v' = v \cdot N(1,0.3) + N(0,0.2)$$

where v is the original value, v' the noised value, and $N(a,b)$ is a value drawn from a normal distribution of mean *a* and standard deviation *b*. The resulting values below 0 were put to 0 and the columns of the resulting matrix were normalized to a mean of 1. With this setup, the average absolute value of the error introduced is 29% of the mean of the original values.

The separation in CT was then performed as described previously with the direct method, with various alpha. The quality of the separation, expressed as the mean squared difference between the recovered CT and the real CT, is shown in figure 1.a. as a function of alpha. A small alpha can dramatically improve the quality of the separation. This amelioration

can be linked to the unicity result. If the solution is uncorrelated the forcing via alpha assure the unicity of the solution. The algorithm is bound to find a solution closer to the real one. When alpha is too high, the decorrelation gets too stringent. The algorithm tries to overly decorrelate the CT, which are slightly correlated. Would the CT be more correlated, the influence of alpha would start to be detrimental for a lower value of alpha.

In figure 1.b. we show the recovered values for every gene in every CT compared with the real values for the best alpha. Even in this very noised setup, the algorithm is able to recover for a large part the signature of the CT.

5 Correlation

As shown before, the no-correlation hypothesis is very helpful to assure the unicity of the solution. Besides, all algorithms but the direct method necessitate un-correlated CT to function properly. The questions asked here are whether this hypothesis is valid or not, and how to handle the case where it is not.

5.1 Validity of the no-correlation hypothesis

The correlation between CT is affected by two things, the technology used for the measurements and the normalization used.

Gene expression measurements can be absolute or relative, depending of the technology used. Absolute values are a direct quantification of the mRNA for each gene (such as in Affymetrix technology), while relative values are the ratio of the absolute values by the absolute values of another sample, the standard (such as in microarrays). The correlations are affected by this difference. In absolute values, it is expected that the genes necessary for the functioning of any cell ("housekeeping genes") are expressed at a comparable level in every CT. Hence, the CT are usually correlated in absolute value. In relative value, correlation is a function of the standard used. If the standard is unrelated to two CT, the correlation between them will be raised by the application of the standard. If the standard is related to at least one of two CT, the correlation between these two CT will be lowered.

Since the genes are not all expressed at a similar level, it is tempting to normalize the values. A common mean of normalization is to divide the values for each gene by the mean of the values for this gene across all experiments. The result obtained is similar to the use of a standard consisting of a mean of the samples. Since by hypothesis the samples are a linear combination of the CT, this standard can also be considered as a mix of the CT. Hence the normalized CT are anti-correlated on average, even though the correlation might have been positive before.

In conclusion, the correlation between the CT should be:

- Positive if the measures are absolute (SAGE or oligochips experiments).
- Undetermined if the measures are relative to a standard (microarray experiments).
- Negative if the measures are normalized.

5.2 Experimental verification

To check if our hypotheses about the correlation between CT were correct, we applied the direct method algorithm to separate real absolute value data (Alon 1999) into CT. The resulting CT were correlated to each other, with a mean correlation of 22%. We normalized the measures gene-wise, and separated again. This time, the correlation between the calculated CT where negative, with a mean correlation of –25%. This shows that the correlations between the CT, as captured by the first algorithm, behave as expected: they are positive on average for absolute measures, and negative after normalization.

5.3 De-correlation of the CT

When tackling the problem of the correlation of the CT, we are facing the following facts:

1. The CT in the original data are usually correlated.
2. If the data are normalized gene-wise, the resulting CT are anti-correlated.
3. The separation in CT using the direct method gives an estimate of the correlations between the real CT.

We have a disease (the positive correlations), a scalpel (the normalization) and a visualization apparatus (the separation in CT using the direct method). A treatment is possible.

To blindly de-correlate the CT, we try to find a normalization of the data gene-wise for which the mean correlation between the recovered CT is minimum. The idea is to normalize using a standard which looks like the mean of the samples (to de-correlate the CT) but not too much (to avoid the anti-correlation). The following transformation was chosen:

$$M'_{ij} = \frac{M_{ij}}{\left(\sum_{i=1}^{NSample} M_{ij} \right)^{v}}$$

where the constant v determine the transformation. For v=0, no transformation is applied while for v=1 the values are normalized with the mean. In order to de-correlate the CT, we search for the value of v between 0 and 1 for which the average correlation between the recovered CT is zero.

5.4 Simulation results

We generated a set of 5 CT, with the correlation between the CT varying between 22% and 61% with a mean of 43%. We generated a matrix of concentration, calculated the matrix of measures and noised this matrix. We then calculated the de-correlation transformation.

Applying this transformation to the real CT, the modified correlations ranged between –32% and +24%, with a mean of –5%. Therefore, on average the modified CT were not correlated anymore, even though relatively large single correlation could still be found.

We tried to separate with the direct method the original data and the de-correlated data. The mean square error on the recovered CT was 0.49 on the original data, and 0.35 on the de-correlated data. This improvement in the quality of the separation can be linked to the unicity result. If the CT are uncorrelated, the algorithm can be told (via alpha) to find uncorrelated CT. The separation is then unique, which helps the algorithm.

6 Results with real data

6.1 Colon cancer data

Alon et al. 1999 have generated data with the Affymetrix technology (absolute measures) on various colon cancers and adjacent tissues. In order to see if a distinction between cancer and normal tissue was apparent from such data, they set up a separation, but this separation was mostly a function of the presence of muscle cells. Those cells were highly present in the normal samples but not in the cancer samples. Based on known biological markers, they designed a "muscle index" which gives an estimation of the amount of muscle tissue present in any sample. This muscle index was indeed usually higher in the normal samples.

In order to validate our technique on these data, we tested three things:

1. That the separation was numerically meaningful, which was checked through bootstrapping.
2. That the separation was biologically meaningful.

3. That we could retrieve the muscle index as determined by Alon et al.

6.1.1 Numerical validation

Table 2 Numerical validation of the method: number of separations with one CT per cluster.

Number of CT	Original data	De-correlated data	Random data
3	80%	96%	21%
4	80%	87%	8%
5	51%	69%	4%

The real data consist of measurements of 1988 genes on 62 samples. In order to see if the separation in CT was a property of the data or just an artifact, we separated these data 100 times using only random subsets of 500 genes. The idea is that the matrices of concentration (C) obtained in all these separations should be close. To check that, we clustered them. If the separation was meaningful, we should have one CT per cluster for each separation. We looked at the fraction of the runs for which this was the case.

As a comparison, we generated a random measurement matrix with sample correlations of about the same magnitude than for the real data, and performed the same calculations.

The separations were done on 3, 4 and 5 CT, with the direct method algorithm. They were performed on the original data, the de-correlated data (with the algorithm presented supra) and the random data. The results are shown in table 2.

This validation shows that the separations are really a property of the measurement matrix. It shows as well that the blind de-correlation of the CT seems to have a positive effect on the separation of real-world data.

Table 3 Markers found in the cellular types.

