**Universidade de Aveiro**
**2009**
Departamento de Electrónica, Telecomunicações e Informática

**Bellina Ribau Teixeira**

# Métodos Computacionais para Análise de Dados de Microarrays

Computational Methods for Microarray Data Analysis


Universidade de Aveiro
2009

Departamento de Electrónica, Telecomunicações e Informática

**Bellina Ribau Teixeira**

# Computational Methods for Microarray Data Analysis

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**o júri**

presidente

Prof. Dra. Maria Beatriz Alves Sousa Santos
Professora Associada com Agregação da Universidade de Aveiro

Prof. Dr. Rui Pedro Sanches de Castro Lopes
Professor Coordenador do Instituto Politécnico de Bragança

Prof. Dr. José Luís Guimarães Oliveira
Professor Associado da Universidade de Aveiro

**agradecimentos**

Quero agradecer ao Prof. José Luís Oliveira e ao Joel Arrais pela orientação prestada para a realização deste trabalho, tanto a nível técnico e teórico, com instruções e directrizes imprescindíveis para a sua realização, como também pelo incentivo e motivação transmitidos sem os quais o resultado apresentado não seria o mesmo.

Agradeço também à minha família, aos meus amigos e a todos os que me apoiaram na realização deste trabalho. A todos eles o meu muito obrigado.

**palavras-chave**

ADN, microarrays, expressão, genes, diferencialmente expressos, ASP .NET, C#, aplicação web

**resumo**

Os *microarrays* de ácido desoxirribonucleico (ADN) são uma importante tecnologia para a análise de expressão genética. Permitem medir o nível de expressão de genes em várias amostras para, por exemplo, identificar genes cuja expressão varia com a administração de determinado medicamento.

Um slide de *microarray* mede o nível de expressão de milhares de genes numa amostra ao mesmo tempo e uma experiência pode usar vários slides, surgindo assim muitos dados que é preciso processar e analisar, com recurso a meios informáticos.

Esta dissertação inclui um levantamento de métodos e recursos de *software* utilizados na análise de dados de experiências de *microarrays*. Em seguida, descreve-se o desenvolvimento de um novo módulo de análise de dados que visa, usando métodos de identificação de genes diferencialmente expressos, identificar genes que se encontram diferencialmente expressos entre dois ou mais grupos experimentais. No final, é apresentado o trabalho resultante, a nível de interfaces gráficas e funcionamento.

**keywords**

DNA, microarrays, expression, genes, differentially expressed, ASP .NET, C#, web application

**abstract**

Deoxyribonucleic acid (DNA) microarrays are an important technology for the analysis of gene expression. They allow measuring the expression of genes among several samples in order to, for example, identify genes whose expression varies with the administration of a certain drug.

A microarray slide measures the expression level of thousands of genes in a sample at the same time, and an experiment can include various slides, leading to a lot of data to be processed and analyzed, with the aid of computerized means.

This dissertation includes a review of methods and software tools used in the analysis of microarray experimental data. Then it is described the development of a new data analysis module that intends, using methods of identifying differentially expressed genes, to identify genes that are differentially expressed between two more groups. Finally, the resulting work is presented, describing its graphical interface and structural design.

# Table of Contents

# List of Figures

# List of Tables

# Acronyms and Abbreviations

| | |
|---|---|
| **ADF** | Array Description File |
| **AJAX** | Asynchronous Javascript and XML |
| **C** | Control |
| **DB** | Database |
| **DNA** | Deoxyribonucleic Acid |
| **GAL** | Gene Annotation List |
| **GUI** | Graphical User Interface |
| **HGP** | Human Genome Project |
| **HS** | Heat Shock |
| **IDE** | Integrated Development Environment |
| **LIMS** | Laboratory Information Management System |
| **MIAME** | Minimum Information About A Microarray Experiment |
| **Mind** | Microarray Information Database |
| **PCR** | Polymerase Chain Reaction |
| **QC** | Quality Control |
| **RNA** | Ribonucleic Acid |
| **mRNA** | messenger RNA |
| **rRNA** | ribosomal RNA |
| **tRNA** | transfer RNA |
| **SNP** | Single Nucleotide Polymorphism |
| **SQL** | Structured Query Language |
| **URL** | Uniform Resource Locator |
| **XML** | Extensible Markup Language |

# Chapter 1

# Introduction

## 1.1 The emerging of Genetics and Bioinformatics

The completion of the Human Genome Project (HGP) in 2003 represented an amazing accomplishment in the genetics and the bioinformatics fields [1]. The sequence of bases of the human DNA (deoxyribonucleic acid) was determined, and the more than 20,000 human genes on the sequence were mapped. The project was finished quicker than expected due to the rapid advancements on the available technology.

The genetics subject has its foundation on the work of Gregor Mendel in the 19$^{th}$ century. He studied the inheritance of several physical traits through generations of pea plants, designating as factors the elements that transmit the phenotype information from living organisms to their offspring [2]. Later, various studies and experiments carried out by different scientists during the first half of the 20$^{th}$ century allowed identifying DNA as the cell component that contains the hereditary information [3]. What Mendel described as factors are the genes present in the DNA. In 1953, Watson and Crick presented the first three dimensional model of the DNA molecule, which has not changed much since then [4]. With these discoveries, the genetics area expanded quickly and is today a major field of research.

The quick progression of the genetic field cannot be dissociated from the evolution of information technology. Not only the HGP depended on it, but most research and work is done with the aid of computerized means. Molecular biology in general (the study of biology at a molecular level) which overlaps genetics, biology and chemistry, is highly dependent on computers. The relationship between these two is so cohesive that it earned its own designation – bioinformatics, a term coined by Paulien Hogeweg in 1978 to express the application of information technology to molecular biology [5].

Although it represents a big milestone, the sequencing and mapping of the human genome leaves much work to be done, in order to use this information for developments in medicine and

other fields of application. Namely, the function of the human genes, and how they work together, is an important topic of research.

Genomes of other organisms have been sequenced and mapped as well, such as the *saccharomyces cerevisiae*, the yeast used in bread, beer and wine production. Although a unicellular living being, its genome has 7,000 genes, and many of these genes overlap in sequence and in function with the human genome, providing a reason for the yeast genome to be studied extensively.

## 1.2 DNA microarrays for gene expression analysis

DNA microarrays are an important technology used in the research of gene function and interactions. They started to be used for gene expression analysis around 1995 [6]. All of our cells have the same DNA, but the genes that are active at a time depend on the cell type, activity and other conditions such as nutrition. Besides, genes are not just simply expressed or not, but rather more or less expressed. Given a sample of cells, DNA microarrays allow to probe the state of a group of genes within that sample.

A typical example is testing a drug for a specific disease. If a group of genes related to that disease is known, one can use this technology to evaluate the effect of the drug in those genes, comparing a group of patients that are being treated with it versus patients that are not. Genes found differently expressed between the two samples may be due to the effect of the medication. If the medication is found to inhibit a gene whose over expression caused the disease while not interfering on the other genes, the experiment will demonstrate evidence of its effectiveness.

DNA microarrays for gene expression analysis use the central dogma of molecular biology, which states that expression of genes is related to the mRNA (messenger ribonucleic acid) molecules produced in the cell during protein synthesis. mRNA molecules are another cell component like DNA. In general each mRNA molecule is one "copy" of an active gene of the DNA. The more expressed a gene is in a cell at a time, the more mRNA molecules will be available in a cell, to allow the protein synthesis necessary to carry out the function coded by the gene. Thus, given a sample of cells, such as blood or tissue, to analyse gene expression using DNA microarrays, what is measured is the mRNA extracted from the cells.

This technology can be used to study gene function and discover genes that work together, known as exploratory analysis, or to probe for genes that are expressed differently between two

(or more) samples, such as the example drug testing experiment described. The focus of this project and this document will be the latter, known as analysis of gene differential expression or gene regulation.

When using DNA microarrays to assess gene differential expression, knowledge from the areas of genetics and biology, computer science and statistics are combined together towards the goal of obtaining a list of genes that are regulated. The DNA microarray itself usually consists of a glass slide onto which the genes or probes are printed. Each slide has up to approximately 20,000 probes and an experiment can have different slides and a few replicates of each one; as a result, this represents an enormous amount of data to be processed, managed and stored.

## 1.3 The Mind Project at the University of Aveiro

The Mind project (Microarray Information Database) was created at the University of Aveiro, to address microarray data management and processing. It is a repository of microarray experimental data that also allows the processing of the data. This web application, hosted at http://bioinformatics.ua.pt/mind/, is a microarray dedicated LIMS (Laboratory Information Management System) and a microarray data analysis application [7].

The LIMS module is responsible for storing and organizing the data according to the MIAME standard (Minimum Information About a Microarray Experiment). As there are many manufacturers of microarray software and laboratorial equipment, this standard ensures that when a microarray experiment is performed, a common set of information is stored, such as the raw data format and the laboratorial practices used, as much information as it is needed to replicate the experiment and to understand how the available data was obtained [8].

The data analysis module processes the raw data that was submitted using the LIMS. It already contains quality control functionalities, including normalization. Quality control is the first step of microarray data analysis (common to both gene regulation assessment analysis and exploratory analysis), and it aims to remove systematic biases and to confirm the quality of the arrays. If after being subject to quality control the data appears to be of bad quality, the microarray experiment may have to be redone. Using the quality control reports generated, which consist mainly of graphical plots, the user can verify if each array was successful [9].

## 1.4 Objectives

The goal of this project is to continue the analysis of the normalized data. Currently, Mind is used to store the raw data files, to perform quality control and to store the normalized files. The subsequent analysis, required to assess the regulated genes, must be done with other software. This project intends to incorporate gene regulation assessment functionalities in Mind. In order to accomplish this, a study of microarray data analysis and an inquiry of available microarray data analysis tools must be done. This dissertation will thereby, besides describing the integration of the new functionalities in Mind, reveal the study that was done on microarray data analysis methods and software.

GeneBrowser is also a project developed at the bioinformatics group of University of Aveiro, like Mind. The web application, hosted at http://bioinformatics.ua.pt/genebrowser2/, given a list of genes, is able to assess their biological function, for example, by identifying them in pathways (a metabolic pathway, in biology, is a series of reactions that occur in succession towards a common goal, for example, the glycolysis pathway in yeast, which transforms sugar into piruvate) [10].

The new Mind data analysis module, object of this dissertation, will allow, in addition to performing quality control (already implemented), post-normalization analysis of the data in order to obtain a list of differentially expressed genes that can be directly and transparently exported to GeneBrowser. As Mind and GeneBrowser are both web applications, Mind can offer a link to send the list of regulated genes as input for GeneBrowser. Mind was initially designed with the inclusion of this data analysis module in it, which will provide a connection between the two web applications.

## 1.5 Structure

The next chapter introduces some biology theory necessary for the understanding of the DNA microarray technology as an estimate of gene expression. The microarray technology is also presented.

The third chapter, DNA Microarray Data Analysis, the longest chapter in this document, describes the methods used to derive significant meaning from the raw microarray data. As there are many methods available, the ones that were chosen for Mind will be described. However, as these correspond to the most commonly used methods, the chapter is a

considerably general microarray data analysis description. Programming libraries that offer the selected methods, and that will be used in Mind, are also presented.

The fourth chapter provides an overview of some existing microarray data analysis software solutions. Together with Mind's users' opinions, these provide an important insight on what is expected in a microarray data analysis application.

The fifth chapter focuses on the development of the Mind microarray data analysis module, and includes technologies used, the modifications done in the Mind application and database and presents the final result with some screen captures.

The last chapter summarizes the work done, presents the most important conclusions of this project and leaves some suggestions of future developments.

# Chapter 2

# DNA Microarrays: Principles and Technology

The DNA microarray technology relies on the complementarity of bases observed in DNA and RNA molecules, the nucleic acids. This principle allows two strands of DNA or RNA to bind, if they are complementary. An mRNA sequence is complementary to the gene it was transcript from and it will bind to a sequence that is identical to the original gene.

By using DNA microarrays for gene expression profiling, the mRNA extracted from sample cells is analyzed and used to infer about the sample's gene expression.

In order to explain these two concepts, binding of complementary sequences and mRNA as an indicator of gene expression, this chapter will begin by describing the DNA and RNA molecules, the nucleic bases, base complementariety and the role of mRNA in protein production. Finally, it will be shown how these biological principles apply to DNA microarrays with the description of a microarray experiment.

## 2.1 DNA and protein synthesis

From the instant we are conceived, our brief existence being unicellular, the DNA molecule residing in the nucleus of that single cell contains all the information needed to build the proteins that will form ourselves. When cell division and replication occur, obviously necessary to our growth, but always needed for tissue regeneration, the DNA molecule too is duplicated, in a much complex process, the DNA replication. Our genetic information, contained in our DNA, defines our physical and psychological traits. DNA is referred commonly as our biological identification because it is unique in each individual. All humans have the same genes, but a small percentage of the genome differs in sequence from person to person; these are the genes that describe each individual's specific characteristics. All living beings have their DNA coding their particular characteristics. The most important biological process that describes how the information of our DNA determines what we are is the protein synthesis.

### 2.1.1 The DNA molecule

The DNA molecule has the form of a double helix, and each of the two strands is a sequence of nucleotides, as represented in Figure 2.1. A nucleotide is made up of a phosphate group, a pentose sugar and a nitrogenous base. The latter, in the DNA molecule, can be an adenine, a thymine, a cytosine or a guanine, and it is what distinguishes the nucleotides. In fact, the nucleotides are commonly called just by the initial of their base, A, T, C or G.

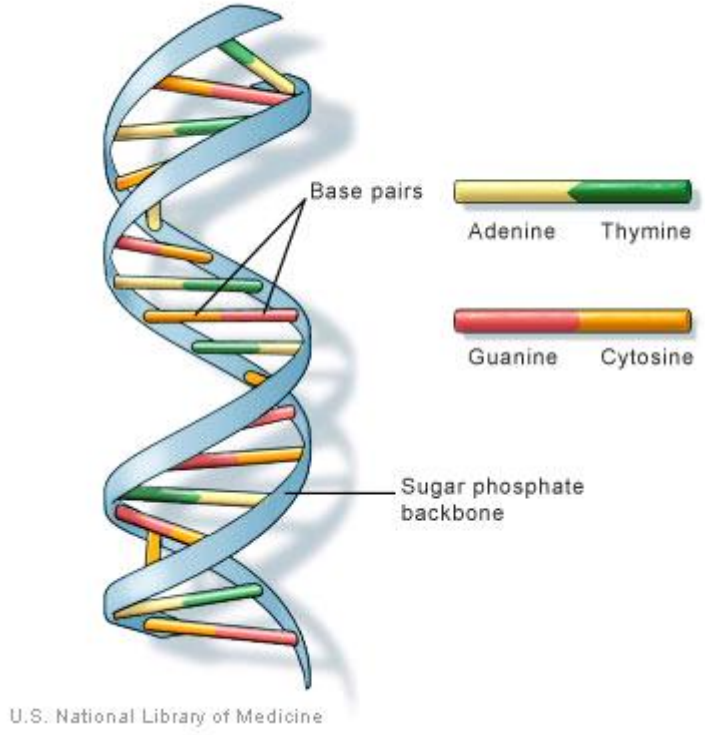



Figure 2.1 – The DNA molecule [11]

On each DNA strand, the pentose sugars and the phosphate groups connect to form the phosphate sugar backbone – the phosphate of the fifth carbon of the sugar of one nucleotide connects to the third carbon of the sugar of the following nucleotide. DNA is actually synthesized in this order, phosphate of the carbon 5 bound to the carbon 3 of the next nucleotide; it is synthesized in a 5' to 3' direction (five prime to three prime), as illustrated in Figure 2.2.

The bases of each strand form non-covalent hydrogen bonds with the bases on the other strand, due to base complementarity. A and T bases are complementary with one another, as are C and G. Each two complementary bases is a base pair. Two strands being complementary to one another means that the base sequence on one strand matches the base sequence on the other, allowing for the hydrogen bonding between each base pair.

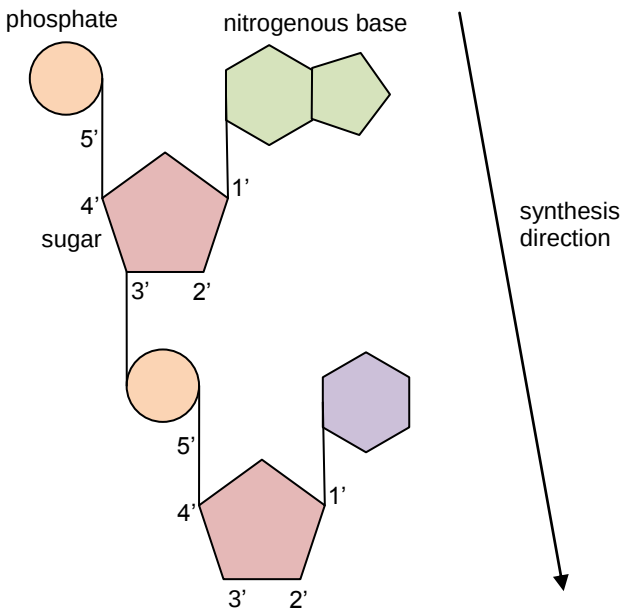



Figure 2.2 – A DNA strand is formed by joining nucleotides 5' to 3'

ATCG is also known as the DNA alphabet, because using these four letters we can describe a DNA molecule. Figure 2.3 shows a simple representation of a DNA fragment. The parallel lines do not need to be present, they represent the backboned formed by the pentose sugars and the phosphate groups on each strand. The numbers 5 and 3 at the end of each strand indicate the direction of the strand. The top strand runs from 5' to 3' (five prime to three prime), the bottom strand runs from 3' prime to 5' prime, which is why the two strands are called anti-parallel.

5' ATTACGGC 3'

3' TAATGCCG 5'

Figure 2.3 – Complementarity of bases between two, anti-parallel, DNA strands

Given any DNA strand, regardless of the length, one strand is expected to bind to it, the complementary strand. However, a single nucleotide polymorphism (SNP) may occur, meaning that even if the strands are not totally complementary they can still bind, so the binding of two strands does not guarantee they are totally complementary.

### 2.1.2 DNA and RNA – the nucleic acids

In an eukaryotic cell, the DNA resides in the nucleus. In a prokaryotic cell there is not a nucleus and the DNA is located in a region called a nucleoid, not delimited by a membrane. In fact, the words prokaryotic and eukaryotic come from the Greek language: prokaryotic means "before

nucleus" or without a nucleus and eukaryotic means "true nucleus". Prokaryotic cells are smaller and less complex than eukaryotic cells. Most organisms are eukaryotes. Humans, animals, plants, mushrooms and yeast are eukaryotes. Bacteria are prokaryote organisms. Figure 2.4 shows where DNA is located in a prokaryotic cell and in an eukaryotic cell.

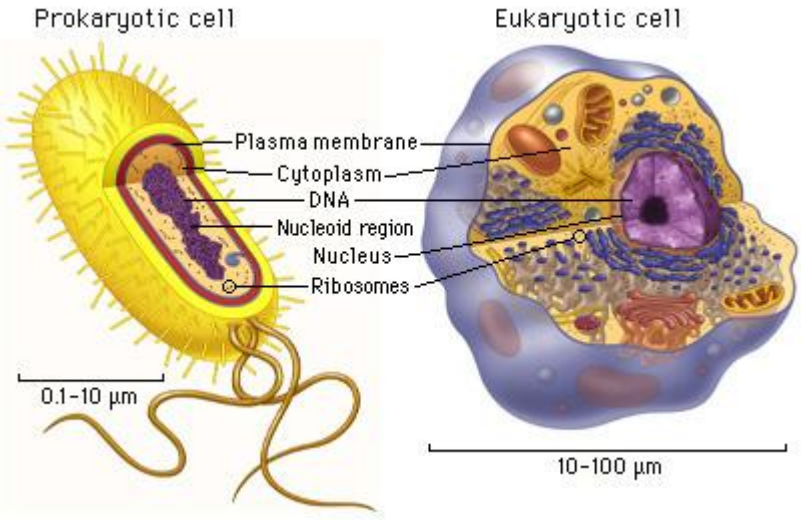



Figure 2.4 – Prokaryotic and Eukaryotic cells [12]

The other nucleic acid present in the cell is the RNA. The designation (nucleic) is due to their role, and not their presence, in the cell nucleus. Although DNA always resides inside the nucleus, RNA can be in the nucleus or in the cytoplasm (in eukaryotic cells). While DNA is double-stranded, RNA is one stranded, at least in the majority of its biological roles (not including the induced mRNA/DNA hybridization on the microarrays). In RNA the nucleotides can have bases A, C and G, like DNA, but instead of thymine, RNA has uracil (U). Also, while DNA contains the sugar deoxyribose, RNA contains ribose.

An RNA molecule can bind with a DNA strand, forming a DNA/RNA hybrid, as long as the two strands are complementary (and anti-parallel). This principle is used on DNA microarray technology – the mRNA extracted from the sample is applied over the slide, and left to bind to the DNA probes.

A prefix is used to indicate the RNA function in the cell. mRNA for messenger RNA, tRNA for transport RNA and rRNA for ribosomal RNA. These three are involved in protein synthesis; there are more types of RNA for DNA replication and other cell functions.

### 2.1.3 Protein synthesis

Messenger RNA, mRNA, is a copy of the DNA, carried from inside the cell nucleus to the ribosomes in the cell cytoplasm, where protein synthesis will occur. This process consists of three main phases: transcription, processing and translation [13].

Protein synthesis starts with transcription, a process that forms a RNA sequence, using a DNA strand as a template. This is done in presence of an enzyme complex, the RNA polymerase. The DNA helix opens up in the region to be transcript, and the RNA polymerase slides along the DNA template strand, building, one nucleotide at a time, a complementary RNA strand. Figure 2.5 shows an example portion of the DNA molecule that is transcript and the resulting mRNA molecule. The RNA polymerase glides along the bottom strand in the direction 3' to 5', so the mRNA synthesis can occur from 5' to 3' (nucleic acids are always synthesized 5' to 3').

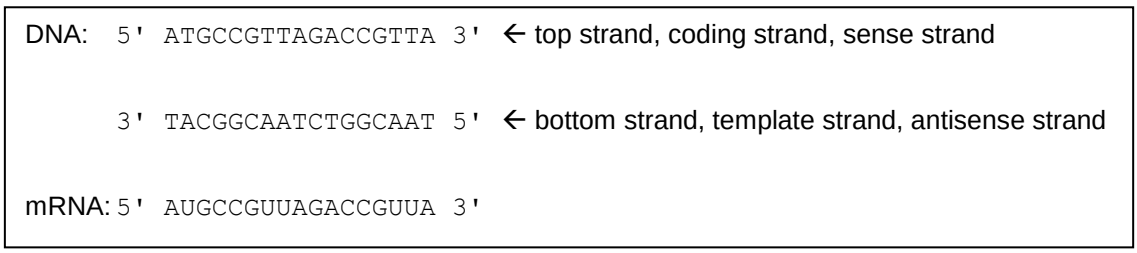



Figure 2.5 – mRNA strand formed identical to the coding strand

The formed mRNA is processed, to remove information that is not further on relevant for protein synthesis and then it is let out into the cell cytoplasm, crossing the cell's porous nucleus (in eukaryotic cells).

Once in the cytoplasm, translation occurs, involving ribosomes and tRNA. The ribosomes are made of rRNA (ribosomal RNA) and protein. The tRNA (transfer RNA) are molecules that contain a sequence of three nucleotides (three of A, U, C, G) called a codon and an amino acid that corresponds to that sequence. Amino acids are the building blocks of proteins. There are around 20 amino acids, and 64 different sequences of 3 nucleotides, so different sequences can code the same amino acid. In the translation process, the ribosome scans the mRNA, matching for each three mRNA nucleotides, a tRNA that is complementary. As the tRNAs are attached to the mRNA, the protein is being formed, the sequence of the amino acids carried by the tRNAs. Once a tRNA's amino acid is connected to the forming protein, the rest of the tRNA is discarded, leaving the three mRNA nucleotides free, which is why the same mRNA molecule can be used by different ribosomes at the same time. Several ribosomes can be at different locations of the same mRNA molecule, each one having its protein at a different stage of development, so one mRNA can allow for multiple copies of the same protein.

After translation, the proteins can undergo many more and complex transformations until they are finally complete. To name some possible final results for these proteins, they may become enzymes and work inside the cell, they may become part of the cell's structure or they may be exported outside of the cell. Not only the protein synthesis process described here is very simplified, many more alterations may be done after the translation.

### 2.1.4 mRNA as an indicator of gene expression

DNA encodes our physical and psychological traits in the manner that it contains the information for the protein synthesis to occur. The DNA is the same in practically all cells of an individual, but we have different cell types, that produce different proteins, according to the genes that are expressed. Gene expression and consequent protein production in a cell also varies according to its condition (nutrition, medication or temperature).

The mRNA present in a cell gives us an indication of what genes are expressed. The DNA microarray technology can be used to analyze the mRNA samples and conclude about gene expression. However, it is not an exact measure of gene expression: the mRNA present in the cell does not indicate exactly what genes are expressed. For cDNA microarrays (definition further in 2.2) there is the possibility that the experiment may not show the presence of some mRNA, or that it shows the presence of mRNA without it being in the sample. The probability of failing to detect present mRNA is around 5% (false negatives) and the probability of detecting mRNA that truly is not present is around 10% (false positives) [14].

It is questioned about which method provides the best picture of functional expression: mRNA expression analysis, also know as transcriptome analysis, or protein expression analysis, referred to as the proteom analysis. The transcriptome analysis scores for high degree of sensitivity, speed and completeness, but there may be discrepancies between the genes and the ultimate phenotype. For example, the butterfly and the caterpillar have identical genomes but very different proteomes. In the end, both techniques are important and complementary. The choice of one or the other depends on the biological question being asked and the resources available, but both methods are important [15].

## 2.2 A DNA Microarray Experiment

A microarray experiment starts with the identification of a problem that can be solved using this technology and by defining a design for a microarray experiment that will provide an answer. The experiment design describes what probes to use on the slides, what mRNA samples to

hybridize to the slide probes, how the samples will be prepared, and what information is to be collected from the experimental procedure. After proper planning, the experiment can be carried out: the microarrays are prepared, hybridization of the targets and the probes occur and the microarrays are scanned. The resulting pictures of the microarrays are converted to numerical data that is handled with data analysis. Figure 2.6 enumerates the steps of a microarray experiment.

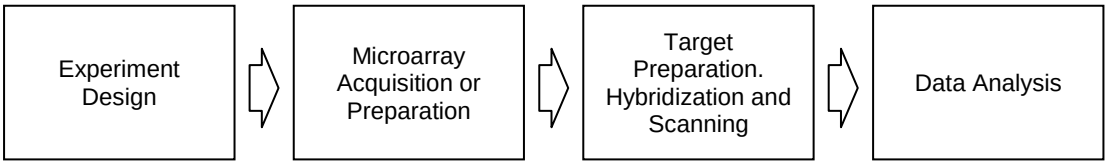



Figure 2.6 – Steps of a Microarray Experiment

### 2.2.1 The microarray slides and probes

The microarray slide can be of different materials, and the probes can be placed on the slide using different methods. The most common are spotted arrays, which use high quality glass slides. The probes are produced and then they are spotted onto the slide, using a spotting robot. This machine dips a grid of fine pins into the wells containing the probes and then lays the tiny drops onto the microarrays slide. Spotted microarrays can be cDNA (most commonly) or oligonucleotides, depending on how the probes were produced [16, 17]. cDNA sequences are cloned from a known gene using a DNA or RNA strand as a template and can be thousands of nucleotides long. These sequences are multiplied, usually using a technique called polymerase chain reaction (PCR), as each spot on the microarray requires thousands of copies of the same cDNA sequence. Oligonucleotide sequences are shorter, up to 20 nucleotides, and are not produced in a laboratory like cDNA sequences. At the University of Aveiro Biology lab, oligo (short for oligonucleotide) sequences are purchased from a manufacturer and then, in the lab, they are spotted onto the microarray slides. Manufacturers also provide complete arrays, with the oligo sequences already printed onto the slides.

There is not a clear definition of cDNA and oligo arrays in the literature. Oligo arrays are sometimes associated to one-color arrays, and cDNA to two-color arrays, but there are two-color arrays with oligo sequence probes. Oligo arrays are usually associated to the commercial arrays, but they can also be printed in the lab, although the oligo sequences must be purchased first. Commercial arrays (purchased with the probes printed) are normally produced using *in situ* synthesis, thus the term oligonucleotide is associated with this manufacture method

too. The *in situ* synthesis is a method, usually based on photolithography (involving light and light-sensitive masks), that synthesizes the probe sequences directly onto the array surface.

At the Biology department of the University of Aveiro, spotted cDNA microarrays are produced in the lab, oligo sequences are purchased to be spotted in the lab and commercial complete oligo arrays are acquired too. In all cases, the arrays are used as two-color, allowing hybridization of two mRNA samples on each slide, labeled with two different color fluorescent dyes.

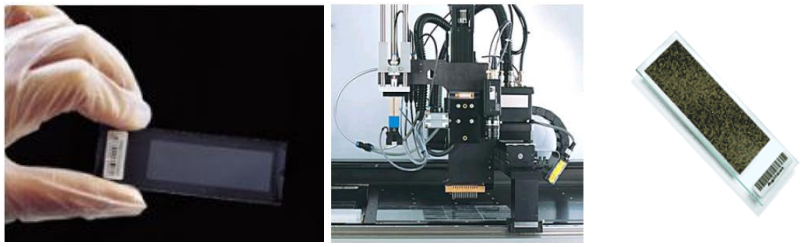


Figure 2.7 – Microarray glass slide, spotter and commercial array [18-20]

### 2.2.2 Target preparation, hybridization and scanning

After planning the experiment and arranging the microarray slides, the biologist prepares the mRNA samples or targets. The mRNA is extracted from the cells to be studied. For each array, two mRNA samples are used and each one is labeled with a different color fluorescent dye. Commonly cyanines cy3 (green) and cy5 (red) are used. The two labeled mRNA samples are applied over the slide and let to hybridize in a hybridization oven, which provides optimum conditions for the process. The mRNA will bind to the spots whose probe sequences are complementary.

Once hybridization is complete, the slides are read by a microarray scanner. The scanner's light excites the red and green fluorescent dyes. Spots to which mostly the red labeled mRNA sample has bound will glow red, spots that contain mostly the green labeled mRNA samples will reveal a green color, and spots that have about the same amount of both will be yellow. Color intensities also vary, from bright colored (red, green or yellow) to black, indicating how much mRNA was actually fixed. The scanner will output an image showing the hybridization that occurred in the slide.

### 2.2.3 Raw data files: the starting point to microarray data analysis

The images obtained from the scanner are processed using microarray image analysis software, provided from the scanner manufacturer or available separately. The software produces one data file for each image and for each scanned array (Figure 2.8). It is usually a tab delimited text file,

with as many rows as the spots of the microarray and several columns for the spot attributes. The most relevant fields are the red and green mean intensities and the red and green background intensities. These values do not have units, but they range from zero to tens or hundreds of thousands. The foreground intensities indicate how much red and green is measured in each spot as a consequence of the competitive hybridization. The background intensities show how much red and green was measured just outside the circular area of each spot, consequence of noise, non-specific hybridization or other factors.

