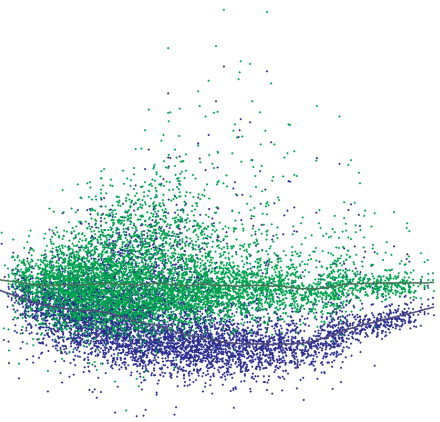

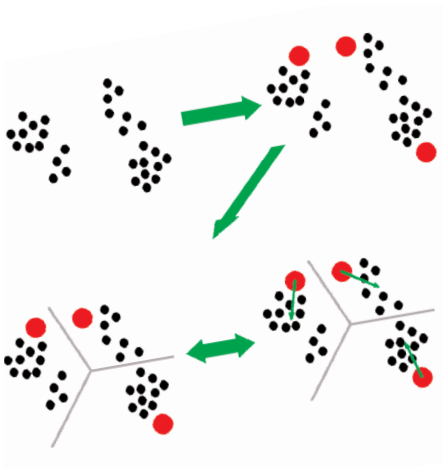

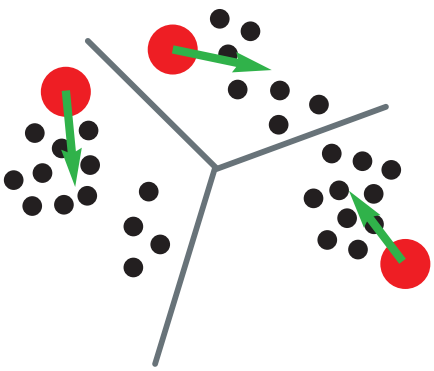

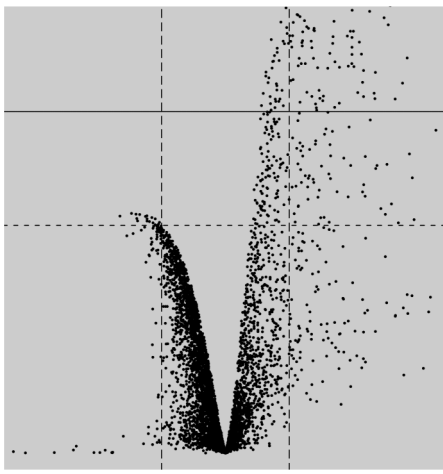

DNA Microarray
Data Analysis

TOMI PASANEN, JANNA SAARELA, ILANA SAARIKKO, TEEMU TOIVANEN,
MARTTI TOLVANEN, MAUNO VIHINEN AND GARRY WONG
EDITORS JARNO TUIMALA AND M. MINNA LAINE
CSC

DNA Microarray Data Analysis

DNA Microarray Data Analysis

Editors
Jarno Tuimala
M. Minna Laine

CSC, the Finnish IT center for Science

CSC – Scientific Computing Ltd. is a non-profit organization for high-performance computing and networking in Finland. CSC is owned by the Ministry of Education. CSC runs a national large-scale facility for computational science and engineering and supports the university and research community. CSC is also responsible for the operations of the Finnish University and Research Network (Funet).

ISBN 952-9821-89-1

http://www.csc.fi/oppaat/siru/

Printed at
Picaset Oy
Helsinki 2003

Preface

This is the first edition of the DNA microarray data analysis guidebook. Although invented in the mid-90s, DNA microarrays are still novelties as biomedical research tools. DNA microarrays generate large amounts of numerical data, which should be analyzed effectively.

In this book, we hope to offer a broad view of basic theory and techniques behind the DNA microarray data analysis. Our aim was not to be comprehensive, but rather to cover the basics, which are unlikely to change much over years. We hope that especially researchers starting their data analysis can benefit from the book.

The text emphasizes gene expression analysis. Topics, such as genotyping, are discussed shortly. This book does not cover the wet-lab practises, such as sample preparation or hybridization. Rather, we start when the microarrays have been scanned, and the resulting images analyzed. In other words, we take the files with signal intensities, which usually generate questions such as: "How is the data normalized?" or "How do I identify the genes which are upregulated?". We provide some simple solutions to these specific questions and many others.

Each chapter has a section on suggested reading, which introduces some of the relevant literature. Several chapters also include data analysis examples using GeneSpring software.

This edition of the book was written by M. Minna Laine (chapters 4, 8 and 14), Tomi Pasanen (chapter 11), Janna Saarela (chapters 2 and 3), Ilana Saarikko (chapter 8), Teemu Toivanen (chapter 14), Martti Tolvanen (chapter 12), Jarno Tuimala (chapters 4, 6, 7, 8, 9, 10, 13 and 15), Mauno Vihinen (chapters 10, 11 and 12), and Garry Wong (chapters 1 and 5).

Juha Haataja and Leena Jukka are warmly acknowledged for their support during the production of this book.

We are very interested in receiving feedback about this publication. Especially, if you feel that some essential technique has been missed, let us know. Please send your comments to the e-mail address Jarno.Tuimala@csc.fi.

Espoo, 19th May 2003

The authors

List of Contributors

M. Minna Laine
CSC, the Finnish IT center for Science
Tekniikantie 15 a D
02101 Espoo
Finland

Tomi Pasanen
Institute of Medical Technology
Lenkkeilijänkatu 8
33520 Tampere
Finland

Janna Saarela
Biomedicum Biochip Center
Haartmaninkatu 8
00290 Helsinki
Finland

Ilana Saarikko
Centre for Biotechnology
Tykistökatu 6
20521 Turku
Finland

Teemu Toivanen
Centre for Biotechnology
Tykistökatu 6
20521 Turku
Finland

Martti Tolvanen
Institute of Medical Technology
Lenkkeilijänkatu 8
33520 Tampere
Finland

Jarno Tuimala
CSC, the Finnish IT center for Science
Tekniikantie 15 a D
02101 Espoo
Finland

Mauno Vihinen
Institute of Medical Technology
Lenkkeilijänkatu 8
33520 Tampere
Finland

Garry Wong
A. I. Virtanen -institute
University of Kuopio
70211 Kuopio
Finland

Contents

Part I

Introduction

1 Introduction

Microarray technologies as a whole provide new tools that transform the way scientific experiments are carried out. The principle advantage of microarray technologies compared with traditional methods is one of scale. In place of conducting experiments based on results from one or a few genes, microarrays allow for the simultaneous interrogation of hundreds or thousands of genes.

1.1 Why perform microarray experiments?

The answers to this question span a wide range from the formally considered, well-constructed hypotheses with elegant supporting arguments to "I can't think of anything else to do, so lets's do a microarray experiment". The true motivation for performing these experiments lies likely somewhere between the two extremes. It is this combination of generating a scientific hypothesis (elegant or not), and at the same time being able to produce massive amounts of data that has made research in microarrays so attractive. Nonetheless, the production and use of microarrays is set with high technical and instrumentation demands. Moreover, the computation and statistical requirements for dealing with the data can be daunting, especially to those scientists used to single experiment – single result analysis. So, for those willing to try this new technology, microarray experiments are performed to answer a wide range of biological questions to which the answers are to be found in the realm of hundreds, thousands, or an entire genome of individual genes.

1.2 What is a microarray?

Microarrays are microscope slides that contain an ordered series of samples (DNA, RNA, protein, tissue). The type of microarray depends upon the material placed onto the slide: DNA, DNA microarray; RNA, RNA microarray; protein, protein microarray; tissue, tissue microarray. Since the samples are arranged in an ordered fashion, data obtained from the microarray can be traced back to any of the samples. This means that genes on the microarray are addressable. The number of ordered samples on a microarray can number into the hundred of thousands. The typical microarray contains several thousands of addressable genes.

The most commonly used microarray is the DNA microrray. The DNA printed or spotted onto the slides can be chemically synthesized long oligonucleotides or enzymatically generated PCR products. The slides contain chemically reactive groups (typically aldehydes or primary amines) that help to stabilize the DNA

onto the slide, either by covalent bonds or electrostatic interactions. An alternative technology allows the DNA to be synthesized directly onto the slide itself by a photolithographic process. This process has been commercialized and is widely available. DNA microarrays are used to determine

1. The expression levels of genes in a sample, commonly termed expression profiling.

2. The sequence of genes in a sample, commonly termed minisequencing for short nucleotide reads, and mutation or SNP analysis for single nucleotide reads.