Cellular type 1		**Cellular type 2 – muscle**	
L33930	b-cells	X86693	SPARC-like 1 (mast9, hevin)
H81864	renal tumor antigen	M63391	muscle cells
M26383	released in response to an inflammatory stimulus	L05144	liver-kidney-adipocytes
D78152	annexin A4	U25138	potassium large conductance calcium-activated channel
T56940	ribosomal protein S5	M26683	chemotactic factor - attracts monocytes and basophils
H87344	ferritin, light polypeptide	X74295	integrin, alpha 7
T49647	---	H06524	phagocytic-platelets-fibroblasts-nonmuscle-muscle
M19045	lysozyme (renal amyloidosis)	M36634	vasoactive intestinal peptide
J02763	calcium-binding protein A6 (calcyclin)	M64110	smooth muscles
D00760	proteasome	U19969	heart and skeletal muscle
H17897	mitochondrial carrier	R48303	dermatopontin
M86553	cathepsin S	H77597	metallothioneins
T70062	interleukin enhancer binding factor 2	X16356	carcinoembryonic antigen-related cell adhesion mol 1
H17969	galectin 6 binding protein	X12369	muscle contraction
R44770	---	D31716	transcription factor
T54303	keratin 8	H43887	adipsin
X80507	---	R44301	aldosterone receptor
T58861	ribosomal protein L30	X68277	dual specificity phosphatase 1
M26481	tumor-associated calcium signal transducer 1	D15049	protein tyrosine phosphatase, receptor
M25108	integrin, beta 3 (platelet glycoprotein IIIa, antigen CD61)	T92451	tropomyosin 2
Cellular type 3 – leukocyte		**Cellular type 4**	
J00231	immunoglobulin gamma 3	Z25521	very broadly distributed on normal adult tissues
M87789	immunoglobulin gamma 3	U26312	chromobox homolog 3
R62549	protein kinase	D00596	thymidylate synthetase
M27749	expressed only in pre-b-cells and a special b-cell line	H17434	nucleolin
T51558	fibrils of tendon, ligaments and bone	R62945	decay accelerating factor for complement
X12876	keratin 18	R11485	proteasome (prosome, macropain) subunit
T72175	Immunoglobulin kappa constant	X13482	small nuclear ribonucleoprotein polypeptide A'
T54767	regul cell growth-morphogenesis-remodeling-wound repair	L38951	karyopherin (importin) beta 1
T57780	immunoglobulin lambda locus	X56597	fibrillarin
D13665	osteoblast specific factor 2	X53586	predominantly expressed by epithelia
Z46389	vasodilator-stimulated phosphoprotein	U04953	isoleucine-tRNA synthetase
M60335	inflamed vascular endothelium - macrophage-like - dendritic	R75843	eukaryotic translation initiation factor 2G
U30498	preferentially expressed in adult hematopoietic tissues	Z29677	Ras homolog enriched in brain 2
T57780	immunoglobulin lambda locus	X70040	keratinocytes
T41204	normal alveolar macrophages and granulocytes	X01060	transferrin receptor
T56350	nucleolin	M31516	decay accelerating factor for complement
R39010	---	H92195	RAN binding protein 7
T62067	---	X54942	CDC28 protein kinase 2
L26494	POU domain, class 3, transcription factor 1	M58050	expressed by almost all cells
L20688	egulates gdp/gtp exchange reaction of the rho proteins	M31516	decay accelerating factor for complement

6.1.2 Biological identification

In order to identify the biological significance of the found CT, we tried to identify which genes could be used as markers. We relaxed the definition of a marker used in part 3. A marker here is a gene that is expressed mostly in one CT. To find them, we divided each row (i.e. gene) of the matrix G (containing the signature of the CT) by the sum of its values. After this transformation, the resulting values are between 0 and 1, a value close to 1 meaning that the gene is expressed mostly in one CT, and so is a good marker. The genes with the highest marker score for each CT could then be identified, and with the information relative to these genes in the literature we tried to assign a meaning to the CT.

Among the various CT recovered, some could be given a clear meaning. In table 3, the 20 genes with the highest marker score for the CT obtained with a separation in 4 CT are shown. The identifications for every gene were obtained with GeneCard (Rebhan 1997).

We looked at the occurrences of patterns in the gene descriptions which could be considered as pertinent to identify certain biological CT: "muscle" or "fibroblast" for the "muscle" cells and "immunoglobulin", "b/t-cell", "hematopoietic" or "macrophage" for the leukocytes. We tried to assess if the distribution of these patterns could be random via a Monte-Carlo simulation (random permutations of the labels). See table 4 for the resullts.
These probabilities show that the results cannot reasonably be due to pure chance. So, at least some expression profile can be assigned clearly to a cell type.

Table 4 Distribution of certain patterns in the markers of the CT for a separation in 4 CT.

Pattern	CT1	CT2	CT3	CT4	Probability
"muscle" or "fibroblast"	0	5	0	0	0.24%
"immunoglobulin" or "b/t-cell" or "hematopoietic" or "macrophage"	1	0	9	0	0.002%

6.1.3 Link with the muscle index

Since we were able to identify one of the CT as representative of muscle tissue, we could compare our results with the muscle index of Alon. This index estimates the amount of muscle tissue in a sample, so it should be correlated with the concentration of the muscle CT through the samples as determined by our algorithm. We plotted our concentration against the index (see fig. 2). The correlation between both is 89%, so there is a good agreement between our results and those from Alon.

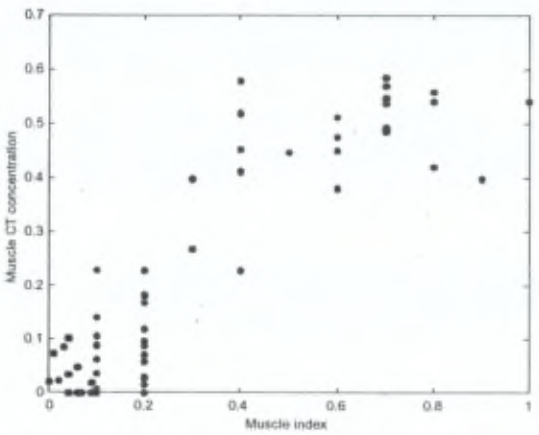


Fig 2. Concentration of the calculated muscle CT as a function of the muscle index given in (Alon 1997).

6.2 Ovarian cancer data

Welsh et al. 2001 have measured 27 ovarian cancer samples with the Affymetrix technology (absolute measures). These samples contained a variable fraction of stromal tissue, as well as infiltrated lymphocytes sometimes. A part was taken from each of these tumors for histological analysis, which permitted to estimate the tumor composition. We tried to see if our algorithm could recover these estimates.

The data concerning the tumors were separated in 4 CTs. The numerical stability of the solutions is comparable to the one obtained with the colon tumors. With the descriptions of the genes, it was possible to identify one CT as leukocyte and another one as smooth muscle, although the statistical significance of those results was lower than for the colon cancer data.