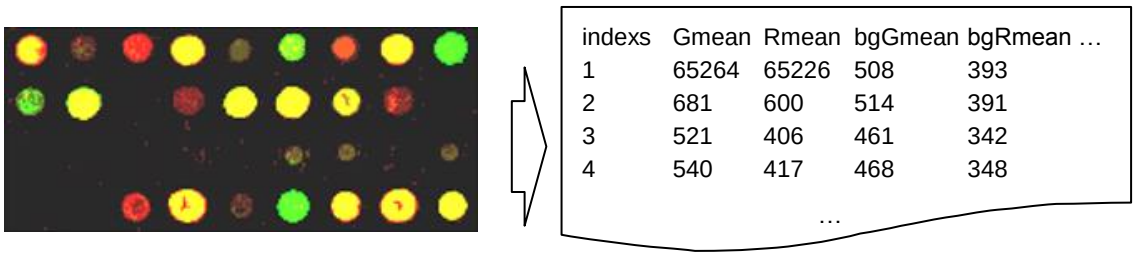



Figure 2.8 – From microarray image to raw data file

A GAL file (Gene Annotation List) must also be created, describing what gene is probed on each spot. The MAGE-TAB equivalent, ADF (Array Description File), contains the same information, where MAGE-TAB is a MIAME-compliant format for organizing microarray experimental data [21]. Both GAL and ADF describe the array layout. Figure 2.9 shows an example array layout, consisting of 4 × 12 squares of 20 × 20 spots each, for a total of 19200 spots. The GAL or ADF file for this array would indicate the name of the gene relative to each spot, just like the data files, it also contains as many rows as the number of spots in the array.

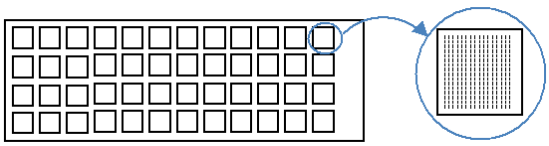


Figure 2.9 – Example array layout

Once the array layout file and the data files are obtained, the microarray data analysis begins. The first part of data analysis is quality control. Quality control includes background correction and normalization procedures, and in addition to correcting the data for systematic biases and noise, it allows acknowledgement that the arrays were successful. If even at the end of this step the data is of bad quality, then the microarray hybridization and/or scanning must be done again until the normalized data for all arrays is acceptable. The second part is gene assessment analysis, which aims to find genes that are likely to be differentially expressed between the samples under study. Alternatively, instead of gene regulation analysis, exploratory analysis could be done, to discover unknown genes or to learn about groups of genes that work together, involving procedures like clustering of the gene expression values; this dissertation will

approach the former. After gene regulation assessment (or exploraroty analysis) functional analysis can take place (Figure 2.10). Exploratory analysis is not contemplated at the moment, but may be added to Mind in future developments. This project adds gene regulation analysis to Mind, constituting itself as a bridge between the existing Mind quality control functionalities and GeneBrowser's functional analysis.

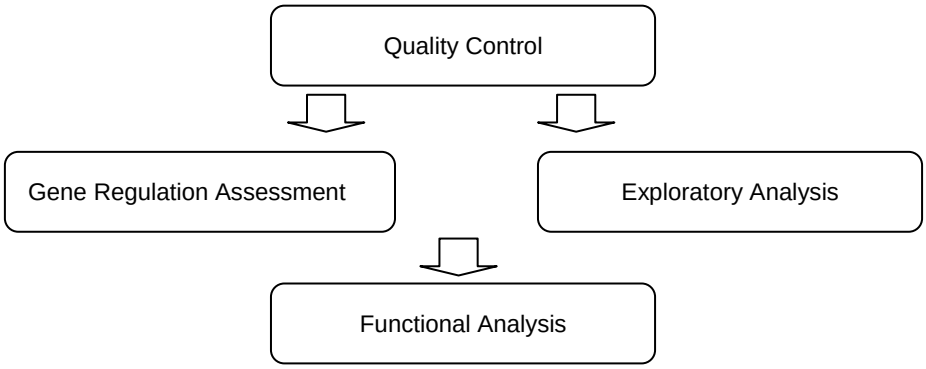



Figure 2.10 – Microarray Data Analysis

## 2.3 Summary

This chapter exposed essential biology theory for the comprehension of the microarray technology. The hybridization on the microarray slides occurs due to complementarity between the probe and the target strands. The mRNA is measured to infer about gene expression in the sample. A microarray experiment was presented, including the preparation of the slides, the probes and the targets, the hybridization and the scanning of the microarrays, and the conversion of the image files to raw data files. With the raw data files and a GAL or ADF file, descriptive of the array layout, the microarray data analysis may begin.

# Chapter 3

# Microarray Data Analysis

The previous chapter presented an overview of the basics of the microarray technology and the initial steps of a microarray experiment, up to the acquisition of the raw data files. The present chapter will go through the fundamental theory of microarray data analysis, for both quality control and gene regulation assessment phases, focusing on the functionalities that were chosen for Mind, as there is not only one exact way of performing microarray data analysis. However, this chapter still exposes a fairly general microarray data analysis approach, and not Mind-specific, because the described methods are some of the most commonly used ones in microarray data analysis and recurrent in related literature. Despite many methods being available for both quality control and gene regulation assessment, only a small percentage is actually used in the majority of experiments.

In Mind, most data analysis functions are offered using R libraries. It is better to use functions that are already available instead of writing new code, especially as the analysis methods are complex algorithms. R is a statistical framework that allows data processing, plot generations, and more [22]. It has its own windows environment or it can be run from the operating system's command line. In both cases data processing and plot generation is done with the R scripting language. This makes R ideal to execute scripts from an application, which is what is done in Mind: the website site provides a GUI to the user, and using his input, processes the microarray data with R. This software is open source and it can be downloaded from the Comprehensive R Archive Network website (CRAN). R base functions are the ones that come with the downloaded software and include arithmetic operations, plots, statistical methods and more. Additional libraries or packages can be downloaded and installed into R, such as libraries specific for the analysis of microarray data. Many bioinformatics related R libraries are released under the Bioconductor project. More information about R and Bioconductor will be presented in the application development chapter (Chapter 5). This brief introduction was required as the current chapter, by introducing the microarray data analysis methods, will present the specific R libraries selected for the Mind R scripts.

Data of a DNA microarray experiment performed at the University of Aveiro Biology department will be used as an example. This experiment aimed to study the result of submitting *saccharomyces cerevisiae* cells to a heat shock. The yeast cells initially were at a temperature of 30 ºC, and they were left to incubate at 37 ºC for twenty minutes. mRNA samples of the cells were taken, before and after the incubation at 37 ºC, leading to two conditions: the "before" or "control" sample and the "after" or "experiment" sample. The heat shock effect on yeast cells is a phenomenon that has been studied thoroughly, either with DNA microarrays or other tools, and many references are available on this matter [23]. This experiment was performed at the lab mainly to check the microarray technology and experimental protocol being used by ensuring the results are as expected. In this chapter, it will be shown how after performing microarray data analysis on the heat shock experiment raw data, a list of regulated genes is obtained.

For the heat shock experiment three biological replicates were used, meaning that the heat shock was applied to three different cultures of yeast, developed under the same conditions, but in three separate containers, yielding the mRNA samples C 1, 2 and 3 (Control), and HS 1, 2 and 3 (Heat Shock). For each distinct culture, two microarrays were created, which are technical replicates. Each pair of technical replicates are also dyes-swaps, meaning that on one array the green channel is assigned to the control and the red channel to the heat shock, and on the other array these are swapped. Dye swap technical replicates are used to make up for different dye efficiencies. Figure 3.1 shows the diagram of the microarray experimental design. Whether the arrows are colored or not, it is a convention that their direction goes from the green labeled sample to the red labeled sample.

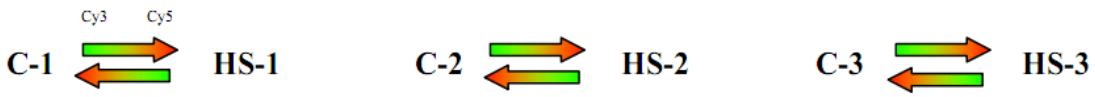



Figure 3.1 – Heat Shock microarray experiment design

The comparison will be done between the control and the heat shock groups, to assess differential expression between the two. A gene that shows a very high intensity on the heat shock group and a low intensity on the control group will be classified as upregulated. A gene that is much less expressed on the experiment group will be downregulated.

Using the heat shock data, this chapter will clarify some necessary concepts like spot and gene expression values and graphical plots used and will then detail the quality control and the gene regulation assessment phases of data analysis. Greater emphasis will be placed on the latter, as quality control was already implemented at the beginning of this project. Although the quality control module is to be revised, its functions, mainly offered through Limma, are suitable for Mind's users and will not be changed.

## 3.1 Expression Values, Scatter Plots and MA Plots

For one microarray, one may plot the green intensities against the red intensities. The originated plot, the scatter plot for that array, shows most spots are scattered along the x = y line, meaning most spots have similar intensities in both samples. Given two conditions, like the control and the heat shock conditions in the example experiment, there are not expected to be many regulated genes. The same applies if comparing a healthy and disease tissue or a medicated and placebo tissue. In fact, gene regulation assessment with DNA microarrays uses this principle as an underlying assumption: most genes will not be differentially expressed among the conditions under study and the goal is to investigate the few genes that are. Scatter plots can be taken for one microarray, for a subset of spots or genes or for multiple arrays (by averaging intensity values across them), as long as two groups of expression values can be plotted one against another.

Scatter plots are taken using the logged intensities. Figure 3.2 shows that this transformation permits a more even distribution of the values along the graphic area and a better reading of the graph too. The last plot, with normalized data, shows the spots scatter better along the x = y line. Before quality control, red and green intensities may show a systematic discrepancy from the x = y line, due to some technical issue that may have affected in the same manner all intensity values.

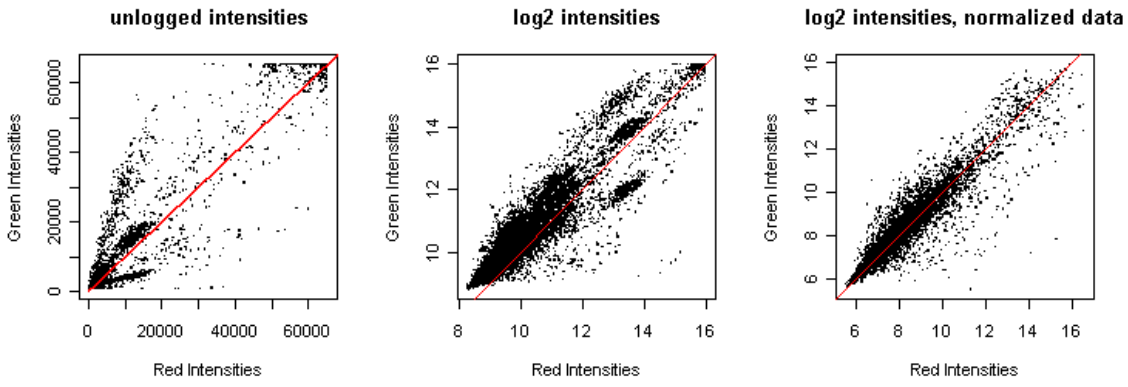



Figure 3.2 – Scatter plots of the first array of the Heat Shock experiment

In two-color microarrays, a spot's or a gene's expression value is the M-value, the base two logged ratio of the red and green intensities. A spot represents a single location on the microarray grid while the gene's expression value is the average of the expression values of all spot replicates that refer to that gene. It should be avoided to consider the individual red or green intensities on their own because they result of competitive hybridization. The intensity a spot shows on the red channel depends on the mRNA present on the red labeled sample, but

also on the mRNA present on the green labeled sample too. For two-color array data, the M-value is the appropriate gene expression value, with the R/G ratio, and not the individual red and green intensities:

$$\text{M-value} = log_2\left(\frac{R}{G}\right) = log_2(R) - log_2(G)$$

The ratio in M-values, along with reflecting best the result of the competitive hybridization in two-color arrays, allows comparison of spot expression on one mRNA sample with the other. A gene with an intensity of 10000 on the red sample and an intensity of 1000 of the green sample has the same relative expression as a gene with a red intensity of 1000 and a green intensity of 100; they both have a 10-fold-change. The ratio is logged so that up and down regulation is accounted similarly. A gene that is 100 times more expressed on the red mRNA sample has a ratio of 100, but a gene that is 100 times more expressed on the green mRNA sample has a ratio of 0.01. By taking the logarithms, the first gene has an M-value of 2, and the second gene has an M-value of -2. If expression values were given by ratios only, there would be a [1,∞[ range for the genes up regulated in the red sample and a [0,1] range for the downregulated genes. M-values allow ranges of equal dimensions for both cases: ]∞,0] for downregulation and [0,∞[ for the upregulation. Logarithms also allow more perceptible plots by distributing the spots more evenly. Figure 3.2 showed how the scatter plot became more perceptible by logging the individual intensities and the same effect is obtained by logging the intensity ratios. Logarithms are taken in base 2 for a matter of convenience, in subsequent data analysis. For example, later in this chapter the fold-change gene regulation assessment method will be explained, a method that easily selects genes that have a fold-change higher than 4, by selecting M-values with absolute value higher than 2; the square of 2, the log fold-change, provides the fold-change value, which is 4 [24-26].

MA plots are two dimensional graphical plots that plot the M-values (expression values, intensity ratios) of a set of genes against the A-values (average intensity values). The average intensity of a spot is given by

$$\text{A-value} = 0.5 \times \left(log_2(R) + log_2(G)\right) = 0.5 \times log_2(R \times G)$$

An MA plot can be generated for the spots of one microarray, for several arrays (by averaging the red and green intensities, or the control and experiment intensities, and then taking the M-values) or for a smaller selection of spots or genes. Figure 3.3 shows the MA plots for the first array of the heat shock experiment, before and after quality control. In both cases, the plot shows a bigger concentration of spots around the M = 0 line, although after quality control

scattering along this line is more evident. Most genes of a sample in two different conditions are not differentially expressed. The aim of gene regulation assessment is to find the small fraction of regulated genes that are.

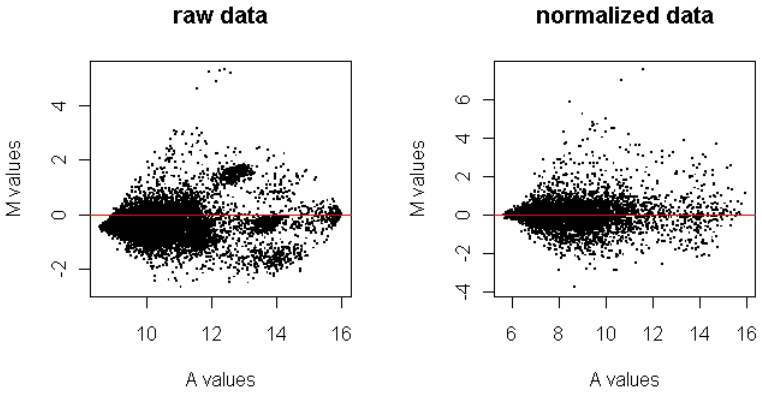



Figure 3.3 – MA Plots of the first array of the Heat Shock experiment

## 3.2 Quality Control

Once the raw data files are obtained, quality control must be performed on the data, involving preprocessing and normalization functions that allow the data to be corrected for systematic biases, such as noise, and the user to check the quality of the arrays. If the data does not look as expected after quality control, which the user can see using the graphical plots generated, than the microarray scanning and/or hybridizations must be done again.

According to some references, normalization can be considered part of preprocessing the data, and thereby quality control and preprocessing would be synonyms. Other references consider preprocessing the filtering and background correction functions and not normalization. It can be simply said that quality control is the collection of all the filtering, background correction and normalization functions applied to the raw data. In this phase, the arrays are separately, and within each array, the spots are handled individually, regardless of the spots being replicates referring to the same gene or not.

It is very important to perform quality control as microarray experiments are typically very noisy and various factors can affect deeply the data. Whether the next step is exploratory analysis or gene regulation assessment, it is important to have the expression values as accurate as possible, in order to infer conclusions out of it. If the influence of systematic factors is minimized, then the random (biological) variability will stand out more clearly [24].

The quality control functions presented in this section (filtering, background correction and normalization) are characteristic of two-color microarray data. These quality control functions were already implemented and available in Mind prior to the beginning of this project, and they have been meeting Mind users' needs adequately. All of them, except for filtering, are offered through the R Bioconductor library Limma. This package, mostly known for its gene regulation assessment functions, also contains a great selection of functions to perform quality control on two-color array data.

### 3.2.1 Filtering

Filtering allows discarding spots that do not seem of interest. This first filtering is not "permanent"; it does not go beyond quality control. Gene regulation assessment will have its own (and a little different) filtering functions. Spots discarded by quality control filtering will still be available for gene regulation assessment. Quality control filtering is used to generate the quality control plots. At the end of quality control, the user will evaluate the quality of the arrays based on MA plots, one plot for each array. A microarray has tens of thousands of spots, so the user may choose to filter out the spots that are barely expressed and that most likely will not be relevant. Figure 3.4 shows how the MA plot of normalized data of the first array of the heat shock experiment, also shown previously in Figure 3.3, is plotted with an intensity filter that selects spots with one or both intensities higher than 1000. Quality control filtering is intended to generate clearer MA plots so the user may concentrate on analyzing the shape of the plot considering the high intensity plots. Again, all data will still be available for gene regulation assessment.

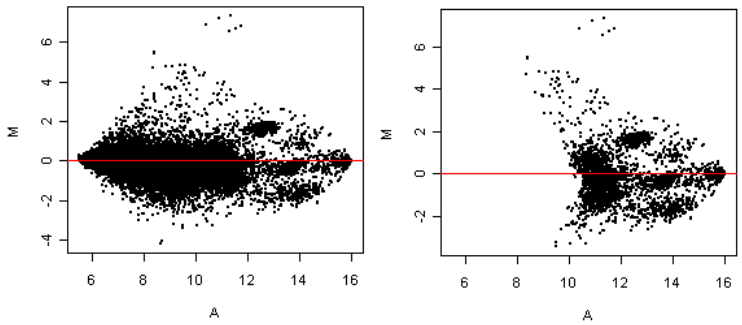



Figure 3.4 – MA Plots: the effect of filtering out low intensity spots

The issue that calls for filtering is as follows. Spots with low red and green intensities can be taken as highly expressed. A spot with a red intensity of 100 and a green intensity of 1 is a low intensity spot, as intensities usually range from zero to tens or hundreds of thousands. However, these intensities suggest the gene is one hundred times more expressed on the red sample and lead to the high M-value of 6.64. In two-color array data, the expression value of a spot is the

ratio of the red and green intensities, but for low intensity spots these ratios may be misleading. The intensity values are so low that they be due to noise and the gene may better be classified as equally and non-expressed in both samples. Low intensity spots, even if they show high M-values, are hardly relevant when the objective is selecting regulated genes. To prevent these genes from figuring in the regulated gene list the user must perform post-normalization filtering, but if he wishes to clear the quality control graphical plots from low intensity spots he must perform filtering in this phase.

Filtering by intensity selects only the spots with big enough red and/or green foreground intensities. Filtering by signal to noise (S2N) ratio selects only the spots whose red and/or green foreground intensity is much bigger than the background intensity. Both these filters are implemented in Mind's quality control module. In practice, both filters, intensity and signal to noise ratio, will lead to similar results – spots with high foreground intensities usually have high signal to noise ratios, and can be selected with any of the filters. For example, in the first array of the Heat Shock experiment, the selection of spots with one or both of the intensities higher than 2000 overlaps considerably with the selection of spots with one or both of the S2N ratios higher than 5, as shown in Figure 3.5.

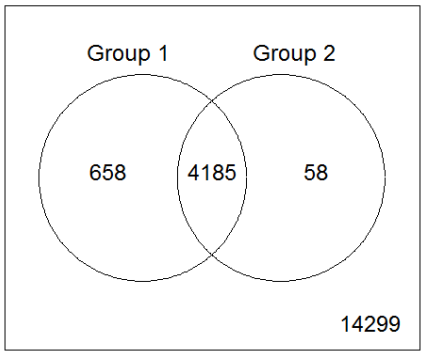



Figure 3.5 – Intensity and S2N ratio filters overlap

One of the columns of the raw data file is a bad spot flag created by the image analysis program to indicate spots that are of bad quality in a way that the intensity values cannot be reliably provided, although there should not be many such spots in an experiment. In the Mind R scripts, this column is read from the raw data files too and treated by Limma as weights that are taken into account during normalization. Limma's handling of bad spots is transparent to the user and independent from the quality control filtering described, and its purpose is also different. The quality control signal to noise ratio and intensity filters do not affect data normalization at all, just the spots that appear on the graphical plots.

### 3.2.2 Background correction

The image analysis program provides data for each and every spot of a microarray (as shown previously in Figure 2.7). The attributes of a spot include red and green foreground and background intensities. The foreground intensity referrers to the spot's total measured intensity in the red or green channel. The background intensity is the ambient signal measured, a result of non-specific binding and spatial heterogeneity across the array.

The background intensity, detected by the microarray scanner and measured by the image processing software, can be due to several factors, such as non-specific binding of the labeled sample to the array surface, natural fluorescence of the glass or its coating or optical noise from the scanner [24, 27].

Background correction is the process of readjusting the foreground intensity values, taking in consideration the detected background values, providing new foreground intensities. Choosing not to use background correction means to consider the new foreground intensities equal to the original ones, thus discarding the background intensity values. In fact, some references advise not to use background correction, as it does not improve the detection of differentially expressed genes, and reduces the precision of data by increasing the variability of low intensity spots values [28].

The most intuitive and common method involves simply subtracting the background values to the foreground values, for both red and green channels, and it is referred to as the subtract or the standard background correction method. Although this method is simple and quite logical, it produces negative intensities when the background intensity detected is larger than the foreground, which in turn will lead to missing expression values (logarithms of negative values cannot be taken). The negative intensities, also referred to as "black holes", occur when the mRNA binds more to the surface than to the spot itself, causing higher non-specific binding than specific binding [29]. Some control probes may have been spotted on the microarray to ensure no mRNA binds to those spots. Although the heat shock data, in study throughout this document, has no such spots, other data may have.

Even when not missing, using background subtract may cause the log-ratios for low intensity spots to be highly variable [27]. The method most commonly subtracts the mean background intensity to the mean foreground intensity, for the red and for the green channel. This is the way it is used in Mind. But other spot attributes can be taken instead of the mean background, such as the spot's morph value, which is a non-linear filter that provides values that are lower and less variable than the background intensity.

Improved versions of the standard method do not perform the subtraction if the result is negative. In the minimum method, in spots that would yield zero or negative values, the new intensity is set to one half of the minimum positive corrected foreground intensity for that array. In the half method, spots that would yield less than 0.5 with the subtract method are set to 0.5. These two methods are available in the Limma library and offered in Mind.

Limma also includes more complex methods, norm exp, edwards and moving minimum, which are offered in Mind. M. Ritchie et al compare and describe several methods in [27] and use them to assess differential expression, concluding that the more complex model-based correction methods perform better than the standard background subtraction.

### 3.2.3 Normalization

The vast majority of spots in an experiment are not differentially expressed between the two RNA samples. When comparing for example, healthy and diseased or placebo and drug treated tissues, not many genes are expected to be regulated between the pair of samples. Therefore, most spots must have equal red and green intensities and M-values around zero. In an MA plot, most genes are expected to be around M = 0. However, some kind of systematic variation that occurred during the microarray experiment may cause data to be constantly distributed differently, for example, around M = 1.

Normalization aims to identify systematic variation in the data of an array and to correct it. If all spots appear to be increased by 1 in their expression values due to some bias, then it will make sense to adjust the red and green spot intensities in order to achieve expression values that are concentrated around M = 0. This is the basic idea of normalization. It corrects the data for systematic variation, caused by different reasons such as different incorporation efficiency of dyes, different amounts of mRNA, printing or print-tip problems, so the data will express better the biological variation. Ideally a normalization method would enable the data to express *only* the unsystematic variation, but as it is not possible, normalization methods attempt to minimize as much as possible the effect of non-biological variation.

The Limma package implements the normalization methods that are offered in Mind: median, loess, print-tip loess and robust spline. The median method subtracts the weighted median from the M-values for each array. The other methods are more complex and are explained further in the Limma user's guide and help manual. Print-tip loess handles data on each print tip at a time. A print-tip is each of the twelve squares shown on the example of Figure 2.8. Print-tip loess generates a curve for each print-tip using local regressions, and adjusts the intensity values according to that curve. Figure 3.6 shows this method aligns the averages per print-tip. Global

loess does a similar procedure but for the entire array. Robust spline implements an idea similar to print-tip loess, but uses regression splines in place of the loess curves.

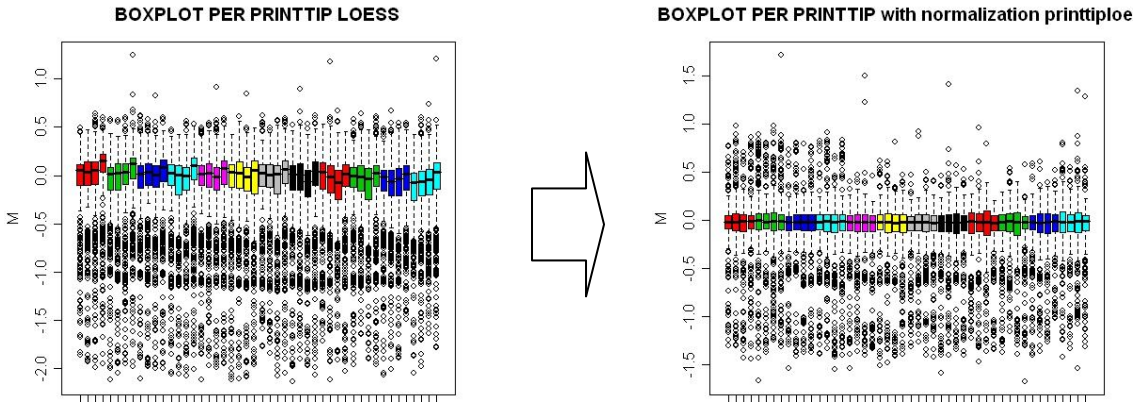



Figure 3.6 – Print-tip loess aligns the averages per print-tip

More details about the normalization methods implemented in Limma are available in Limma help and in the article by G.K. Smyth about normalization of cDNA microarray data [30]. More information about microarray data normalization in available in many good sources, such as the introduction to Microarray Data Analysis by M. Babu [9], the book by J. Quackenbush et al [24] and the book by S. Draghici [25].

Normalization concludes quality control. Limma contains functions that generate MA plots and other useful graphics for the raw, background corrected and normalized data. Mind quality control produces one report for each array with these graphics, allowing the user to assess the quality of the arrays. It may happen that the data does not look at all like the MA plots that have been presented in this chapter, in that case something went wrong with the experiment and it must be redone. Otherwise, if after background correction and normalization the data looks adequate, the differential gene expression analysis can begin.

## 3.3 Differential Gene Expression Assessment

Differential gene assessment starts with averaging the spot replicates. This was not required in quality control, which intended to check the quality of each array as a whole and correct each of its individual spots for systematic bias. The present phase, using the normalized data, aims to select genes that are differentially expressed. Spots that refer to the same gene (identified by its name in the GAL/ADF file) should be averaged. This step is not strictly obligatory, although highly recommended. If spot replicate averaging is not done, than the spots will be considered independently. In this phase, the spots are no longer considered, but genes instead. Without the averaging, each spot will be regarded as a distinct gene. Simply averaging the spots disregards replicate variability, but it is currently the most commonly used method [31].

After the spot replicates averages are taken, filtering is (optionally) performed on the data. The filtering done in quality control is no longer considered, as it filtered only the spots to appear on the quality control graphical plots. Filtering on this phase will select the group of genes onto which the statistical method is applied. Finally, on the filtered genes, or on all genes if filtering was not done, a statistical method is applied to output a list of regulated genes. This section will describe filtering and will present several statistical methods. The chosen R libraries to offer the statistical methods in Mind will also be presented. The statistical methods will be exemplified using the non-filtered data, with spot replicates averaged (background corrected with the subtract method and normalized with the print-tip loess).

### 3.3.1 Filtering

The reason for filtering is to discard low intensity genes that can otherwise figure in the regulated gene list if they have high M-values. Quality control filters only selected spots for inclusion in the graphical plots that helped the user assess the quality of each individual microarray. A spot would be selected or not to figure in the MA plots of its array. Those were per array plots and the filters selected or excluded each individual spot.

Now the goal is to identify regulated genes. The spots are no longer considered individually, but the genes (spot replicate averages). The arrays are no longer considered separately as the genes will be declared as regulated or not based on their expression values throughout all the arrays. Filtering is now done based on the values the genes show in all the arrays. A gene cannot be selected on one array and excluded on another, either it is selected or it is not.

The genes are filtered to apply a statistical method on the selection, so the number of selected genes cannot be very low. In fact, it is questionable whether filtering may be recommend, since it often interferes in the results of the statistical test which will declare the genes as regulated with higher confidence than it would if no filtering was previously done [32]. The user must decide what the best option is.

Figure 3.7 shows an example of an experiment with three RNA samples, different from the heat shock example, where each one is compared with the other two. Filtering by the expression values of a gene throughout all the arrays is non-specific, meaning the filter considers the expression values of the gene on all six arrays, regardless of what mRNA samples were hybridized on the arrays.

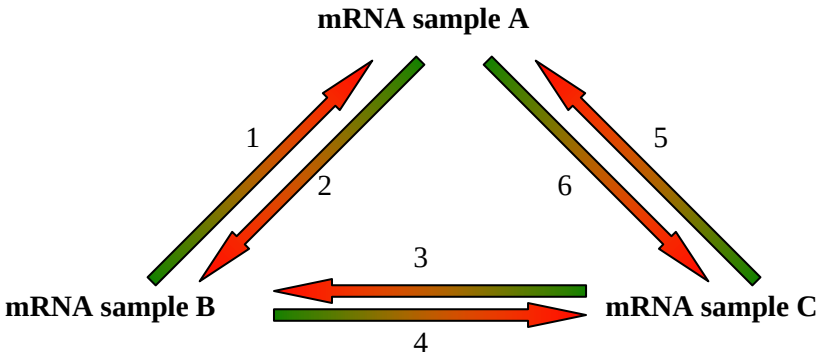



Figure 3.7 – A microarray experiment design with 3 mRNA samples and 6 arrays

The filters chosen to be implemented in Mind are the intensity and the standard deviation filters. The intensity filter will select genes that have red and/or green intensity values higher than a certain unlogged intensity value, in a minimum number of arrays. The minimum red and green intensity values and the minimum number of arrays are both user-specified. The input provided is similar for the quality control intensity filter (also in order to provide a similar GUI for both) but now the gene is selected based on the values on all arrays. The standard deviation filter will select genes that have a high variability throughout the several arrays.

Other possible filters would be the variation filter and the fold-change filter. The variation filter would select genes whose minimum and maximum expression values have a minimum gap between them, where that minimum difference is user specified. The fold-change filter would select genes that have a specified number of measurements with fold-change above a certain minimum, being each spot's fold-change calculated as the difference between its expression value and the gene's mean or median expression value. These two filters will not be offered in Mind. The fold-change filter was found to be rather slow, and the two previous filters were found to suffice.

### 3.3.2 Statistical methods

On the group of genes that passed the filtering, or on all genes in case of no filtering applied, a statistical method is chosen to output a list of regulated genes. In a microarray experiment, thousands of genes are evaluated at once and only a small group is to be selected as regulated among the conditions under study. A microarray experiment also has noise, systematic variation, and other random factors that decrease the reliability of the expression measurements. Besides, there usually are not many replicates to compensate for this non-biological variability – each condition usually includes around 2 to 8 replicates. Putting together thousands of genes, few replicates and lots of noise, it can be seen why microarrays offer a big statistical challenge for assessing regulated genes.

The most basic methods are not truly statistic, although they are often included in the same group as the true statistical methods – they are all referred to as statistical methods (possibly because "gene regulation assessment methods" is a name a bit too big for a menu item in a software application). These basic gene regulation assessment methods do not take variability into account and can be used even when there are no replicates. It will be explained how the fold-change method can be applied to the data of a single array, or to the data of several arrays by taking the means of the expression values.