1.3 Microarray production

Printing microarrays in not a trivial task and is both an art and a science. The job requires considerable expertise in chemistry, engineering, programming, large project management, and molecular biology. The aim during printing is to produce reproducible spots with consistant morphology. Early versions of printers were custom made with a basic design taken from a prototype version. Some were built from laboratory robots. Current commercial microarray printers are available for almost every size application and have made the task of printing microarrays feasible and affordable for many nonspecialized laboratories. The basic printer consists of: a nonvibrating table surface where the raw glass slides are placed, a moving head in the x-y-z plane that contains the pins or pens to transfer the samples onto the array, a wash station to clean the pins/pens between samples, a drying station for the pins/pens, a place for the samples to be printed, and a computer to control the operation. Some of these procedures can be automated such as replacing the samples to be printed, although most complete systems are semi-automated. Samples to be printed are concentrated and stored in microtitre plates.

The printers are operated in dust-free, temperature and humidity controlled rooms. Some printer designs have their own self-contained environmental controls. Printing pen designs have been adapted from ink methods and include quill, ball-point, ink-jet, and P-ring techniques. The pens can get stuck and need to be cleaned frequently. Multiple pens placed on a printing head can multiplex the printing operation and speed up the process. Thousands of samples in duplicate or triplicate are printed in a single run over perhaps a hundred or more slides; thus, printing times of several days are common. Since printing times can be long and sample volumes are small, sample evaporation is a major concern. As a result, hygroscopic printing buffers, often containing DMSO have been developed and are highly useful to alleviate evaporation. A typical printer design is shown in Figure 1.1.

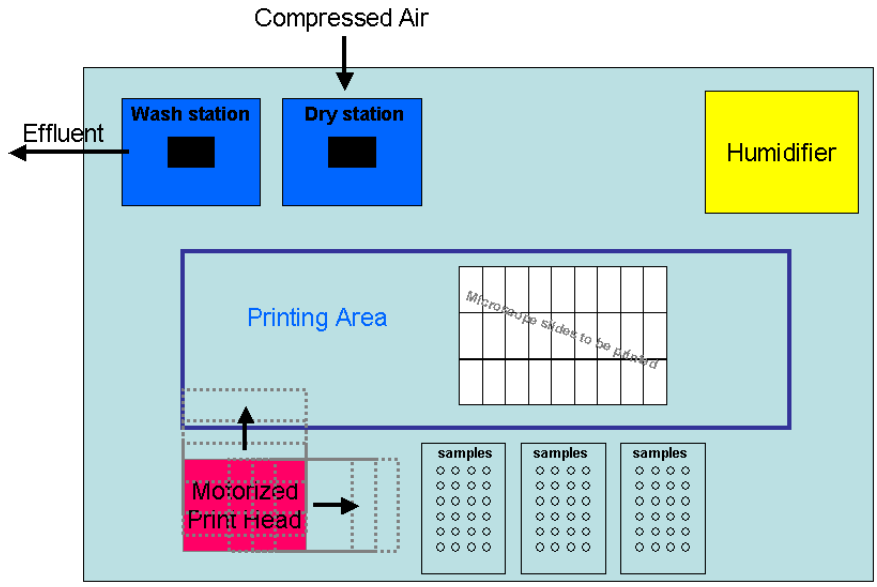

Figure 1.1: *Typical picture of a printer.*

1.4 Where can I obtain microarrays?

Microarrays can be obtained from a variety of sources. Commercial microarrays are of high quality, good density and available for the most commonly studied organisms including human, mouse, rat, and yeast. Many companies also have specialized arrays available for more focused investigations. Also microarray core facilities located in academic and government institutions produce microarrays that are available for use (Table 1.1). The microarrays from these centers have definite cost advantages and excellent quality standards, but tend to have a more restricted focus in terms of the types of arrays and species covered. Custom microarrays that are designed and produced for individual or limited use are a growing application. Custom fabrication of a microarray still requires a microarray printer or spotter facility to produce the array, but has the advantage of producing smaller batches of slides and more flexibility in the genes on array. If you are fortunate to have your own facility, microarrays can be produced at your own convenience. Microarrays once made store well in dark dessicated plastic slide boxes. Some manufacturers suggest storage at $-20\,^{\circ}\mathrm{C}$ while others find room temperature adequate. The shelf life of microarrays has been claimed to be up to 6 months although this has not been empiracally tested.

Table 1.1: *Places where microarrays can be obtained.*

Center name	Website
Biomedicum Biochip Center	www.helsinki.fi/biochipcenter
Finnish DNA microarray Center	microarrays.btk.utu.fi
Norwegian Microarray Consortium	www.med.uio.no/dnr/microarray/
Ontario Cancer Institute	www.microarrays.ca

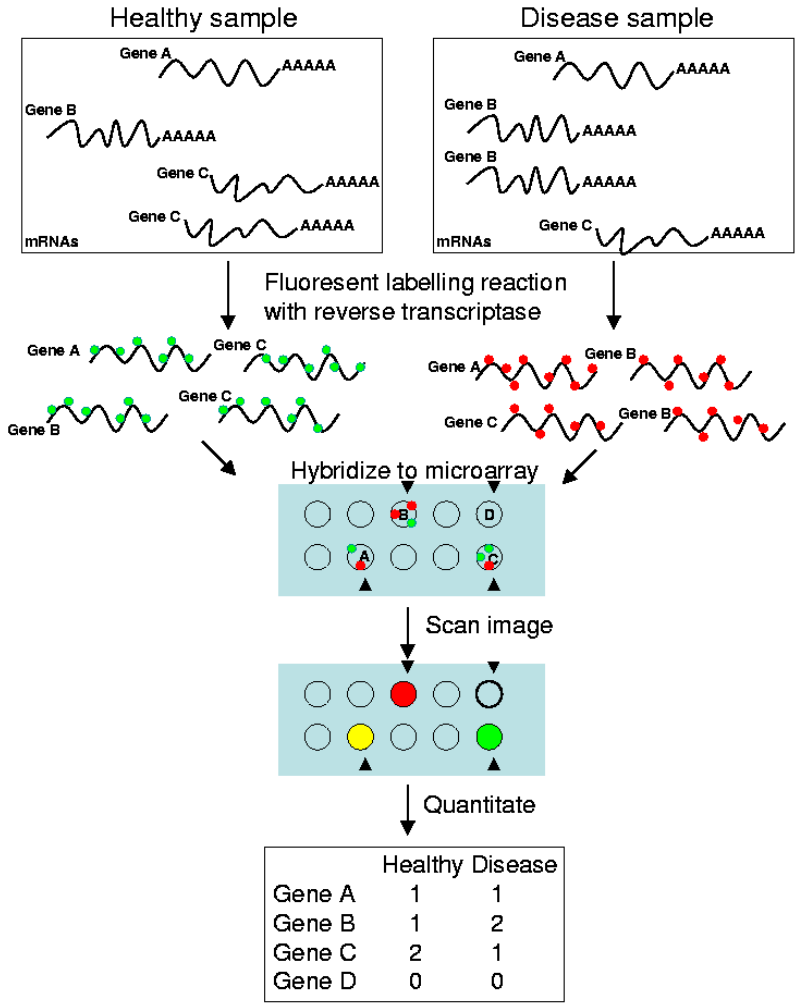

Figure 1.2: *Work flow of a typical expression microarray experiment.*

1.5 Extracting and labeling the RNA sample

A typical workflow of the microarray experiment has been summarized in Figure 1.2. Once microarrays have been made and obtained, the next stage is to obtain samples for labeling and hybridization. Labeling RNA for expression analysis generally involves three steps:

1. Isolation of RNA.
2. Labeling the RNA by a reverse transcription procedure with fluorescent markers.
3. Purification of the labeled products.

RNA can be extracted from tissue or cell samples by common organic extraction procedures used in most molecular biology labs. Many commercial kits are available for this task. Both total RNA and mRNA can be used for labeling, but the contaminating genomic DNA must be removed by DNAase treatment. The amount of total RNA necessary for a single labeling reaction is about 20 μg while the amount of mRNA necessary is about 0.5 μg. Lesser amounts are known to work, but require extreme purity and well developed protocols. Thus, while the absolute amounts may vary, the purity and integrity of the RNA is an absolute must. It is generally a good idea to check the RNA samples before using them in microarray experiments. In fact, for many core facilities it is a requirement. This can be done by assaying the absorption ratio 260/280 lambda and/or running a sample on an ethidium bromide stained agarose gel.