The concentrations in leukocyte were given in the paper as "abundant", "rare" or "0". The average concentrations as given by our algorithm for these three categories were 18%, 17% and 11%, respectively. However, huge discrepancies could be found, for instance a sample in the "abundant" category had a concentration of only 11%, while a sample in the "rare" category had a concentration of 69%. The estimates in Welsh were based on the quantity of cells, and not of mRNA, and the samples examined were not exactly the same as the samples on which gene expression profiling were performed, so it might be that our estimates are indeed correct. To check that, we created a (very rough) leukocyte index, as the mean expression of all the genes with "immunoglobulin" in their description. This index presented larger discrepancies with Welsh's estimates than our concentrations. Our concentrations were however correlated to this index, with a coefficient of 75%.

The concentration of stromal tissue is given as a percentage in Welsh. The correlation between our concentrations, after suppression of the effect of the leukocytes, and theirs is 69%. To see again if estimates based on gene profiles would be more closely related to our concentrations, we created an estimator as the mean of the expression of all the genes belonging to the "stromal" cluster in Welsh. This new estimator correlates reasonably with the concentrations in Welsh (correlation: 69%) but the correlation is much better with our concentrations (correlation: 85%).

7 Conclusion

An approach and a set of algorithms are presented allowing to mathematically separate samples consisting of many cellular types into their constituents. This advance should make it possible to treat experimental cases which seem out of reach without complex biological methods.

The techniques shown have some weaknesses that should be addressed in the future. The blind de-correlation of the CT is an important part of the techniques and is presently only a very rough method which can certainly be improved. An algorithm should be developed to automatically determine the number of CT out of the data.

A more thorough biological validation of the technique could also be carried out, by verifying if the genes predicted as being markers were really only expressed in the predicted CT. This could be checked using in situ hybridization.

Nevertheless, even with those limitations and uncertainties, the techniques presented can already be an important help for researchers having to deal with complex cases of cell populations composition, where it is never clear in which cellular type a given gene is expressed.

We view this work as a step toward applying mathematical theories and computer sciences techniques to biology, and allowing to extract hidden relevant information from the huge set of data produced by modern biology.

8 Bibliography

Alizadeh A.A., Eisen M.B., Davis R.E., Ma C., Lossos I.S., Rosenwald A., Boldrick J.C., Sabet H., Tran T., Yu X., Powell J.I., Yang L., Marti G.E., Moore T., Hudson J., Lu L.,

Lewis D.B., Tibshirani R., Sherlock G., Chan W.C., Greiner T.C., Weisenburger D.D., Armitage J.O., Warnke R., Staudt L.M., et al 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403:** 503—11.

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D. and Levine A.J. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA* **96:** 6745—6750.

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* **95:** 14863—14868.

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

Perou C.M., Sorlie T., Eisen M.B., van de Rijn M., Jeffrey S.S., Rees C.A., Pollack J.R., Ross D.T., Johnsen H., Akslen L.A., Fluge O., Pergamenschikov A., Williams C., Zhu S.X., Lonning P.E., Borresen-Dale A.L., Brown P.O., Botstein D. 2000. Molecular portraits of human breast tumours. *Nature* **406:** 747—52

Rebhan, M., Chalifa-Caspi, V., Prilusky, J., Lancet, D.: GeneCards: encyclopedia for genes, proteins and diseases. Weizmann Institute of Science, Bioinformatics Unit and Genome Center (Rehovot, Israel), 1997. World Wide Web URL: http://bioinfo.weizmann.ac.il/cards

Welsh, J.B., Zarrinkar, P.P., Sapinoso, L.M., Kern, S.G., Behling, C.A., Monk, A.J., Lockhart, D.J., Burger, R.A., Hampton, G.M. 2001. Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer. *Proc Natl Acad Sci USA* **98:** 1176—1181.

6 - Genetic network inference

1 Introduction

The availability of a large amount of gene expression data has raised the hope that a complete regulatory network in a cell type could be inferred in a systematic and comprehensive way. Different models of such networks exist. The most common ones are Boolean networks [1,8], qualitative model [14], Bayesian networks [12], weight matrices [4,7,15], systems of linear or non-linear differential equations [3,16] and hybrid models [2,10]. This work focuses on Boolean networks.

In the Boolean network model, each gene has only two possible states, "on" and "off". The state of every gene is a Boolean function of the states of some other genes. Usually, the number of genes necessary to determine the state of a gene is limited in order to avoid overly complex solutions.

This model, like most other models of gene regulation, has an important limitation. The variables which control the expression of the genes are supposed to be gene expressions themselves. In real biological systems, this is often not the case. Gene expressions can also be controlled by the concentration and the activity of proteins, or by properties of the cell environment (glucose concentration, temperature...). Those parameters are hard to introduce in the framework of a Boolean network in a clean and consistent manner. They can sometimes be estimated using *a priori* knowledge of the experimental conditions, but this is not always possible. For instance, in a study relative to the temporal evolution of gene expression during the cell cycle, the experimental conditions should be labeled as a function of state of the cells. Such labeling is hard to do, and prone to error and bias.

We propose here a modification of the Boolean network paradigm which addresses this issue while still keeping its inherent simplicity (figure 1). In a Boolean network (figure 1A), each gene is a binary function of other genes, or of an external parameter like temperature. In order to find the corresponding "binary switch model", the elements controlling the network (genes or external) are singled out, and called "switches". The evolution of the system is by definition controlled by, and only by, those switches. This model is a generalization of the Boolean network model. When the network model is applicable, the switches can be assimilated to the expression of particular genes. This generalization allows the treatment of a much wider panel of experiments in a systematic fashion, as external parameters can be included naturally.

The identification of this model is a three-fold process: first the genes which depends on the same switches must be grouped. The is equivalently to a clustering. Setting a model of gene regulation imposes the existence and the mathematical form of this clustering. The values of the switches and the dependence between switches and groups of genes are then determined. FInally, if possible, the switches can be identified with genes.

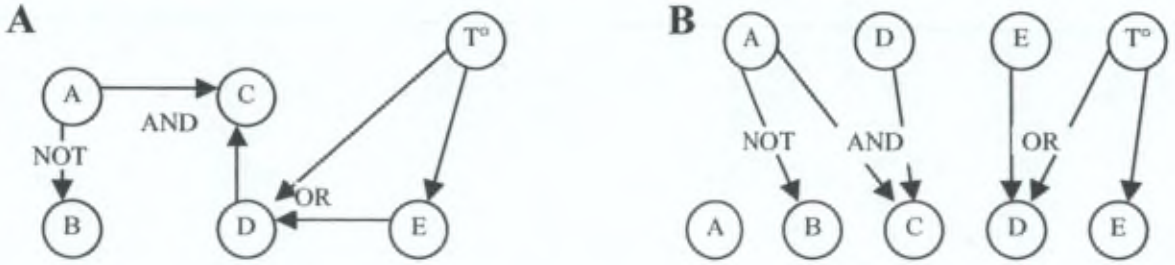


Figure 1. **A**. Boolean network. A, B, C, D and E are genes, T° is temperature. **B**. The corresponding binary switch model. The switches are on the upper line, the controlled genes on the lower line.

In order to apply the binary switch model to real biological data, a binarization must be performed. The same applies of course to other models which use discrete values, like Boolean and Bayesian networks. Such discretization is usually done by another, independent, algorithm [12]. We show here that the binarization can be performed at the same time than the determination of the regulations. This approach leads to a better definition of the problem, and potentially to better solutions.