It is strongly recommended that the arrays do have replicates. Even if they cannot be many (arrays are expensive), replicates help ensure the gene expression values considered are not due to chance. And while a method like the fold-change can only output regulated genes based on the means of the expression values, a statistical method takes the variability of the expression levels of the genes across the several replicates into account. They do not say for sure what genes are regulated, but they quantify the probability of a gene to be regulated, so the user is able to select genes with, for example, a probability of 95% or 99% of being regulated, based on expression values and variability. There are many statistical methods used for gene regulation assessment, new methods are being developed and existing methods are being perfectionned continuously to allow gene regulation analysis to be done quicker, easier and more reliably. The vast majority of microarray data analysis is done with a small number of common statistical tests, including the t-test, ANOVA, Limma and SAM, which have been selected to be offered in Mind. T-test and ANOVA are classical statistical tests that have been found very useful for microarray data. Limma and SAM are methods created specially for microarray data. These methods will be explained further.

Methods like the fold-change are not very reliable because they do not take into account the variation of the data through the several array replicates, and they are not enough for drawing

conclusions about the conditions under study [33]. However, they are still commonly used to have a first look at the data, or to analyze non-replicated data. Specially for these methods, it is extremely important to ensure that adequate background correction and normalization procedures were applied on the data, otherwise a lot of genes will appear as regulated when they are not [34]. A common approach for the biologist is to the use the fold-change method to gain an intuitive perception of the data but then use a statistical method to obtain results he can include in a scientific report.

### 3.3.3 The fold-change

In a microarray experiment, most ratios of gene expression levels are expected to be around one, and most log ratios to be around zero. In the histogram of Figure 3.8, relative to the first array of the heat shock experiment (normalized data, spot replicates averaged), the peak at M = 0 is evident. The regulated genes have one of the two intensity values very high relative to the other and are at the tails of the histogram. A common choice is to select genes that have a fold-change greater than 4, which is done by setting thresholds at ± 2 (a gene 4 times more expressed on the red sample has an expression value of $\log_2(4) = 2$, a gene 4 times more expressed on the green sample has an expression value of $\log_2(0.25) = -2$) [26]. In the histogram, this means tracing vertical lines at y = –2 and y = +2 and selecting the genes that are outside of those lines. In the first heat shock array, there are 34 genes with M-values below 2, and 112 genes with M-values above 2. As the red sample is the heat shock sample, genes in the right tail of the histogram are upregulated, so 112 genes. The 34 genes of the left tail are downregulated.

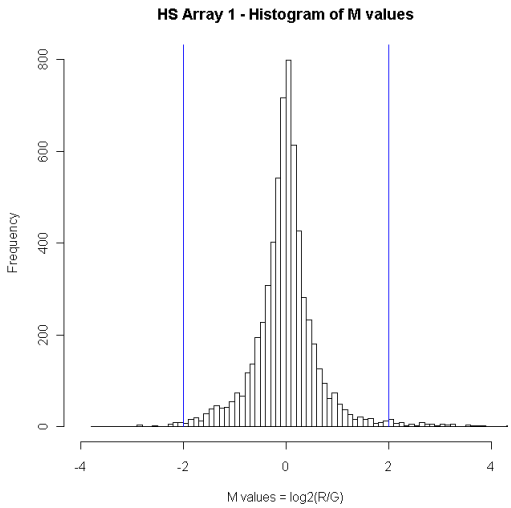



Figure 3.8 – Histogram of M-values of the first array of the HS experiment

A gene is up or downregulated when comparing two conditions, meaning it is always upregulated in one condition relative to the other. When the conditions are not specified, a gene is characterized as upregulated if it is more expressed in the experiment condition relatively to the control. A downregulated gene is less expressed in the experiment when comparing to the control.

Using a scatter plot to apply the fold-change selection method to the same data, the log experiment intensities are plotted against the log control intensities (Figure 3.9). The same number and list of regulated genes can be obtained as with the histogram, but the scatter plot allows for a better perception of how many genes are regulated and by how much. In the figure, the heat shock sample intensities are plotted along the x-axis and the control sample intensities are plotted along the y-axis. Genes that are more expressed on the experiment are colored red. Genes that are more expressed on the control are colored green. The colors are not related to the fluorescent dyes, although in the present case, as the heat shock sample was hybridized to the red channel and the control sample was hybridized to the green channel, the upregulated genes coincide with the genes that show higher red intensities. It is a convention to plot the upregulated genes red, and the downregulated genes green.

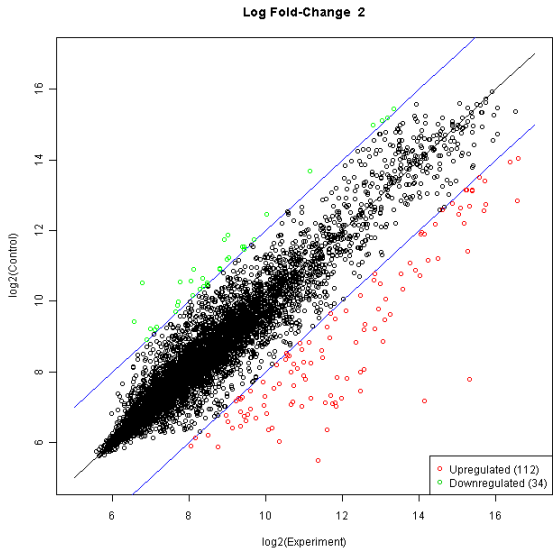



Figure 3.9 – Fold-change: Scatter plot of the first array in the HS experiment

The fold-change method can also be applied when there are array replicates. In the heat shock experiment, the control logged intensities and the experiment logged intensities are averaged, regardless if they are red intensity values or green intensity values, and the two conditions plotted in a scatter plot (Figure 3.10). Considering the same thresholds, there are 83 upregulated genes and 16 downregulated genes.

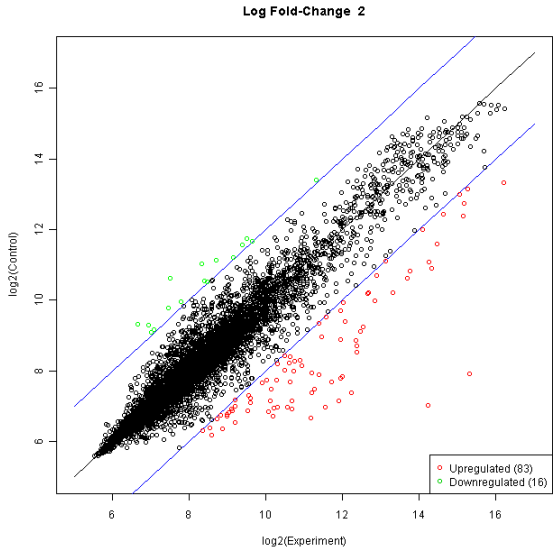



Figure 3.10 – Fold-change: Scatter plot of the Heat Shock experiment (six arrays)

Selecting a different fold-change will depart or approximate the fold-change threshold lines from the x = y line, selecting less or more regulated genes. Considering the same data, selecting genes that have a log fold-change superior to 4.25 (fold-change higher than 18) provides a small group of 6 regulated genes (Figure 3.11). The six genes can be identified using the information of the GAL or ADF file, and their names are shown in Table 3.1.

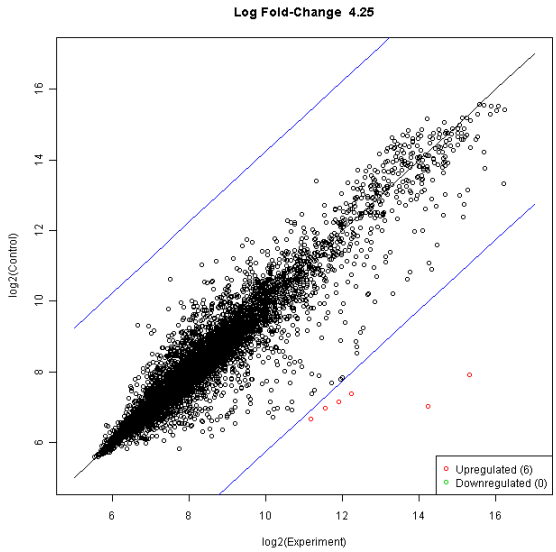



Figure 3.11 – Fold-change: Scatter plot of the Heat Shock experiment (six arrays), log fold-change higher than 4.25

| Gene Number | Gene Name | Log Fold-change |
|---|---|---|
| 1770 | YFL014W | 7.38 |
| 3059 | YBR072W | 7.20 |
| 349 | YLR178C | 4.86 |
| 4074 | YMR169C | 4.73 |
| 1461 | YPR160W | 4.57 |
| 4426 | YGR248W | 4.49 |

Table 3.1 – Fold-change: HS experiment top 6 genes

The importance of averaging replicated spot should be remembered. If spots replicates were not averaged, first a more crowded scatter plot would be obtained, as shown if Figure 3.12 (19200 spots instead of 6342 genes). Then, the six spots that would stand out most would be: two replicates of gene YFL014W, two of YER103W and two of YBR072W. The YER103W shows up in the top six spots, but does not show up in the top six genes; in fact, the gene shows an average fold-change inferior to 2.0. The reason is that this gene has 34 spots replicates on the array, and only two of them show very high fold-changes. Averaging spot replicates, this gene goes down on the ranking.

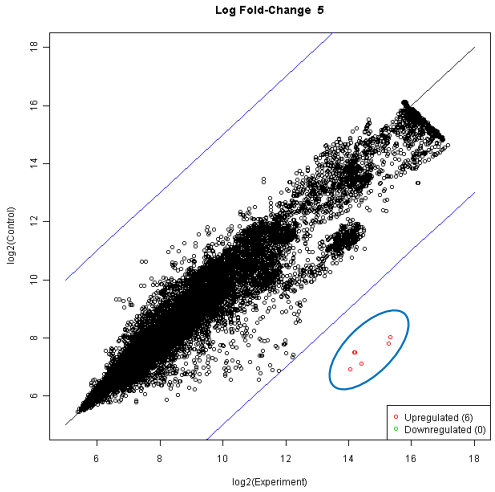



Figure 3.12 – Not averaging spot replicates brings up different "regulated genes"

Although the conclusions of a study should not be based on fold-change analysis, knowing a few genes to look out for before proceeding to a statistical test can be helpful. However, this method does have disadvantages: besides not taking into account gene expression variability among the samples of a condition, the fold-change method does not handle low intensity genes specially. The microarray technology tends to have a bad signal to noise ratio for genes with low expression levels, meaning that genes with low intensities have higher variability. On a scatter plot this is show by a funnel shape distribution, showing higher variability in the low end. In Figure 3.13 the curved lines attempt to show this effect, although it is not very evident in

the heat shock data. Genes further from the diagonal are less reliable if they are in the low expression level region of the graph (inferior-left corner) [25, 26].

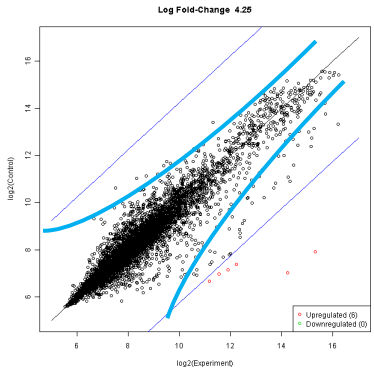



Figure 3.13 – The funnel shape

The term fold-change has two different definitions that should be clarified (Table 3.2). The standard definition of a gene's fold-change is the ratio of the experiment intensity by the control intensity, and the ratio log fold-change is the base two logarithm of this ratio. For more than one array, the ratio log fold-change is not the logarithm of the ratio "average of the experiment intensities / average of the control intensities", but the difference between the mean log two experiment intensities and the mean log two control intensities (the logarithm of a mean does not equal the mean of the logarithms), as it is calculated in Mind. The other definition is the difference between the mean M-values in the experiment and the mean M-values in the control samples [35]. The fold-change method had to handle red and green intensities individually as a method applicable to a single array. However, as it was clarified in section 3.1, the true expression value for a gene is the M-value, since it is two-color array data. The M-value is the expression value most often used in two-color array data analysis. With M-values, the fold change is the difference between the mean of the M-values of the experiment arrays and the mean of the M-values of the control arrays. The explanation of t-statistics will help clarify both.

| Ratio Log Fold-change |
| --- |
| $\log_2\left(\frac{\text{Experiment}}{\text{Control}}\right) = \log_2(\text{Experiment}) - \log_2(\text{Control})$, for one array<br>$\text{mean}\left(\log_2(\text{Experiment Red or Green Intensities})\right) - \text{mean}\left(\log_2(\text{Control Red or Green Intensities})\right)$, for several arrays |
| Difference Log Fold-change |
| $\text{mean}\left(\log_2\left(\frac{\text{Red}}{\text{Green}}\right)\right)_{\text{Experiment Arrays}} - \text{mean}\left(\log_2\left(\frac{\text{Red}}{\text{Green}}\right)\right)_{\text{Control Arrays}}$ |

Table 3.2 – Ratio and Difference Log Fold change

### 3.3.4 T-statistics

T-statistics, based on the t-distribution, are the simplest of the statistical methods used to assess differential expression [25, 36]. The t-distribution resembles the normal distribution, which is one of most important distributions ever. This section will start with a brief description of the normal distribution in order to present the t-distribution and t-statistics more clearly. Then, hypothesis testing using t-statistics, or the t-test, will be detailed.

## Hypothesis testing using the normal distribution (z-test)

Many natural phenomena, including gene expression levels, follow the normal distribution. The probability density function of the normal distribution has a bell shape. A normal distribution is characterized by a mean and a standard deviation, which is typical of the population. Although the bell shape is constant for all normal distributions, the curve may be widen with a larger standard deviation or narrowed with a smaller standard deviation. The curve is positioned on the x-axis according to the mean of the distribution. The standard normal distribution or z-distribution has mean zero and standard deviation one (Figure 3.14).

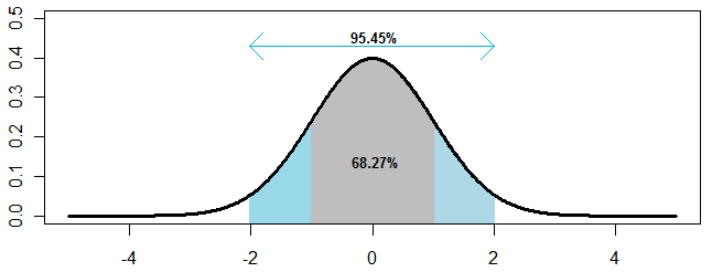



Figure 3.14 – Standard Normal Distribution Probability Density Function

In a normal distribution, 68.27% of the values are within 2 standard deviations of the mean (between one standard deviation to the left and one standard deviation to the right). 95.45% of the values are within 4 standard deviations of the mean. These values are tabled, in the standard normal distribution table, also called z-table. The reverse can also be asked: outside of what limits reside 5% of the distribution's values? The answer is -1.96 standard deviations left from the mean and 1.96 standard deviations right from the mean. These questions are answered with the aid of z-tables, which tell the amount of distribution that fall between a given z-value and the distribution's mean (zero), for example: for the z-value of 1.0 the amount is 0.3413, allowing to calculate $2 \times 0.3413 \times 100 = 68.27\%$. The actual paper z-table is not actually used as often, with the many computational tools, like Excel, R and scientific calculators that provide these values.

The percentages of the z-table apply to any normal distribution: all the normal distributions have 68.27% of the values between -1 and 1 standard deviations from the mean. But the z-table provides values that are based on the standard normal distribution, so the normal distributions must be mapped onto the standard in order to use the z-table. Any normal variable can be mapped onto the standard normal distribution by applying a z-transformation: subtracting the population mean to each element and dividing the result by the population standard deviation always yields the standard normal distribution (z-value=$(x-\mu)/\sigma$). Standardizing a normal sample means to convert its measurements to z-values or z-scores using the z-transformation.

Given a sample of a population, and known the population mean and standard deviation, it can be tested whether the mean of a sample is different from the population mean, and if so, if the difference is significant. This is performing hypothesis testing using the standard normal distribution, or a z-test. The null hypothesis is stated – that the sample's mean is equal to the population mean, $h_0: \mu_1 = \mu_P$. The alternative hypothesis is that they are different, $h_a: \mu_1 \neq \mu_P$. A level of significance must be chosen, for example 0.05. This represents a maximum chance of 5% of rejecting the null hypothesis when it is in fact true. If it is known in advance that the sample's mean can only be higher or that it can only be lower than the population's mean, a one-tail test is performed, otherwise, a two-tail test must be conducted.

For a two-tailed test at a significance level of 0.05, the z-value of 0.025 (0.05 divided by two) provided by a z-table is 1.96. This means that in a standard normal distribution, 2.5% of the values are 1.96 standard deviations left from the mean and 2.5 % of the values are 1.96 standard deviations right of the mean (Figure 3.15). These are the distribution's tails. Then the z-value of the sample under study is looked up. If it falls on one of the tails of the distribution, the null hypothesis is rejected, and it can be concluded that there is a significant difference between sample and population. Otherwise there is not enough evidence to reject the null hypothesis at the chosen significance.

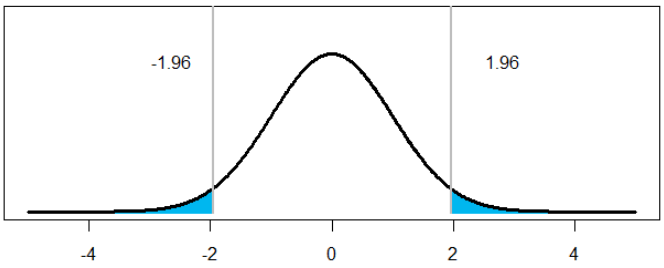



Figure 3.15 – Standard Normal Distribution: two-tailed z-test at a significance of 0.05

This brief summary of the normal distribution and hypothesis testing was only to introduce the t-distribution and hypothesis testing with t-statistics. In microarray data analysis, the normal distribution will not be used because the population mean and the standard deviation for a gene's expression value are unknown; the only information is the one provided by the measurements of the sample. Besides, the hypothesis testing using z-test compares a sample mean with a population mean. In microarray data analysis, usually two (or more) samples are compared with one another.

## Hypothesis testing using the t-student distribution (t-test)

The t-distribution was discovered by W. S. Gosset while working in Guinness Brewery in Ireland. He published his work under the pseudonym Student, which is why the distribution is also referred to as the Student's t-distribution. The t-distribution is similar to the normal distribution, but it has an additional parameter – the number of degrees of freedom. Generally, the more the number of measurements or replicates in each sample, the higher the number of degrees of freedom, and the closer the distribution is to the normal. If there are few degrees of freedom, a higher percentage of values reside in the tails of the t-distribution.

For assessing differential expressed between two samples using t-statistics, a common choice is to use t-statistics with assumption of equal variances. In this case, the number of degrees of freedom is given by $d.f. = n - 2$, where n is the number of samples in both groups together (the two groups may not be equally sized).

Hypothesis testing using the t-test compares the two groups with one another. Only the means and the variances (or standard deviations) of the two samples are known. The goal is to conclude if the two samples can be considered from the same population or not, meaning, if they are equally expressed or if they are regulated.

Just like it was done in the fold-change method, for each gene, the twelve individual intensities are divided into experiment and control, the base two logarithms of the individual intensities are taken and finally the mean experiment value and the mean control value are calculated, as indicated by the ratio log fold-change formula for multiple arrays of Table 3.2. These mean values, exemplified in Table 3.1 for the gene YFL014W, are taken just like they were for the fold-change method, but for the t-test the standard deviations are also required.

| | Array 1 | Array 2 | Array 3 | Array 4 | Array 5 | Array 6 |
|---|---|---|---|---|---|---|
| Red | 40653.98 | 273.6825 | 27630.72 | 266.8208 | 46991.98 | 276.9533 |
| Green | 221.8654 | 42391.34 | 196.3839 | 45337.64 | 224.0685 | 42338.3 |
| | | | | | | |
| Control (log2): | 7.793541 | 8.096359 | 7.617533 | 8.059727 | 7.807796 | 8.113499 |
| Experiment (log2): | 15.31111 | 15.37148 | 14.75399 | 15.46842 | 15.52013 | 15.36968 |
| | | | | | | |
| | | | | | **Mean** | **Std. Dev.** |
| | | | | **Control** | 7.9147426 | 0.203946 |
| | | | | **Experiment** | 15.299134 | 0.277502 |

Table 3.3 – T-test: division of the twelve intensities of YFL014W into the control and the experiment groups

A hypothesis test can be performed to see if the gene is differentially expressed between the two groups, similar to the z-test presented previously, but now, instead of using the normal distribution, the t-distribution with 10 degrees of freedom is regarded. The null hypothesis is that the means of these samples are equal. The alternative hypothesis is that they are different. There are no assumptions of which one can be bigger, so it is a two-tailed test. For a significance level of 0.05, and given that it is a two-tail test, each tail of the t-distribution has 2.5% of the values. Reading from a statistical table, the t-statistic value of 0.025 under 10 degrees of freedom is 2.228. The formula to calculate the t-statistic for the gene, with equal sample sizes and assuming equal population variance, is

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{(n_1 - 1)S_{X_1}^2 + (n_2 - 1)S_{X_2}^2}{n_1 + n_2 - 2}} \times \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where $\overline{X_1}$ and $\overline{X_2}$ are the means of the two samples, $S_{X_1}^2$ and $S_{X_2}^2$ are the standard errors of the the samples and $n_1$ and $n_2$ are the sample sizes. When $n_1$=$n_2$, the formula becomes

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{S_{X_1}^2 + S_{X_2}^2}{\frac{n}{2}}}}$$

where $n$ is the total number of measurements in both samples. Applying the formula with the values for gene YFL014W (Table 3.3) this yields

$$t = \frac{15.2991 - 7.9147}{\sqrt{\frac{0.2775^2 + 0.2039^2}{6}}} = \frac{7.384391}{0.140595} = 52.5225$$

If the t-statistic falls on one of the tails, the null hypothesis can be rejected. If it falls on the non-rejection region, there is not enough evidence to conclude the gene is regulated. The example t-statistic is clearly outside of –2.228 and +2.228, so the null hypothesis is rejected, and the gene is declared as deferentially expressed.

Instead of determining the rejection area relative to a chosen significance and seeing whether the t-statistics falls in or out the rejection area, a p-value can be calculated for the gene's t-statistic, before a significance level is chosen. The p-value represents the minimum significance with which the null hypothesis can be rejected and the gene considered regulated. The p-value is the probability of rejecting the null hypothesis when it is in fact true. It is the probability of concluding that the gene is regulated when it is not, the probability of a regulated gene being a false positive. If a gene's p-value is higher that the chosen significance, the null hypothesis cannot be rejected and therefore the gene cannot be accepted as regulated.

The p-values for a t-statistic, given the number of degrees of freedom of the t-distribution and the test type (two-tail or one-tail) can be obtained from a t-table or from a software tool such as R. For the example YFL014W gene, for the t-statistic of the 52.52, considering a two-tail t-test and the t-distribution with 10 degrees of freedom the p-value is $1.5\times10^{-13}$; this is an extremely low p-value and the gene is regulated at a significance level of 0.05, 0.01 and 0.001.

Multtest is an R Bioconductor library that, among many functions, calculates t-statistics for a set of gene expression values [37]. For the heat shock example, once obtained the expression values and defining the control and experiment groups, the t-statistic is calculated for all the 6,342 genes at once. The expression values matrix is the 6,342 × 12 logged intensities (data), and the groups are defined by a vector of integers that describes to what group belongs each of the twelve columns of data (y). If the twelve columns of the matrix are ordered by ascending array number, the odd columns refer to red intensities and the even columns to green intensities, then the grouping vector would be "1,0,0,1,1,0,0,1,1,0,0,1".

```
data <- cbind(A1R,A1G,A2R,A2G,A3R,A3G,A4R,A4G,A5R,A5G,A6R,A6G)
y <- c (1,0,0,1,1,0,0,1,1,0,0,1)
tstatistics <- mt.teststat(data,y,test="t")
```

Obtained the t-statistics, p-values are calculated for the genes. This is done with a function of the statistics R package (and not Multtest), which comes with the R framework, the probability distribution function for the t-distribution (pt). The p-value for the gene YFL014W can be obtained with

```
2 * pt(52.5225,lower.tail=F,df=10) #result: 1.515109e-13
```

The p-values for all the 6,342 t-statistics of the experiment are obtained with

```
rawpvalues <- 2 * (pt(abs(tstatistics),lower.tail=FALSE,df=10))
```

Besides the p-values, the user should also consider the fold-change of the genes in order to consider them as regulated. In a volcano plot, the p-values of the genes are plotted against their fold-change. The negative logarithms of the p-values (base 2 or 10) are taken so the graph has the V shape. Figure 3.16 shows the volcano plot illustrative of performing the t-test on the heat shock experiment. The most interesting genes have low p-values and high fold-changes, so they are in the upper corners of the plot. The regulated genes are exactly the same ones as the fold-change method selected, minus the genes that have a p-value higher than the chosen one.

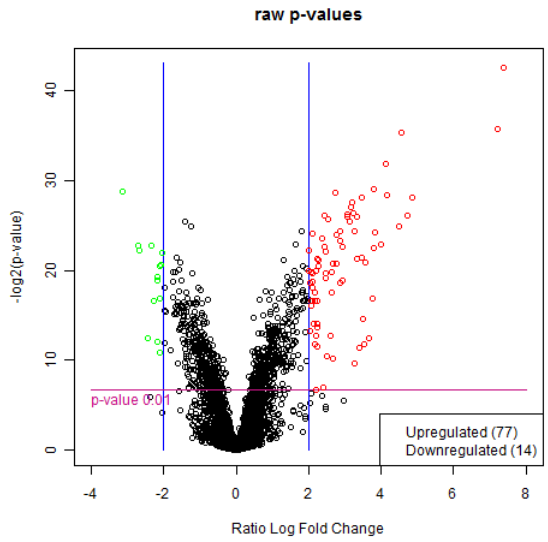



Figure 3.16 – Volcano plot: t-test, p-values vs ratio log fold changes

The six genes with highest log fold-changes also have very low p-values (Table 3.3). These are the same top six genes found through fold-change analysis, the ratio log fold-change values are the same, but now a confidence value is assigned to each gene's log fold-change value.

| Gene Number | Gene Name | Log Fold-change | t-statistic | Raw p-value |
|---|---|---|---|---|
| 1770 | YFL014W | 7.38 | 52.52 | 1.52E-13 |
| 3059 | YBR072W | 7.20 | 32.44 | 1.82E-11 |
| 349 | YLR178C | 4.86 | 19.10 | 3.36E-09 |
| 4074 | YMR169C | 4.73 | 16.50 | 1.39E-08 |
| 1461 | YPR160W | 4.57 | 31.56 | 2.40E-11 |
| 4426 | YGR248W | 4.49 | 15.06 | 3.36E-08 |

Table 3.4 – Top genes: t-test using ratio log fold changes

With two-color array data, it is generally preferable to perform a statistical test with the M-values, instead of the individual logged red and green intensities. Instead of using the ratio log fold-changes, the difference log fold-changes will be considered (formulas presented previously in Table 3.2). In the heat shock experiment, the arrays can be divided into two groups, control and experiment, according to the mRNA samples hybridized on the red and green channels of each array. In arrays 1, 3 and 5, the experiment group, the control was hybridized on the green channel (Cy3) and the heat shock sample on the red channel (Cy5). On the even arrays, the control group, it was the opposite. Taking one of the genes as an example, the YFL014W gene, the following table, Table 3.5, divides its M-values by the control group (arrays 2, 4 and 6) and the experiment group (arrays 1, 3 and 5).

| | | | | | |
|---|---|---|---|---|---|
| Control Group | -7.27512 | -7.40869 | -7.25618 | Mean: | -7.31333 |
| | | | | Std deviation: | 0.08313 |
| Experiment Group | 7.517568 | 7.136453 | 7.712331 | Mean: | 7.45545 |
| | | | | Std deviation: | 0.29291 |

Table 3.5 – T-test: division of the six M-values of YFL014W into the control and the experiment groups

Using the M-values, the t-statistic for the YFL014W gene is 84.01, performing a two-tail t-test and using the t-distribution with 4 degrees of freedom, and its corresponding p-value is $1.20\times10^{-7}$. It still is a very low p-value.

$$t = \frac{7.45545 - (-7.31333)}{\sqrt{\frac{0.29291^2 + 0.08313^2}{3}}} = \frac{14.76878}{0.17579} = 84.01354$$

The Multtest library and the base R pt function are used in the same way, but now the data matrix is the 6,342 × 6 M-values, the grouping vector is "1,0,1,0,1,0" and the number of degrees of freedom of the t-distribution used is 4:

```
data <- cbind(A1M,A2M,A3M,A4M,A5M,A6M)
y <- c (1,0,1,0,1,0)
tstatistics <- mt.teststat(data,y,test="t")
rawpvalues <- 2 * (pt(abs(tstatistics),lower.tail=FALSE,df=4))
```

The volcano plot and the top six genes obtained are shown in Figure 3.17 and Table 3.6. The volcano plot x-axis range is different, as the fold-change is calculated differently; it is the difference log fold-change. Nevertheless, the top six genes are the same, and still have low p-values.

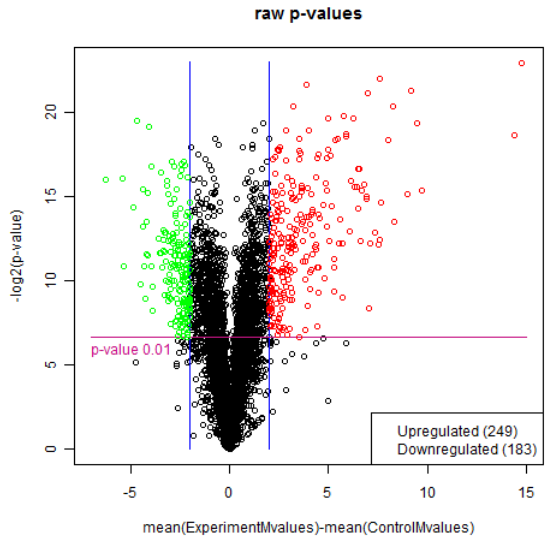



Figure 3.17 – Volcano plot: t-test using difference log fold-changes (M-values)

| Gene Number | Gene Name | $mean(M\text{-}values)_{Exp} - mean(M\text{-}values)_{Control}$ | t-statistic | Raw p-value |
|---|---|---|---|---|
| 1770 | YFL014W | 14.77 | 84.01 | 1.20E-07 |
| 3059 | YBR072W | 14.40 | 39.75 | 2.39E-06 |
| 349 | YLR178C | 9.71 | 22.49 | 2.31E-05 |
| 4074 | YMR169C | 9.46 | 45.16 | 1.44E-06 |
| 1461 | YPR160W | 9.15 | 62.97 | 3.81E-07 |
| 4426 | YGR248W | 8.98 | 21.51 | 2.76E-05 |

Table 3.6 – Top genes: t-test using difference log fold changes (M values)

The t-test was shown applied to the unratioed intensities to compare it with the fold-change method, as in that case it provides the same fold-change values, just adding a confidence value to each gene. However, with two-color microarray data, the post-normalization data analysis is commonly performed with M-values. It should be avoided to use the individual red intensity and green intensities for analysis purposes, as the two mRNA samples underwent competitive hybridization for the spots to reveal their final color. This implies considering each array as a whole (instead of considering each of the two RNA samples) so the user must be careful how he defines the two groups of arrays to run the t-test.

## The multiple comparison issue in hypothesis testing

When analyzing microarray data, all the genes are probed at once. In the Heat Shock experiment, this procedure must be applied for the 6,342 genes. This rises what is known in statistics as the multiple comparison issue. When declaring a gene regulated, by rejecting the null hypothesis with 95% of confidence, there is a 5% of committing an error. In 100 genes that are declared as regulated, 5 are false positives, declared as regulated when in fact they are not. The more genes that are tested for differential expression, the higher the probability of obtaining at least one false positive (Table 3.7). Of the probed genes, 5% are expected to be false positives, meaning, detected as regulated just by chance.

| Number of genes being probed<br><br>*(N)* | Significance of each hypothesis test<br><br>*(sig)* | Excepted number of false positives<br><br>*(N × sig)* | Probability of getting at least one false positive<br><br>*(1 – (1– sig)^N)* |
|---|---|---|---|
| 1 | 0.05 | 0 | 0.05 |
| 10 | | 1 | 0.40 |
| 100 | | 5 | 0.99 |
| 1000 | | 50 | 1.00 |

Table 3.7 – The multiple comparison problem: probability of getting at least one FP raises with the number of independent tests.