Direct labeling of the RNA is achieved by producing cDNA from the RNA by using the enzyme reverse transcriptase and then incorporating the fluorescent labels, most commonly Cy3 and Cy5 . Other fluorophores are available (*e.g.* Cy3.5, TAMRO, Texas red) but have not yet found widespread use. In the indirect procedure, a reactive group, usually a primary amine, is incorporated into the cDNA first, and the Cy3 or Cy5 is then coupled to the cDNA in a separate reaction. The advantage of the indirect method is a higher labeling efficiency due to the incorporation of a smaller molecule during the reverse transcription step. Once fluorescently labeled probes are made, the free unincorporated nucleotides must be removed. This is typically done by column chromatography using convenient spin-columns or by ethanol precipitation of the sample. Some protocols perform both purification steps. As a small aside, radioactivity is still around and may even make a comeback in microarrays. Incorporation of ^{33}P- or ^{35}S-labeled nucleotides into cDNAs have high rates and provide more sensitivity than fluoresecently labeled probes. These features have been exploited in plastic microarrays that have the gene density and size of microarrays but require far less instrumentation and are reusable.

1.6 RNA extraction from scarse tissue samples

For many microarray applications there is a scarcity of tissue available for RNA extraction. This seems to be the case especially in human tissue studies. In response, many scientists have developed techniques aimed at getting around this

problem. These procedures generally involve either PCR amplification of the cDNAs made from the original RNAs, or production of more RNA from the original RNA sample by hybridization of a T7 or T3 promoter followed by RNA sythesis with RNA polymerase. As usual for any amplification procedure, proper controls and interpretation of the results need to be considered.

A related issue in isolating tissues for microarray studies is the dissection of small populations of cells or even single cells. Sophisticated instruments have been developed for this application and many are commercially available. These laser-assisted microdissection machine, while expensive, are nonetheless fairly straightforward to use and provide a convenient method for obtaining pure cell samples.

1.7 Hybridization

Conditions for hybridizing fluorescently labeled DNAs onto microarrays are remarkably similar to hybridizations for other molecular biology applications. Generally the hybridization solution contains salt in the form of buffered standard sodium citrate (SSC), a detergent such as sodium dodecyl sulphate (SDS), and nonspecific DNA such as yeast tRNA, salmon sperm DNA, and/or repetitive DNA such as human Cot-1. Other nonspecific blocking reagents used in hybridization reactions include bovine serum albumin or Denhardt's reagent. Lastly, the hybridization solution should contain the labeled cDNAs produced from the different RNA populations.

Hybridization temperatures vary depending upon the buffers used, but generally are performed at approximately $15-20\,^{\circ}\mathrm{C}$ below the melting temperature, which is $42-45\,^{\circ}\mathrm{C}$ for PCR products in 4X SSC and $42-50\,^{\circ}\mathrm{C}$ for long oligos. Hybridization volumes vary widely from 20μl to several mLs. For small hybridisation volumes, hydrophobic cover slips are used. For larger volumes, hybridization chambers can be used.

Hybridization chambers are necessary to keep the temperature constant and the hybridization solution from evaporating. Hybridization chambers vary substantially from the most expensive high-tech automated instruments to empty pipette boxes with a few wet paper towels inserted. The range of solutions for providing a thermally stable, humidified enviroment for a microscope slide is only virtually unlimited. Some might even consider a sauna as a potential chamber. In small volumes, the hybridization kinetics are rapid so a few hours can yield reproducible results, although overnight hybridizations are more common.

1.8 Scanning

Following hybridization, microarrays are washed for several minutes in decreasing salt buffers and finally dried, either by centrifugation of the slide, or a rinse in isopropanol followed by quick drying with nitrogen gas or filtered air. Fluorescently labeled microarrays can then be "read" with commercially available scanners. Most microarray scanners are basically scanning confocal microscopes with lasers exciting at wavelengths specifically for Cy3 and Cy5, the typical dyes used in experiments. The scanner excites the fluorescent dyes present at each spot on the

microarray and the dye then emits at a characteristic wavelength that is captured in a photomultiplier tube. The amount of signal emitted is directly in proportion to the amount of dye at the spot on the microarray and these values are obtained and quantitated on the scanner. A reconstruction of the signals from each location on the microarray is then produced. For cDNA microarrays one intensity value is generated for the Cy3 and another for the Cy5. Hence, cDNA microarrays produce two-color data. Affymetrix chips produce one-color data, because only one mRNA sample is hybridized to every chip (see chapter 3). When both dyes are reconstructed together, a composite image is generated. This image produces the typical microarray picture.

1.9 Typical research applications of microarrays

The types and numbers of applications for microarray experiments are quite variable and constantly increasing. Microarrays used to monitor the expression level of genes in comparison between two conditions remains one of the most widespread uses of microarrays. This type of study, termed gene expression profiling, can be used to determine the function of particular genes during a particular state, such as nutrition, temperature, or chemical environment. Such results could be observed as up- or down-regulation, or unchanged during particular conditions. For example, a group of genes could be up-regulated during heat shock, and as a group, these genes could be assigned as heat shock responsive genes. Some genes in this group may have already been identified as heat shock responsive, but other genes in the group may not have been assigned any function. Based on a similar response to heat shock, new functions are then assigned to the genes. Therefore, extrapolation of function based on common changes in expression remains one of the most widespread applications of microarray research. By assumption, genes that share common regulatory patterns also share the same function.

In agriculture, microarrays have been used to identify genes which are involved in the ripening of tomatos, for example. In this type of study, RNAs are isolated from raw and ripened fruit, and then compared to determine which genes are expressed during the process. Genes which are down-regulated during the ripening may also provide useful information about the process.

On a basic scientific level, microarrays have been used to map the cellular, regional, or tissue-specific localization of genes and their respectively encoded proteins. Microarrays have been used: at the subcellular level to map genes that encode membrane or cytosolic proteins; at the cellular level to map genes that distinguish between different types of immune cells; at the tissue region level to disinguish genes which encode hippocampus or cortex brain region specific proteins; and at the tissue level to identify genes which are expressed in muscle, liver, or heart tissues.

Pharmacological studies have also used microarrays as a means of discerning the mechanism of action of therapeutic agents and as a corrollary to develop new drug targets. The guiding principle in this endeavor is that genes regulated by therapeutic agents result from the actions of the drug. Identification of the genes that are regulated by a certain drug could potentially provide insight into the mechanism

of action of the drug, prediction of toxicologic properties, and new drug targets.

One of the most exciting areas of application is the diagnosis of clinically relevent diseases. The oncology field has been especially active and to an extent successful in using microarrays to differentiate between cancer cell types. The ability to identify cancer cells based on gene expression represents a novel methodology that has real benefits. In difficult cases where a morphological or an antigen marker is not available or reliable enough to distinguish cancer cell types, gene expression profiling using microarrays can be extremely valuable. Programs to predict clinical outcome and to design individual therapies based on expression profiling results are well underway.

A very recent application of microarrays has been to perform comparative genomic analysis. Genome projects are producing sequences on a massive level, yet there still does not exist sufficient resources to sequence every organism that seems interesting or worthy of the effort. Therefore, microarrays have been used as a shortcut to both characterize the genes within an organism (structural genomics) and also to determine whether those genes are expressed in a similar way to a reference organism (functional genomics). A good example of this is in the species *Oryza sativa* (rice). Microarrays based on rice sequences can be used to hybridize cDNAs derived from other plant species such as corn or barley. The genome sizes in the latter are simply too large for whole genome projects, so hybridization with microarrays to rice genes presents an agile way to address this question.

Single nucleotide polymorphism (SNP) microarrays are designed to detect the presence of single nucleotide differences between genomic samples. SNPs occur at frequencies of approximately 1 in a 1000 bases in humans and underlie the genomic differences between individuals. Mapping and obtaining frequencies of identified SNPs should provide a genetic basis for identifying disease genes, predicting effects of the environment as well as responses to therapeutic agents. Minisequencing, primer extension, and differential hybridization methods have been developed on the microarray platform with all the advantages of expression arrays: high throughput, reproducibility, economy, and speed.

Indeed, the use of microarrays to determine whether a gene is present and whether it goes up or down under certain conditions will continue to spawn even more applications that now depend only upon the imagination of the microarray researcher.