We firstly present the binary switch model in more detail and discuss some of its implications. Secondly, we show a technique to infer the parameters of the model from the data. Thirdly, we demonstrate this technique on simulated data. Finally, the models are determined for two real data sets, showing the applicability of the method.

2 The binary switch model

The binary switch model describes the regulation of the genes in a simple and understandable way. We present here the hypotheses behind this model in some detail.

Gene expression data can be organized as a matrix, G. A value g_{ij} of this matrix corresponds to the level of expression of the gene i in the experimental condition j. A first hypothesis is that the matrix G can be reduced to a binary matrix, B. This means that there is a threshold t_i for each gene that can be used to binarize the matrix G:

$$b_{ij} = g_{ij} > t_i \qquad (1)$$

where b_{ij} is the binary expression of gene i in condition j. Its value is "1" if g_{ij}, the real valued expression of gene i in condition j, is higher than a gene specific threshold t_i and "0" otherwise.

A second hypothesis is the existence of N binary switches. The switch k has the value sw_{jk} in condition j. Those switches describe the regulatory state of the cells. They could be a function of the presence or absence of a given protein, or of its activity, or of high temperature, or of anything else.

Finally, the model implies that the binarized values of gene expression are Boolean functions of the switches:

$$b_{ij} = \mathrm{f}_i\left(sw_{j1}, sw_{j2}, \ldots, sw_{jN}\right) \qquad (2)$$

where f_i is the function linking the value of the gene i to the switches.

This model is a generalization of the Boolean network model. In the network model, it is supposed that the states of the genes are a function of the state of other genes. When the Boolean network model is applicable, the switches can be assimilated to the expression of certain genes. The binary switch model generalizes the Boolean network model to the case where the variables responsible for the evolution of the profile of gene expression are not measured. Since this is often the case, the binary switch model is more realistically applicable than the Boolean network model.

Table 1. Example of genes respecting the binary switch model. "Val" are the real values measured. "Bin" are the binarized values: "1" if the real value is over the gene's threshold ("Thresh") and "0" otherwise. Gene 1 is Switch 1. Gene 2 is NOT(Switch 1 AND Switch 2). Gene 3 is XOR(Switch 1, Switch 2). Gene 4 is NOT(Switch 2).

	Exp 1		Exp 2		Exp 3		Exp 4		Exp 5		Exp 6		Exp 7		Exp 8		
Switch 1	1		1		1		0		0		0		1		0		
Switch 2	1		0		1		0		1		0		1		0		
	Val	Bin	Val	Bin	Val	Bin	Val	Bin	Val	Bin	Val	Bin	Val	Bin	Val	Bin	Thresh.
Gene 1	4.1	1	3.5	1	5.1	1	2.0	0	1.5	0	.23	0	2.7	1	1.1	0	2.35
Gene 2	.5	0	1.3	1	.21	0	1.2	1	.9	1	2.1	1	.17	0	1.5	1	0.7
Gene 3	.5	0	1.2	1	.8	0	.4	0	1.7	1	.7	0	.8	0	.1	0	1
Gene 4	1	0	1.3	1	.5	0	1.4	1	.3	0	8.2	1	.1	0	1.5	1	1.2

The inference of the switches from the data makes it possible to determine a large part of the regulatory network. In order to determine the rest of this network, the switches must be tentatively identified with an underlying biological reality. If this biological reality is not among the quantities measured, then no identification is possible.

In the special case of a kinetic study (temporal evolution), if a regulation is performed via a gene regulated at the mRNA level, then it should be possible to identify a switch with this gene. In that case, the switch represents the activity of a transcription factor while the gene expression measured is the corresponding mRNA level. The evolution of the activity of a gene should be similar to the evolution of its mRNA level, with a certain delay caused by the time taken for the translation, the folding and sometimes the activation of the corresponding protein. Since the activity of a transcription factor should be represented as a switch, such switch should be correlated with the level of expression of the gene, with a certain delay. Hence, the gene could be identified using its delayed correlation with a switch.

Methods similar to this one have been proposed to identify Boolean networks. In these approaches, the complexity of the identification is much higher, since many possible links are taken into consideration. Also, the delays between the cause and the effect are usually arbitrarily set to the time between the measurements taken, which limits the applicability of the techniques. And of course, those methods are not resilient at all to missing values.

The model presented has implications concerning the maximum possible number of different experimental conditions and gene profiles. For a given number N of switches, there are at most 2^N different experimental conditions. The same limitation applies to Boolean network models as a function of the number of transcription factors. This maximum number of conditions could be considered as too high or too low, depending on the point of view.

In the framework presented here, the experimental conditions which have the same combination of switches must be grouped for the determination of the switches. This is done by clustering the samples in at most 2^N groups. In order to render this clustering meaningful, the number of experimental conditions should be much higher than the number of groups, hence usually higher than 2^N. This means that the number of experiments needed to identify the parameters of the model grows exponentially with the number of switches, and so must be quite large for even a moderate number of switches.

However, taking a different point of view, the maximum number of conditions can be considered as low. Each experiment being done in a different setup, the results are different. If the switches are supposed to explain the cell's entire behavior, their number should be such that 2^N is higher than the number of experiments. But then, no identification is possible using the framework presented.

In this work, we consider that the number of switches is such that 2^N is much smaller than the number of experimental conditions. The hypothesis is that there exists a sufficient amount of similarity between the conditions with the same values of the switches for those switches to predict the behavior of many genes. We do not consider that we are in a situation

where everything can be deduced from the measurements, but that switches nevertheless exist, which explain a large part of the cell's behavior.

The model does not constrain at all the functions linking the switches and the genes. For a given number of switches N, the number of possible different gene functions is 2^{2^N}. This number rises very fast with N. Four switches are enough to allow every gene in the human genome to have a different behavior. Since the expressions of many genes are correlated, this level of freedom seems too high. The gene functions should probably be constrained somehow. Such constraints would also allow the determination of N switches with less than 2^N experimental conditions.

The binary switch model can also be viewed as a high-level description of the state of the cells. The switches often represent understandable experimental conditions, like temperature or starvation. Our technique may then offer an explanation of a large part of the gene expression measurements in terms of simple concepts. Such simplicity makes it a useful tool for biological understanding.

3 Identification of the model

The binary switch model implies that all experimental conditions sharing the same combination of switches have the same binarized profiles of expression. This means that the conditions can be grouped according to their switch configuration. However, there is an indetermination in the values of the switches. Once the conditions are grouped, any solution for which each group has a different combination of switches fits as well the model. The switches are not determined by the data, only the grouping of the conditions is. This is due to the lack of constraint put on the form of the Boolean functions linking the switches and the genes. A criterion will be defined later in order to choose the "right" switches among all possible combinations.