The problem has been addressed using different procedures. The simplest and also the most conservative approach is the Bonferroni correct method. For 100 genes and 100 hypothesis tests, to assure a family-wise error rate (FWER) of 0.05, only genes with a p-value inferior to 0.0005 are considered as regulated. Other multiple correction methods are Holm, Hochberg, Benjamini & Hochberg and Benjamini & Yuketieli. These five methods are implemented in Multtest and will be offered in Mind using this library. A multiple correction procedure can then be applied with the Multtest `mt.rawp2adjp` function:

```
adjpvalues<-mt.rawp2adjp(rawpvalues,"Bonferroni")
```

Because all these methods tend to highly lower the genes' p-values, the raw p-values will be presented to the user as well, regardless of the multiple correction method chosen, so that the user can have a view of both. Besides the p-values, the user should also consider the fold-change of the genes is order to consider them as regulated.

Other important multiple correction testes are permutation based methods, which assign the arrays at random to groups under study, hundreds of times, to conclude how likely their expression values can be due to chance. Although these methods are less conservative, they consume much more time and processing.

## Other variants of t-statistics for hypothesis testing

Additional variants of the t-statistics hypothesis testing include one-sample, paired and Welch. All of these choices are used in gene regulation assessment.

Performing t-test hypothesis testing using one group compares each gene expression value with a numerical value, for example, zero. In this case, the null hypothesis is that the gene's expression value is zero. The alterative hypothesis is that it is not. In the heat shock experiment, considering the group of the odd arrays (numbers 1, 3 and 5) and comparing each gene's M-value with zero brings out the genes that are differentially expressed between heat shock and control. Genes for which the null hypothesis can be rejected, and that have mean expression values further away from zero are regulated. The formula to compute the t-statistic for comparing a gene's mean expression value with a numerical value is

$$t = \frac{\overline{X} - \mu_1}{\frac{S_x}{\sqrt{n}}}$$

where $\overline{X}$ is the sample's mean, $S_x$ is the sample's standard error, $n$ is the sample size (number of arrays) and $\mu_1$ is the numerical value.

Paired t-statistics take advantage of the within-subject variability to obtain more reliable conclusions. The heat shock experiment, by applying a heat shock to three biological replicates of yeast cells, can and should be analyzed using paired t-statistics. Another example is when analyzing the effect of a medication on patients – each control sample matches an experiment sample. With pairing, the two related groups of gene expression values are reduced to one single sample of differences. The t-statistic for a sample of differences is given by

$$t = \frac{\overline{X_D} - \mu_0}{\frac{S_D}{\sqrt{n}}}$$

where $\overline{X_D}$ is the mean of the sample of differences, $S_D$ is its standard error, $n$ is the sample size (number of arrays) and $\mu_0$ is zero.

Performing a t-test on the heat shock experiment expression values, now considering the three pairings, provides the following results (Table 3.8). The gene t-statistic and corresponding p-values are different, but the top six genes remain the same (it still is the difference log fold-change).

| Gene Number | Gene Name | mean(M-values)$_{Exp}$ – mean(M-values)$_{Control}$ | t-statistic | Raw p-value |
|---|---|---|---|---|
| 1770 | YFL014W | 14.77 | 120.27 | 2.87E-08 |
| 3059 | YBR072W | 14.40 | 37.15 | 3.13E-06 |
| 349 | YLR178C | 9.71 | 17.91 | 5.71E-05 |
| 4074 | YMR169C | 9.46 | 34.29 | 4.32E-06 |
| 1461 | YPR160W | 9.15 | 56.93 | 5.70E-07 |
| 4426 | YGR248W | 8.98 | 60.50 | 4.47E-07 |

Table 3.8 – Paired T-Test

The Welch t-test is applied when equal variance cannot be assumed for the two samples. This variant also contemplates unequal sample sizes. The formulas for the Welch t-statistic and respective degrees of freedom are:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad d.f. = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{(n_1 - 1)} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{(n_2 - 1)}}$$

If gene YFL014W (M-values) is analyzed with the Welch t-test, considering the same division by control group (arrays 2, 4 and 6) and experimental group (arrays 1, 3 and 5), the t-statistic value is the same as if equal variances were assumed, but the number of degrees of freedom is no longer 4:

$$t = \frac{7.45545 - (-7.31333)}{\sqrt{\frac{0.08313^2 + 0.29291^2}{3}}} = 84.01 \qquad d.f. = \frac{\left(\frac{0.08313^2}{3} + \frac{0.29291^2}{3}\right)^2}{\frac{\left(\frac{0.08313^2}{3}\right)^2}{(3 - 1)} + \frac{\left(\frac{0.29291^2}{3}\right)^2}{(3 - 1)}} = 2.32$$

The p-value for the t-statistic of 84.01, under 2.32 degrees of freedom and performing a two-tail test, is $4.3 \times 10^{-5}$.

All these t-tests assume that the gene expression values, or the sample of differences, follow a normal distribution. If this assumption cannot be made, then non-parametric tests are used instead. No non-parametric tests were chosen to be implemented at this time in Mind, as the selected methods were found to be the most relevant for microarray data analysis.

### 3.3.5 ANOVA

While the t-test is performed to assess the differential expression between two experimental groups, ANOVA (Analysis Of Variance) is the most well known statistical technique for multiple group comparison. Just like the t-test allows concluding if two sample groups have the same mean (at a chosen significance level), ANOVA allows evaluating if three or more groups have equal means. When performing hypothesis testing using t-test, a t-statistic is calculated for each gene, leading to a p-value that allows selecting the gene as regulated or not depending on the significance level. Similarly, ANOVA calculates an F-statistic and a p-value for each gene to select it as regulated or not. The null hypothesis for both t-test and ANOVA is that the considered groups are different. Rejection of the null hypothesis means that at least one group is different. ANOVA and the F-statistic only allow concluding if there is a difference among the groups or not – it does not say which group is different, or what groups are different. In case of rejecting the null hypothesis, to know what groups are different from which ones, hypothesis testing using t-statistics must be performed between all the possible pairs of groups.

Analysis of the variance, the square of standard deviation, to assess differences between groups, was first used by Ronald A. Fisher in the early 20$^{th}$ century. As he worked at an agriculture experiment station, this researcher and statistician dedicated most of his work to addressed agriculture concerns. Fisher, using ANOVA and some innovative experimental design methodologies that he also introduced, like replication and randomization, investigated the effect of different fertilizers on crop growth. ANOVA is currently the most common statistical method for multiple group comparison, and it can be applied to microarray experiment data as long as the multiple comparison issue is taken into account.

One-way ANOVA allows comparing multiple groups at once. Two-way ANOVA permits comparing multiple groups and multiple factors. Different factors represent different independent variables, whereas different groups represent different levels of an independent variable. For each factor, the arrays are divided into groups. The groups are compared within the factor and then the factors are compared. Two-way ANOVA describes the interactions between the factors on gene expression. For example, in an experiment that compared a control with two differently medicated samples (three groups) at different time points, there would be two factors influencing gene expression: the medication and the time course. Such an experiment could be analyzed with two-way ANOVA, finding genes that are regulated due to the medication, genes that are regulated due to time and genes that are regulated due to the interaction of both factors. Currently only one-way ANOVA will be added to Mind.

The heat shock experiment is not a good example of an experiment that could be analyzed using ANOVA. There are two experimental groups, the control and the heat shock groups, that refer to the measurements taken before and after the heat shock. Nevertheless, one could perform ANOVA on the data using the following group definition: arrays 1 and 2 for group A, arrays 3 and 4 for group B and arrays 5 and 6 for group C. These three groups represent the three biological replicates that were used to extract the RNA samples and no genes are expected to be regulated between them. However, instead of using data of a different experiment, the ANOVA method will be explained using the heat shock experiment and the three groups as described, also with the purpose of seeing what kind of answers a statistical method can provide when the input does not make sense, at least from a point of gene regulation assessment. Algorithmically, this is still valid input. A user may actually define these three groups and request the list of regulated genes to be outputted with the ANOVA method.

The same gene as before, YFL014W, which was seen as highly differentiated between the control and heat shock groups, will now be analyzed considering the described groups A, B and C (Table 3.9). Apart from having three groups that are in fact replicates, a gene with extremely different values within the same group in not a common situation in gene regulation assessment, which is the case for the groups A, B and C formed. This is though still valid input for the statistical method.

| Array 1 | Array 2 | Array 3 | Array 4 | Array 5 | Array 6 |
|---|---|---|---|---|---|
| 7.518 | -7.275 | 7.136 | -7.409 | 7.712 | -7.256 |
| Group A | | Group B | | Group C | |

Table 3.9 – Gene YFL014W: division of the M values into groups A, B and C

The step-by-step procedure to calculate ANOVA analysis on this gene was based on Draghici's book ANOVA chapter [25]. The ANOVA test for this gene begins with the calculation of the group and overall means, standard deviations and variances (Table 3.10). The grand mean is the mean of the three group means or the six measurements. The group SD is the standard deviation of the six measurements. Each variance is the square of the corresponding standard deviation.

| | Group A | Group B | Group C |
|---|---|---|---|
| | 7.518 | 7.136 | 7.712 |
| | -7.275 | -7.409 | -7.256 |
| Group mean $\overline{X}$ | 0.1212 | -0.1361 | 0.2281 |
| Group standard deviation $SD$ | 10.46 | 10.29 | 10.58 |
| Group variance $SD^2$ | 109.41 | 105.78 | 112.03 |
| Grand mean = 0.07106, Grand standard deviation = 8.092, Grand variance = 65.472 | | | |

Table 3.10 – ANOVA: startup data for gene YFL014W

Let G be the number of different groups, M the number of measurements per groups and N the number of total measurements. N is not equal to G x M if there are different number of measurements per group. In this case, G = 3, M = 2 and N = 6.

ANOVA distinguishes within group variability from inter group variability. Measurements in each condition vary around the condition mean and the mean of each condition varies around the overall mean. As a consequence, each measurement varies around the overall mean too.

The sum of squares is a descriptor of the overall, the within group or the inter group variability. The overall sum of squares is given by the sum of the squares of all differences of the individual measurements to the global mean:

$$SS_{Total} = \sum_{i=1}^{N} \sum_{j=1}^{K_i} \left( X_{ij} - \overline{X}_{grand} \right)^2$$

$$SS_{Total} = \left(7.518 - 0.07106\right)^2 + \left(-7.275 - 0.07106\right)^2$$

$$+ \left(7.136 - 0.07106\right)^2 + \left(-7.712 - 0.07106\right)^2$$

$$+ \left(7.712 - 0.07106\right)^2 + \left(-7.256 - 0.07106\right)^2 = 327.36$$

The within group variability is also referred to as the error or residual variability, as a gene that has different expression values for replicates of the same group must be due to error, while the inter group variability is biological and explained by the differences between the conditions, so it is the condition or model variability. As the overall variability results of the within group and the inter group variabilities, the total or overall sum of squares is the sum of the within group and inter groups sum of squares or the condition and error sum of squares.

$$SS_{Total} = SS_{Cond} + SS_{Error}$$

The condition or model sum of squares is given by the sum of the differences of each group mean to the grand mean. Each individual measurement leads to one parcel in this sum:

$$SS_{Cond} = \sum_{i=1}^{G} \sum_{j=1}^{M_i} \left( \overline{X}_{group_i} - \overline{X}_{grand} \right)^2$$

$$SS_{Cond} = 2 \times \left(0.121 - 0.00712\right)^2 + 2 \times \left(-0.136 - 0.00712\right)^2 + 2 \times \left(0.228 - 0.00712\right)^2$$

The error or residual sum of squares is given by the sum of the differences of each individual measurement to its respective group's mean:

$$SS_{Error} = \sum_{i=1}^{N} \sum_{j=1}^{K_i} \left( X_{ij} - \overline{X}_{group_i} \right)^2$$

$$SS_{Error} = (7.518 - 0.1212)^2 + (-7.275 - 0.1212)^2$$

$$+ (7.136 + 0.1361)^2 + (-7.712 + 0.1361)^2$$

$$+ (7.712 - 0.2281)^2 + (-7.256 - 0.2281)^2 = 327.36$$

The overall number of degrees of freedom is given by the number of individual measurements minus one (N – 1), the number of degrees of freedom in a group is given by the number of measurements within that groups minus one (M – 1) and the number of degrees of freedom across the several groups is given by the number of groups minus one (G – 1). The overall number of degrees of freedom equals the sum of the condition and the error degrees of freedom.

With the sums of squares and degrees of freedom determined, an ANOVA table can be built. Each mean square (condition and error) is calculated by dividing the respective sum of squares by the corresponding number of degrees of freedom. The F-statistic or the F-ratio (capital letter after R. A. Fisher) is the ratio of the condition mean square divided by the error mean square (Table 3.11).

| | Sum of Squares | df | Mean Square | F-statistic | p-value |
|---|---|---|---|---|---|
| Between Groups (Condition, Model) | 0.140189 | 2 | 0.070095 | 0.0006426 | 0.9993577083 |
| Within Groups (Error, Residual) | 327.2206 | 3 | 109.0735 | | |
| Total | 327.3608 | 5 | | | |

Table 3.11– ANOVA table for gene YFL014W

The p-value is obtained from the F-distribution with degrees of freedom equal to the overall number of degrees of freedom. For the t-test, the p-value was obtained using the t-distribution at the appropriate number of degrees of freedom, and the p-value is obtained similarly in the ANOVA or F-test (both designations are used). However, the F-distribution has a different shape and it depends on two numbers of degrees of freedom. As it can be seen by Figure 3.18, the F-distribution approaches the normal shape with higher degrees of freedom relative to both numerator and denominator. The F-statistic, resulting of a ratio of sums of squares, is always positive, so contrary to the t-distribution, the F-distribution has only positive values.

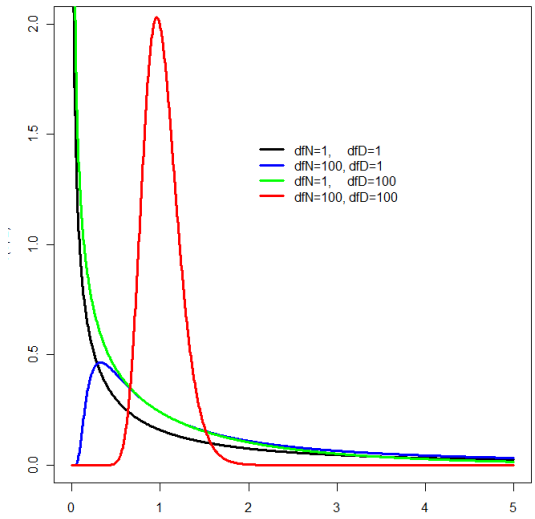



Figure 3.18 – The F-distribution

For the gene YFL014W, the low F-ratio and the high p-value strongly do not allow for the null hypothesis to be rejected. This is expected, as this gene, and all genes of the experiment, should behave similarly in the three groups. The F-statistic and the raw p-value of Table 3.9 were used as a reference to ensure that the same values were obtained for this gene using the R script written.

However, by running ANOVA on the heat shock data set and selecting all genes with p-value lower than 0.05, 60 genes are selected. With p-value lower than 0.01, 15 genes are selected, even though, like it was said, these three groups are biological replicates, and thus genes are not expected to behave differentially among them.

Type I error, inherent to hypothesis testing, implies some genes will appear as regulated when they are not – false postives. When working at a significance level of 0.05, each twenty regulated genes will contain a false positive. This is not enough however to explain how this method outputted 15 genes as regulated at a significance level of 0.01. These 15 genes have their 6 M-values close to zero, in the [-0.5; 0.5] interval. ANOVA does not analyze if the measurements are high or low, regardless of that, it just assesses if there is a difference between groups or not. In fact, the M-values close to zero biologically mean that this gene is expressed similarly in the control and the heat shock RNA samples, but those are not the groups under consideration. The statistical methods do not interpret the expression values, so the results outputted by these methods must be confirmed by the user. Noise and the partial inaccuracy that is characteristic of microarray experiments also account for these results. Analyzing the heat shock experiment data with the groups A, B and C served as proof that a statistical test is not

enough to declare a gene as regulated or not, the interpretation of the user is always of the most fundamental importance.

Available R libraries to incorporate ANOVA analysis of microarray data into Mind include Maanova (Microarray Analysis Of Variance) [38] and the Multtest package introduced previously. Both packages seem appropriate for the functionality that is to be added. They permit division of the arrays into several groups, calculation of the F-ratios and the p-values and multiple comparison correction procedures. Maanova, using a matrix of expression values and the information about the groups, obtains the desired results with the following lines of code:

```
design <- data.frame(Array=1:nrArrays,GroupName=gMulti) # gMulti = group info
A1 <- read.madata(datafile=matrixMValues, designfile=design,
arrayType="oneColor")
B1 <- fitmaanova(A1, formula = ~GroupName)
C1 <- matest(A1, B1, term="GroupName", n.perm=2)
C2 <- adjPval(C1, method="adaptive")
idx.fix2 <- volcano(C2,threshold=c(minPValue),method="unadj")
```

The `arrayType` parameter set as `"oneColor"` is so that the function does not attempt to obtain log ratios from the expression values, meaning, to make it use the expression matrix as is. All data processed by Mind is relative to two-color data. The results obtained were checked to be consistent with the expected values, for example, the values obtained using the R code above for the YFL014W gene match the values obtained by the step-by-step calculation that was shown in this section. The return of the `matest` function includes the F-ratios, the raw p-values and the permutation corrected p-values. Permutation correction requires much more processing time, proportional to the number of permutations done. At the moment it was chosen to offer only non-permutation correction for both t-test and ANOVA, which is much quicker. In the `matest` function, the number of permutations parameter is set to 2 to override the default number of 1000 permutations. The library offers the Benjamini and Hochberg ("adaptive"), Hochberg and Benjamini ("stepup") and Westfall and Young ("stepdown") correction methods.

This library also includes a volcano plot function that returns the number of regulated genes that are regulated in the plot. The resultant graph is a plot of the minus base ten p-value logarithms versus the log fold-changes, with a horizontal line above which the low p-value genes are selected as regulated, and where log fold-change is the root mean square of the relative expression values [39]. Along with the Maanova volcano plot, it was also chosen to create a new graphic by plotting the minus base ten p-value logarithms against the highest difference fold-change of all possible pair comparisons. With three groups, there are three different possible pair comparisons, the absolute difference fold-change between each pair of groups is

calculated (absolute value of mean M-values Group i – mean M-values Group j), and the highest for each gene is selected for the volcano plot. For the heat shock data set, the volcano plots (Maanova plot and new plot) for the 3 groups mentioned and a minimum p-value of 0.01, with the 15 genes highlighted, are shown in the following Figure 3.19.

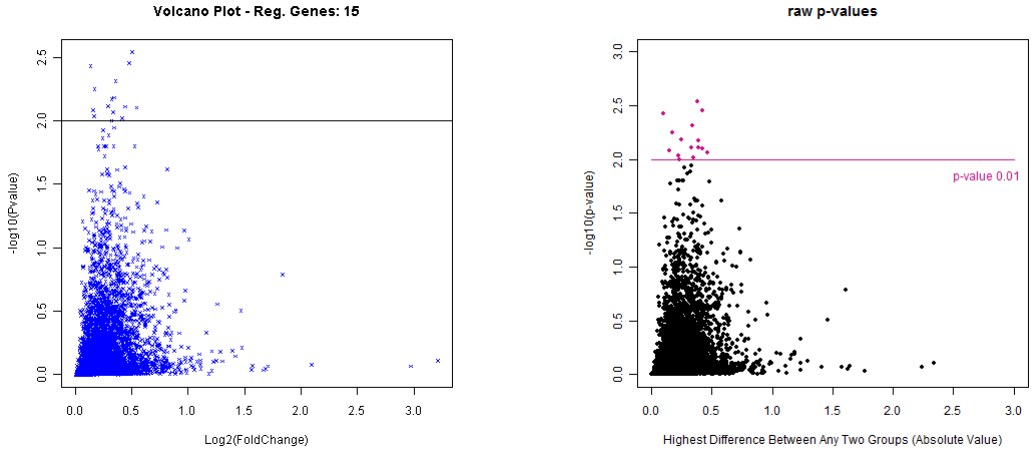



Figure 3.19 – Volcano plot: ANOVA

The Multtest function `mt.teststat` also calculates the F-ratios, by setting the test type to F. Then, the p-values can be taken from the F-distribution using the base R probability distribution function for the F-distribution (pf). The raw p-values can be corrected then using the function `mt.rawp2adjp`, just like it was done for the t-test.

```
fstat <- mt.teststat(expressionValuesMatrix,divisonByGroups,test="f")
rawp <- 2 * pf(abs(fstat),lower.tail=F,df1=2,df2=3))
adjp <- mt.rawp2adjp(rawp, proc="Bonferroni") # "Holm", "Hochberg", "BH", "BY"
```

The volcano plot would then be created using the raw or corrected p-values and the highest difference log fold-change among groups. As it could be seen, both Maanova and Multtest allow to calculate F-ratios, raw p-values and corrected p-values to include ANOVA in Mind, although each library offers some other specific features. As both packages provide the intended results, Maanova was chosen, as it may be helpful for addition of two-way ANOVA to Mind in future developments.

### 3.3.6 Limma

Limma is an R Bioconductor library, with many functions for the analysis of microarray data, for both preprocessing and gene regulation assessment [40, 41]. Due to its innovative approach for gene regulation analysis, it is also commonly referred to as a methodology.

Limma implements various functions to perform quality control of two-color array data, some of which are offered in Mind. This library is most known for the approach it proposes for differential expression assessment, implemented for both one-color and two-color array data. In fact, the word Limma stands for Linear Models For Microarray Data. The two key points of the Limma approach are the usage of linear models to describe the experimental design and the application of an empirical Bayes to moderate the standard errors of the log fold-changes [42]. Moderated t-statistics lead to raw p-values in the same way that ordinary t-statistics do, except degrees of freedom are increased, providing more reliable results, due to the shrunk standard errors. The result is a robust method that, even considering the statistical challenges of a microarray experiment (large number of genes, small number of replicates and considerable amount of noise), can provide a list of regulated genes with reliable p-values.

The linear models approach involves specifying one or two matrices for the experiment, the design matrix which is obligatory, and the contrast matrix which is only required in more complex experiments. Having specified the design matrix, one uses Limma's function `lmFit` to fit that linear model to the matrix of M-values of the experiment. With that, the systematic part of the data is modeled, so log fold-changes can be calculated and eBayes is applied to correct the standard errors of the log fold-changes and associated p-values, in order to finally obtain the regulated gene list.

The design matrix has always one row for each array. The number of columns is the same as the number of different RNA samples in one-color arrays or two-color arrays with a common reference, and it is equal to the number of RNA sample minus one for two-color arrays without a common reference (direct-design two-color arrays). The implementation in Mind will only focus on the latter, as it will not pose a problem for two-color arrays with a common reference. The columns represent the coefficients or comparisons between RNA samples. In the heat shock example there is only one comparison so the design matrix will have one column (Table 3.12).

| Array Nr | RNA Samples | | "Control – Heat Shock" |
|---|---|---|---|
| | Cy3 (Green) | Cy5 (Red) | |
| Array 1 | Control | Heat Shock | 1 |
| Array 2 | Heat Shock | Control | -1 |
| Array 3 | Control | Heat Shock | 1 |
| Array 4 | Heat Shock | Control | -1 |
| Array 5 | Control | Heat Shock | 1 |
| Array 6 | Heat Shock | Control | -1 |

Table 3.12 – Design Matrix: Heat Shock Experiment

A simple dye-swap experiment with two groups like heat shock has a design matrix of only ones. The coefficient being "Control – Heat Shock" or "Heat Shock – Control" affects the signal of the values. "Control – Heat Shock" was chosen, so genes that are upregulated in the Heat Shock sample will have positive log fold-changes. Arrays with the heat shock on the red channel will have a one on the design matrix and their dye swaps will have a minus one.

There are different ways of building a design matrix, all correct and suitable for Limma differential expression analysis. The different possibilities, however, are not easily perceptible considering a simple experiment like the heat shock one. The implementation chosen for Mind is the one that allows the analysis to be carried out with higher odds of not needing the contrast matrix. Each column of the design matrix (coefficient) represents one or comparison between two RNA samples. In the heat shock experiment, the only possible comparison is "control – heat shock". In an experiment with three different RNA sample, say A, B and C, three different pair comparisons are possible, A-B, B-C and C-A, but only two will be represented in the design matrix. The simple design matrix construction method would always compare with, for example, the first sample: B-A and C-A. If the user wishes to compare C-B, he would need to build a contrast matrix to compare B-A with C-A, so a contrast matrix with a coefficient [C-A]-[B-A] = C-B. This is to explain that, if the user can specify from the start he would like to compare B with C, this comparison is specified by a coefficient in the design matrix and he does not need the contrast matrix.

For the purpose of explaining the design matrix calculation approach implement in Mind, a different experiment will be exposed now, the ApoAI experiment, whose data is used in the Limma User's Guide and in many Limma tutorials [43]. This experiment, carried out by Callow et al. in 2000, compares tissue from mice that had their ApoAI (apolipoprotein AI) gene knocked out, with mice that had not [44]. This is a gene that is involved in lipid metabolism. The objective is to see what other genes are affected by the inhibition of this gene and that therefore may work together with ApoAI. Those are the differentially expressed genes between Control and ApoAI. As these two RNA samples were always hybridized in the red channel, the green channel had a common reference RNA.

The user specifies that he would like to compare "Control – Reference" and "ApoAI – Control". For the first eight arrays, the design matrix values are straightforward as the arrays have Control on the green channel and Reference on the red channel. For the last eight,

$$ApoAi - Reference = 1 \times (Control - Reference) + 1 \times (ApoAi - Control)$$

which explains the last sixteen ones in the design matrix (Table 3.13). By knowing the RNA samples hybridized on each array and channel and the comparisons the user wishes, the values of the design matrix are found. As it can be seen, this eliminates the need of a contrast matrix, which is not needed in the large majority of cases. By not offering the contrast matrix in Mind, if the user wishes to make all possible comparisons ("Control – ApoAI", "Control – Ref" and "ApoAI – Ref"), he will need to run a second Limma analysis, but often it is not necessary.

| Hybridizations | Array | Control - Ref | ApoAi - Control |
|---|---|---|---|
| Cy3: Reference<br>Cy5: Control | Array 1 | 1 | 0 |
| | Array 2 | 1 | 0 |
| | Array 3 | 1 | 0 |
| | Array 4 | 1 | 0 |
| | Array 5 | 1 | 0 |
| | Array 6 | 1 | 0 |
| | Array 7 | 1 | 0 |
| | Array 8 | 1 | 0 |
| Cy3: Reference<br>Cy5: ApoAI Knockout | Array 9 | 1 | 1 |
| | Array 10 | 1 | 1 |
| | Array 11 | 1 | 1 |
| | Array 12 | 1 | 1 |
| | Array 13 | 1 | 1 |
| | Array 14 | 1 | 1 |
| | Array 15 | 1 | 1 |
| | Array 16 | 1 | 1 |

Table 3.13 – Design Matrix: ApoAI Experiment

After fitting the linear model (function `lmFit`) and applying eBayes to the fit (function `eBayes`) the top regulated genes can be requested (function `topTable`). When there are more than two RNA samples in the experiment, Limma provides a top table ranked by F-statistics. By asking for the top table based on one of the coefficients of the design matrix, Limma provides the top table ranked by t-statistics for that comparison. The number of genes to figure in the top table must also be specified by the user. The topTable function also provides adjusted p-values to correct for multiple testing (the p-values provided by eBayes are raw, although resulting of moderated t-statistics), using by default Benjamini and Hochberg's method. All the options, in order of increasing conservatism, are: none, Benjamini and Hochberg, Benjamini and Yekutieli and Holm's step-down Bonferroni. Mind will use the default one.

The top table with F-statistics includes one column for the log fold-change of each comparison specified by the design matrix coefficients, the F-statistic, the p-value and the adjusted p-value (Table 3.14).

| Name | Coef1 | Coef2 | AveExpr | F | P.Value | adj.P.Val |
|---|---|---|---|---|---|---|
| ApoAI,lipid-Img | 0.09847 | -3.166 | 12.5 | 540 | 1.02e-16 | 6.50e-13 |
| EST,WeaklysimilartoC | -0.27457 | -1.027 | 12.6 | 232 | 1.62e-13 | 5.18e-10 |
| CATECHOLO-METHYLTRAN | 0.00217 | -1.848 | 12.9 | 154 | 5.22e-12 | 1.01e-08 |
| EST,HighlysimilartoA | 0.16388 | -3.049 | 12.3 | 151 | 6.33e-12 | 1.01e-08 |
| ApoCIII,lipid-Img | -0.14592 | -0.933 | 13.7 | 132 | 1.99e-11 | 2.54e-08 |
| similartoyeaststerol | -0.42894 | -0.955 | 13.3 | 127 | 2.62e-11 | 2.79e-08 |

Table 3.14 – ApoAI Experiment: TopTable of Fit, 6 genes

For a specific comparison or coefficient, a top table with t-statistics is shown, including t-statistics for the comparison considered, p-values, adjusted p-values and B-statistics (Table 3.15). It is ranked by the B-statistic. The B-statistics are the log-odds that a gene is differentially expressed. With B = 1.5, the log-odds that it is differentially expressed is exp(1.5) = 4.48, meaning, about 4.5 to 1. The probability that the gene is differentially expressed is 4.48/(1+4.48) = 0.82 (82%). A B-statistic of zero corresponds to a probability of 50% of the gene being differentially expressed. A negative B-statistic means that there is a very low probability of the gene being differentially expressed.

| Name | logFC | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| ApoAI,lipid-Img | -3.166 | 12.5 | -23.98 | 4.77e-15 | 3.05e-11 | 14.93 |
| cDNA EST,HighlysimilartoA | -3.049 | 12.3 | -12.96 | 1.57e-10 | 5.02e-07 | 10.81 |
| CATECHOLO-METHYLTRAN | -1.848 | 12.9 | -12.44 | 3.06e-10 | 6.51e-07 | 10.45 |
| EST,WeaklysimilartoC | -1.027 | 12.6 | -11.76 | 7.58e-10 | 1.21e-06 | 9.93 |
| ApoCIII,lipid-Img | -0.933 | 13.7 | -9.84 | 1.22e-08 | 1.56e-05 | 8.19 |
| ESTs,Highlysimilarto | -1.010 | 13.6 | -9.02 | 4.53e-08 | 4.22e-05 | 7.30 |

Table 3.15 – ApoAI Experiment: TopTable of Fit, 6 genes, coefficient 2

Back to the heat shock example, the top ten genes are represented in Table 3.16. Only three of the top six genes previously determined with the fold-change and the t-statistics methods are in the Limma top ten.

| Gene Nr | Gene Name | logFC | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|---|
| 1770 | YFL014W | 7.38 | 11.61 | 88.63 | 7.17E-14 | 4.55E-10 | 20.80 |
| 21 | YDL204W | 3.81 | 9.29 | 58.37 | 2.46E-12 | 6.86E-09 | 18.56 |
| 3827 | YOR374W | 3.48 | 10.60 | 53.52 | 5.11E-12 | 6.86E-09 | 17.99 |
| 2609 | YML100W | 4.13 | 9.92 | 53.33 | 5.26E-12 | 6.86E-09 | 17.97 |
| 3059 | YBR072W | 7.20 | 10.62 | 53.03 | 5.52E-12 | 6.86E-09 | 17.93 |
| 4074 | YMR169C | 4.73 | 9.53 | 52.02 | 6.49E-12 | 6.86E-09 | 17.80 |
| 3749 | YFR053C | 4.00 | 9.70 | 44.42 | 2.47E-11 | 2.23E-08 | 16.69 |
| 557 | YDR214W | 3.12 | 11.76 | 41.74 | 4.16E-11 | 3.30E-08 | 16.23 |
| 4463 | YLR216C | 2.94 | 12.29 | 40.52 | 5.35E-11 | 3.77E-08 | 16.00 |
| 3062 | YBR054W | 2.93 | 9.17 | 37.55 | 1.02E-10 | 5.87E-08 | 15.42 |

Table 3.16 – Limma: HS experiment top table (10 genes)

Table 3.17 shows how the ranking of those six genes changes with Limma. Limma still calculates the same ratio log fold-changes, but ranks the genes by B-statistics, a measure that takes into account other factors such as variability.

| | Ranking in Fold-change | Ratio Log Fold-change (Fold-change and Limma) | Ranking in Limma (ordered by B-statistic) | T-statistic (Limma) | B-statistic (Limma) |
|---|---|---|---|---|---|
| YFL014W | 1 | 7.38 | 1 | 88.63 | 20.80 |
| YBR072W | 2 | 7.20 | 5 | 53.03 | 17.93 |
| YLR178C | 3 | 4.86 | 18 | 30.86 | 13.85 |
| YMR169C | 4 | 4.73 | 6 | 52.02 | 17.80 |
| YPR160W | 5 | 4.57 | 13 | 36.24 | 15.14 |
| YGR248W | 6 | 4.49 | 90 | 18.87 | 9.63 |

Table 3.17 – Ranking of genes by fold change method and by Limma

Typically a volcano plot is also created, where the log odds of differential expression (B-statistics) are plotted against the log fold-change values. Figure 3.20 shows the 10 genes with top B-statistics, presented in Table 3.13.