1.10 Experimental design and controls

Good experimental design in a microarray project requires the same principles and practices that are part of any scientific investigation. Appropriate controls are the foundation to any experiment. Both positive controls and negative controls can provide confidence in the results and even provide insight into the success or failure of the experimental protocol. Sufficient replicates should be planned to decrease experimental error and to provide statistical power. Forethought and consultation on the correct statistical practices and procedures for the design are always advantageous. Attention to experimental parameters is a must, so that whatever treatment, time, dose, individual, or tissue location is being studied, the results will be

interpretable with minimum number of confounders. Indeed, probably the only difference between good experimental design in microarray and other experiments is that the time budgeted for data analysis seems always to be underestimated. As a final suggestion, attention to the mundane but critical statistical and data analysis elements of a microarray experiment will greatly increase your ratio of joy to pain at the end of your microarray journey.

1.11 Suggested reading

1. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., Vingron, M. (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat. Genet. 29, 365-371.

2. Brown, P. O., and Botstein, D. (1999) Exploring the new world of the genome with DNA microarrays. Nat. Genet. 21, 33-37.

3. Chee, M., Yang, R., Hubbell, E., Berno, A., Huang, X. C., Stern, D., Winkler, J., Lockhart, D. J., Morris, M. S., Fodor, S. P. (1996) Accessing genetic information with high-density DNA arrays. Science 274, 610-614.

4. Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. U S A 95, 14863-14868.

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

6. Hacia, J. G., Makalowski, W., Edgemon, K., Erdos, M. R., Robbins, C. M., Fodor, S. P., Brody, L. C., Collins, F. S. (1998) Evolutionary sequence comparisons using high-density oligonucleotide arrays. Nat. Genet. 18, 155-158.

7. Humpherys, D., Eggan, K., Akutsu, H., Friedman, A., Hochedlinger, K., Yanagimachi, R., Lander, E. S., Golub, T. R., Jaenisch, R. (2002) Abnormal gene expression in cloned mice derived from embryonic stem cell and cumulus cell nuclei. Proc. Natl. Acad. Sci. U S A 99, 12889-94.

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

9. Pastinen, T., Kurg, A., Metspalu, A.,Peltonen, L., Syvanen, A-C. (1997) Minisequencing: a specific tool for DNA analysis and diagnostics on oligonucleotide arrays. Genome Res. 7, 606-614.

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

This chapter was written by Garry Wong.

2 Affymetrix Genechip system

2.1 Affymetrix technology

There are two principal differences between the Affymetrix Genechip system and "traditional" cDNA microarrays in studying gene expression. First, instead of hybridizing two RNAs labeled with different fluorophores competitively on one cDNA microarray, a single RNA is hybridized on the array in the Affymetrix system, and the comparisons are then made computationally. Second, in the Affymetrix arrays each gene is represented as a probe set of 10–25 oligonucleotide pairs instead of one full length or partial cDNA clone. The oligonucleotide pair (probe pair) comprises of one oligonucleotide perfectly matching to the gene sequence (Perfect Match, PM) and a second oligonucleotide having one nucleotide mismatch in the middle of it (Mismatch, MM) . Probes are designed within 500 base pairs of the 3' end of each gene to hybridize uniquely in the same, predetermined hybridization conditions. Some housekeeping genes are represented as three probe sets, one set designed to the 5' end of the gene, second set to the middle of the gene and the third to the 3'end. In addition to species specific genes, some spiked-in control probe sets are introduced to facilitate the controlling of the hybridization. In the experiment, biotin-labeled RNA is hybridized to the array, which is stained with phycoerythrin-conjugated streptavidin after washing and scanned with the Gene Array Scanner. A grid is automatically laid over the array image and the intensities of each probe pair are used to make expression measurements with the Affymetrix Microarray Suite , version 5 software.

2.2 Single Array analysis

This analysis generates qualitative and quantitative values from one gene expression experiment and provides initial data required to perform comparisons between experiments. A quantitative value, a Detection call, indicates whether a transcript is reliably detected (Present) or not detected (Absent) in the array. The Detection call is determined by comparing the Detection p-value generated in the analysis against user-defined cut-offs. A quantitative value, a signal, assigns a relative measure of abundance to the transcript.

2.3 Detection p-value

The detection algorithm uses a two-step procedure to determine the Detection p-value. First it calculates Discrimination score (R) for each probe pair. The Discrimination score measures the target-specific intensity difference of the probe pair relative to its overall hybridization intensity:

$$R = \frac{PM - MM}{PM + MM}$$

Second, the Discrimination score (R) is tested against the user-defined threshold value Tau, which is a small positive number (by default 0.015). Those probe pairs with the discrimination score higher than Tau vote for the presence of the transcript, while the scores lower than Tau vote for the absence. The voting results of all probe pairs are summarized as a p-value, which is generated by using the One-sided Wilcoxon's Signed Rank test as a statistical method. The higher the discrimination scores are above Tau, the smaller the p-value and the more likely the transcript is present. The lower the scores are below Tau, the larger the p-value, and more likely the transcript is absent. Additionally, prior to the two-step Detection p-value calculation, the level of photomultiplier saturation is evaluated for each probe pair. If all probe pairs in a probe set are saturated, the corresponding transcript is given a Present call. If the Mismatch (MM) probe is saturated, that probe pair is rejected from further analysis. Tau can be used to modify the specificity and the sensitivity of the assay: increasing the value of Tau increases the specificity, but lowers the sensitivity, while decreasing Tau lowers the specificity and increases the sensitivity.

2.4 Detection call

A qualitative value, the Detection call is determined by comparing the Detection p-value with the user-modifiable p-value cut-offs (alpha1 and alpha2, $\alpha 1 = 0.04$, $\alpha 2 = 0.06$ by default). Any p-value smaller than alpha1 is assigned a Present call, and that larger than alpha2 an Absent call. Marginal calls are given to values between alpha1 and alpha2. The p-value cut-offs can also be adjusted to modify specificity and sensitivity.

2.5 Signal algorithm

Signal is a quantitative value calculated for each probe set and it represents the relative level of expression of a corresponding transcript. Signal is calculated as a weighted mean using the One-Step Tukey's Biweight Estimate. The specific signal for each probe pair is estimated by taking the log of the Perfect match intensity after subtracting the stray signal estimate. It does not make physiological sense if the intensity of the Mismatch probe cell is higher than the Perfect match intensity, and therefore an imputed value called Change Threshold (CT) is used instead of the uninformative Mismatch intensity. Three rules are employed to form a stray signal estimate:

- If the Mismatch value is less than the Perfect match, the Mismatch value is

considered informative and the intensity is used directly as an estimate of stray signal.

- If the Mismatch probe cells are generally informative across the probe set except for a few Mismatches, an adjusted Mismatch value is used for uninformative Mismatches based on the bi-weight mean of the Perfect match and the Mismatch ratio.

- If the Mismatch cells are generally uninformative, the uninformative Mismatches are replaced with a value slightly smaller than the Perfect match. The transcripts represented by such probe sets are generally called Absent by the Detection algorithm.

Each probe pair is considered as having a potential vote in determining the Signal value. The probe pair is weighted more strongly if the signal of the probe pair is closer to the median value for that probe set. Once the weight of each probe pair is determined, the mean of the weighted intensity values for that probe set is identified. This mean is the quantitative value Signal.

2.6 Analysis tips

For each experiment check that:

- Background is 40–70 (using current scanner settings).

- Scaling Factor is close to one (using current scanner settings and Target Intensity 100) or at least in the same level for those probe arrays you are planning to compare with each other.

- All positive hybridization controls (BioB, BioC, BioD, CreX) are present.

- Signal ratios of 3', middle and 5' probe sets (GAPDH, Beta-actin) are close to one.

- Percentage of present calls is between 30-50

2.7 Comparison analysis

Comparison analysis provides a computational comparison between two RNA samples hybridized to two GeneChip probe arrays of a same type. For the analysis one array is designated as an experiment and the other as the baseline and compared with each other in order to detect and quantify differences in gene expression between the two RNAs. Two sets of algorithms are used to generate qualitative (Change, Change p-value) and quantitative (Signal Log Ratio) estimates of the potential change.

2.8 Normalization

Before a comparison of two probe arrays can be made, variations between the two experiments caused by technical and biological factors must be corrected by scaling or normalization. Main sources of technical variation in an array experiment are quality and quantity of the labeled RNA hybridized as well as differences in reagents, stain and chip handling. Biological variation, though irrelevant to study question, may arise from differences in genetic background, growth conditions, time, sex, age, etc. Either scaling or normalization should be used to minimize this variation.