The model does not determine the number of groups of samples. The only limitation is that this number is at most 2^N, where N is the number of switches. The number of groups is expected to be lower than this maximum, because usually not every combination of switches is experimentally available. If for instance one switch is relative to excessive heat and another to excessive cold, it is unlikely that there exists an experimental condition where both of these conditions are simultaneously "on". As the number of switches increases, the number of missing combinations increases as well. The choice of the optimal number of groups and switches has to be done by judging the solutions obtained and by using biological insights concerning the data.

The determination of the parameters of the model is divided in two parts: firstly, the thresholds and the groups are determined, such that a maximum of genes has a constant value inside each group. Secondly, the best values of the switches for each group are determined.

3.1 The thresholds and the groups

3.1.1 Function to maximize

We show here how the grouping of the samples and the binarization of the data are done. A quality function is defined whose maximum should correspond to the best possible groups and thresholds. Those are then determined by maximizing the function.

Trivial solutions for which every gene respects the model always exist. For instance, if the thresholds are sufficiently low all binary values become "1" and any grouping gives a solution which perfectly fits the model. The quality function should be very low for such trivial solutions. A more interesting solution should be more informative concerning the regulation of the cells.

It is not expected anyway that every gene fit the model. Many reasons can prevent a gene from doing so. The binarization of the values might not be a reasonable hypothesis for certain genes. Some genes could be regulated by other, less important, switches which

control only small groups of genes in the experiments performed. Noise can also prevent certain genes from following the model. A gene which fits the model is called a predicted gene. By definition, the binarized values of the predicted genes are constant inside each group of experimental conditions.

The solution should maximize the number of predicted genes while keeping their profiles as interesting as possible. There are many different ways to quantify the relevance of a profile. This could be done using an information function:

$$I = -\sum_{i}^{PG} (p1_i \log(p1_i) + p0_i \log(p0_i)) \quad (3)$$

where the sum is on the *PG* predicted genes. $p1_i$ is the fraction of "1" in the predicted gene *i* and $p0_i$ the fraction of "0". This function effectively sets a trade-off between the number of genes predicted and the information each of those carries.

In practice, it is necessary to modify the information function (3). The penalty for less informative genes in (3) is not large: with 10 samples, a gene with five "1" and five "0" is only 2.1 times more informative than a gene with nine "1" and one "0". The fits on small groups being likely to emerge from random fluctuations, this penalty seems too small. In order to widen the difference, the cube of the information is taken:

$$I = -\sum_{i}^{PG} (p1_i \log(p1_i) + p0_i \log(p0_i))^3 \quad (4)$$

This modification increases the importance of the information in the trade-off between the number of genes predicted and the information they carry. Even with this modification, solutions with groups formed of just one experimental condition are still a concern. To address this issue, the genes for which only one condition has a different binary value than the others are excluded from the calculation of the information function.

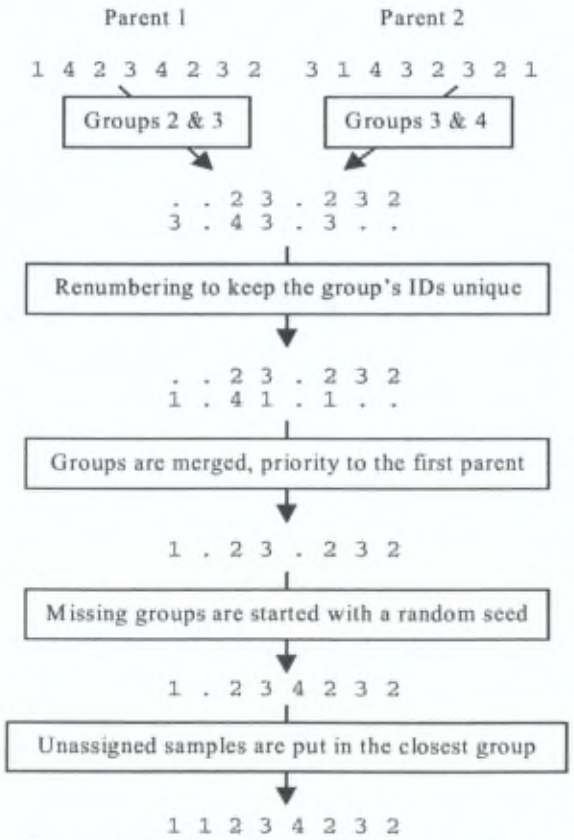


Figure 2. Example of the creation of a child in the grouping genetic algorithm

The maximum of (4) lies on top of a narrow hill. The function decreases very fast when the grouping is not perfect because one bad group may be enough to render the

information predicted null. Any non-exhaustive search algorithm will tend to converge to a local maximum, consisting of many groups containing just one sample. In order to help the search for the global maximum, a widening function is added to the information function to soften the base of the hill. The idea is to evaluate not only the quality of a complete solution, but also the quality of the groups forming the solution.

Certain genes may follow the model, i.e. have constant values inside each group, only on some subsets of groups. The value of the widening function for each gene is the information function obtained on these subsets. This information is multiplied by the fraction of the conditions which appears in that subset, in order to lower the values for the solutions based on few groups. The best possible subset is kept for each gene. This leads to the following widening function:

$$W = -\sum_{i}^{NPG} \max_{S_i} \left(\frac{n_i(S_i)}{n} \left(p1_i(S_i) \log\left(p1_i(S_i)\right) + p0_i(S_i) \log\left(p0_i(S_i)\right) \right) \right)^3 \quad (5)$$

where the sum is on the *NPG* non-predicted genes. S_i are the various possible subsets of groups for which the gene *i* is correctly predicted, $n_i(S_i)$ is the number of experimental conditions in the groups in S_i, and $p1_i(S_i)$ (resp. $p0_i(S_i)$) is the number of "1" (resp. "0") in the binarized gene *i* while keeping only the groups in S_i.

The function maximized is the sum of the information function (4) and the widening function (5), multiplied by a constant alpha:

$$F = I + \alpha W \quad (6)$$

Alpha should be small enough so that the maximum of (6) corresponds to the maximum of (4), but large enough so that αW is larger than the values of (4) obtained with a solution comprising many small groups.

It is of course necessary to check that the maximum of (6) found is indeed a maximum of (4). If it is not the case, a new search is performed with a lower alpha.

3.1.2 Maximization of the function

Two different things must be determined in order to maximize (6): the thresholds for the binarization of the genes, and the clustering of the experimental conditions. The thresholds are determined, for a given clustering, by an exhaustive search. The clustering is determined using a grouping genetic algorithm [5].

For this algorithm, a population of individuals is created. Each individual is encoded as a vector, with as many elements as there are experimental conditions. The values of the individuals represent the group membership of the conditions. For instance, an individual encoded as [1 2 1 2] has a first group consisting of the first and third conditions, and a second group consisting of the second and fourth conditions. At each round, the best individuals are selected using a tournament. Offspring is created from the best individuals. The offspring replaces the worst individuals from the last round. Mutations are then applied to the population.