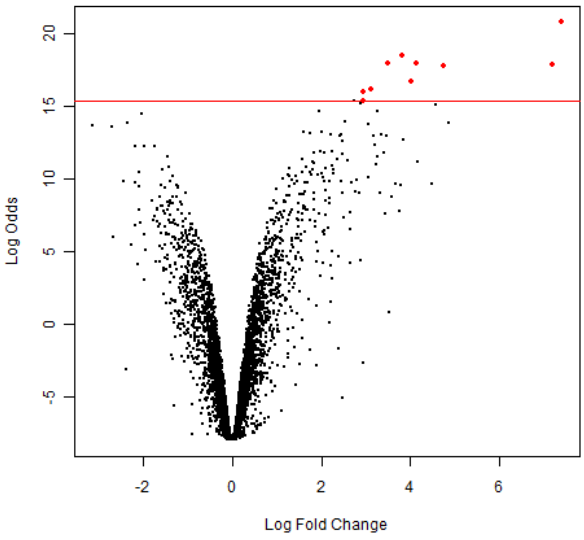



Figure 3.20 – Volcano plot: limma

### 3.3.7 SAM

SAM, Significance Analysis of Microarrays, is a technique proposed in 2001 by Tusher, Tibshirani and Chu [45]. This method considers the multiple comparison issue and assigns a score to each gene fold-change based on the standard deviation of repeated measures. For gene with high scores, the false discovery rate (FDR) is calculated using permutations of the measurements, meaning, what percentage of genes was declared as regulated by chance,

providing a permutation based correction method. SAM has been found to identify regulated genes in a experiment with lower FDR as opposed to using other traditional statistical methods.

SAM is available from the Stanford University website, as an Excel plug-in. Using the SAM Excel plug-in requires, besides Microsoft Excel, the installation of R and the R Samr package. This library is downloadable from the website or it can be installed directly from R. In order to offer SAM in Mind, the Samr package will be used [46].

To run SAM, considering two groups or classes of data, SAM can be run with the two-class or two-class paired options. It is required to define the two groups of data and to specify a delta. The delta represents how much false positives the user is willing to accept; the higher the delta, the less genes will be selected as regulated. Another option is SAM multi-class, for more than one group.

Performing a SAM two-class analysis on the Heat Shock experiment data (M-values), considering the even arrays as the control group and the odd arrays as the experiment group, choosing a delta of 25.0 provides the following results.

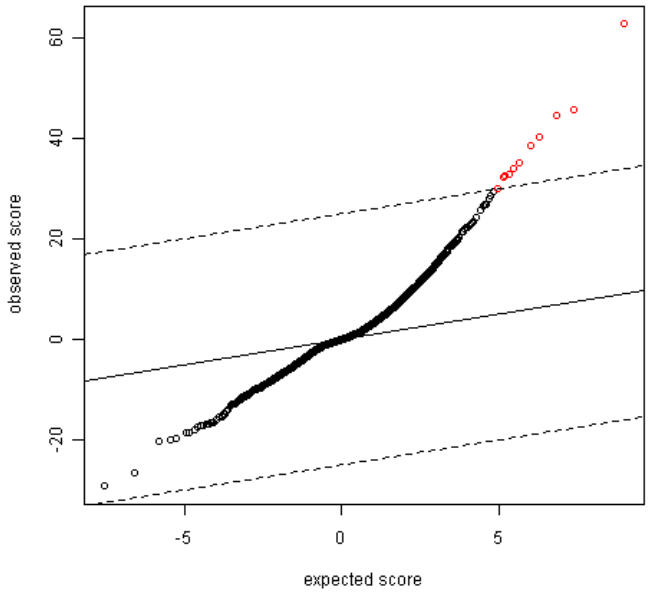



Figure 3.21 SAM Plot for the Heat Shock experiment, with delta 25.0

| GeneNr | Name | Score(d) | Numerator(r) | Denominator (s+s0) | Log Fold-change | q-value(%) |
|---|---|---|---|---|---|---|
| 1770 | YFL014W | 62.79 | 14.77 | 0.24 | 14.77 | 0 |
| 21 | YDL204W | 45.85 | 7.62 | 0.17 | 7.62 | 0 |
| 1461 | YPR160W | 44.70 | 9.15 | 0.20 | 9.15 | 0 |
| 3827 | YOR374W | 40.29 | 6.97 | 0.17 | 6.97 | 0 |
| 2609 | YML100W | 38.57 | 8.27 | 0.21 | 8.27 | 0 |
| 4074 | YMR169C | 35.19 | 9.47 | 0.27 | 14.77 | 0 |
| 3059 | YBR072W | 34.15 | 14.40 | 0.42 | 7.62 | 0 |
| 1873 | YBR149W | 32.94 | 3.88 | 0.12 | 9.15 | 0 |
| 557 | YDR214W | 32.54 | 6.23 | 0.19 | 6.97 | 0 |
| 606 | YKL096W | 32.22 | 5.74 | 0.18 | 8.27 | 0 |
| 3019 | YMR196W | 30.13 | 4.95 | 0.16 | 14.77 | 0 |

Table 3.18 – HS Experiment: SAM regulated genes (delta 25.0)

Only three of the eleven regulated genes were selected with t-statistics and fold-change, but eight of these were already selected with Limma. The score (d) is the t-statistic value. It is an adjusted t-statistic value. The numerator (r) and the denominator (s + s0) refer to that t-statistic value. This is a moderated t-statistic, as s0 was added to the denominator to compensate the high variability in low intensity genes. The q-value has a meaning similar to the p-value, but adapted to multiple comparison testing: it represents the lowest FDR at which the gene is called significant.

SAM has been a method very commonly used for microarray data analysis, so it was found a valuable addition to the Mind microarray data analysis module. It will be included in the variants: two-class, two-class paired and multi-class.

## 3.4 Summary

This section provided an overview of microarray data analysis methods for gene regulation assessment, using two-color array data. First the heat shock experiment was introduced, as its data is used for the analysis results shown. Then the most commonly used graphical plots were explained and the M-values were distinguished from the individual intensities. Next, commonly used quality control procedures were introduced, mostly offered by the R Bioconductor Limma library. Finally gene regulation assessment techniques were exposed: spot replicate averaging, filtering and common statistical methods used (fold-change, t-test, ANOVA, Limma and SAM). For all of the statistical tests, several R libraries (Multtest, Maanova, Limma and Samr) are used, except for the fold-change method. The multiple comparison issue in microarray experiments has also been presented along with the need to include correction procedures in a data analysis application.

# Chapter 4

# Data Analysis Module for Mind

At the beginning of this project, Mind already contained the data analysis module, with the quality control functions (filtering, background correction and normalization). The user could normalize the data of his experiment stored in the Mind LIMS and obtain the quality control reports and the normalized data files. This project aims to present a new data analysis module by revising the existing quality control functions and by adding gene regulation assessment functions.

To establish the requirements for this project, Mind's initial state of development was considered as well as feedback from the Mind users. The overview of microarray data analysis presented in Chapter 3 also provided important insight to decide upon Mind's features. This chapter will begin by presenting a study of existing applications, which is also important to understand what is valued in data analysis software. It will then aim to explain the development work that was done, by describing the new workflows, application architecture, technologies used and data-base modifications. The final application is then presented.

## 4.1 Existing Data Analysis Software Solutions

This project aims to add a new gene regulation assessment module to Mind and to revise the existing quality control module. To define the requirements, including selecting the most common statistical methods for gene differential expression assessment presented in the previous chapter, it is important to understand what features are valued in microarray data analysis software, from the user's point of view. This section will present a general overview of important microarray data analysis software, will observe in more detail some popular applications and will point some important aspects of microarray data analysis web applications.

In the original Mind web application, no issues were declared regarding the quality control functionalities themselves: the background correction, normalization and plots, all offered through Limma, are just right for the user conducting the experiment. The alterations done in the quality control section of Mind relate, not to the R scripts and the normalized data files and graphical plots generated, but to the GUI (graphical user interface), processing of the R scripts and information present in the reports.

### 4.1.1 Overview of Microarray Data Analysis Solutions

The current users of Mind provided valuable information about what is important in a microarray data analysis application, specifically for this new module being developed. Several institutions make available on-line the software they use for microarray data analysis [47-49]. The Department of Microbiology of the Montana State University offers a commented list aimed to help others users that are looking for microarray data analysis tools [48]. After knowing the user's opinions and taking a closer look to some applications, a better perception of what is required for the module being developed can be gained. Table 4.1 presents some popular microarray data analysis programs, summarizing their features.

Flex Array is an application for the statistical analysis of microarray data. It is maintained by Genome Quebec Innovation Centre. It can be downloaded from their website, at http://genomequebec.mcgill.ca/FlexArray/, and it is free for academic and governmental use. It contains functionalities for normalization of one-color microarray data, and it includes several statistical tests for gene regulation assessment: t-test, LPE (Local Pooled Error), SAM, Empirical Bayes, Bayes T, cyber-T, ANOVA, fold-change. It also offers methods to control the multiple comparison issue: Benjamini Hochberg, Benjamini Yekutieli, Holm, Hochberg, Sidak Single Step, Sidak Step Down, and Bonferroni. It generates a large variety of plots which include Venn diagrams to compare gene lists, volcano plots, histograms, scatter plots, QQ plots, MA plots, box plots and others, which depend on the phase of the data analysis and the methods chosen. FlexArray has also a very appealing GUI and superb video tutorials that allow the user to start performing his analyses right away. It allows also importing and exporting data in tab delimited text files, readable by other microarray data analysis or spreadsheet software. It uses R Bioconductor libraries, making their functionalities available to the user in a friendlier and graphical environment.

TM4's MIDAS and MeV together cover microarray data analysis – MIDAS allows performing quality control, MeV allows gene regulation assessment analysis and exploratory analysis. MIDAS contains several filters and functions for data normalization (www.tm4.org). MeV, for statistical tests, provides t-tests (one class, between subjects, paired), SAM (two-class unpaired,

two-class paired, multi-class, censored survival, one-class) and ANOVA. It also includes non-parametric tests: Wilcoxon Mann-Whitney Test, Kruskal-wallis Test, Mack-Skillings Tests, Fisher Exact Test. Besides these, MeV includes exploratory analysis functions (gene clustering and more).

| Features | FlexArray | MeV and MIDAS (TM4) | BASE | Limma | LimmaGUI | WebArray and WebArrayDB |
|---|---|---|---|---|---|---|
| Platform and Software Requirements | MS Windows, R and R (D)COM Server (automatic installation) | Multi-platform (Java) | None (web application) | R | R, Limma | None (web application) |
| Storage | No (desktop application) | No (desktop application) | Yes | No (desktop application) | No (desktop application) | Yes, WebArrayDB |
| MIAME compliance | | | Yes | | | Yes, WebArrayDB |
| One-color / Two-color arrays | One-color only | | Two-color only | Quality Control: Two-color, Regulation assessment: both | Two-color | WebArray both, WebArrayDB any number of colors |
| Quality Control | Many functionalities | Many functionalities (MIDAS) | Basic functions | Many functionalities | Many functionalities | Many functionalities |
| Statistical Analysis | Many functionalities | Many functionalities (Mev) | Basic functions | Limma | Limma | Good selection of functionalities: Limma and more |
| Clustering tools | No | Yes (MeV) | No | No | No | No |
| Documentation tutorials | Yes, video tutorials | Yes, quickstart guide and manual | | Yes, user's guide and HTML help | Yes, detailed HTML help | |
| Data Data Import/Export, Sharing | Yes | Yes | Yes | | Yes | Yes |
| Notes: | "GUI to bioconductor for one-color microarray data" Very intuitive GUI | Very intuitive GUI | Unproven scalability of MySQL | Command line interface | | |

Table 4.1 – Features of Several Microarray Data Management and Analysis Programs

BASE, BioArray Software Environment, is a web-based database solution for storage and management of microarray data that also offers some data analysis functionalities (http://base.thep.lu.se). It is a very broad microarray data storage facility, and it offers some features for microarray data analysis. It does not have many data analysis features, and it is curious to know how such an application chose a few functionalities to offer – functionalities that are commonly used and that are satisfactory in a large number of experiments. BASE offers simple filtering, normalization, averaging and statistics, for two-sample comparison (the most common microarray experiment): filter out bad spots, adjust low intensities, and normalize

correcting for non-linearities and dye inconsistencies. BASE is a good application to look at because it shows that an application can offer a smaller amount of data analysis functionalities, as long as those functions are chosen carefully. BASE brilliantly excels for its LIMS, and is a reference platform for microarray data storage and management.

Limma itself is a fully functional software package for microarray data analysis. The user who wishes to perform microarray data analysis may install R and Limma, and run the analysis using the command line. Limma offers a broad range of functions: loading the raw data files, GAL file, targets information and spot types information files, spot quality weighting functions, background correction methods, within array normalization, between array normalization and gene regulation assessment by performing the fit of a linear model and applying empirical Bayes statistics. It generates various types of plots. As these features are executed using a command line, other applications such as limmaGUI, WebArray and Mind sought to provide a graphical interface to the user for the execution of these methods.

limmaGUI is, exactly how it is named, is a GUI to the Limma package. It offers the quality control and differential expression assessment functions for two-color array data. It runs over R, so this framework needs to be installed. limmaGUI itself is an R Bioconductor package, and when loaded in R, opens up a windows where the Limma functions can be accessed in a graphical fashion: loading the raw data files and the GAL file, setting the spot types information and the targets information, background correction, normalization, computing and fitting a linear module and selecting the top regulated genes. Various plots can also be requested by the user. The results, top regulated gene list and all the expression values (M-values) can also be exported to a text-file and an HTML report can be generated.

WebArray and WebArrayDB are web applications. Although distinct, they are part of the same project, hosted online at www.webarraydb.org. WebArray contains microarray data normalization and different expression assessment with Limma. WebArrayDB is currently under beta testing and adds data storage facilities with MIAME compliance, parallel computing and more statistical analysis methods for gene differential expression assessment (SAM, ANOVA and non-parametric analysis). Another interesting feature is that, while WebArray processes one- and two-color data, WebArrayDB supports data with any number of colors. Being WebArray and WebArrayDB web applications they constitute cross-platform applications. WebArray is also a graphical user interface for Limma, in the sense it offers background subtractions functionalities (subtract, half, minimum, moving min, edwards, normexp and rma), normalization (print-tip loess, global loess, median, robust spline) and differential expression assessment through the Limma library. WebArrayDB includes more data analysis functions that are not available in the Limma package, such as SAM and ANOVA.

The functionalities of these applications are summarized in Table 4.1. It is found that, if all the important microarray data software features are to be ordered, user-friendly GUIs and good documentation and tutorials come first, as they are vital for the software to be satisfactory to the user. After these are met, diverse and good functionality offered for quality control and analysis, good data import, export and storage and report generation are valued.

All the six applications presented are free for academic use and some are open source. There is plenty of commercial software available that requires a license to enable full use. For example, J-Express, owned by MolMine, is a complete microarray data analysis package. Although the full annual license costs $800, at the time of writing, a laboratory can purchase licenses only for the needed modules, for example, fold-change analysis ($80) and SAM ($120). The decision of purchasing data analysis solution or using a free one is ultimately up to the user. It was found that the free solutions analyzed (for academic use) and presented in Table 4.1 are all of excellent quality. Although different, and none of the applications offer all the listed features, each one corresponds nicely to what is its goal. J-Express is also a great application, with much functionality, an attractive GUI and good documentation. The choice of using a commercial or a free application is up to the user, according to what best to fit his needs.

### 4.1.2 Valued Features in Microarray Data Analysis Software

The previous section exposed briefly several different microarray data analysis applications, with the goal of understanding what may be important to include in Mind. It can be seen that while having a lot of functionalities is good, it is not essential for the microarray application to be a valuable tool to the biologist or the geneticist. In fact, none of the previous applications have all of the features listed in Table 4.1; each application lacks at least one of: storage, one- or two-color data support, quality control, statistical analysis or clustering functionalities.

Intuitive GUIs and good documentation are very important, as they are essential for the user to start using the software quickly and comfortably. An application can focus on data storage, and in this case must offer MIAME compliance or it can focus on data analysis, offering a good selecting of data processing functionalities. In both cases data import and export facilities are valued, as the user may want to save data produced with one application and open it with a different one. An application's data analysis functions may include gene regulation assessment, gene clustering, or both. The user must also see whether the software support one- or two-color data, depending on the microarray technology he uses in the laboratory. Generation of reports and plots is also important.

In Mind, data storage features, MIAME compliance and data import and export facilities have already been effectively addressed by the Mind LIMS. The Mind data analysis module already includes, for quality control, filtering, background correction, normalization and generation of plots using the Limma library. Quality control HTML reports are also generated, which include the plots generated for each array. The goal of this project is to add gene regulation functions and to revise the existing quality control (GUI, processing architecture…), as adequate.

The possible functions for quality control and for differential expression assessment are very vast. In fact, not only so many methods are available, but new methods are created as the microarray technology evolves. However, while offering a very broad range of analysis functions makes an application attractive, an application can still excel offering fewer features (the most commonly used ones), if it provides good user interfaces and documentation. After all, a small number of methods are used in the majority of data analysis.

### 4.1.3 Considerations for the Mind Data Analysis Module

Mind is a web application. As in any field (other than microarrays), desktop and web applications have different advantages and disadvantages. Web applications require no installation and consume minimal hardware and software resources, as the processing is done on the server. Web applications with storage capabilities such as Mind also handle storage on a server data-base. Desktop applications, although faster, depend on the client's resources for both processing and data storage. Creating a web application has specific requirements, as opposed to its desktop equivalent. The user interfaces, the way the input parameters are asked, the output generated, must be thought differently. From the applications described in Table 4.1, BASE, WebArray and WebArrayDB are web applications. BASE focused more on data storage and management, WebArray focuses more on data processing and WebArrayDB provides both types of capabilities. Mind provides data storage and data analysis means too.

With the various software solutions analyzed, and many more great solutions available, the reason for this project may be questioned. Mind was originally developed to address data storage and provide a LIMS with MIAME compliance that is transparent to the user. The MIAME standard object model is not simple and providing the biologist an environment to allow him to organize his experiment data with MIAME compliance, in a way that is transparent to him, is a challenging problem. Mind effectively addresses this problem with a LIMS into which the user can upload his data files. Allowing data processing functionalities directly from Mind, after the user stored the experiment files in the Mind LIMS, is indeed a practical and useful feature, which is why the Data Analysis module has already been created with the quality control functionalities. Permitting gene expression assessment is the next step –

allowing the user to derive conclusion from his data directly from Mind. With GeneBrowser, the gene functional analysis web application, also developed at University of Aveiro, the gene regulation assessment module seems like the bridge that is called for between Mind and GeneBrowser.

While the Mind data analysis module cannot include all possible features, the overview of existing software done made it clear that as long a subset of features is well chosen and coupled with clear GUIs, good documentation and report generation, the module may still be very useful. Data import facilities do not apply for the Mind data analysis module, as it gets the input files from the Mind LIMS, however data export must be considered to generate tab delimited text files of data the user may like to view in other software. Of course, each gene regulation assessment analysis done must be able to output the regulated gene list in a way it can be loaded into GeneBrowser.

## 4.2 New Mind Data Analysis Workflow

At the beginning of this project, the Mind microarray data analysis module was composed only of the quality control functions. The following diagram (Figure 4.1) summarizes the steps a user would take to perform quality control in Mind. The user would select a subset of measurements of one of his experiments stored in the Mind LIMS to perform quality control. He would then define the quality control parameters in three distinct steps: first the spot types information, then the filtering, background correction and plot choices and finally the normalization type and normalized plot choices. The four processor activity times represented with ellipses indicate that the R scripts are being executed for the user's data, with the specified parameters, and that he must wait before proceeding to the next step. At the end of normalization the user could view the reports and the normalized data files, which were stored in the Mind LIMS.

The aim of this project is to add a new gene regulation assessment sub-module, while reorganizing the quality control sub-module as appropriate. The diagram of Figure 4.2 summarizes the steps a user will go through to perform the data analysis, with the new Mind microarray data analysis module. The first novelty is that the data analysis module has now three sub-modules: quality control, gene regulation assessment and data set.

The **data set** sub-module is the section where the user can create and manage data sets. A data set is created based on an existent Mind LIMS experiment belonging to the user and it consists of a subset of measurements to perform the microarray data analysis on. Initially, a user would perform the choice of measurements as a as a step of quality control itself, a step called "Define

Data Set". However, now the data set becomes more than just a choice of measurements of an experiment, it is the entity under which the reports, normalized data and regulated gene lists are saved. A major advantage is that now, the user can create multiple data sets for one experiment. The analysis functions are only run on one data set at a time, the one the user defines as the "Current Data Set", but all the data sets the user created are available, until he chooses to delete them. With the new data set sub-module, the normalized data files are no longer stored under the LIMS. All microarray data analysis information is associated to the data set.

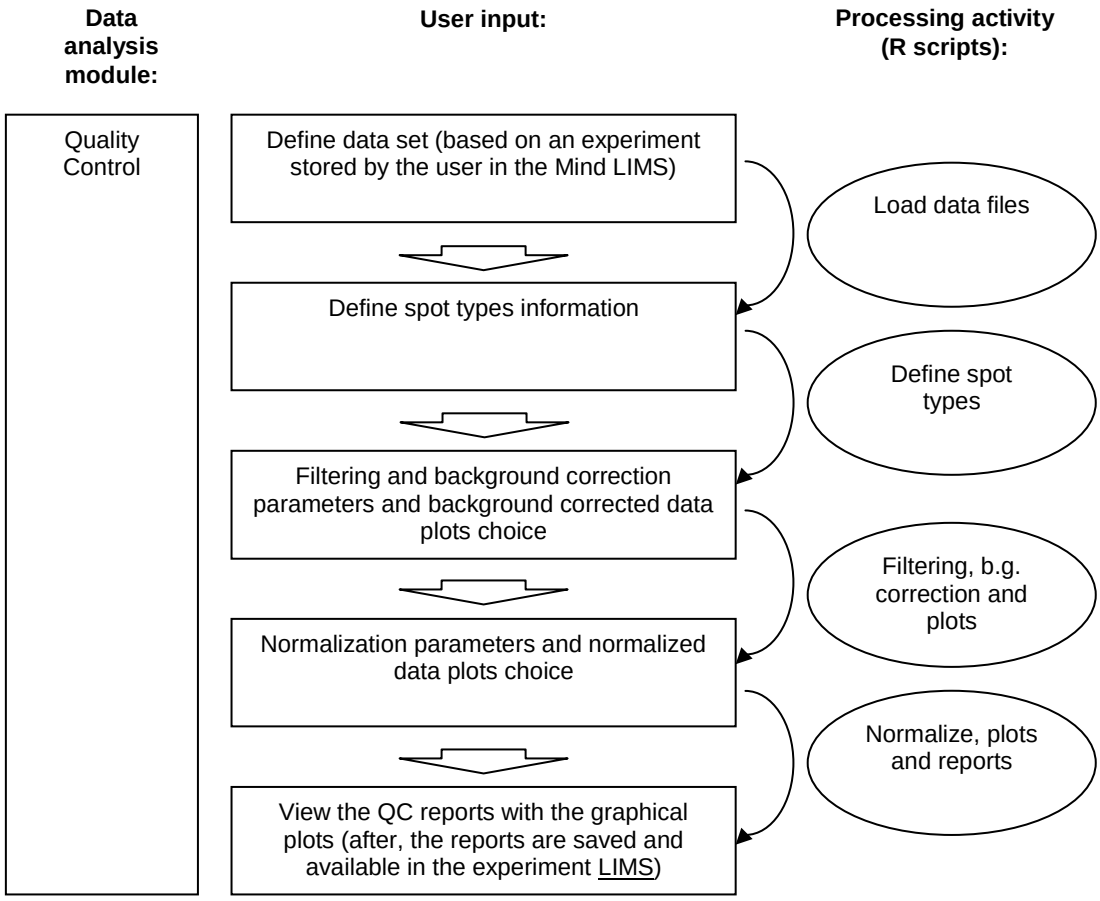



Figure 4.1 – Original Mind data analysis workflow

The **quality control** includes the same functions as before. It performs filtering, background correction and normalization on each array individually to correct the data for systematic biases and to assess the quality of each array, and generates the graphical plots and reports. However, all user input is inserted in one single step, except for the data set definition that now belongs to the data set module. Quality control has now one, but longer, processing activity time where the user waits for the results. The new quality control sub-module has a new GUI, but the R scripts executed are the same as before, although they are now executed sequentially.

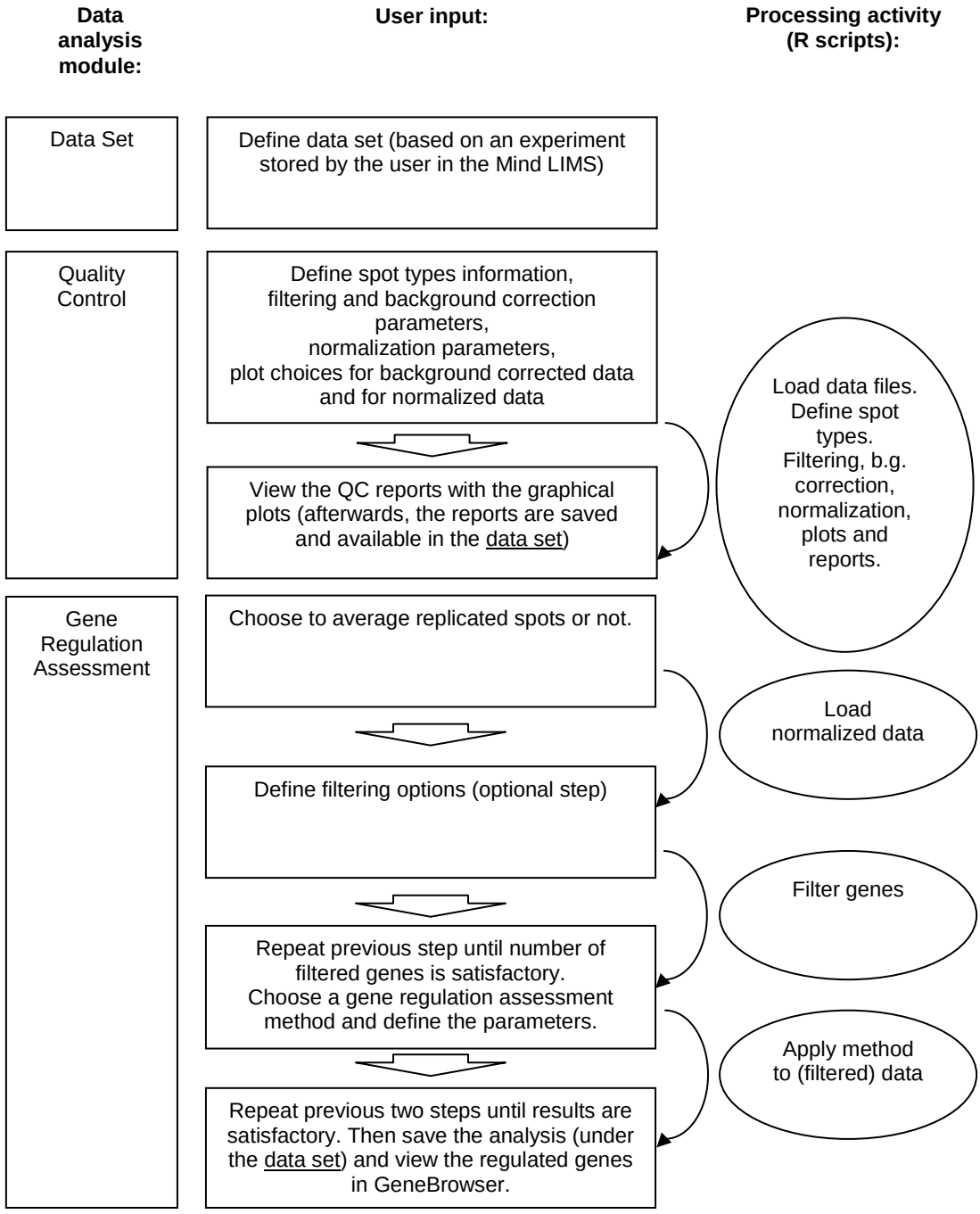



Figure 4.2 – New Mind data analysis workflow

The **gene regulation assessment** sub-module is done on the "current data set", whose data must have been normalized, meaning quality control must have already been run on the data set. It includes averaging replicated spots (optionally, but highly recommended), filtering (optionally) and applying a statistical method.

User input in quality control can be concatenated into one single step but the same cannot be done in the gene regulations assessment sub-module. Figure 4.2 shows three user input insertion

steps and three processor activity times for this sub-module. In gene regulation assessment the user must repeat the filtering until the number of filtered genes is satisfactory, as the statistical method will only be applied on the selected genes. Once the data is filtered, one of the statistical methods is applied with the desired parameters outputting a list of regulated genes. The number of regulated genes can be too high or too low, so the user may try several parameters and several methods until obtaining an analysis he will want to save and/or view in GeneBrowser. This is why in gene regulation assessment, data loading, filtering and statistical method application must be three different steps: they are divided so that each step the user wishes to repeat will be much quicker. In quality control the same did not happen, and the user would often prefer to leave all the data processing at once and read the reports only in the end.

To perform a data analysis, the user must have created an experiment in the Mind LIMS, and uploaded there the raw data files and the ADF or GAL file. LIMS stores the raw data files in their original format, as outputted by the image analysis software, supporting at the moment Spot, Quantarray and Agilent file formats (all two-color), and converts them to a Spot file (or copies them if they are originally Spot files already). The microarray data analysis will receive the Spot files as input. The LIMS also retains the ADF file and the GAL file for each experiment; it is the GAL file that is the input for the data analysis.

Data analysis begins by defining the data set in the data set sub-module. The user must create a new data set based on one of his LIMS experiments, or he can select an existing data set if he has already some created. The data set creation steps are related to the way an experiment is organized in LIMS. One experiment contains several bioassays and each bioassay relates to one actual microarray experiment. What happens is that the same array could have been scanned several times, with different settings, leading to several images. And each image may have been converted several times to raw data files, with different settings. Thus, one microarray can lead to several raw data files. The user can store all the generated raw data files in LIMS, organizing the experiment by bioassays, images and measurements. Each measurement corresponds to one only raw data file. For data analysis, only one raw data file can be considered for each array or bioassay. The data set creation allows the user to select what bioassays or arrays of the experiment to use, and if each one has multiple images and measurements, permits the use to choose one image and one measurement per bioassay. The user then types a name for his new data set and saves it. Upon creation, by default, it is set as the "Current Data Set".