Scaling (recommended by Affymetrix) and normalization can be made using either data from user-selected probe sets or all probe sets. When using selected probe sets (for example a group of housekeeping genes) for scaling/normalization, it is important to have *a priori* knowledge of gene expression profiles of selected genes. Thus in most cases less bias is introduced to the experiment when the data of all probe sets is used for scaling/normalization. When scaling is applied, the intensity of the probe sets from experimental and baseline arrays is scaled to a same, user-defined target intensity (recommendation using current scanner setting is 100). If normalization is applied, the intensity of the probe sets from experimental array is normalized to the intensity of the probe sets on the baseline array.

An additional normalization factor, Robust Normalization, is also introduced to the data. It accounts for probe set characteristics resulting from sequence-related factors, such as affinity of the probe set to the RNA and linearity of the hybridization of each probe pair. More specifically, this factor corrects for the inevitable error of using an average intensity of all the probes (or selected probes) on the array as a normalization factor for every probe set. Robust normalization of the probe set is calculated once the initial scaling/normalization factor is determined, by calculating a slightly higher and a slightly lower factor than the original. User-modified parameter, Perturbation, which can vary between 1.00 (no perturbation) and 1.49, defines the range by which the normalization factor is adjusted up and down. The perturbation value directly affects the subsequent p-value calculation, since of the p-values resulting from applying the normalization factor and its two perturbed variants, the most conservative is used to evaluate whether any change in level is justified by the data. Increasing the perturbation value can reduce the number of false changes, but may also block the detection of some true changes.

2.9 Change p-value

Differences between Perfect Match and Mismatch intensities as well as between Perfect Match and background intensities are used to calculate the Change p-value by the Wilcoxon's signed rank test. Also the level of photomultiplier saturation for each probe pair is evaluated. In the computation, any saturated probe cell is rejected from the analysis (number of used cells can be determined from the Stat common pairs column). The Change p-value ranges from 0.0 to 1.0 providing an estimate of the likelihood and direction of the change. For values close to 0.5, true change in the level of expression between the two RNA samples is unlikely, while

values close to 0.0 indicate probability for an increase in gene expression level in the experiment array compared to baseline, whereas values close to 1.0 indicate likelihood for a decrease in gene expression level.

2.10 Change call

To make the Change Call, Change p-value is categorized by cutoff values called gamma1 ($\gamma 1$) which distinguishes between Change calls: Increase, Marginal Increase and No Change, and gamma2 ($\gamma 2$) which draws the line between Change calls: Decrease, Marginal Decrease and No change (Figure 2.1). These values are not directly user-definable, but are derived from two user-adjustable parameters, $\gamma 1$L and $\gamma 1$H, which define the lower and upper boundaries for gamma1. Gamma 2 is computed as a linear interpolation of $\gamma 2$L and $\gamma 2$H.

It is possible to compensate for effects that influence calls based on low and high signals by adjusting the stringency of calls associated with low and high signal ranges (γL and γH) independently. This option is not used by default, since gamma values are set as $\gamma 1$L = $\gamma 1$H and $\gamma 2$L = $\gamma 2$H.

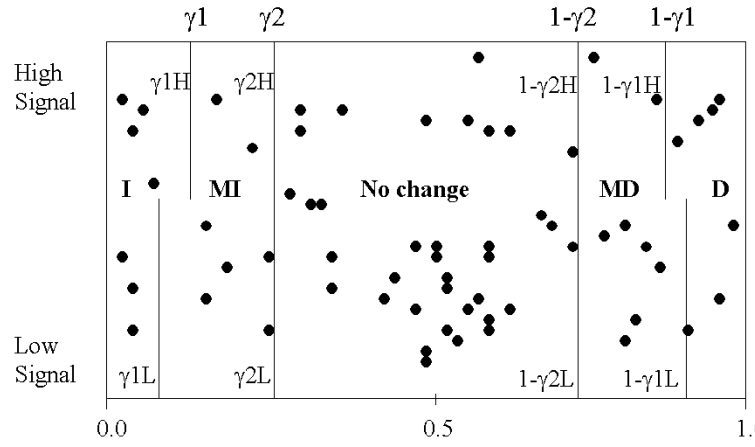

Figure 2.1: *A representation of p-values in a data set. The Y-axis is the probe set signal and the X-axis is the p-values. Values $\gamma 1$L, $\gamma 2$L, $\gamma 1$H, $\gamma 2$H describe user-adjustable values used in making the Change Call (I= Increased, MI = Marginal Increase, MD = Marginal Decrease, D = Decrease).*

2.11 Signal Log Ratio Algorithm

Signal Log Ratio algorithm estimates the measure and the direction of change of a gene/transcript when two arrays are compared. Each probe pair on the experiment array is compared to the corresponding probe pair in the baseline arrays in the calculation of Signal Log Ratio. This process eliminates differences due to different probe binding coefficients. A One-Step Tukey's Biweight method is used in computing the Signal Log Ratio value by taking a mean of the log ratios of probe pair intensities across the two arrays. The base 2 log scale is used, translating the Signal

Log Ratio of 1.0 to a 2-fold increase in the expression level and of -1.0 to a 2-fold decrease. No change in the expression level is thus indicated by a Signal Log Ratio value 0.

Tukey's Biweight method also gives estimate of the amount of variation in the data. Confidence intervals are generated from the scale of variation of the data. A 95% confidence interval shows a range of values, which will include the true value 95% of the time. Small confidence interval implies that the expression data is more exact, while large confidence intervals reflect more noise and uncertainty in estimating the true level. Since the confidence intervals attached to Signal Log Ratios are computed from variation between probes, they may not reflect the full width of experimental variation.

This chapter was written by Janna Saarela.

3 Genotyping systems

3.1 Introduction

SNP (single nucleotide polymorphism) microarrays provide an efficient and relatively inexpensive tool for studying several genetic variations in multiple samples simultaneously. Numerous methods have been developed for SNP genotyping and two of them, namely single base extension (SBE) followed by tag-array hybridization and allele-specific primer extension, are shortly described here.

3.2 Methodologies

For the single base extension reaction, multiple genomic DNA-regions flanking SNP sites are amplified by PCR using SNP-specific primers (Figure 3.1:A). After enzymatic removal of the excess primers and nucleotides single base extension reactions are carried out with detection primers each containing a sequence complementary to the predefined TAG sequence spotted on the array and the SNP-specific sequence just preceding the variation. One allele-defining nucleotide, each labeled with different fluorescent dye, is added to the detection primer in the SBE reaction. Finally, detection primers are hybridized to the TAG sequences spotted on an array. Genotypes are determined by comparing the signals of two possible nucleotides labeled with different fluorescent dyes on each spot.

For the second method, allele-specific primer extension, two amino-modified detection primers, each containing one of the variable nucleotides of the SNP as their 3' nucleotide, are spotted and covalently linked to chemically activated microscope slides (Figure 3.1:B). Genomic DNA flanking the SNP is amplified with SNP-specific primers containing T7 or T3 RNA polymerase recognition sequence in their 5' end. Multiple SNPs can be amplified in one multiplex PCR reaction simultaneously. By using T7 (or T3) RNA polymerase, PCR products are subsequently transcribed to RNA, which is then hybridized to the SNP array containing the two detection primers for each SNP. Reverse transcriptase enzyme and fluorescently labeled nucleotides are then employed to visualize the sequence-specific extension reaction of the allele-defining detection primer(s). Up to 80 different samples can be analyzed on one slide when as many identical subarrays of detection primers are spotted on a slide, which is partitioned into small hybridization chambers. Genotypes are determined by comparing the signals of the two detection primers representing the SNP on the array.