The success of such algorithm depends on the choice of the mutation and crossover operators. Its effectiveness can be raised with the use of an appropriate heuristic. Since the problem is a modified clustering, k-means is chosen as the heuristic. It is used for the creation of the starting individuals. For the creation of a child, two parents are chosen (see figure 2). Some randomly chosen groups from each parent are inherited in the child, the groups being renumbered so that each group has a unique identifier. Conditions which belong to groups inherited from both parents are set to the first parent's group. The number of groups being a parameter of the problem, if the number of groups in the child is too low, one randomly chosen condition among the ones which are not assigned to any group is assigned to each missing group. The conditions which do not belong to any group are assigned to the closest group, *i.e.* the group whose members have on average the highest correlation with the condition to assign. Another child is made with the parents inverted.

In order to widen the search, mutations are made on randomly chosen individuals. Two types of mutations are used. In the first, three groups are randomly selected. The samples belonging to those groups are clustered into three new groups using a k-means. In the second type of mutation, the group membership of a sample is randomly modified.

The number of groups is not determined by the model. When this number is too high, the solutions found tend to overfit the data. Insights concerning the experiments must be used to detect such overfitting and determine the real number of groups.

3.2 The switches

Many different combinations of switches can fit the clustering found. Using Occam's razor, the simplest solution should be favored. Simplicity here lies in the functions linking the switches and the predicted genes.

Those functions are, in general, in the form of equation (2):

$$b_{ij} = \mathrm{f}_i\left(sw_{j1}, sw_{j2}, \ldots, sw_{jN}\right)$$

Often, b_{ij} does not explicitly depend upon all switches. A subset of switches can be enough to determine b_{ij}. We introduce simplicity here as the number of switches necessary to predict the value of gene. In the most extreme case, b_{ij} can be a value of just one switch, sw_{js}:

$$b_{ij} = \mathrm{f}_i\left(sw_{js}\right)$$

The solution should present as many of those simple functions as possible.

In practice, the information predicted directly by the switches is maximized. This information is calculated in a fashion similar to (4), except that here only the directly predicted genes are taken into account. Directly predicted genes are genes whose values correspond to the values of a switch, up to a "NOT" transformation. For example, in table 1, the switch 1 directly determines gene 1 and the switch 2 directly determines gene 4.

In complex cases, when the number of switches is very high, no gene is determined by a single switch anymore. In that case, some other type of simple functions (e.g. conjunctions of a few switches) should be considered. We suppose here that the situation is simple enough for a large number of genes to be directly determined by each switch.

For the search of the switches, the data can be simplified. Since the thresholds are known from the grouping, the data matrix can be considered as being binarized. Only the genes fitting the model are kept. The switches being only determined up to a "NOT" transformation, their values can be set to "0" in an arbitrary group. It remains then to find for each other group the values of the switches.

In a valid solution, each group of samples must have a different combination of switches. This constraint is stronger when the number of groups is the maximum possible for a given number of switches. In that case, among other things, every switch must have as many groups with "1" as groups with "0". These strict constraints are a reflection of the unlikeness of having the maximum possible number of groups, especially when the number of switches is high.

Since the switches should maximize the information directly predicted, they necessarily correspond to the binarized values of a gene or its opposite. The information directly predicted by each of these switches is easy to calculate. The different possible switches are then sorted in function of the information they predict, in order to try the most likely solutions first. See the first part of figure 3 for an example of possible switches with the corresponding information.

The solution is calculated using a branch and bound algorithm (see figure 3 for an example). This is done by firstly creating partial solutions consisting of just one switch, starting from the most informative switch. The validity and the quality of these partial solutions are checked. If by completing a partial solution with any other switch it is impossible

to create a valid solution of a better quality than the best actual solution, the partial solution can be discarded. Otherwise, another switch is added to generate a new, more complete, partial solution and the process is started again. When a generated partial solution is indeed a full solution, it replaces the current best solution. Its quality can then be used as a new bound on the quality of the partial solutions.

A partial solution is valid only if it can be part of a complete solution in which each group has a different combination of switches. This implies that the largest number of groups sharing the same combination of switches in the partial solution should be at most 2^N, where N is the number of switches which still have to be determined.

Groups	1	2	3	4	5	6	Number	Info
SW1	1	1	0	0	0	0	30	6.45
SW2	1	1	1	0	0	0	15	5.00
SW3	1	0	0	1	0	0	17	4.38
SW4	1	1	0	1	1	0	12	3.09
SW5	1	0	1	0	1	0	5	1.67
SW6	1	0	0	0	0	0	15	1.37

Possible switches information data. Each group is supposed to have the same number of conditions. "Number" is the number of genes directly determined by the switches. "Info" is the total information predicted by a switch: it is the product of the information of the switch with the number of genes directly determined. The switches are sorted in function of "info".

Current switches in the partial solution	Best possible quality	Comments
1	19.35	OK
1, 2	-	Not valid, backtrack
1, 3	-	Not valid, backtrack
1, 4	12.63	OK
1, 4, 5	11.21	New best solution
1, 5	9.79	Insufficient quality, backtrack
2	15	OK
2, 3	13.76	OK
2, 3, 4	12.47	New best solution
2, 4	11.18	Insufficient quality, backtrack
3	13.14	OK
3, 4	10.56	Insufficient quality, backtrack
4	9.27	Insufficient quality, finished

Trace of the program.

Figure 3. Example of the determination of the switches on an artificial data set.

4 Application: artificial data

The algorithms were applied on artificial data sets, in order to check that they effectively determine the switch structure when such structure exists in the data. In those data sets, as is expected in the real data sets, there are three categories of genes: random genes, genes directly determined by one switch and genes which are a Boolean function of more than one switch.

The grouping algorithm was applied to a first artificial data set, consisting of 100 experimental conditions organized in 16 groups using 4 switches. 10 genes are directly determined by each switch, 60 genes are determined by a combination of switches and 900 genes act as noise. The algorithm was able to recover the right grouping and thresholds in a

few hours. This setup being much more difficult that what is expected with real data, the performance of the grouping algorithm seems satisfying, although a faster version is certainly desirable. The switch determination algorithm was able to recover the switches from the grouping very quickly.

Secondly, a more reasonable case was created. In this setup, there are 1000 genes and 50 experimental conditions clustered into 6 groups using 3 switches. 20 genes are directly determined by each switch, 240 are determined by a combination of switches and the 700 remaining are random.

In this setup, when the algorithm is asked to find six groups in the data, it does so in a relatively short time (a couple of minutes). The switches are correctly recovered from the group information. When the algorithm is asked to find seven or eight groups, the best solutions have single conditions as new groups, the rest fitting the real solution. The quality of the solutions as estimated by (4) does not raise much when the number of groups is increased. When the algorithm is asked to find four or five groups, it finds solutions similar to the real one, except that some groups are merged. The quality of these solutions is much lower than the quality of the correct solution.

We expect that on real data, the quality of the solutions will keep raising as the number of groups is increased, because there are probably many "small" switches which explain small parts of the data. Nevertheless, the pattern of small groups should still appear, showing the likeliness of overfitting. We use this as a clue that the number of groups is too high.

5 Application: real data

We present here two applications of the technique to real world data. We have not tried to make breath-taking new discoveries, but simply to show that the binary switch model can be used to explain a large part of real regulatory networks. The switches discovered are also identified with simple, high level concepts, showing the power of the technique as a tool for biological understanding.