Now the data set, defined as the current one, will undergo the quality control functions. The user needs to define the quality control parameters for spot types, filtering, background correction, normalization and plot creation (spot types information describes the colors used in the graphical plots created by Limma). The user can define the various parameters to perform

quality control, or he can use one of his favorite settings that he has saved previously. When a user runs quality control, he can choose to save the chosen settings. Next time, instead of choosing the spot types, filtering, background correction, normalization and plotting options one by one, he can just check to use one of the saved settings.

At the end of quality control, a report is generated for each array. The user is able to evaluate the quality of each array, and check if background correction and normalization was enough to correct for disturbances in the data. If not, the user can redo quality control with other options. Only the last normalized data files and quality control reports are saved for the data set, just like previously only the last normalized data files and quality control reports were saved under the measurements of the experiment, in the LIMS.

If the data is not acceptable after quality control, the user must redo the microarray experiment and/or the microarray image processing, and upload the new data files to the experiment in LIMS. In that case, if the user just replaces files in the same experiment, the Data Set will still be okay. If the user creates a new experiment in LIMS, a new Data Set must be created in the Data Analysis, because a Data Set is based on the experiment.

Once a quality control has been performed on the data set, it is ready for the gene regulation assessment. This begins with loading the normalized data files, selecting to average the replicated spot files or not. It includes then filtering (optional) and the application of a statistical method. After the data files are loaded, filtering can be performed several times, with different settings, until a satisfactory number of genes is obtained. Then a statistical method can be applied on the filtered data. Several methods with different settings can be tried, one at a time, on the filtered data. After trying a statistical method, the data can be filtered again without reloading the normalized files. This type of usage calls for the three input steps in gene regulation sub-module, already described, and suggests a menu for the GUI, which will be shown in this chapter. Appendix A enumerates all the steps and options in performing a Mind microarray data analysis.

## 4.3 Development

Mind consists of the web application itself and the database. All modules of the web application, LIMS, Data Analysis, Tools and Public Repository use the database. The Data Analysis module uses some tables of the database to retrieve and store data and it also uses R and R script to process data (Figure 4.3). This sub-chapter will describe the technologies used and some development matters relative to the creation of the new Data Analysis module.

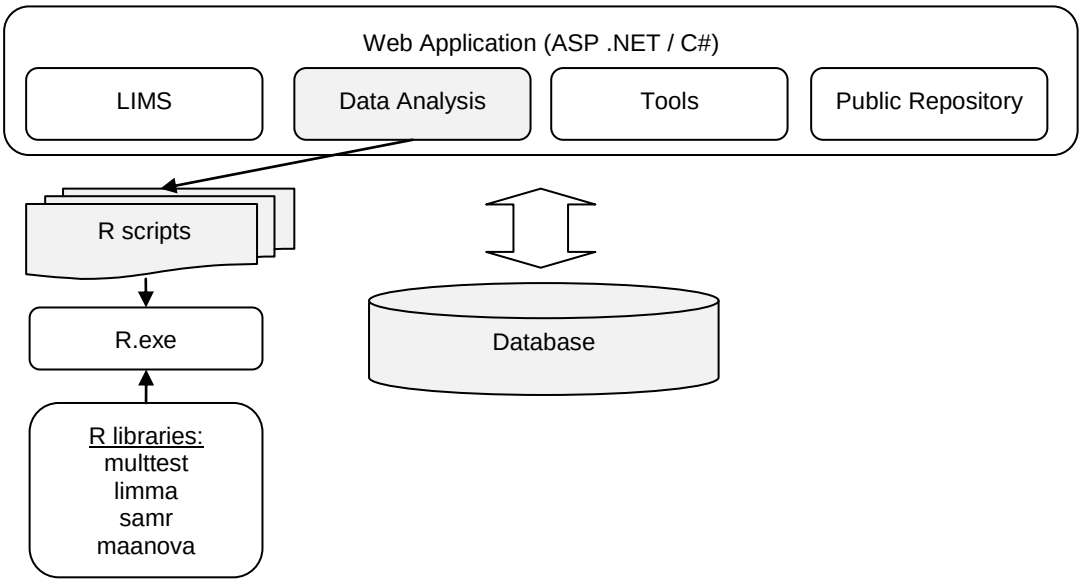



Figure 4.3 – The Mind application architecture

### 4.3.1 ASP .NET and C#

The Mind web application was originally created using the Microsoft Visual Studio 2005 environment and the database using Microsoft SQL Server 2005. It is natural to continue development with the new releases of these tools: Visual Studio 2008 and SQL Server 2008. Web development using Visual Studio is done using a choice of ASP .NET and C# or ASP .NET and Visual Basic. The application was created using ASP .NET and C# so development was continued in these programming languages.

Each page in a web application contains a markup file (.aspx file) with ASP .NET controls (buttons, radio buttons, data table views and many more) and a code-behind file (.cs file) with the C# code relative to the page and its controls. For example, a button will have a function in the code-behind file with code on what to do when it is clicked. The markup file contains the markup tags representative of the controls and it can also be viewed graphically. Web pages, besides ASP .NET controls, can contain user controls created by the developer, where each user control is a collection of ASP .NET controls with its own code-behind file. The web application also contains C# classes for functions that are common to several controls, for example, several buttons with similar code-behind behavior should all use the same function of a C# class, whenever possible, to avoid repeating the same code in the on-click functions of all buttons [50].

The Mind web application contains four main folders, which naturally reflect its architecture: LIMS, Data Analysis, Tools and Public Repository. This project worked mostly with the Data Analysis folder, which contains pages and user controls for the data set, quality control and gene regulation assessment sub-modules. A few of the original quality control user controls were maintained but most of them were modified or added. This project, maintaining all the functionality of the quality control sub-module, offered mainly through the Limma library, kept the R scripts intact but changed greatly the module's web pages.

Two folders were created in the web application to store temporary files. The first folder stores images generated in run time for display. After performing a statistical test, a plot is generated by R in the R work directory, which is not a folder belonging to the Mind web application project. Therefore it needs to be copied first to a web application folder in order for a graphical preview of the statistical test to be presented to the user. The second folder, stores temporary files for communication with GeneBrowser. When, from the Data Set, the user requests to view an analysis' genes in GeneBrowser, the text file containing the gene names is downloaded from the database and converted to an XML file. By passing the URL of the XML file to GeneBrowser, it can access the file and begin functional analysis. A folder is needed to store the temporary text and XML files, generated on each request for viewing the analysis regulated genes in GeneBrowser.

### 4.3.2 AJAX

AJAX (Asynchronous Javascript and XML) is a methodology followed to develop web applications, using JavaScript, XML, dynamic HTML and cascading style sheets in order to retrieve data from the server asynchronously, resulting in interactive and quick-responding web applications for the user. AJAX is better recognized by its applications then by its definition; whenever a particular section of a web page is refreshed without the rest of the page being reloaded, it may be AJAX in action.

Visual Studio 2008 comes will good support to this technology, allowing developers to include AJAX in their web applications just by adding items from the toolbox. AJAX was first seen as a need for the data analysis module for Limma's design matrix construction: as the user chooses different pairs of comparisons, by using an AJAX "UpdatePanel" from the IDE's toolbox in the web page, the design matrix is refreshed without the page being reloaded. The user does not need to know how to build the design matrix, but if he would like to see the design matrix originated by different parameters, it is presented as he chooses the parameters (Figure 4.4).

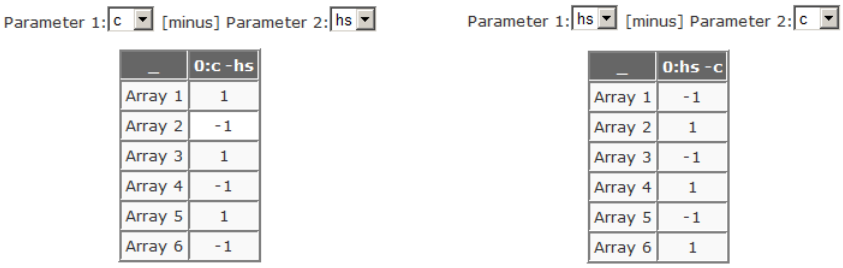



Figure 4.4 – AJAX for the design matrix creation

AJAX was then seen as a tool to enable background processing of the R scripts. Originally, in Mind, while the R scripts were running, the web browser would be waiting for the processing to complete in order to show "done" on the status bar and to present the results. With Visual Studio 2008's AJAX extension "UpdateProgress", while the R scripts are being executed, the web browser window is not waiting ("done" shown in the status bar) and an animated GIF is shown to indicate server activity (Figure 4.5). When the R scripts are done, the web application knows and replaces the GIF animation by a message indicating processing has been completed.

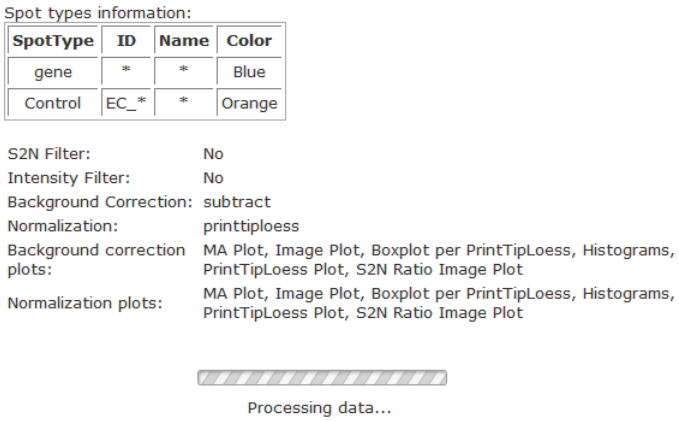



Figure 4.5 – AJAX for background processing

AJAX is also used to choose between defining parameters or using existing parameters for quality control. Two radio buttons are used to switch between the two options; the controls for one or the other are hidden and shown very quickly using the "UpdatePanel".

By using the AJAX extension offered in the Visual Studio IDE, some HTML and the JavaScript code is auto-generated in the web pages. As it is expected with auto-generated code, it may not be as easy to read (for debugging purposes), but all ASP controls, from radio button lists to tab containers, from data grid views to expression validators create auto-generated code, that, while not being as legible as typed HTML and JavaScript, is rather readable for debugging purposes. And creating a web application using an IDE like Visual Studio allows, after learning how to use the IDE, to build more powerful complications much more quickly than without them. With

inclusion of AJAX extension in the Visual Studio environment, a developer can create interactive and attractive web applications with much ease.

### 4.3.3 Running processes in the background

The initial quality control module asked three times for user input. The new module requires only one input screen, as was presented in section 5.1. The other issues pointed out relatively to the initial quality control module were that the R scripts were not executed in the background and quality control was performed, on the server side, for one data file at a time, when the files can actually be normalized in parallel.

Background processing was solved using AJAX. Parallel processing was enabled using a C# function that allows launching several processes in parallel, each of which is responsible for running the R script to normalized one data file. Using parallel processing, fantastic gains were obtained. Performing quality control initially for the six data files of the heat shock experiment would take 385 seconds, but now the normalization is done in 260 seconds, representing a 30% gain. These times were obtained (using the trace function) running the web application in debug mode, in the development dual core station containing two processors. With six R consoles running in parallel it is visible that the computer's processing capacity is used at its maximum. More gains are expected with the application not running in debug mode and in a server station with more processors.

### 4.3.4 R language and environment

Mind uses R to run the quality control and gene regulations assessment scripts. The quality control scripts were already existent and new scripts were created for gene regulation assessment. In the web application there is a class that handles running the R scripts with the R application. This class was already existent as of the beginning of this project and new functions were added to it as appropriate.

R is a software environment for statistics, plotting and data analysis. It is open-source, it is continuously being improved with new releases and there are many R libraries many of which are also being updated regularly. R libraries or packages provide additional functions that can be invoked when using R. Bioconductor provides many R libraries specific for the bioinformatics field. This is why the R and Bioconductor combination are a great choice for biologists. Limma, Multtest and Maanova are three of the many Bioconductor packages (Samr is not a Bioconductor package).

For gene regulation assessment, several scripts were created: one to load the normalized data files into R (averaging the spot replicates or not), one to filter the genes and one for each statistical test. The filtering script works on an R workspace file (.RData) created by the file loading script with the ready-to-use expression values. The statistical test scripts work on this same workspace file, whether it has been modified by filtering, or not (if no filtering was done). The quality control scripts, already existent, use Limma, and the new gene regulation assessment scripts use the Multtest, Limma, Samr and Maanova libraries (libraries introduced in Chapter 3).

## 4.4 Database

As a microarray LIMS, the Mind application includes a database to store and organize all the information related to the experiments. This information is used not only by LIMS, but by all application modules. This section will explain some of the most important tables of the microarray database (in the context of this project) and enumerate the alterations done to the database.

### 4.4.1 Existing tables required for this project

Some new tables need to be created to support all the modifications and new functionalities described and some fields of existing tables are no longer to be used. The Mind's database contains over forty tables dedicated to store information relative to the experiment, the microarrays used, the raw data resultant of the hybridizations and the biomaterial and labeled extracts. Figure 4.6 presents the tables whose knowledge is of particular importance to the development of the Mind data analysis module. Although all the tables of the database contain information relevant to a microarray experiment, not all of them are relevant in the context of this project, and only the ones that are used will be exposed.

The Experiments uploaded by a user in the Mind LIMS retain a reference to their Submitter, so the data analysis module is able to retrieve the Experiments belonging to the logged user. An Experiment includes several BioAssays, each BioAssay includes several Images and each Image contains several measurements or BioAssayData. Each measurement is associated to one raw data file (the original file uploaded by the user) and to the spot file to which the raw file is converted to.

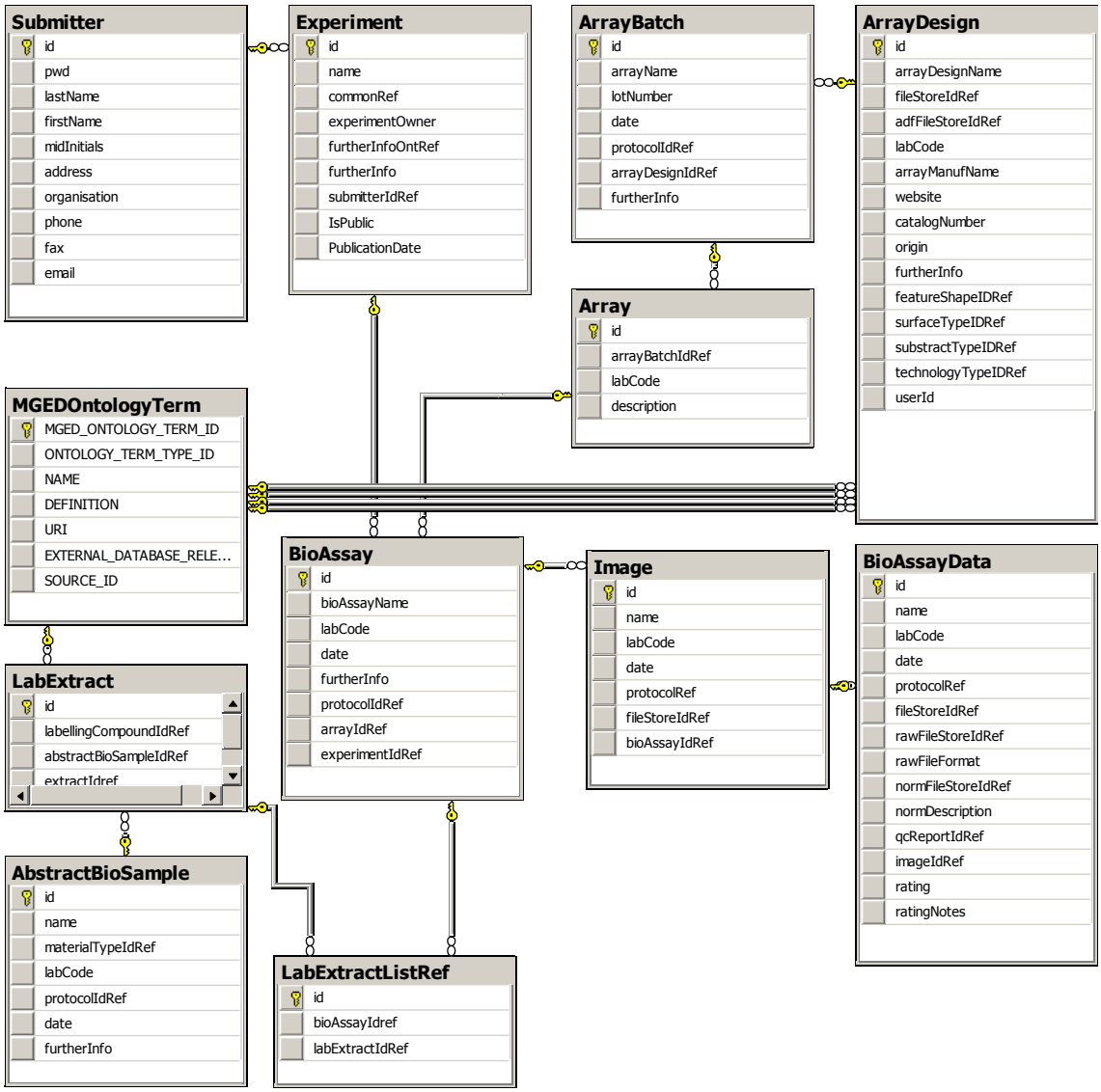



Figure 4.6 – Existing tables used for the development of the data analysis module

The BioAssay refers to the actual hybridization, the microarray itself. A BioAssay may originate several Images of the microarray is scanned with different settings, and each Image may originate various raw files for different conversion settings. The user may store all the raw data files relative to one BioAssay, by organizing them by Images and by Measurements in the Mind LIMS. The BioAssayData includes a reference to a row entry in the FileStore table (not shown in the diagram) where the raw and spot files are stored. The FileStore table stores, for a given file, the name, the length, and the data in byte format.

The GAL and ADF file references are stored in the ArrayDesign table. The array layout in the GAL format is required for the data analysis. Each BioAssay is associated to one Array of an ArrayBatch, being the ArrayDesign common for all the BioAssays.

The LabExtractListRef, the Lab Extract, the AbstractBioSample and the MGEDOntologyTerm tables are used in the data analysis module to retrieve the two labeled extract names that the user associated to a BioAssay, one for the red channel and one for the green channel.

### 4.4.2 New tables

The need to store the gene regulation assessment analysis immediately justifies the creation of additional tables in the Mind database. The creation of the data set and saving the quality control and gene regulation assessment reports under this entity (and not associated to the LIMS experiment) needs new tables too. This project requires the addition of six new tables to the database: DataSet, DataSetItem, QCParameters, CurrentDataSet, Analysis and TargetsInfo (Figure 4.7). These tables will now be explained and how they are related to the existing tables. The table creation script is included in Appendix B.

The user can create data sets for an experiment he owns. Each entry of the DataSet table contains a reference describing what Experiment the data set belongs to and a field with the creation data to allow for easier identification of the DataSets. The table contains also two fields to describe the background correction and normalization methods applied on the data last time quality control was run.

Each DataSet refers to a subset of measurements, being each measurement represented as an entry in the BioAssayData table. Each BioAssayData item contains a reference to the Image it belongs to, each Image item contains a reference to the experiment it belongs to, and each Image contains a reference to the Experiment it belongs to. This may suggest some redundancy as the DataSet table also references the experiment. It is found however that this is justified, as it allows, for example, listing an experiment's data sets to be a more efficient query. A data set is composed of several data set items. Each DataSetItem table entry contains a reference to the data set it belongs to, a reference to the measurement that contains the raw data and two references to the FileStore table (not shown in the diagram) where the normalized data file and the QC report are now to be stored. In the BioAssayData table, the normFileStoreIdRef, normDescription and qcReportIdRef fields are no longer needed: the normalized data files and the quality control reports are now associated to the data set items, and the description of the normalization and background methods used are properties of the data set.

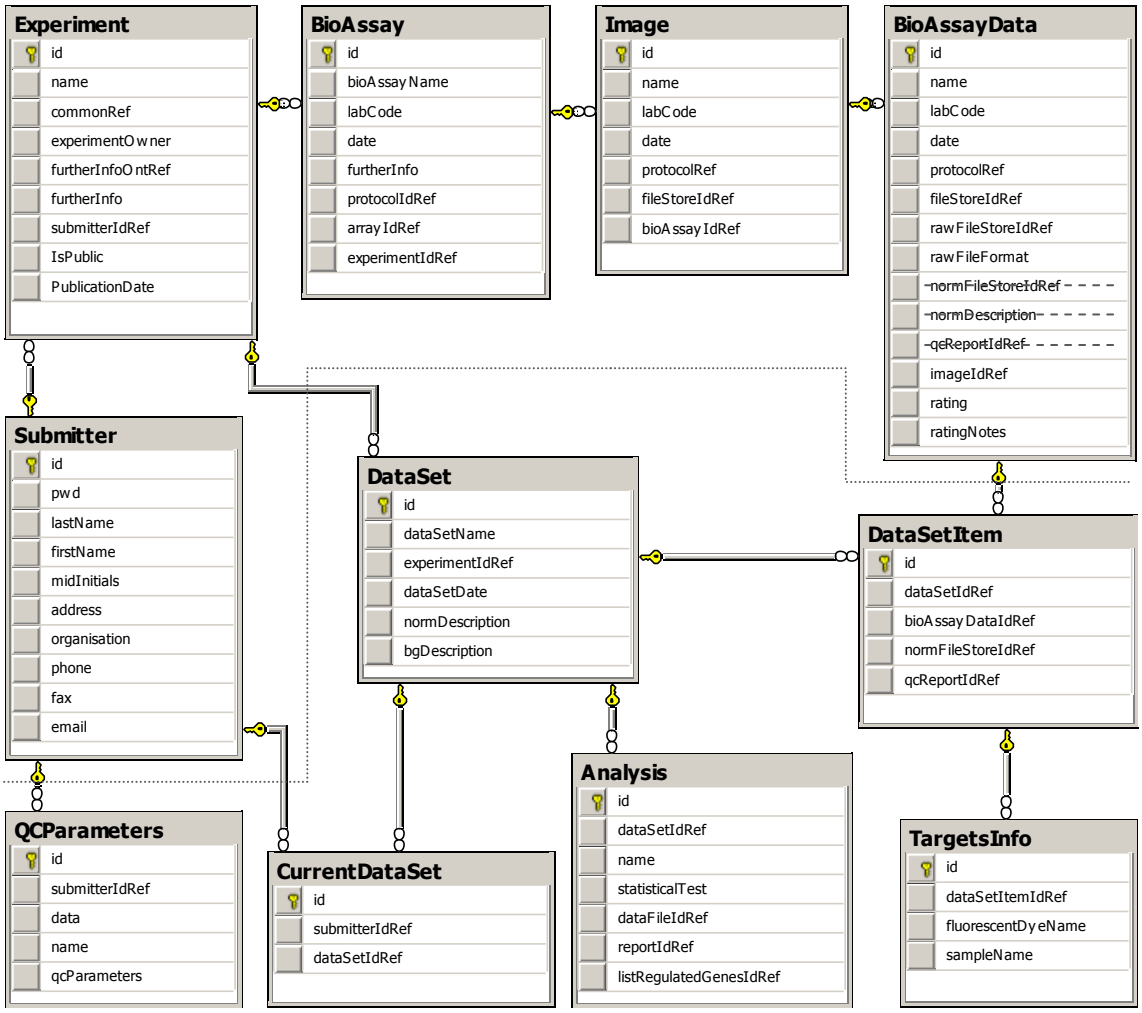



Figure 4.7 – New tables and associations to existing tables

The current data set will be associated with the user. The user can only have one of his data sets defined as the current one at a time, so the CurrentDataSet table will only have one row per user (and zero rows for users who have not yet defined their current data set). Quality control and gene regulation assessment are always performed on the current data set.

Each time quality control is run on a data set, the parameters used can be saved as one of the user's favorite quality control parameters with an entry in the table QCParameters. It includes fields to store a name chosen by the user for the parameters, the creation data (for easier identification), a reference to the user they belong too, and a long string containing all the parameters (up to 1000 characters long). The user may delete all his data sets and still have favorite parameters stored, as he can do several normalizations and never define any favorite parameters.

Once its data is normalized, several regulated gene assessment tests may be done on one data set. Each test will be saved with an entry in the table Analysis. One data set can contain zero or more gene regulation assessment analysis; therefore, each Analysis must reference the DataSet it belongs to. The Name is chosen by the user to identify the analysis within the data set. The Type just describes the method used, like "foldchange" or "t-test". An Analysis also contains references to the FileStore table, where the reports and data files are stored.

To assess differential expression using Limma, the data set must have the targets information defined, meaning, on each array, what mRNA sample is on the red channel and what mRNA sample is on the green channel. Each channel on each array belongs to a group, "experiment" or "control", "A" or "B", "1" or "2". The data set can involve two or more mRNA samples. The name of the samples itself is not important, but the user is advised to define the targets information in a meaningfully way (e.g., name the samples "heatshock" or "control" rather than "A" or "B"). The targets information is mandatory for Limma and it is useful on some other tests too. For example, in the t-test method, considering the heat shock example, if the targets information has been defined, it will be presented to assist the user in defining the groups; otherwise he will have to form the groups just using the bioassay or measurement names. There is some information on LIMS regarding the labeled extracts used, but this information is not mandatory (some experiments stored in the database as of the beginning of this project do not contain it). Even if an experiment's bioassays have the labeled extracts described, these may not reflect the groups that the user wishes to analyze. Therefore it was thought better to have the user define the targets information in the data analysis module, as a characteristic of the data set. The TargetsInfo table, for each DataSetItem, contains two entries: one with the mRNA sample associated to the green channel (cy3) and one with the mRNA sample associated to the red channel (cy5). This approach of having two rows for each DataSetItem, instead of one, was chosen to avoid having "Cy3" and "Cy5" as columns names. The supported labeled extract names at the moment are only two, but they can be more in the future, and it was thought best to build the table this way. The supported labeled extracts are contained on the MGEDOntologyTerm table, and more can be added in the future, such as Cy2, a blue cyanine. Although the application does not support more than two-color arrays, the database is designed with this future possibility in mind.

Stored procedures to delete completely an analysis and a data set were also created and added to the database. These two stored procedures are in Appendix C.

## 4.5 Application Overview

The Mind Data Analysis module is divided into three sub-modules: Data Set, Quality Control and Gene Regulation. Once the user initiates the Data Analysis module, the default sub-module presented is the Data Set. This is where the user creates his Data Sets based on his LIMS experiments, views them and manages them. Once the user has a data set selected as the Current Data Set, he can move to the Quality Control sub-module where he can perform quality control on the raw data files. The normalized data files and the reports are associated to the Current Data Set and downloadable from there. After normalizing the data files, Gene Regulation Assessment can begin, that will elaborate a list of regulated genes. The analysis reports and data files are saved under the Current Data Set too.

### 4.5.1 Data Set

The Current Data Set is the default page of the Data Set sub-module. The first time using Mind Data Analysis, the user has no created Data Sets and therefore no Current Data Sets, so he is prompted to create a new one first in the New Data Set page. The list of the user's data sets is under the My Data Sets sections. The user can define the Current Data Set by creating a new one or by choosing one of the created data sets.

Figure 4.8 shows a view of the Current Data Set, on which quality control and several analyses have already been run. The header includes general information about the data set, such as its name, the background correction and normalization methods applied when quality control was performed and the LIMS experiment it is associated to. The Data Set Items section enumerates all the Data Set Items, and each one has a link to the LIMS, which opens, in a new browser window or tab, the Measurement in the Mind LIMS, showing the details about the raw data file and the spot data file (to which the raw data file was converted, if of another format). Since QC has been performed, the Data Set Items provide links to their normalized data files and QC reports. Appendix D presents a sample QC report. The QC reports contain the same graphical plots (generated by Limma) as the original ones, but a header has been added with information relevant to the data set and with the QC parameters chosen. The Targets Information indicates what samples were hybridized to each array on what channel, and must be set for each Data Set created. Finally, the Analysis table lists all the gene regulation assessment analyses performed on the normalized data, including the data files (regulated genes file and all genes file), the analysis report and a link to view the regulated genes on GeneBrowser, by opening in a new browser window or tab GeneBrowser loaded with the regulated genes.

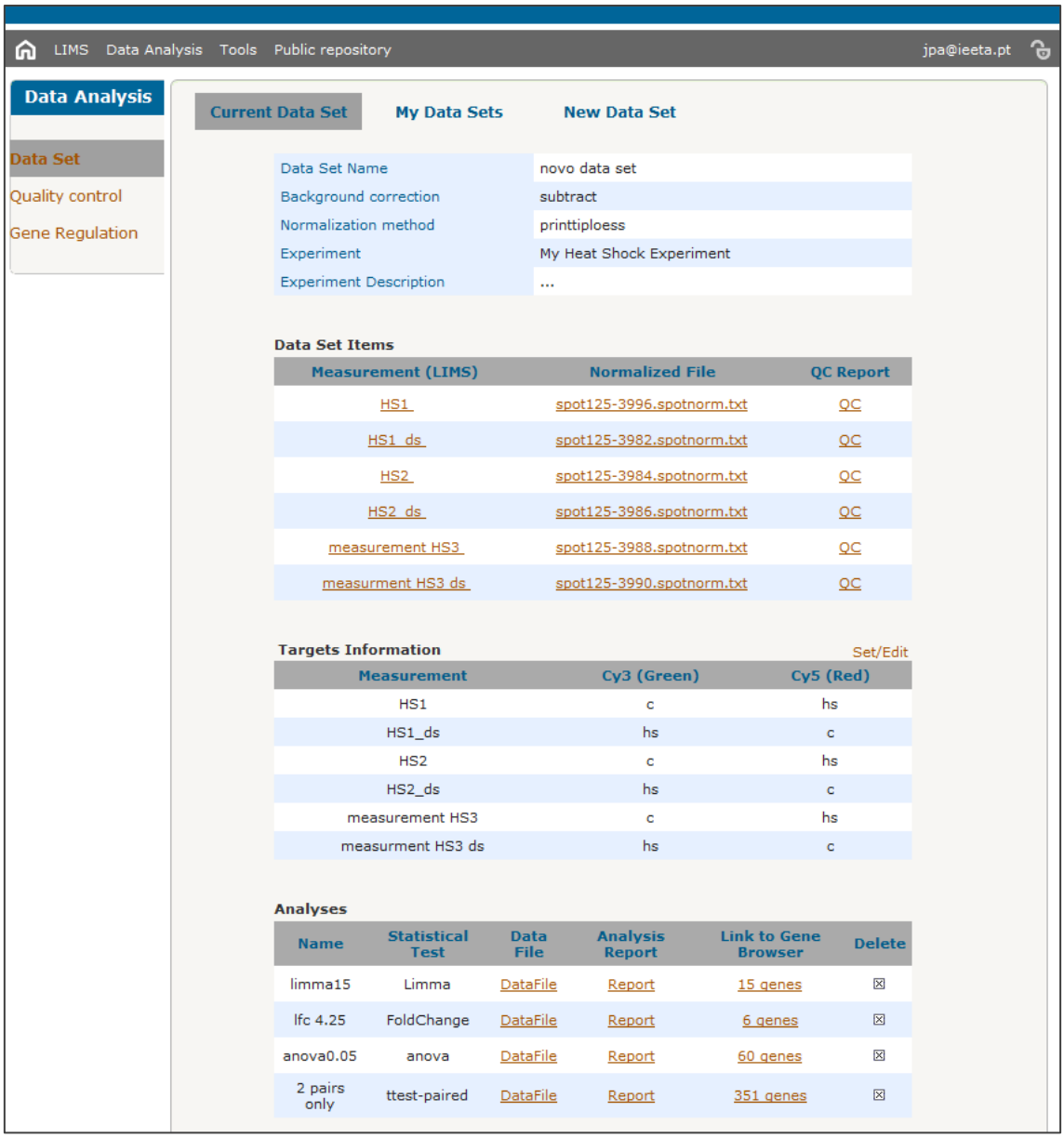



Figure 4.8 – Data Set: Current Data Set

As only the last quality control information (normalized data files and QC reports) is stored, and gene regulation assessment analysis are run on the most recent normalized data available for the Current Data Set, each gene regulation assessment report also contains the information about the background correction and normalization methods that were used. The data set items and the groups considered may be different for each analysis, so this information is registered in the analysis reports too. Appendix E shows a sample analysis report. The analyses that are no longer relevant may be deleted from a data set.