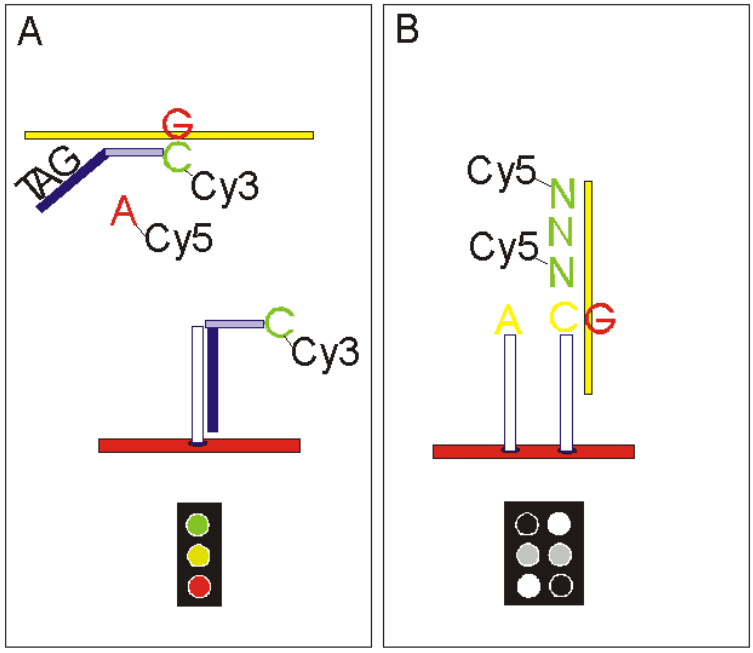

Figure 3.1: *Principles of the two SNP genotyping methods. A. Single base primer extension followed by TAG-array hybridization. B. Allele-specific primer extension on array. Traffic lights at the bottom of the figures describe three possible genotypes with each method.*

3.3 Genotype calls

To make the genotype call, signals of the two possible variants are compared. In case of the SBE reaction, intensities of the two different fluorophores in a spot representing the SNP are measured. In case of the allele-specific primer extension method, each SNP is represented by two spots, both putatively labeled (with the same fluorophore). The intensities of the two spots are measured and compared to each other to determine the genotype. The same methods can be used for both comparisons. One method is to calculate the part of the signal of one allele over the total signal intensity:

$$P = \frac{signal\,of\,allele\,1}{(signal\,of\,allele\,1 + signal\,of\,allele\,2)}$$

All the values are between zero and one (Figure 3.2). Values close to one represent the homozygote allele 1 genotype, and those close to zero represent the other homozygote genotypes (allele 2). Those in-between (close to 0.5) represent the heterozygote genotypes. A scatter plot can be formed having the values for P in the X-axis and logarithm of the total intensity as the Y-axis. Three clear groups of signals should show clearly, each representing one of the possible genotypes.

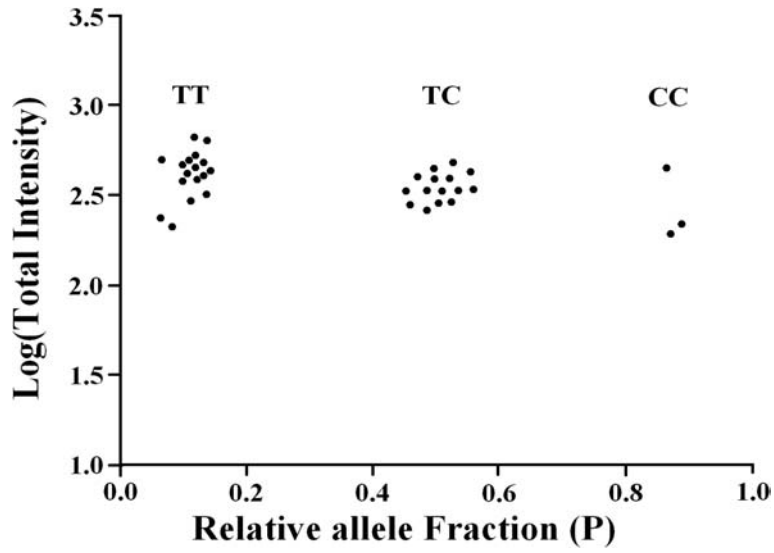

Figure 3.2: *An example of genotype calls. TT and CC denote homozygous individual, and TC denotes heterozygous individuals.*

3.4 Suggested reading

1. Syvanen, A-C. (1994) Detection of point mutations in human genes by the solid-phase minisequencing method. Clin. Chim. Acta. 226, 225-36.

2. Guo, Z., Guilfoyle, R. A., Thiel, A. J., Wang, R., and Smith, L. M. (1994) Direct fluorescence analysis of genetic polymorphisms by hybridization with oligonucleotide arrays on glass supports. Nucleic Acids Res. 22, 5456-5465.

3. Pastinen, T., Raitio, M., Lindroos, K., Tainola, P., Peltonen, L., Syvanen, A-C. (2000) A system for specific, high-throughput genotyping by allele-specific primer extension on microarrays. Genome Res. 10,1031-42.

4. Hirschhorn, J. N., Sklar, P., Lindblad-Toh, K., Lim, Y. M., Ruiz-Gutierrez, M., Bolk, S., Langhorst, B., Schaffner, S., Winchester, E., Lander E. S. (2000) SBE-TAGS: an array-based method for efficient single-nucleotide polymorphism genotyping. Proc. Natl. Acad. Sci. U S A 97, 12164-9.

This chapter was written by Janna Saarela.

4 Overview of data analysis

In this book, we emphasize to microarray data analysis after the microarrays have been hybridized, scanned and the images have been analyzed with an image analysis software. Before any experiments in the laboratory have been initiated, the experiment and its analysis should be planned carefully. *Chapter 5* describes points that need to be considered when designing the experiments. The following schematic workflow outlines the basic analysis steps of exploratory data analysis.

4.1 cDNA microarray data analysis

We start data analysis from the results of scanned images. At this point, images have been evaluated, bad spots have been investigated and the spots have preferably been scored with flags indicating whether the spot was good, bad, or borderline. This is crucial, because in the later stages of the analysis the visual inspection of individual spots is not possible.

Next steps in the analysis are preprocessing (*Chapter 7*), normalization (*Chapter 8*), and quality control (*Chapters 6, 7, and 8*). The goal of these analyses is to organize results in a meaningful order: flags, controls, and experiments are pointed out and checked. Variation due to systematic errors are removed, and data from different chips is made comparable.

Statistics is needed in many steps during the analysis. Therefore, the whole *Chapter 6* has been dedicated to basics of statistics. Statistical tools are used for evaluation of the raw data during the above-mentioned steps, and in the further analysis to find significantly differentially expressed genes.

In these further analysis steps, statistically significant, quality-checked data is separated from not interesting and not-trustworthy data (*Chapter 7*). The next step is to find the differentially expressed genes using statistical tools (*Chapter 9*), or to group the good quality data (usually only a small fraction of the original raw data) into meaningful clusters by *e.g.* clustering (*Chapter 10*). The goal of clustering is to find similarly behaving genes or patterns related to time scale, time point, developmental phase or treatment of the sample.

At this point, we have already manipulated our data quite a lot, and spent considerable time with computer. Despite that and depending on what we are looking for, we may be at the very beginning of the challenging part of the data analysis. Next we need to link the observations to biological data, to regulation of genes, and to annotations of functions and biological processes. This part, data mining, is described in *Chapters 11, 12 and 13*.

With an enormous amount of data, we need standardized systems and tools for data management in order to publish the results in a proper and sound way, as well as to be able to benefit from other publicly available gene expression data. These aspects are discussed in *Chapter 14*. Data file manipulation and analysis tools are introduced in *Chapter 15*.

4.2 Affymetrix data analysis

Putting model-based methods aside, exploratory data analysis using Affymetrix chips is very similar to cDNA microarray data analysis. The biggest difference is normalization. If comparison analysis (see 2.7) is conducted, the Affymetrix data can be treated similarly to a background corrected cDNA data. If single array analysis is performed, the basic normalization scheme can be similar to the one presented in section 8.9.

4.3 Data analysis pipeline

To get an overview of the data analysis pipeline, consult Figure 4.1. It covers the basic methods introduced in this book. The flow chart also helps to choose the right method for the situation. The Table 4.1 contains a short list of analysis methods, and links to the chapters, where more details are available.

The flow chart should be read flexibly, though. For example, there can be several filtering steps instead of just one shown on the chart. Or, when there are more than two conditions in the experiment, the data can be analyzed using two conditions route. Note that all possible orders of analysis have not been shown for clarity, and the schema is only meant to help you in the analysis process, not to be taken as an absolute truth.

There are different opinions on which order the steps should be taken, and which portion of data should be transferred to the next step, so please, study the relevant literature to find out more about limitations and characteristics of different analysis steps. We would very much like to encourage you to make your own educated decisions, and not to take this book with a face value!

This chapter was written by M. Minna Laine and Jarno Tuimala.