5.1 Cell cycle

The first set of data comes from the study of Spellman *et al.* [13] concerning the cell cycle in the yeast. There are a few different experiments in that study, which differ in the technique used to synchronize the cells. The one discussed here is alpha factor. Similar results were obtained with cdc15.

The data being very noisy, some pre-processing had to be done before the identification of the parameters of the model could be performed. Firstly, a low-pass filter was used in order to remove some noise from the data. The filter was an acausal, zero-phase filter of order 10, determined from a Butterworth filter [11]. The cutting frequency was half of the Nyquist frequency. This filtering rendered the data much smoother. Secondly, certain genes were excluded from the data set. The selection criteria were that at least 30% of the gene's derivative was conserved after smoothing, and that the smoothed gene had at least a 2-fold variation across the samples. Using this filter, only 608 genes were kept.

The identification of the parameters of the model was performed with four groups and two switches. The results can be seen in table 2. The first switch could be understood as standing for genes which are controlled by the arrest of the cell cycle necessary to synchronize the cells. The second switch is relative to the cell cycle itself. This shows that the technique is able to recover the expected structure in the data.

Among the 608 genes taken into consideration, 251 (41%) fit the groups. 45 genes are directly determined by the first switch, 36 by the second. Group 1, which is Switch 1 AND Switch 2, has quite a lot of success. This could be due to random fitting of noise (it is a small group) or to the sharp raises or falls which seem to be present for many genes in the first time points.

Table 2. Results obtained with the cell cycle experiment.

Time-min	0	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119
Group	1	1	4	4	4	2	2	2	2	2	3	3	3	2	2	2	2	2
Switch 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Switch 2	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Table 3. Results obtained with the metal experiments.

	WT Zn-	**WT Zn=**	**WT Zn+**	**Zap Zn-**	**Zap Zn=**	**Zap Zn+**	**Mac C**	**Mac B**	**WT Cu-**	**Cu+ 30**	**Cu+ 60**
3 groups	1	2	2	1	2	2	2	2	3	3	3
4 groups	1	4	4	1	3	3	2	2	1	1	1
Switch 1	**1**	0	0	**1**	0	0	0	0	0	0	0
Switch 2	**1**	0	0	**1**	0	0	0	0	**1**	**1**	**1**

In order to deduce the regulation network of the 41% of the genes which fits the model, it would only remain to identify the switches with biological counterparts. This task might be performed using gene expression alone if these counterparts are gene expressions. For the others, like probably for the switch relative to the cell cycle arrest, no identification is possible using only the data available. The limits of the deductions that can be made using gene expression data alone are reached.

5.2 Metal work

We have taken two 2-colors microarray data sets concerning the reactions of yeast to variations in the concentration of zinc [9] and copper [6] in the environment. These two data sets were chosen because the experimental conditions are somewhat similar, raising the hope to find some common trends. Nevertheless, the strains of yeast as well as the details of the experimental protocols are different. This demonstrates the applicability of the technique on heterogeneous data.

After some transformations, it is possible to reduce the measures of the first data set to six experimental conditions and the measures of the second to five other conditions. The two data sets are then merged (see table 3). WT is the wild type yeast, different in the two experiments. Zap is a WT yeast with the ZAP gene knocked out. This reduces its reaction to the lack of zinc. Mac is a WT yeast, with the gene MAC1 constitutively expressed. MAC1 regulates the expression of high affinity copper intake genes. There are two Mac experiments, based on two different strains: MacC, taken from yeast strain CM66J grown exponentially, and MacB, taken from yeast strain BR10 grown to late log phase. In the zinc experiment, WT and Zap cells were cultivated in deplete zinc (Zn-), replete zinc (Zn=) and excess zinc (Zn+) conditions. In the copper experiment, Mac cells were cultivated in normal condition and WT cells were cultivated in deplete copper medium (WT Cu-), and in excess copper medium for 30min (Cu+ 30) and 60min (Cu+ 60).

Since the standards in the two groups of experiments are not identical, the genes were normalized separately in each group. This was done by dividing, separately in the zinc and the copper experiment, the values for each gene by the mean of its values across the conditions of the experiment.

The group search algorithm was applied to the data set, with three and four groups. As shown in table 3, with four groups the best solution has three groups made of pairs of conditions. The experimental conditions in those pairs are very similar. The excess zinc condition is similar to the replete zinc condition, and the two Mac experiments are also similar. This solution can be considered as having groups formed of essentially the same samples, and so is probably due to overfitting. The solution with three groups was kept.

The switch determination algorithm was then run on those three groups, leading to the two switches shown in table 3. It is possible to assign a simple meaning to these switches. The first one is "on" only in the two experiments where the cells are lacking zinc. This switch could be understood as a "zinc starvation" signal. The second switch is "on" in

the experiments where the cells are in a difficult situation, lack or excess of something. This switch could be understood as a "sickness" signal. The excess of zinc is not very harmful for the yeast, which explains why those conditions are in the same group than the replete zinc conditions.

Among the 960 genes which are expressed at a reasonable level in all experiments and show more than 3-folds variation across the conditions, 369 (38%) fit the simple explanation given. Among those genes, 161 (44%) are directly determined by the "sickness" switch and 203 (55%) by the "zinc starvation" switch. Since the algorithm is sensitive to noise, as one noisy measurement is sufficient to consider a gene as not explained, and since the experiments were done using different standards, those are quite high figures, showing that a large part of the behavior of a cell can be understood with simple concepts.

The switches found are explained here in a "high-level" way, but we expect that some biological means exist in the yeast which pilot the regulations described. The identification of these means is impossible with gene expression data alone. However, the distribution of the switches provides some clues which might prove useful for such identification.

In a case like the one shown here, where one switch is included in the other, a hierarchical clustering algorithm should give a comparable result (see Figure 4). Nevertheless, the clustering algorithm does not offer a high-level explanation like the technique outlined here does, nor does it establish a link between the variation of the expression of the genes and the clustering. Besides, in this case, the clustering algorithm does not discover the "zinc starvation" switch.

Figure 4. Clustering of the metal experiments.

To our knowledge, this is the first time two different data sets are compared in order to deduce something about the samples. Such comparisons have only been done in order to predict gene's functions using clustering or classification techniques. We have shown here that such comparison can be performed and be meaningful. The distribution of the switches and the links established between those switches and the expression of certain genes might prove useful for a biologist. For instance, to an observer interested in the result of a lack of zinc in the environment, the genes which react in a similar way to a lack of copper may be of little interest. A comparison between the two experiments permits to spot such genes and thus to suggest unsuspected relations.

6 Conclusion

The model presented here allows the description and determination from the data of a large part of the gene expression regulatory network in a simple and consistent fashion. The rest of the network can be determined by identifying the switches with biological realities and discover how they are regulated. This complex task is simplified by the pattern of expression of the switches, which should permit to identify them with measured genes when such identification is possible.

We have demonstrated here that it is possible to deduce the value of the variables which regulate gene expression even when those variables are not measured. This identification is possible because a small number of causes create a large number of

different effects, and because the possible links between the causes and the effects are modeled precisely. This is performed here with regulations modeled as Boolean functions, but the same could certainly be done for other, more realistic, models of regulation of gene expression.