Under My Data Sets, all the user's data sets are listed, ordered by experiment and then by creation date. Data sets can be deleted here. It can be chosen to select the detailed view of the

data sets in the list. When a detailed view of a data set that is not the Current Data Set is presented, that data set can be defined to be the current one.

To create a New Data Set, the user chooses one of his LIMS experiments, then the desired bioassays and the measurements (applicable if at least one bioassay has multiple measurements). Finally the user types a name for the data set and unchecks a checkbox if the data set is not to be set as the current one.

### 4.5.2 Quality Control

By selecting the quality control sub-module, the user is prompted to set the desired QC setting or to retrieve existing ones, in order to run quality control on the normalized data files. QC parameters include spot types definition and filtering options for the graphical plots, the selection of the plots, and the background correction and normalization methods that are to be applied to the data (Figure 4.9).

If the user prefers to retrieve existing settings, a list is presented with the list of settings, with their names and creation dates, and the user just needs to select one.

Since quality control normally takes a few minutes, the user is requested to confirm the parameters before proceeding. Parameters are presented for confirmation as shown previously in Figure 4.5. While quality control is processing, an animated GIF indicates normalization is being carried out. Once terminated, the user is informed that the normalized data files and the reports are available under the Current Data Set.

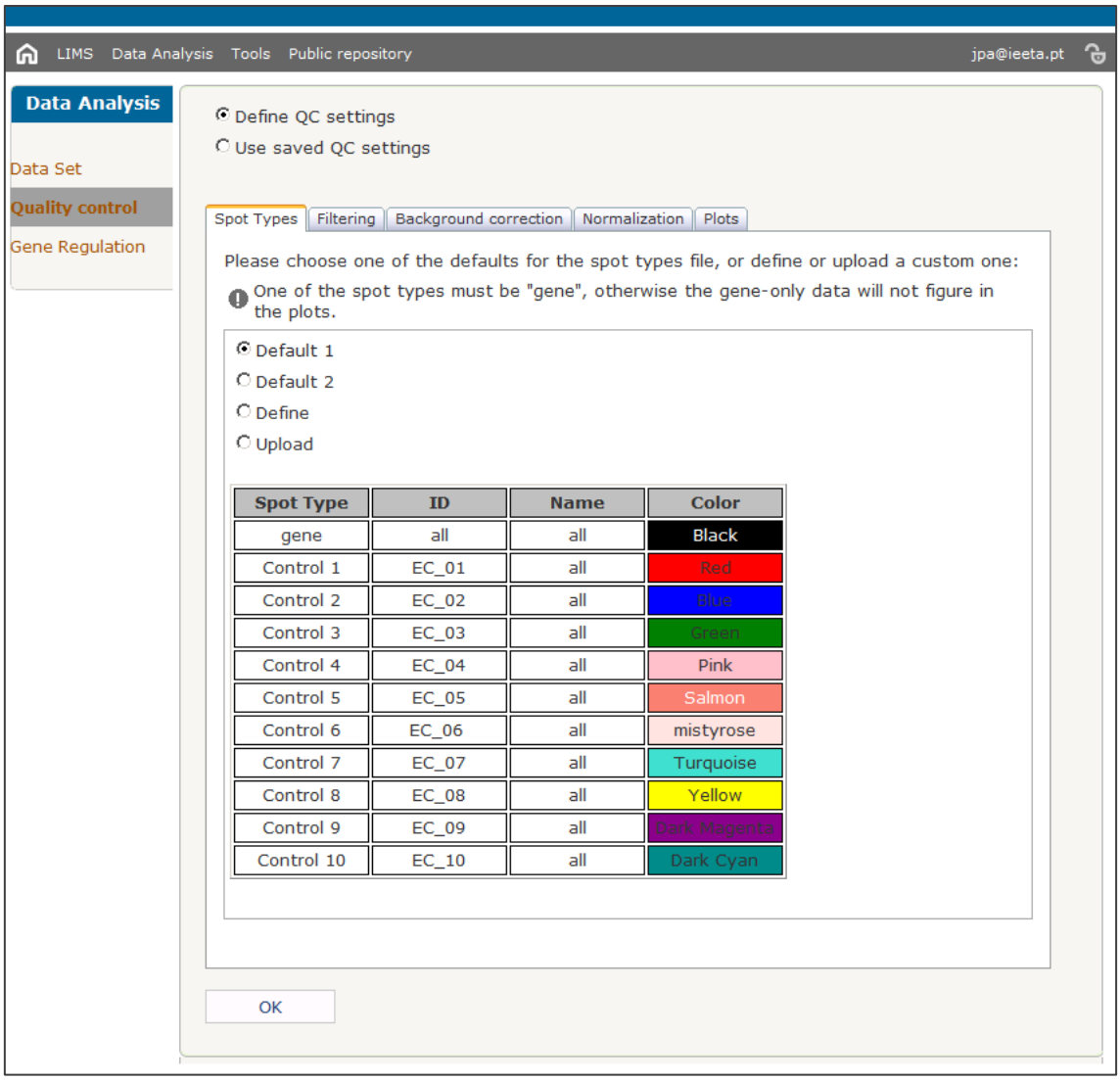



Figure 4.9 – Quality Control: Parameters

### 4.5.3 Gene regulation assessment

Gene regulation assessment starts with loading the normalized data files, with or without the replicated spots averaged (Figure 4.10).

If any filtering options are intended, they are set (Figure 4.11). Filtering can be applied as many times until the number of genes that are selected is adequate, and then the user may choose a statistical test. If no filtering is wished, the user can move from file loading to the statistical test desired. The selection of the statistical test is done using the menu.

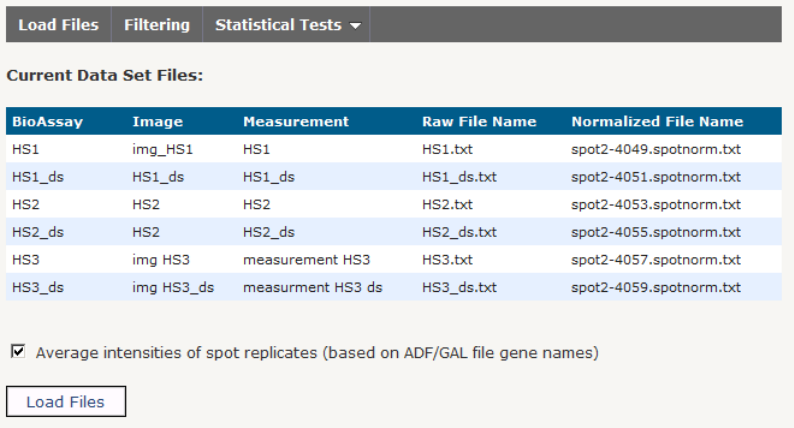



Figure 4.10 – Gene Regulation: Load normalized data, averaging spot replicates

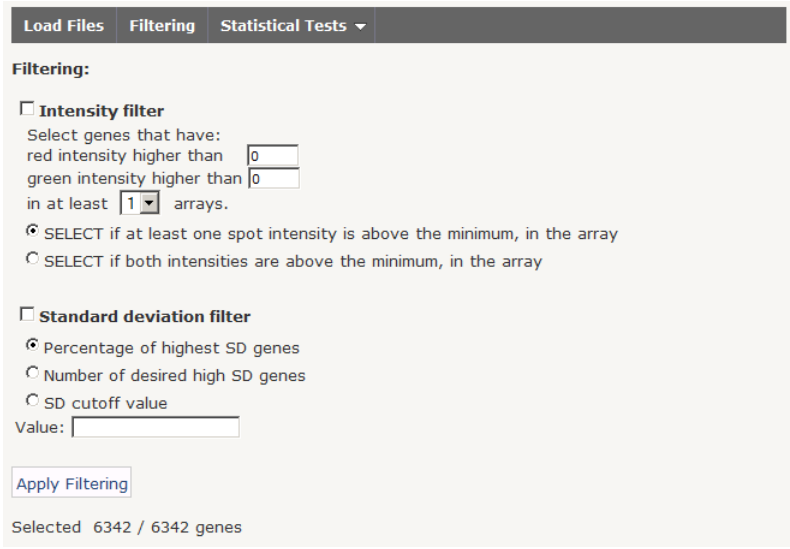



Figure 4.11 – Gene Regulation: Filtering

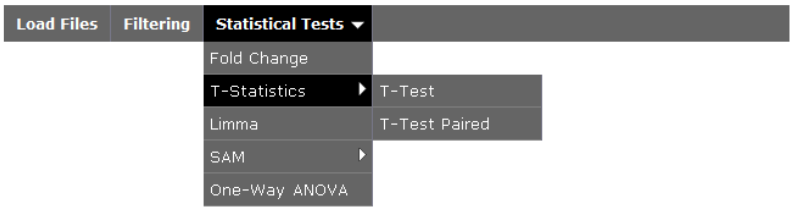



Figure 4.12 – Gene Regulation: Menu

A statistical test starts with the group definition. For Limma, the groups are pre-defined based on the mRNA sample names (targets information of the data set), but for the other tests, the groups must be formed for the test to proceed. Figure 4.13 shows how to form the control and experiment groups by dividing the channels (for fold-change and t-test with ratio log fold-change) and Figure 4.14 shows how to form two groups by dividing the arrays (for t-test with difference log fold-change and SAM two-class).

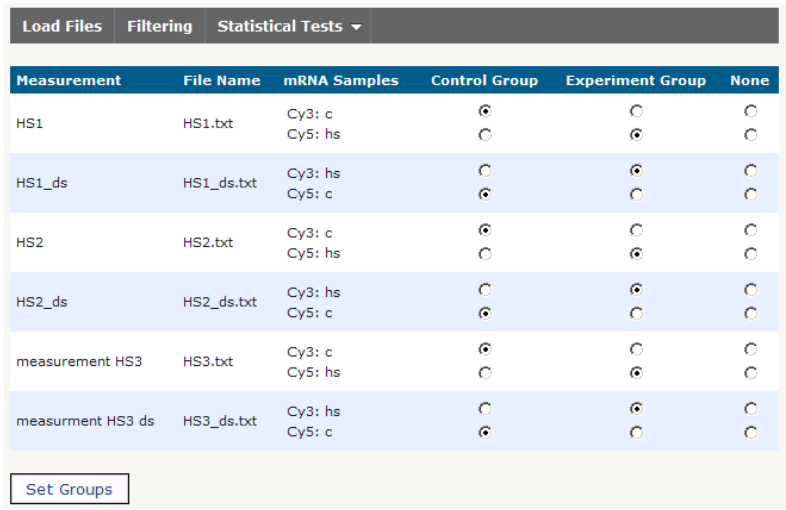



Figure 4.13 – Gene Regulation: divide by channels

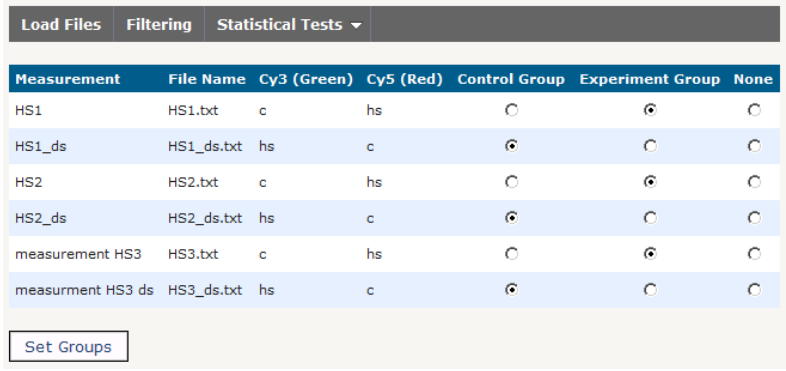



Figure 4.14 – Gene Regulation: divide by arrays

Then, the parameters specific for the statistical test can be inputted. Figure 4.15 shows the input to run the t-test. When the t-test is completed, the volcano plots are presented with the number of regulated genes correspondent to the chosen parameters. If these are too many or too little, the parameters can be chosen again and the test repeated, until a satisfactory number of genes is obtained. If the number of genes is adequate, the analysis can be saved. It will then be available under the Current Data Set, where the report and analysis data can be downloaded and the regulated gene list can be open in GeneBrowser.

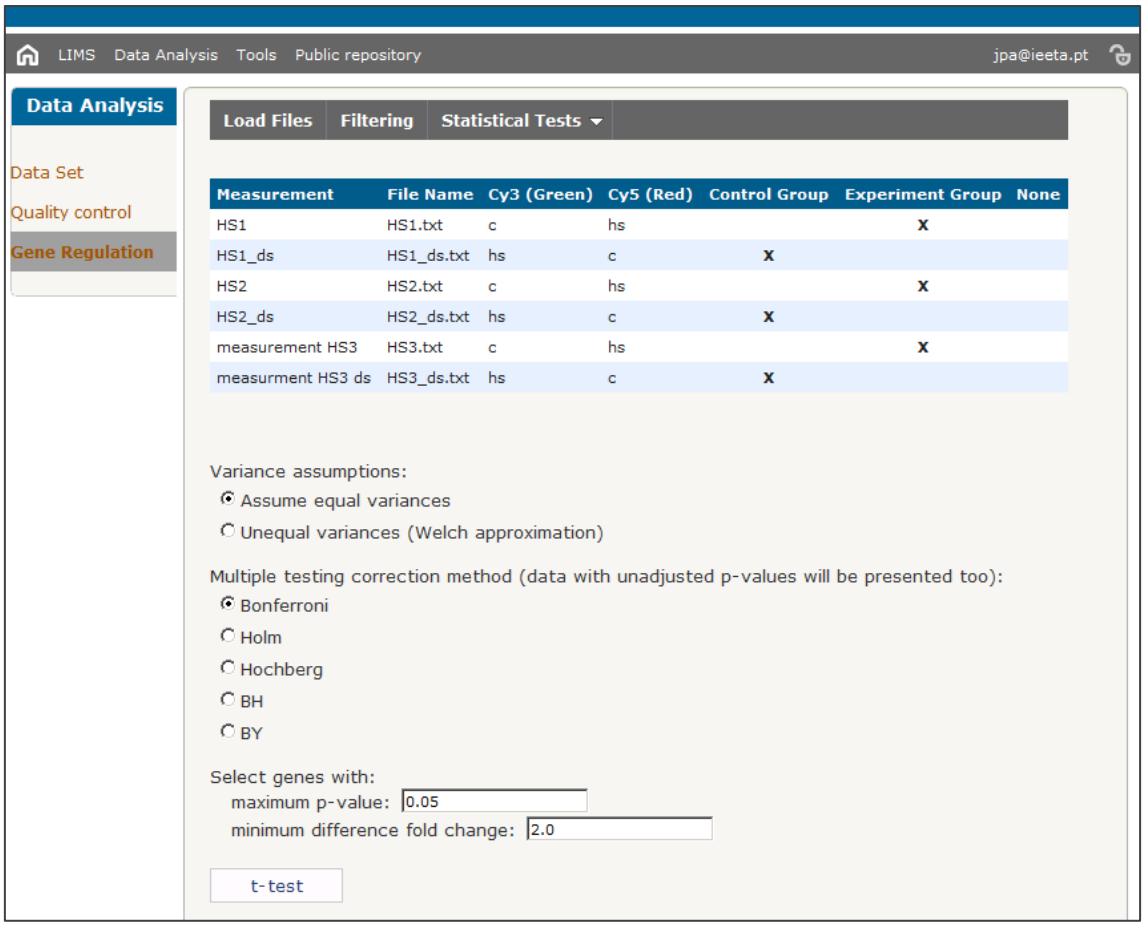



Figure 4.15 – Gene Regulation: statistical test t-test

## 4.6 Summary

This chapter presented the resulting final application, while exposing development issues, technologies used and workflow philosophy. The old workflow was described in order to better introduce the new one and the technologies and programming languages used in the development of the data analysis module were mentioned. The new database tables created and their relation to existing ones were also illustrated. Finally, an overview of the final application, including some screen captures, was shown.

# Chapter 5
# Conclusion

The DNA microarray technology has been proven to be a very valuable tool in genetic research. By probing the expression levels of genes in different conditions, their biological function and the interactions between them can be learned. As thousands of genes can be probed at once, the entire genome of an organism can be probed in one array or in a few arrays. A sample of patients with a certain disease can be compared with a sample of healthy patients to identify the few genes that appear differentially expressed as a result of the disease. A new drug can be developed to target those few genes, without affecting the rest of the human genome. In fact, the application of microarrays in medicine shows great potential for diseases to be better understood and for cures to be more rapidly learned.

In a DNA microarray experiment for assessment of regulated genes, the mRNA extracted from the sample cells is measured and used to infer about gene expression, as the mRNA produced for protein synthesis in an indicator of gene expression. Two-color arrays probe, on the same slide, for the expression levels of thousands of genes in two mRNA samples: one that has been labeled with a green fluorescent dye and the other labeled with a red fluorescent dye. The experiment generates raw data files that describe, for each gene, the expression levels measured for the red and the green sample. Each microarray in an experiment should have a few replicates, as microarrays are typically noisy, and having at least two replicates of each array helps to obtain more reliable gene expression values (there cannot be many replicates, as each microarray is costly). Together, the noise, the few number of arrays replicates and the thousands of genes being probed at once, pose a statistical challenge in finding the genes that are truly regulated. Therefore, this technology is still far from being completely accurate, in the sense in cannot say exactly what genes are differently expressed, but it has been a great guide in genetic research. New methodologies are being continuously being developed and improved to perfection the microarray technology and the microarray data analysis process.

The microarray data analysis includes all the procedures that are undergone to obtain, from the raw data files, the list of regulated genes. Data analysis can be divided into quality control and gene regulation assessment. Quality control involves correcting the data for unwanted

systematic biases, with background correction and normalization and confirming the quality of the data with graphical plots and reports. Gene regulation assessment, from the normalized data, uses filtering and statistical methods, such as fold-change, t-test, Limma, SAM and ANOVA to obtain a list of regulated genes.

The goal of this development project was not to improve any methods, nor to increase the reliability of results obtained, but to integrate several existing microarray data analysis tools and methods into Mind. Many users rely on Mind for easy and MIAME-compliant data storage and on GeneBrowser for intuitive gene functional analysis. This new data analysis module allows quick differential expression of experimental data stored in the Mind LIMS and transparent exporting of the regulated gene lists to GeneBrowser, therefore facilitating obtaining conclusions from the data.

## 5.1 Results

From the beginning, this project aimed to add gene regulation assessment to the Mind data analysis module, revising the existing quality control functions as necessary.

The first part of this work involved studying DNA microarray technology, data analysis methods and data analysis software. This research was necessary in order to understand the application to be developed and included selecting the gene regulation assessment methods for Mind.

The quality control functionalities offered in Mind through the Limma library were found adequate by the users and needed no alterations. These include methods for background correction (half, minimum, subtract, norm exp, edwards and moving minimum) and for normalization (median, robust spline, loess and print-tip loess). The plots generated with Limma (MA plots, histograms, and more) are also satisfactory.

For the gene regulation assessment sub-module, spot replicate averaging, filtering and several statistical methods were chosen. Spot replicate averaging is highly recommended because on a microarray, several spots can refer to the same gene. Filtering is optional and is done if the user would like to perform a statistical test on a smaller number of genes. The statistical tests selected were: fold-change, t-test (paired and unpaired), Limma, SAM (two-class, two-class paired and multi-class) and one-way ANOVA. The multiple comparison issue characteristic of microarray data was also addressed and some correction methods presented.

The second part of this project involved the actual development. The quality control sub-module had its GUI renewed and it now allows sequential processing of each raw data files and parallel processing of several data files, in the background. Parallel processing allows for data to be normalized in much less time.

The quality data processing functionalities themselves (background correction, normalization and plots) were maintained, meaning that the R scripts are the same as the initial ones. For the gene regulation assessment, the interfaces for file loading and spot replicate averaging, filtering and statistical tests were created, as well as the R scripts. Quality control scripts use Limma and the gene regulation scripts use Limma, Multtest, Samr and Maanova R libraries.

A new entity was created, the data set, on which quality control and gene regulation assessment are run. The data set is a subset of an experiment's bioassays, on which data analysis is performed, and to which all files resultant of the analysis are associated (quality control reports, normalized data files, gene regulations assessment reports and lists of differentially expressed genes). As a result, the Mind data analysis module has now three sub-modules: data set, quality control and gene regulation.

A new format was created for both quality control and gene regulation HTML reports, including important information relative to the experiment and to the data set and the user-inputted parameters used to generate the final results. A chapter for the Mind User's Guide will also be provided to introduce the user to the Mind Data Analysis module. Exporting the list of regulated genes to GeneBrowser was also addressed, as for each gene regulation analysis saved in the data set, a link is created to allow opening GeneBrowser in a new browser window with the regulated genes already loaded.

Development involved working with several different development technologies: ASP .NET and C#, AJAX, SQL and R. In fact, performing this project required learning concepts specific to computer science, biology and statistical fields, as these three fields are combined in DNA microarray data analysis.

## 5.2 Further developments

The gene regulation assessment module establishes a bridge between Mind and GeneBrowser, but it does not conclude Mind nor does it conclude the data analysis module.

For the gene regulation sub-module, an easy suggestion is the addition of more statistical tests. An effort was made to include methods that were widely used, since the depth if this project did not allow for implementation of many methods. Two-way ANOVA for the analysis of several groups and factors is indeed a very attractive feature to add. Non-parametric tests are also very useful as they permit statistical analysis without assumptions about the distribution of the data.

It is recommended to address the multiple comparison issue too. This is a complex statistical problem, and this dissertation presented the need to address it and the module included correction methods for the statistical tests, but this should be better looked into. Namely, the addition of permutation-based correction methods, that require more processor time, but that often provide better and less conservative results.

Another point is the spot replicates. This document emphasized how important array and spot replicates are. However, while array replicates can be analyzed with statistical tests, the most common way to handle spot replicates is simply by averaging them, which disregards variability throughout the spot replicates. It is recommended to look out for a new library or function, or to create a new function, for spot replicate averaging.

Exploratory data analysis, which includes clustering algorithms such as hierarchical clustering and K-means, is a great feature to add to Mind as a new Data Analysis sub-module.

Data mining for the bioinformatics field, especially for microarray data, has been a popular topic, and it is agreed that adding data mining to Mind should be thought about. This dissertation and project present methodologies for gene regulation assessment, but it has been left clear that they to not indicate exactly what genes are regulated. The likelihood of false positives and false negatives is far from being eliminated. Data mining intends to offer a more robust analysis in order to detect more reliably the regulated genes.

The most important suggestion that can be left, however, is to pay attention to valuable user opinions. His feedback over time is fundamental, to note some adjustments that could be made to the application and to suggest further developments.

With only these topics, it can be seen much can still be done for the Mind data analysis module, and much can be done in the DNA microarray area in general. Microarrays offer the possibility of very exciting applications, and have been so far a great guide for researchers who use them. All work done towards perfectionning this technology or making it more accessible is undoubtedly to feel very rewarding as it may lead us towards more advancements in health care and a deeper knowledge of the universe around us.

# References


1. Watson, J.D., *The human genome project: past, present, and future.* Science, 1990. **248**(4951): p. 44-9.

2. Dunn, P.M., *Gregor Mendel, OSA (1822-1884), founder of scientific genetics.* Arch Dis Child Fetal Neonatal Ed, 2003. **88**(6): p. F537-9.

3. O'Connor, C., *Isolating Hereditary Material: Frederick Griffith, Oswald Avery, Alfred Hershey, and Martha Chase.* Nature Education, 2008 (www.nature.com/scitable/).

4. Pray, L.A., *Discovery of DNA Structure and Function: Watson and Crick.* Nature Education, 2008 (www.nature.com/scitable/).

5. Hogeweg, P. and B. Hesper, *Interactive instruction on population interactions.* Comput Biol Med, 1978. **8**(4): p. 319-27.

6. Schena, M., et al., *Quantitative monitoring of gene expression patterns with a complementary DNA microarray.* Science, 1995. **270**(5235): p. 467-70.

7. Arrais, J.P., et al., *A Microarray Information Database*, in *Biocomputation, Bioinformatics, and Biomedical Technologies, 2008. BIOTECHNO '08. International Conference on*. 2008: Bucharest.

8. Brazma, A., et al., *Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.* Nat Genet, 2001. **29**(4): p. 365-71.

9. Babu, M., *An Introduction to Microarray Data Analysis*, in *Computational Genomics: Theory and Application*, R.P. Grant, Editor. 2004, Horizon Bioscience.

10. Arrais, J., et al., *GeneBrowser: an approach for integration and functional classification of genomic data.* Journal of Integrative Bioinformatics, 2007.

11. ImageDNA. *DNA molecule image*. 2009 [cited November 2009]; Available from: http://ghr.nlm.nih.gov/handbook/basics/dna.

12. ImageCellOrganells. *Prokaryotic and Eukaryotic Cells*. [cited November 2009]; Available from: http://www.nslc.wustl.edu/courses/Bio2960/labs/04Microscopy/index.html.

13. Silva, A.D.d., et al., *Terra, Universo de Vida - Ciências da Terra e da Vida, 11º ano*. 2000: Porto Editora.

14. Lee, M.L., et al., *Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations.* Proc Natl Acad Sci U S A, 2000. **97**(18): p. 9834-9.

15. Lorkowski, S. and P. Cullen, *Analysing Gene Expression: A Handbook of Methods: Possibilities and Pitfalls*, P.C. Stefan Lorkowski, Editor. 2003, Wiley-VCH.

16. Seki, M., et al., *Gene expression profiling in plants using cDNA microarrays*, in *DNA microarrays BIOS advanced methods*, U. Nuber, Editor. 2005, Taylor&Francis.

17. Stekel, D., *Microarray Bioinformatics*. 2003: Cambridge University Press.

18. ImageCommercialArray. *Microarray service provider*. 2009 [cited November 2009]; Available from: http://www.labnews.co.uk/product_archive.php/1602/3/microarray-service-provider.

19. ImageGlassSlide. *Glass Slide Microarrayer*. 2009 [cited November 2009]; Available from: http://www.vp-scientific.com/glass_slide_microarrayer.php.

20. ImageSpotter. *Core Unit DKFZ: High throughput Gene Expression Analysis*. 2009 [cited November 2009]; Available from: http://www.science.ngfn.de/10_312.htm.

21. Rayner, T.F., et al., *A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB.* BMC Bioinformatics, 2006. **7**: p. 489.

22. R Development Core Team *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. 2009, Vienna, Austria.

23. Causton, H.C., et al., *Remodeling of yeast genome expression in response to environmental changes.* Mol Biol Cell, 2001. **12**(2): p. 323-37.

24. Quackenbush, J., *Microarray gene expression data analysis: a beginner's guide*. 2003: Blackwell Publishing.

25. Draghici, S., *Data Analysis Tools for DNA Microarrays*. 2003: Chapman & Hall/CRC.

26. Draghici, S., *Statistical intelligence: effective analysis of high-density microarray data.* Drug Discov Today, 2002. **7**(11 Suppl): p. S55-63.

27. Ritchie, M.E., et al., *A comparison of background correction methods for two-colour microarrays.* Bioinformatics, 2007. **23**(20): p. 2700-7.

28. Yang, Y.H., M.J. Buckley, and T.P. Speed, *Analysis of cDNA microarray images.* Brief Bioinform, 2001. **2**(4): p. 341-9.

29. Brown, C.S., P.C. Goodwin, and P.K. Sorger, *Image metrics in the statistical analysis of DNA microarray data.* Proc Natl Acad Sci U S A, 2001. **98**(16): p. 8944-9.

30. Smyth, G.K. and T. Speed, *Normalization of cDNA microarray data.* Methods, 2003. **31**(4): p. 265-73.

31. Smyth, G.K., J. Michaud, and H.S. Scott, *Use of within-array replicate spots for assessing differential expression in microarray experiments.* Bioinformatics, 2005. **21**(9): p. 2067-75.

32. Tuimala, J., *DNA microarray data analysis using Bioconductor* 2008: CSC, IT Center for Science.

33. Sreekumar, J. and K.K. Jose, *Statistical tests for identification of differentially expressed genes in cDNA microarray experiments.* Indian Journal of Biotechnology, 2008. **Volume 7**(October 2008): p. 423 - 436.

34. Steinhoff, C. and M. Vingron, *Normalization and quantification of differential expression in gene expression microarrays.* Brief Bioinform, 2006. **7**(2): p. 166-77.

35. Witten, D.M. and R. Tibshirani, *A comparison of fold-change and the t-statistic for microarray data analysis.* Department of Statistics, Stanford University, 2007.

36. Petrie, A. and C. Sabin, *Medical Statistics at a Glance* 2000: Blackwell Science.

37. Pollard, K.S., et al., *multtest: Resampling-based multiple hypothesis testing.* R package version 2.1.1. http://CRAN.R-project.org/package=multtest.

38. Wu, H., et al., *maanova: Tools for analyzing Micro Array experiments.* R package version 1.14.0. http://research.jax.org/faculty/churchill.

39. Wu, H., et al., *Maanova: A software package for the analysis of spotted cdna microarray experiments.*, in *The Analysis of Gene Expression Data*. 2003, Springer London. p. 313-341.

40. Smyth, G.K., *Limma: linear models for microarray data*, in *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, R. Gentleman, et al., Editors. 2005, Springer. p. 397-420.

41. Smyth, G.K., Y.H. Yang, and T. Speed, *Statistical issues in cDNA microarray data analysis.* Methods Mol Biol, 2003. **224**: p. 111-36.

42. Smyth, G.K., *Linear models and empirical bayes methods for assessing differential expression in microarray experiments.* Stat Appl Genet Mol Biol, 2004. **3**: p. Article3.

43. Pirooznia, M. *Practical Microarray Data Analysis*. February 2008 [cited August 2009]; Microarray data analysis tutorial, based on Limma and LimmaGUI.]. Available from: http://mcbc.usm.edu/pirooznia/microarray-practical/index.htm.

44. Callow, M.J., et al., *Microarray expression profiling identifies genes with altered expression in HDL-deficient mice.* Genome Res, 2000. **10**(12): p. 2022-9.

45. Tusher, V.G., R. Tibshirani, and G. Chu, *Significance analysis of microarrays applied to the ionizing radiation response.* Proc Natl Acad Sci U S A, 2001. **98**(9): p. 5116-21.

46. Tibshirani, R., et al., *samr: SAM: Significance Analysis of Microarrays.* R package version 1.26, http://www-stat.stanford.edu/~tibs/SAM.

47. JCVI. *JCVI Research Software*. [cited; Available from: http://www.jcvi.org/cms/research/software/.

48. Montana. *Microarray Data Analysis*. 2009 [cited; Commented list of existing microarray software analysis tools.]. Available from: http://www.montana.edu/wwwmb/index.php?page=microarray-data-analysis.

49. SMD, *Standford Microarray Database: Microarray Resources: Software and Tools.* 2008.

50. Abreu, L. and Carreiro, João Paulo, *ASP.NET 2.0 curso completo com a colaboração de João Paulo Carreiro*. 2006, Lisboa: FCA - Editora de Informática. XXIV, 714.