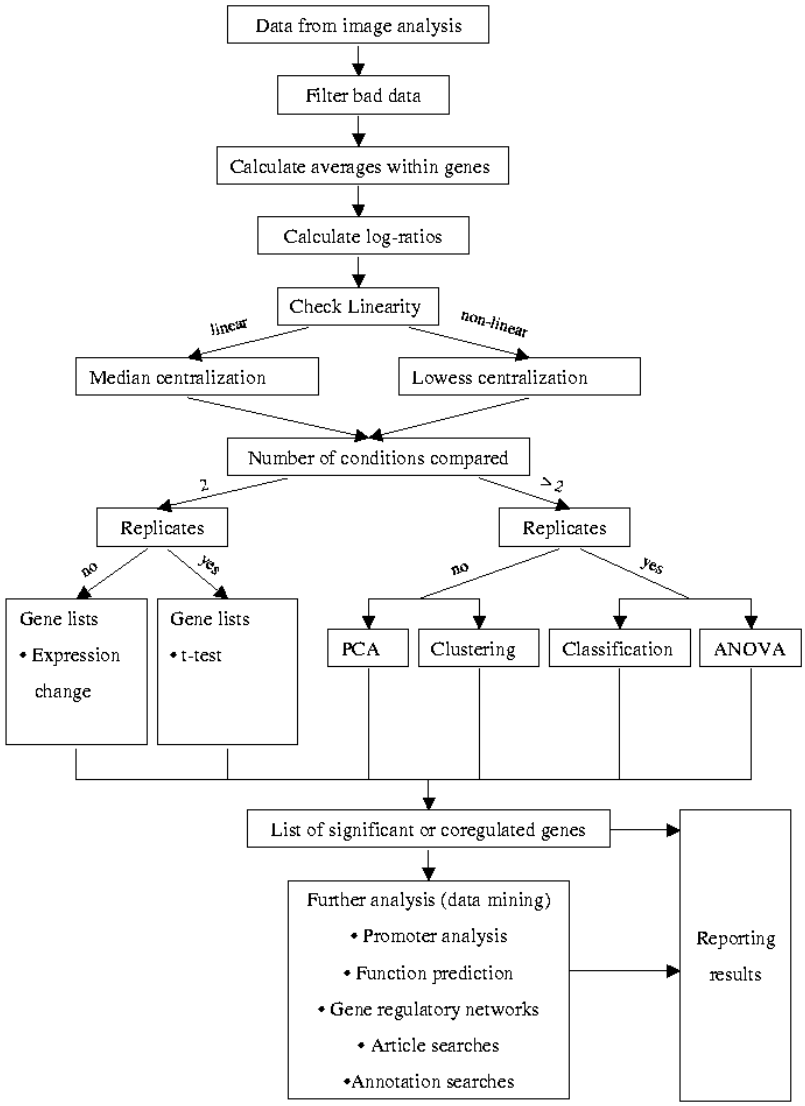

Figure 4.1: *Overview of data analysis methods discussed in this book. Note that all possible orders of analysis have not been shown.*

Table 4.1: *Chapters for methods shown in Figure 4.1.*

Method	Chapter
Experimental design	5
Basic statistics	6
Background correction	7
Calculation of expression change	7
Quality checking	7
Filtering	7
Normalization	8
Single slide methods	9
t-test	9
ANOVA	6
Clustering	10
Classification	10
PCA	10
Function prediction	10
Gene regulatory networks	11
Promoter sequence analysis	12
Annotations	12, 13
Ontologies	12, 13
Article mining	13
Data management	14
MIAME standard	14
MAGE standard	14

5 Experimental design

5.1 Why do we need to consider experimental design?

Good experimental design, above the myriad of other considerations, will likely provide you with the greatest amount of satisfaction and least amount of frustration in executing a microarray project. In the early days of microarray research, a simple design of control v. treatment and replicates of each would suffice. However, those days are rapidly going by the wayside and 30–80 chip experiments in a single design are becoming common. While the design depends primarily on the scientific question that is being proposed (hypothesis), other considerations, such as appropriate controls, platforms, and statistical issues merit serious thought. This is to say nothing about costs! While many projects have similar objectives, it would be foolish for us to design protocols in a guidebook without intimate knowledge of the system and the intricacies involved in the study. Moreover, many fields have specialized and/or traditional methodologies that require non-typical designs. Nonetheless, there are several variables in the experimental design that are common to most studies, and we will try to describe them here. We hope that the consideration of the strengths and weaknesses of each variable on both the biological and statistical level would help the individual investigator better design his/her own microarry study. Most of the discussion concerns expression arrays, although places where SNP arrays are relevant are pointed out.

5.2 Choosing and using controls

The *control* serves as a reference for an experiment commonly termed the *treatment*. The treatment can be chemical (drugs, toxins), biological (viruses, microbes, transgenes) or environmental (stress, irradiation) in nature. The individual treatments can be given in a time series (time course) or at different doses (dose response). In all these cases, the control should be matched as closely as possible genetically. This can mean that the controls are sibs or that the animal used is an inbred strain, or a combination of the two. Controls with the same environmental influences are usually dealt with by using litter siblings that have been raised identically. Physiological matching can be done by taking the same sex, age, and health status.

Controls for human tissues are problematic since age, cause of death, and post-mortum interval are difficult to match. Moreover, human tissues for research purposes are often stored for unequal intervals, which greatly effect RNA quality.

Because of this potential confounder, many human studies have been done using human cell lines rather than tissues. Another solution is to match tissue by taking different regions from the same human tissue, one that is healthy to serve as the control, and one that carries the disease as the treatment sample.

Controls for mice studies have a different problem in that some transgenic mice are crosses between xxx and yyy strains. Therefore, both transgenic and non-transgenic littermates may have quite different genetic backgrounds. The solution would be to backcross to one of the parental strains until both the control and transgenic mouse have the same genetic background. This is somewhat time consuming since mice have a reproductive cycle of 2–3 months. Another solution would be to make sure that the transgenic lines that are produced have a homogeneous background.

Controls for cell based experiments generally consist of identical cultures without the physiological, physical, or chemical treatments. The controls may also include cells derived from other sources such as the equivalent or healthy tissues. When cells are cocultured, the controls become more complicated. Cultures of each of the individual constituent cell populations in addition to a coculture could be used as a control. In practice this means that if a coculture of A and B cells is made, then the controls will be A alone, B alone, A and B together.

Because of all these potential variables that could effect the microarray results, it often makes sense to create a design with more than one control within the study.

5.3 Choosing and using replicates

Replicates are repeated experiments with the same sample (Table 5.1). Replicates provide a measure of the experimental variation, such as in RNA isolation, labeling efficiency, or in chip quality. It is highly recommended to include replicates in your design, otherwise you will not know the inherent experimental variation. Duplicates are common practice and triplicates provide even more information. Some designs incorporate combinations of duplicates and triplicates. In some cases, it may not be practical to perform replicates due to lack of tissue or other reason. In these cases, replicated spots on the chip can provide some idea of the variation in hybridization. Including a large number of both controls and treatments without replicates is another design strategy that has been used. Confirmation of results from non-replicated experimental designs using independent methods such as quantitative PCR and northern blotting is also one way to circumvent the need for replicates.

5.4 Choosing a technology platform

For expression arrays, there are two principle platforms: spotted arrays and *in situ* synthesized arrays such as Affymetrix. The basic principles have been presented in the introduction and detailed aspects are covered later in the book. Apart from the experimental design issues presented here, the Affymetrix platform uses one labeling reaction per chip, so a control reaction is performed and hybridized and then the treatment reaction is performed on a different chip. The results are then

Table 5.1: *Typical experimental designs and number of microarrays needed.*

Experimental design	Samples	Replicates	Cy3/Cy5	Affymetrix
Control v. Treated	1	1	1	2
Control v. Treated	1	3	3	6
Control v. Treated	5	2	10	12
Control v. Treatments (4)	1	1	4	5
Control (2) v. Treatments (2)	1	1	4	4
Control (2) v. Treatments (2) vs. time points (4)	1	1	16	16
Control (2) v. Treatments (2) vs. time points (4)	1	3	48	48
Control (2) v. Treatments (2) vs. time points (4)	4	2	64	96-112
Controls (1) v. Treatments (2) vs. time points (4)	6	2	96	144

compared after having been normalized by independent spiked standards. This is in contrast to expression arrays where there is a direct comparison between control and treated. While not completely obvious, the advantage of the Affymetrix platform is that multiple experiments can be compared directly if they are performed on the same chip series. Thus, many of the steps that can cause experimental variation are controlled. The labeling, hybridization, and readout is done on identical instruments with identical protocols on identically designed chips, so other than normalization to an external standard, all experiments performed on the same chip series should be comparable. This adds tremendous power to the experimental design. Since control and treatment are hybridized on different Affymetrix chips, the design requires twice as many chips compared to expression arrays. However, controls can be compared with all other samples so there is a small benefit in this design if many controls are needed. Moreover, comparison to publicly available data can be performed. Affymetrix platform has the added advantage of multiple probes for each gene, so in a sense, you get 16–25 measurements for each gene rather than the 1 or 2 from typical expression arrays. This is especially useful if you wish to hybridize to genes with degenerated sequences such as splice forms, highly polymorphic, or related species. The selection of gene chips available from Affymetrix is more limited than currently available expression arrays, but contains the major species used in scientific research including *Saccharomyces cerevisea* (yeast), *Arabidopsis Thaliana* (mustard weed), *Caenorhabditis elegans* (worm), *Mus musculus* (mouse), and *Homo Sapiens* (human).