We have also demonstrated that it is possible to perform the binarization of the data using the same framework as the one used for the determination of the gene regulations. Such method should lead to better solutions than a discretization based on some pre-processing algorithm.

The binary switch model, like any Boolean model, is only a very simplistic description of the possible links between the regulators and the genes regulated. Nevertheless, we have shown here that this simplicity does not preclude the ability of the model to fit real data and to suggest interesting links. Simpler models have the advantage of being more understandable and less prone to overfitting. As long as they are expressive enough to describe the systems studied, they usually outperform more complex models. This may explain the maybe surprising success of the real world applications presented.

With the technique presented, it is possible to perform a comparison of different data sets. This way, an explanation of common traits between them can be obtained. As shown in the metal work example, this could be useful to focus the search on genes whose regulations are specific to certain experiments. The simplicity and understandability of the explanations given by the switches should prove useful for the selection of the most interesting genes.

A limitation of the model is that the links between the switches and the predicted genes must be perfect. As the number of samples increases, the likeliness of having at least one measure which does not fit because of the noise raises dramatically. This prevents the application of the framework on large data sets. A better version should allow for errors in the predictions, for instance by using a probabilistic Bayesian model instead of the Boolean model presented here.

In order to compare data sets obtained by different laboratories, with different protocols, it might be better to discretize independently each group of experiments. It our framework, that could be done by using a different threshold for each group of experiments. This should make the comparison of heterogeneous experiments less dependent upon the normalization of the gene expressions. This shows as well the importance of performing the discretization as a part of the estimation of the regulations, and not as a separate process.

Finally, the determination of the groups and the determination of the switches are done independently, which is probably not optimal. The simultaneous determination of the clustering, the thresholds and the switches should lead to better-defined problems.

Even with those limitations, the technique presented here is already applicable to real data sets, offering some interesting results. Would those limitations be lifted, it might be a useful tool for the discovery of regulation networks and the interpretation of biological processes.

7 References

[1] Akutsu, T., Miyano, S., and Kuhara, S. Identification of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Network Model. In Proceedings of PSB 1999, 17-28

[2] Akutsu, T., Miyano, S., and Kuhara, S. Algorithms for Inferring Qualitative Models of Biological Networks. In Proceedings of PSB 2000, 290-301.

[3] Chen, T., He, H.L., and Church, G.M. Modeling Gene Expression with Differential Equations. In Proceedings of PSB 1999, 29-40.

[4] D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. Linear Modeling of mRNA Expression Levels During CNS Development and Injury. In Proceedings of PSB 1999, 41-52.

[5] Falkenauer, E. Genetic Algorithms and Grouping Problems. Wiley (1998)

[6] Gross, C., Kelleher, M., Iyer, V.R., Brown, P.O., Winge, D.R. Identification of the copper regulon in Saccharomyces cerevisiae by DNA microarrays. Biol. Chem. 41 (2000), 32310-32316

[7] Holter, N.S., Maritan, A., Cieplak, M., Fedoroff, N.F., and Banavar, J.R. Dynamic modeling of gene expression data. Proc. Nat. Acad. Sci. USA 98 (2001), 1693-1698.

[8] Liang, S., Fuhrman, S. and Somogyi, R. REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. In Proceedings of PSB 1998, 18-29.

[9] Lyons, T.J., Gasch, A.P., Gaither, L.A., Botstein, D., Brown, P.O. and Eide, D.J. Genome-wide characterization of the Zap1p zinc-responsive regulon in yeast. Proc. Nat. Acad. Sci. USA 97 (2000), 7957-7962.

[10] Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., and Eguchi, Y. Development of a system for the inference of large scale genetic networks. In Proceedings of PSB 2001, 446-458.

[11] Matlab Signal Processing Toolbox. MathWorks inc. (1999)

[12] Pe'er, D., Regev, A., Elidan, G., and Friedman, N. Inferring subnetworks form perturbed expression profiles. In Proceedings of ISMB 2001, Bioinformatics 17, Suppl 1, S215-S224.

[13] Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9 (1998), 3273-3297.

[14] Thieffry, D., and Thomas, T. Qualitative analysis of gene networks. In Proceedings of PSB 1998, 77-88.

[15] Weaver, D.C., Workman, C.T. and Stormo, G.D. Modeling Regulatory Networks with Weight Matrices In Proceedings of PSB 1999, 112-123.

[16] Wessels, L.F.A., Van Someren , E.P., and Reinders, M.J.T. A Comparison of Genetic Network Models. In Proceedings of PSB 2001, 508-519

Conclusion

The analysis of microarray data proved to be a very challenging task, in part because of the sheer size of the data. This made such simple things as the determination of p-values and clustering complicated enterprises. The other difficulty lied in the imprecision of the questions asked. The biologists often have qualitative questions, which are not always easy to formulate mathematically. Also, they often do not correctly assess what is possible and what is not. For these reasons, it has often been necessary to determine what a biologist might be interested in and could be obtained from the data. This can be the hardest part of a work.

The works presented in this thesis span a large spectrum. The first half was mainly concentrated on technical issues: how to estimate significance levels, how to improve the data quality and how to store the data. For these parts, the questions where quite clear, and most of the difficulties came from the size of the data and its particularities. In this first half, the techniques presented are mostly improvements of existing techniques. The main original points were:

1. In the statistical chapter, the comparison of different scoring functions, the merging of false discovery rates determined on different intensity windows and the determination of a local false discovery rate.
2. In the data improvement chapter, the creation of two efficient data quality criteria. The merging of different scans to avoid saturation was original at the time it was created, although it has been published by others since. The assessment of the different modifications in a thorough and consistent manner is obvious but surprisingly unusual.
3. The data storing chapter present a model of database which is essentially the development of simple design ideas.

The second half was concerned with questions which the biologists did not ask directly, but could be inferred from their frustrations. Since the problems those chapters try to solve are original, the techniques used tend to be original also. The three original points raised in those chapters were:

1. The discovery of the composition of complex samples. The expression profile of complex samples can depend more on their composition than on their pathological status. A mean to mathematically dissect those samples was proposed. This work is completely original.
2. The discovery of different clustering. The samples in an experiment can often be organized in more than one way, depending on the criterion chosen. A mean to cluster the gene in function of the clustering they give on the samples is given. The idea of this work stems from a paper showing the existence of different superimposed clusterings. The re-definition of this problem as a clustering problem on the genes is original, as is the algorithmic means to perform said clustering.
3. The last chapter is concerned about the link between genetic network and clustering. It is shown that to suppose the existence of a form of genetic network implies the existence of a clustering, whose form is determined by the form of the genetic network. This is illustrated with a Boolean network, showing that its identification can be separated in two tasks, a clustering and an identification of the clustering with the network. This way of linking clustering and genetic network is original, as are the algorithms proposed.

Table of content

1 Determination of differentially expressed genes

2 Data quality improvement

3 A database for microarray data

4 Discovery of overlapping clustering

5 Mathematical dissection of heterogeneous samples

6 Genetic network inference

Conclusion