# Glossary

**Data set**

In microarray data analysis, a data set refers to a collection of experimental data files. In the Mind application, it is the name of the sub-module where data sets can be managed, and it is an entity, that contains of a collection of data files, to which data analysis results are associated to.

**Dye-swaps**

Dye-swaps refer to a pair of microarray technical replicates with the RNA targets switched. For example, the first array may have the experiment sample on the red channel and the control sample on the green channel. The second array will then have the experiment sample on the green channel and the control sample on the red channel. Dye-swaps are used to account for the higher efficiency of the green fluorescent dye. In a self-self hybridization, which hybridizes the same sample to both array channels, the measured values will indicate that the genes are more expressed on the green sample than on the red same, when they both refer to the same sample, meaning green intensities are higher than they should be. To compensate for this effect, dye-swaps are the most common and effective procedure.

**Exploratory analysis**

Exploratory analysis is the other major application of DNA microarrays for gene expression assessment, besides gene regulation analysis. Gene regulation assessment aims to find genes that are differentially expressed between two or more defined groups. Exploratory analysis aims to define groups by analyzing the experimental data – groups of related samples, groups of related genes. Clustering algorithms play an important role in exploratory analysis as they groups together data based on observed patterns and expression values.

**Fold-change**

A gene that has an intensity 16 times bigger in the experiment sample, when comparing to the control sample, is said to have a **16-fold-change**, or a base two logarithm fold-change of 4. This is the **ratio (log) fold-change** which compares unratioed control and experiment intensities. The

difference fold-change considered M-values, for example, a gene has a **difference log fold-change** of 2 in relation to another array if its M-value is 2 units higher on the first array.

**Gene regulation analysis**

Gene regulation assessment aims to find genes that are differentially expressed between two or more samples, and it may be done using DNA microarrays. DNA microarrays measure the mRNA present in a sample to infer about gene expression levels, and the experimental data may be used to identify regulated genes between the conditions under study.

**Genome**

The genome is the collection of genes present in an organism's DNA.

**Hybridization**

Nucleic acid hybridization occurs when two complementary nucleotide strands bind, establishing the hydrogen bond between each base pair. On the microarray slides the binding of the mRNA targets with the DNA probes forms mRNA/DNA hybrids.

**Probe**

As a verb, to probe a sample for thousands of genes at once means to measure the expression value of thousands of genes within that sample. As a noun, probes are the nucleotide sequences, representative of genes that are spotted, printed or synthesized on the spots of a microarray slide, so a sample can be probed for the expression levels of those genes using the microarray.

**Proteom**

The proteome is the collection of proteins present in a cell, resultant of the expression of genes, among other factors. Gene expression analysis study how the genes relate to the phenotype, proteom ananlysis studies how the proteins reflect in the phenotype. There is not a better way; both are important and complementary methodologies in gene expression analysis.

**Regulated**

A gene that shows different expression values in two or more different RNA samples is differentially expressed or regulated. A gene is **upregulated** in one sample in relation to another when it is more expressed on this sample. A gene is **downregulated** in one sample in relation to another when it is less expressed on this sample. When a gene is simply said to be up- or

downregulated, without comparing it with another sample, it means that it is up- or downregulated in the Experiment sample in relation to the Control sample. Also by convention, upregulated genes are usually plotted red while downregulated genes are usually plotted gene (no relation with the red and green fluorescent dyes).

**Replicate**

**Spot replicates** on a microarray refer to two or more spots that represent the same gene. Identical nucleotide sequences or probes were printed onto all of a spot's replicates, and they will all investigate the target RNA samples for the expression of that gene. Spot replicates should be placed at random locations on the array and not next to each other, for example to avoid bias due to the print-tip. **Microarray replicates** refer to two or more arrays onto which identical mRNA samples were hybridized. Often each two microarray replicates are dye-swaps to account for the different dye efficiency. They can be biological replicates, such as when having three different cultures of yeast, although bred under identical conditions, to extract three RNA samples. Several technical replicates can be extracted from one yeast culture. Both spot and array replicates permit more measurements of a gene's expression, and these help to ensure the gene's expression value, whatever it may be, is not due to chance.

**Target**

Targets are the mRNA samples that are applied over the microarray slides to hybridize with the probe sequences. In two-color arrays, two mRNA samples are applied on one slide, one labeled with a green fluorescent dye and the other labeled with a red fluorescent dye. The **targets information** for one array indicates what mRNA sample was associated to the green channel and what mRNA sample was associated to the red channel.

**Transcriptome**

The transcriptome is the collection of all RNA molecules, including mRNA, present in a cell. DNA microarrays perform transcriptome analysis, measuring the mRNA, in order to infer about gene expression levels in the cell.

# Appendix A – Detailed Microarray Data Analysis Workflow

**[1] Create a Data Set (based on an experiment of the Mind LIMS)**

[1.1] Select the experiment.
[1.2] Select the bioassays of the experiment (at least one).
[1.3] For each selected bioassay, if there is more than one measurement, choose one. A bioassay may have several images, each one with several measurements. Only one measurement is to be chosen for each bioassay. (Each measurement corresponds to one data file.)
[1.4] Define a name to identify this Data Set within the experiment.

**[2] Perform Quality Control**

[2.1] Set the spot types (to color the values in the QC plots accordingly).
  [2.1.1] Use a default spot types definition.
  [2.1.2] Define the spot types.
  [2.1.3] Upload an existing spot types file.
[2.2] Set the QC options.
  [2.2.1] Filtering
    [2.2.1.1] Intensity filter
      [2.2.1.1.1] Define Red and Green intensity minimums
      [2.2.1.1.2] Exclude spot if [both] / [at least one] (choose!) intensities are below the minimum
    [2.2.1.2] S2N ratio filter
      [2.2.1.2.1] Define Red and Green ratio minimums
      [2.2.1.2.2] Exclude spot if [both] / [at least one] (choose!) S2N ratios are below the minimum
  [2.2.2] Background correction method
    [2.2.2.1] Half
    [2.2.2.2] Minimum
    [2.2.2.3] Subtraction
    [2.2.2.4] Norm Exp
    [2.2.2.5] Edwards
    [2.2.2.6] Moving minimum

[2.2.2.7] None

[2.2.3] Normalization method

[2.2.3.1] Median

[2.2.3.2] Robust Spline

[2.2.3.3] Loess

[2.2.3.4] Print-tip Loess

[2.2.3.5] None

[2.2.4] Choose plots (of the raw data, after filtering and background correction)

[2.2.4.1] Plots of background corrected data (filtered spots plotted only):

[2.2.4.1.1] MA Plot

[2.2.4.1.2] Image Plot

[2.2.4.1.3] Boxplot per Print-tip Loess

[2.2.4.1.4] Histograms

[2.2.4.1.5] Plot Printiploess

[2.2.4.1.6] S2N Ratio Imageplot

[2.2.4.2] Plots of normalized data (filtered spots plotted only):

[2.2.4.2.1] MA Plot

[2.2.4.2.2] Image Plot

[2.2.4.2.3] Boxplot per Print-tip Loess

[2.2.4.2.4] Histograms

[2.2.4.2.5] Plot Printiploess

[2.2.4.2.6] S2N Ratio Imageplot

[2.3] When processing is done, view results. The reports and normalized data will be available under the current data set.

**[3] Gene Regulation Assessment**

[3.1] Choose to average gene intensities (normalized red and green unlogged intensity values) by the gene name, specified in the ADF/GAL file, or not.

[3.2] Set the filtering options (optionally), as a combination of:

[3.2.1] Intensity filter

[3.2.1.1] Define Red and Green intensity minimums

[3.2.1.2] Define minimum number of arrays

[3.2.1.3] Choose to select if [both] / [at least one] intensities are above the minimum

[3.2.2] Standard deviation filter

[3.2.2.1] Input SD cutoff value

[3.3] Select the method to assess gene regulation to be applied to the filtered data (only one, they can all be tried, one at a time):

[3.3.1] Fold-change

[3.3.1.1] Divide the red and green channels of the data set by the experiment and control groups (option "none" for channels that belong in none).

[3.3.1.2] Input a log fold-change value (default is 2.0) and apply the fold-change method.

[3.3.1.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.2] t-test

[3.3.2.1] With ratio log fold-changes (like the fold-change method)

[3.3.2.1.1] Divide the red and green channels of the data set by the experiment and control groups (option "none" for channels that belong in none).

[3.3.2.1.2] Input a maximum p-value, a minimum log fold-change, indicate equal variances are assumed or not, the desired multiple comparison correction method and run t-test.

[3.3.2.1.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.2.2] With difference log fold-changes

[3.3.2.2.1] Divide the arrays of the data set by the experiment and control groups (option "none" for channels that belong in none).

[3.3.2.2.2] Input a maximum p-value, a minimum log fold-change, the desired multiple comparison correction method and run t-test.

[3.3.2.2.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.3] t-test paired

[3.3.3.1] Divide the arrays of the data set by the experiment and control groups, forming pairs.

[3.3.3.2] Input a delta value and perform SAM.

[3.3.3.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.4] Limma

[3.3.4.1] Define targets file information if it has not been defined yet.

[3.3.4.2] Choose the comparisons to build the design matrix

[3.3.4.3] Select the number of genes for the top table

[3.3.4.4] In case of multiple groups, specify one comparison to view the empirical Bayes t-statistics for that comparison.

[3.3.4.5] Repeat previous four steps until results are adequate.

[3.3.5] SAM Two-class

[3.3.5.1] Divide the arrays of the data set by the experiment and control groups.

[3.3.5.2] Input a delta value and perform SAM.

[3.3.5.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.6] SAM Two-class paired

[3.3.6.1] Divide the arrays of the data set by the experiment and control groups, forming pairs.

[3.3.6.2] Input a delta value and perform SAM.

[3.3.6.3] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.7] SAM Multi-class

[3.3.7.1] Specify number of groups.

[3.3.7.2] Divide the arrays of the data set by the groups.

[3.3.7.3] Input a delta value and perform SAM.

[3.3.7.4] Repeat the previous step until the number of regulated genes is satisfactory.

[3.3.8] One-Way ANOVA

[3.3.8.1] Specify number of groups.

[3.3.8.2] Divide the arrays of the data set by the groups.

[3.3.8.3] Input a delta value and perform SAM.

[3.3.8.4] Repeat the previous step until the number of regulated genes is satisfactory.

[3.4] Save results. Once the number of genes is satisfactory, choose a name to save the analysis under the Current Data Set. The report and data files will be available for download under that Data Set.

# Appendix B – Table Creation Script

```
CREATE TABLE DataSet (
    id int IDENTITY PRIMARY KEY,
    dataSetName nvarchar(100) NOT NULL,
    experimentIdRef int NOT NULL FOREIGN KEY REFERENCES Experiment(id),
    dataSetDate datetime NULL,
    normDescription nvarchar(100) NULL,
    bgDescription nvarchar(100) NULL
);

CREATE TABLE DataSetItem (
    id int IDENTITY PRIMARY KEY,
    dataSetIdRef int NOT NULL FOREIGN KEY REFERENCES DataSet(id),
    bioAssayDataIdRef int NOT NULL FOREIGN KEY REFERENCES
                                                    BioAssayData(id),
    normFileStoreIdRef int NULL FOREIGN KEY REFERENCES FileStore(id),
    qcReportIdRef int NULL FOREIGN KEY REFERENCES FileStore(id)
);

CREATE TABLE Analysis (
    id int IDENTITY PRIMARY KEY,
    dataSetIdRef int NOT NULL FOREIGN KEY REFERENCES DataSet(id),
    name nvarchar(100) NULL,
    statisticalTest nvarchar(100) NULL,
    dataFileIdRef int NULL FOREIGN KEY REFERENCES FileStore(id),
    reportIdRef int NULL FOREIGN KEY REFERENCES FileStore(id),
    listRegulatedGenesIdRef int NULL FOREIGN KEY REFERENCES FileStore(id)
);

CREATE TABLE TargetsInfo (
    id int IDENTITY PRIMARY KEY,
    dataSetItemIdRef int NOT NULL FOREIGN KEY REFERENCES DataSetItem(id),
    fluorescentDyeName nvarchar(100) NOT NULL,
    sampleName nvarchar(100) NULL
);

CREATE TABLE CurrentDataSet (
    id int IDENTITY PRIMARY KEY,
    submitterIdRef int NOT NULL FOREIGN KEY REFERENCES Submitter(id),
    dataSetIdRef int NOT NULL FOREIGN KEY REFERENCES DataSet(id)
);

CREATE TABLE QCParameters (
    id int IDENTITY PRIMARY KEY,
    submitterIdRef int NOT NULL FOREIGN KEY REFERENCES Submitter(id),
    data datetime NULL,
    name nvarchar(100) NULL,
    qcParameters nvarchar(1000) NULL
);
```

# Appendix C – Stored Procedures

### Delete Analysis

```
CREATE PROC DeleteAnalysis
    @AnalysisID int
AS
BEGIN

    SET NOCOUNT ON;

    DECLARE @dataFile int
    SET @dataFile = (SELECT Analysis.dataFileIdRef
             FROM Analysis WHERE Analysis.id = @AnalysisID)

    DECLARE @reportFile int
    SET @reportFile = (SELECT Analysis.reportIdRef
            FROM Analysis WHERE Analysis.id = @AnalysisID)

    DECLARE @listRegGenesFile int
    SET @listRegGenesFile = (SELECT Analysis.listRegulatedGenesIdRef
            FROM Analysis WHERE Analysis.id = @AnalysisID)

    if exists (select Analysis.name
            from Analysis where Analysis.id = @AnalysisID)
    begin
        delete from Analysis from Analysis.id = @AnalysisID
        delete from FileStore where FileStore.id=@dataFile
        delete from FileStore where FileStore.id=@reportFile
        delete from FileStore where FileStore.id=@listRegGenesFile
    end
END
```

### Delete Data Set

```
CREATE PROC DeleteDataSet
    @deleteDSid int
AS
BEGIN

    SET NOCOUNT ON;

    -- delete all analysis:
    DECLARE cursorAnalysis CURSOR FOR
    SELECT id FROM Analysis WHERE dataSetIdRef = @deleteDSid
    DECLARE @analysisID int

    OPEN cursorAnalysis
    FETCH NEXT FROM cursorAnalysis INTO @analysisID
    WHILE (@@fetch_status <> -1)
    BEGIN
    IF (@@fetch_status = 0) -- FETCH statement successful:
    BEGIN
    -- delete that analysis:
        EXEC DeleteAnalysis @analysisID        END
        FETCH NEXT FROM cursorAnalysis INTO @analysisID
     END
    CLOSE cursorAnalysis
    DEALLOCATE cursorAnalysis

    -- delete targets info (if existing), normalized data and qc report files:
    DECLARE cursorDSI CURSOR FOR
    SELECT id FROM DataSetItem WHERE dataSetIdRef = @deleteDSid
    DECLARE @dsitemid int

    OPEN cursorDSI
    FETCH NEXT FROM cursorDSI INTO @dsitemid
    WHILE (@@fetch_status <> -1)
    BEGIN
        IF (@@fetch_status = 0) -- FETCH statement successful:
        BEGIN
            delete from TargetsInfo
                where dataSetItemIdRef = @dsitemid

            -- get normalized data file and qc report file:
            DECLARE @normdatafile int
            DECLARE @qcreportfile int
            SET @normdatafile = (SELECT normFileStoreIdRef from DataSetItem
            where id = @dsitemid)
            SET @qcreportfile = (SELECT qcReportIdRef from DataSetItem where
            id = @dsitemid)
            -- delete entry in DataSetItem table
            delete from DataSetItem where DataSetItem.id = @dsitemid
            -- delete normalized data file and qc report file:
            delete from FileStore where FileStore.id = @qcreportfile
            delete from FileStore where FileStore.id = @normdatafile
        END
        FETCH NEXT FROM cursorDSI INTO @dsitemid
    END
    CLOSE cursorDSI
    DEALLOCATE cursorDSI

    -- if current data set:
    delete from CurrentDataSet where dataSetIdRef  = @deleteDSid

    delete from DataSet where id = @deleteDSid
END
```

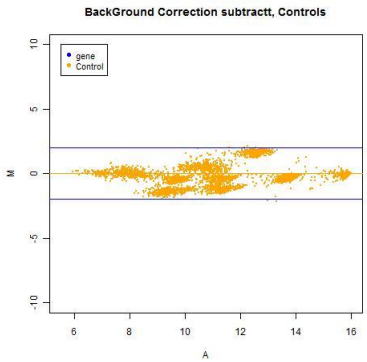
# Appendix D – Example QC Report

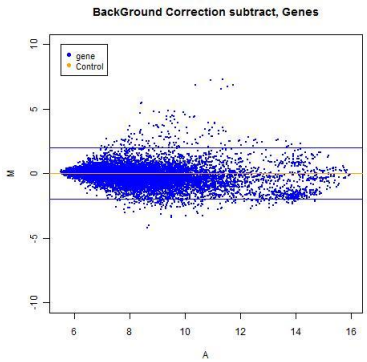
| Quality Control Report | | | |
|---|---|---|---|
| **Data Set** | | | |
| Data Set Name: | novo data set | | |
| Experiment Name: | My Heat Shock Experiment | | |
| **Data Set Items** | | | |
| **BioAssay** | **Image** | **Measurement** | **Raw File Name** |
| HS1 | img_HS1 | HS1 | HS1.txt |
| HS1_ds | HS1_ds | HS1_ds | HS1_ds.txt |
| HS2 | HS2 | HS2 | HS2.txt |
| HS2_ds | HS2 | HS2_ds | HS2_ds.txt |
| HS3 | img HS3 | measurement HS3 | HS3.txt |
| HS3_ds | img HS3_ds | measurment HS3 ds | HS3_ds.txt |
| **Quality Control Parameters** | | | |
| Background correction method | subtract | | |
| Normalization method: | printtiploess | | |
| Filtering done for data plots: | | | |
| Intensity filter: | none | | |
| S2N ratio filter: | none | | |

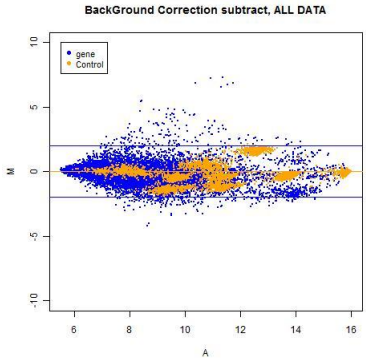
**QC report related to the raw data file: HS1.txt**

## Background Corrected Data Plots

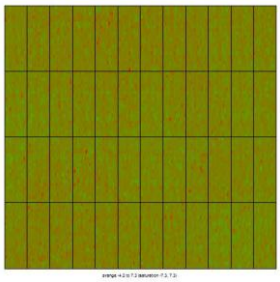
**MA Plots**



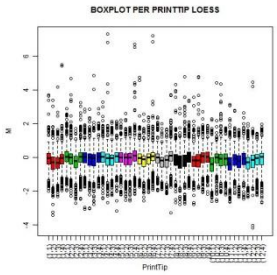






## Image Plot

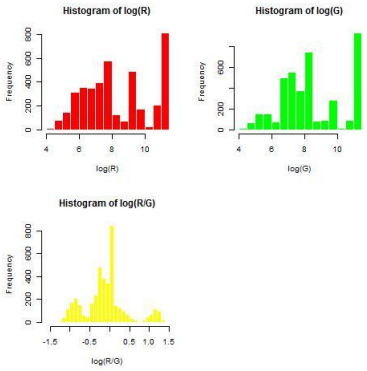


## Boxplot per print-tip loess

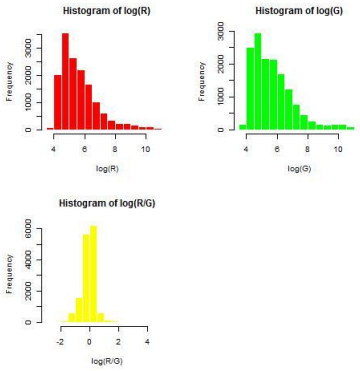



## Histograms

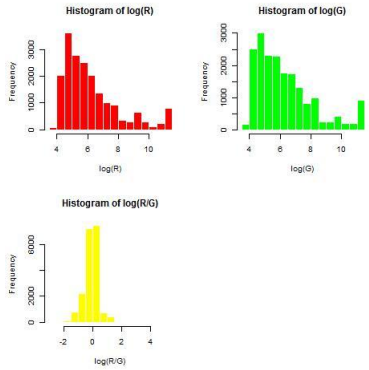
### Just Controls



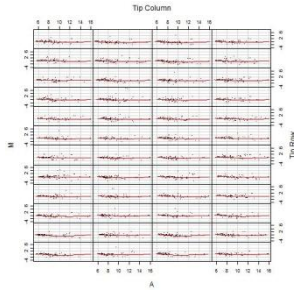

### Just Genes



## All Data



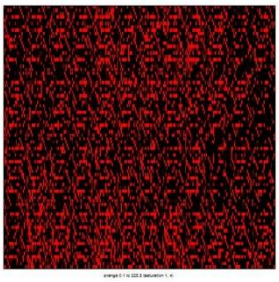

## Plot print-tip loess




## S2N Ratio Image Plots

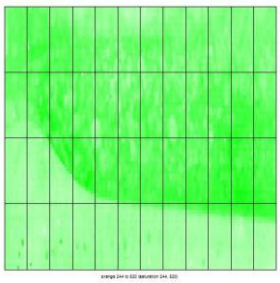
### S2N Ratio Green

### S2N Ratio Red

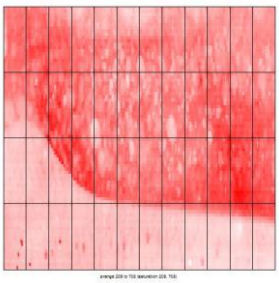


## Background Images

### Background Green



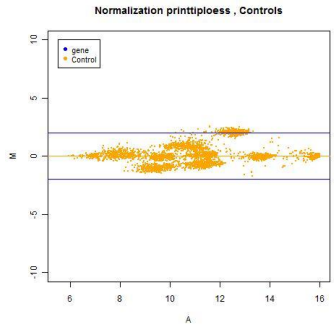
### Background Red



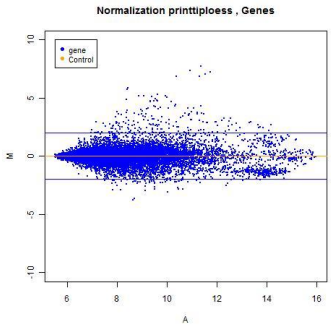
# Normalized Data Plots

## MA Plots - Normalize All Data

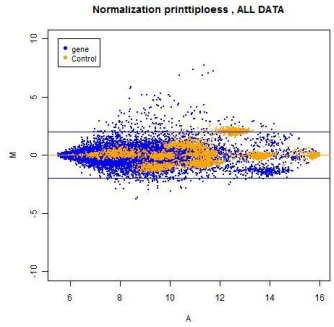
### Controls




### Genes

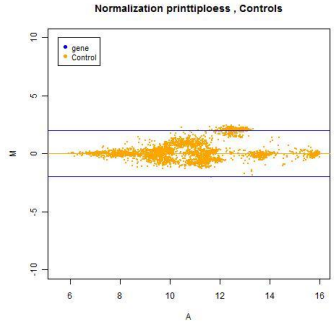



### All data

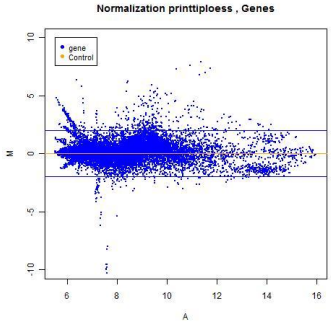



## MA Plots - Normalize Controls

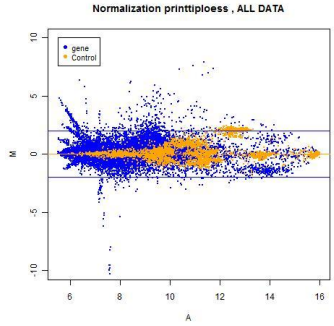
### Controls




### Genes

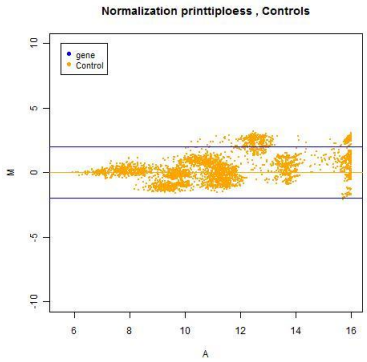



### All data

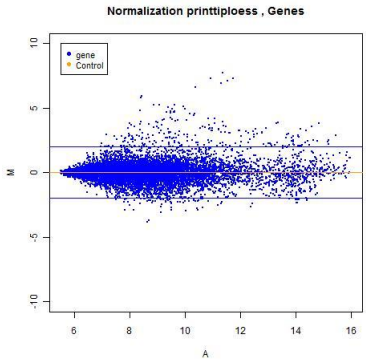


## MA Plots - Normalize Genes

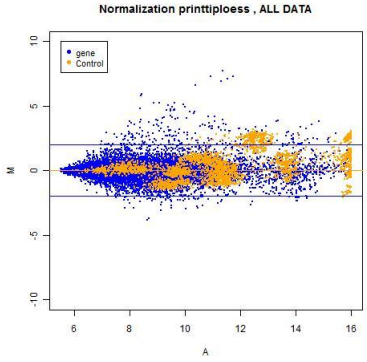
### Controls



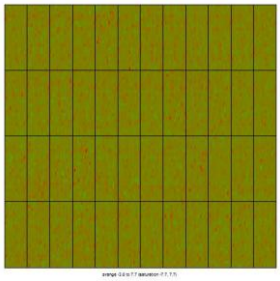

### Genes



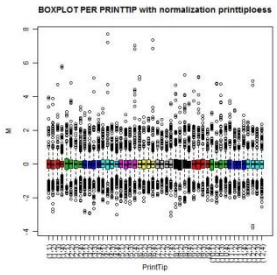

### All data




## Image Plot

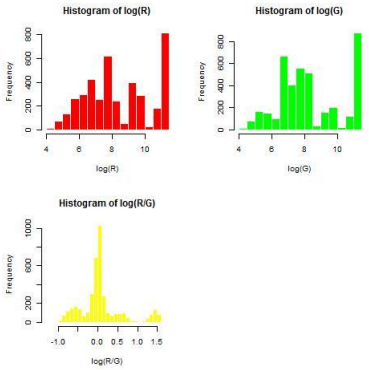


## Boxplot per print-tip loess

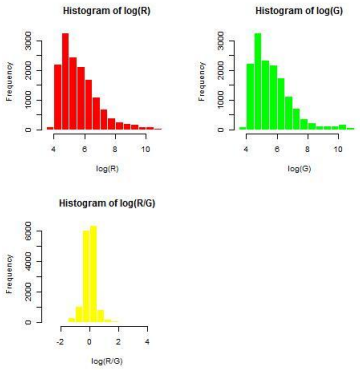


## Histograms

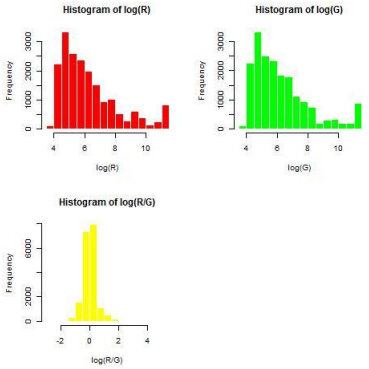
### Just Controls



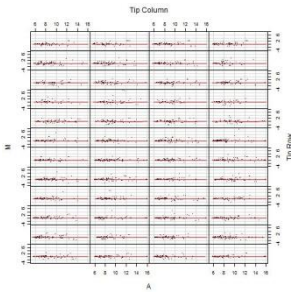

### Just Genes




### All Data



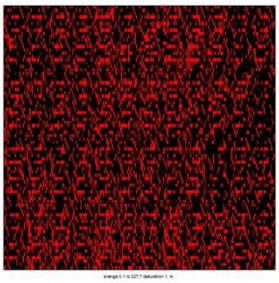

## Plot print-tip loess




## S2N Ratio Image Plots

### S2N Ratio Green

### S2N Ratio Red

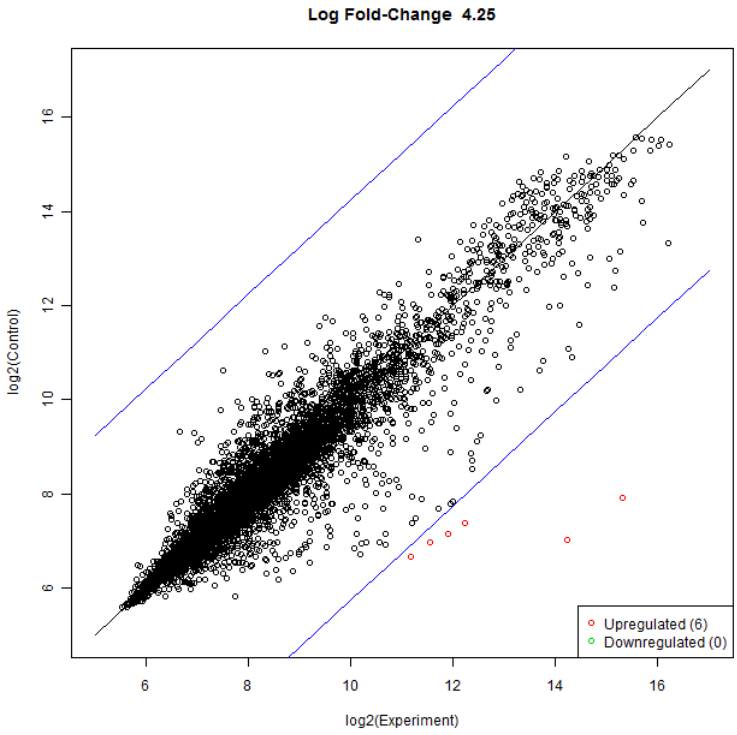


# Appendix E – Example Analysis Report

| Gene Regulation Assessment Report | | | |
|---|---|---|---|
| **Data Set** | | | |
| Data Set Name: | novo data set | | |
| Experiment Name: | My Heat Shock Experiment | | |
| **Data Set Items** | | | |
| **BioAssay** | **Image** | **Measurement** | **Raw File Name** |
| HS1 | img_HS1 | HS1 | HS1.txt |
| HS1_ds | HS1_ds | HS1_ds | HS1_ds.txt |
| HS2 | HS2 | HS2 | HS2.txt |
| HS2_ds | HS2 | HS2_ds | HS2_ds.txt |
| HS3 | img HS3 | measurement HS3 | HS3.txt |
| HS3_ds | img HS3_ds | measurment HS3 ds | HS3_ds.txt |
| **Quality Control Parameters** | | | |
| Background correction method: | subtract | | |
| Normalization method: | printtiploess | | |
| **Targets Information** | | | |
| **BioAssay** | | **Cy3 (Green)** | **Cy5 (Red)** |
| HS1 | | c | hs |
| HS1_ds | | hs | c |
| HS2 | | c | hs |
| HS2_ds | | hs | c |
| HS3 | | c | hs |
| HS3_ds | | hs | c |

| **Gene Regulation Assessment Method: Fold-change** | |
|---|---|
| Minimum ratio log fold-change: | 4.25 |
| Control group: | HS1/Cy3, HS1_ds/Cy5, HS2/Cy3, HS2_ds/Cy5, HS3/Cy3, HS3_ds/Cy5 |
| Experiment group: | HS1/Cy5, HS1_ds/Cy3, HS2/Cy5, HS2_ds/Cy3, HS3/Cy5, HS3_ds/Cy3 |

## Fold-change Plot




## Regulated Genes

| GeneNr | Name | FoldChange |
|---|---|---|
| 1770 | YFL014W | 7.38439106031169 |
| 3059 | YBR072W | 7.19930613301369 |
| 349 | YLR178C | 4.85621267908528 |
| 4074 | YMR169C | 4.73215446551329 |
| 1461 | YPR160W | 4.57364120675199 |
| 4426 | YGR248W | 4.48778091184962 |