5.5 Gene clustering v. gene classification

It often makes more sense to identify genes and their related expression levels that predict the experimental group they are derived from (gene classification) than to find groups of genes that act in a particular way after treatment (gene clustering).

For classification studies, large numbers of samples for each treatment groups are needed in order to robustly find a set of genes that predicts their original treatment group. The large number of samples is also necessary to validate the predicted classification based on training sets derived from the data. In this case of using gene expression to predict human tumor type as a diagnostic tool, individual variation from human subjects, and variation in the samples due to which part of the tumor was dissected, how the tissue was obtained, and the other experimental variables typically suggests that random variation in the profiling is unavoidable, further indicating that sample sizes should be large.

In contrast, the number of genes profiled need not be large if the genes consistently predict the classification group. This is unfortunately not the usual case.

5.6 Conclusions

The experimentalist is likely to know the most about the scientific question and therefore have the greatest input into the experimental design. Nonetheless, consultations with a statistical expert prior to experiments could help to decrease experimental variation derived from the limitations listed above. Recently, two excellent review articles have become available and are listed at the end of this chapter. In the end, there are many factors influencing the successful outcome of a microarray study, and attention to as many as possible will greatly aid in the final outcome.

5.7 Suggested reading

1. Churchill, G. A. (2002) Fundamentals of experimental design for cDNA microarrays. Nature Genetics supplement 32, 490-495.

2. Yang, Y. H. (2002) Design issues for cDNA microarray experiments. Nature Reviews, 3, 579-588.

This chapter was written by Garry Wong.

6 Basic statistics

6.1 Why statistics are needed

One important use of statistics is to summarize a collection of data in a clear and understandable way. There are two basic methods: numerical and graphical. Graphical methods are best at identifying patterns in the data. Numerical approaches are more precise and objective. Since numerical and graphical approaches complement each other, both are needed.

6.2 Basic concepts

6.2.1 Variables

A *variable* is any measured characteristic that differs between subjects. For example, if the weight of 50 individuals were measured, then weight would be a variable. The numerical values of weight for the subjects are called measurements, scores, data points or observations.

Variables can be quantitative or qualitative. Quantitative variables describe certain quantities of subjects, and can be measured on a continuous or discrete scale. On the continuous scale, there is an infinite number of values variables can take. Discrete variables can take on only a limited number of values. For example, weight is measured on a continuous scale, and school grades on a discrete scale.

Qualitative variables describe the attributes of the subjects, for example, hair colour. They are always measured on a discrete scale.

When an experiment is conducted, some variables are manipulated by the experimenter, and others are measured from subjects. The former are factors or independent variables, the latter, dependent variables.

6.2.2 Constants

Measures of subjects, i.e. variables, which can take on only one value during the experiment are called *constants*. For example, if only women are included in a study, then the variable sex is constant during the study.

6.2.3 Distribution

Distribution describes the scores and the number of scores the variable can take. A continuous variable can be described with a histogram, which is a graphical representation of the distribution (Figure 6.5). Distributions are more thoroughly

covered in the subsequent sections.

6.2.4 Errors

Errors are inaccuracies in measurements. In the laboratory analyses, there are often an innumerable amount of potential sources of errors, but only a few are usually of paramount importance.

Errors can be systematic (biases) or random. Systematic errors affect the measurements such a way that you get a wrong estimate of whatever you are measuring. Systematic errors affect all the subjects similarly. Random errors lack such predictivity. For example, one laboratory technician can be biased, if his/her analysis results are always consistently higher than the results acquired by other technicians. If the same technician drops the tube so that the liquid is spilled on the floor, but he/she then slurps the liquid back up and transfers it to the tube with a pipet (a procedure known as linoleum blot), a random error has been created. If all the tubes are always cast on the floor, that would systematically bias the results.

6.3 Simple statistics

6.3.1 Number of subjects

Number of subjects (N) or sample size both describe the same thing: How many subjects there are included in the study. Often the sample size is fixed before the experiment is conducted. For example, if 1 000 genes are spotted on the DNA microarray, then the sample size would be 1 000 genes.

6.3.2 Mean (m)

The arithmetic mean of a variable is calculated as the sum of all measurements divided by the number of measurements. The mean is a good measure of central tendency for roughly symmetric distributions, but it can be misleading in skewed distributions due to probable influence of extreme scores. For skewed distributions, the trimmed mean or median usually gives more meaningful results.

6.3.3 Trimmed mean

A trimmed mean is calculated by discarding a certain percentage of the lowest and the highest scores and then computing the mean of the remaining scores. A trimmed mean is obviously less susceptible to the effects of extreme scores than the arithmetic mean, but it is also a less efficient descriptor than the mean for normal distributions.

6.3.4 Median

The median is the middle of a distribution: half the scores are above the median and half are below. The median is less sensitive to extreme scores than the mean and this makes it a better measure of central tendency for highly skewed distributions.

6.3.5 Percentile

The percentile is a certain cut-off, under which the specified percentage of the observations lie. The median is 50th the percentile of the distribution. Similarly, the 80th percentile is a point of the distribution under which 80% of the observations lie.

6.3.6 Range

The range is the difference between the largest and the smallest value of the distribution. It is the simplest measure of spread, but it is very sensitive to extreme scores, because it is based only on two values.

6.3.7 Variance and the standard deviation

The variance (d) and standard deviation (s or sd) are both measures of spread. The variance is calculated as the averaged squared deviation of observations from their mean. For example, for the scores 1, 2, and 3, the mean is two, and the variance is: $[(1-2)^2+(2-2)^2+(3-2)^2]/3 = 0.667$. The standard deviation is the square root of the variance: $\sqrt{(0.667)} = 0.827$. Standard deviation is often more easily interpreted than variance, because its unit is the same as the unit of measuments.

6.3.8 Coefficient of variation

The coefficient of variation (CV) describes the relative variability of the data. It is especially good in situations, where distributions of unequal magnitude are compared. For example, the absolute variability measured in kilograms is very small in a mouse population, but very large in an elephant population. In such cases, the coefficient of variation gives a better description of the variability than the standard deviation, because we are usually interested in how the variability is related to the mean of the species, and because the CV is independent of the absolute values of the variable. The coefficient of variation is calculated as

$$CV = \frac{s}{m},$$

where s is the standard deviation and m is the mean.

Often the CV is reported as a percentage. In biological research, it is a widely-used measure of reliability for replicated experiments.

6.4 Effect statistics

6.4.1 Scatter plot

The Scatter plot is an important graphical tool for studying the spread and linearity of the data. In its simplest form, two variables are plotted along the axes, and marks are drawn according to these coordinates. For example, green and red channel intensities of DNA microarray experiments can be depicted as a scatter plot (Figure 6.1).

6.4.2 Correlation (r)

The correlation of two variables represents the degree to which the variables are related. When two variables are perfectly linearly related, the points in the scatter plot fall on a straight line. Correlation measures only linear relationship. Two summary measures or correlation coefficients, Pearson's correlation and Spearmans's rho, are most commonly used. Both of these measure range from perfectly positive linear relationship to perfectly negative linear relationship, or from -1 to 1.

It is not wrong to calculate the correlation between variables, which are not linearly related, but it does not make much sense. If the variables are not linearly related, the correlation does not describe their relationships effectively, and no conclusions can be based on the correlation coefficient only.

Correlation and scatter plot are a good example of how numerical and graphical tools effectively complement each other.

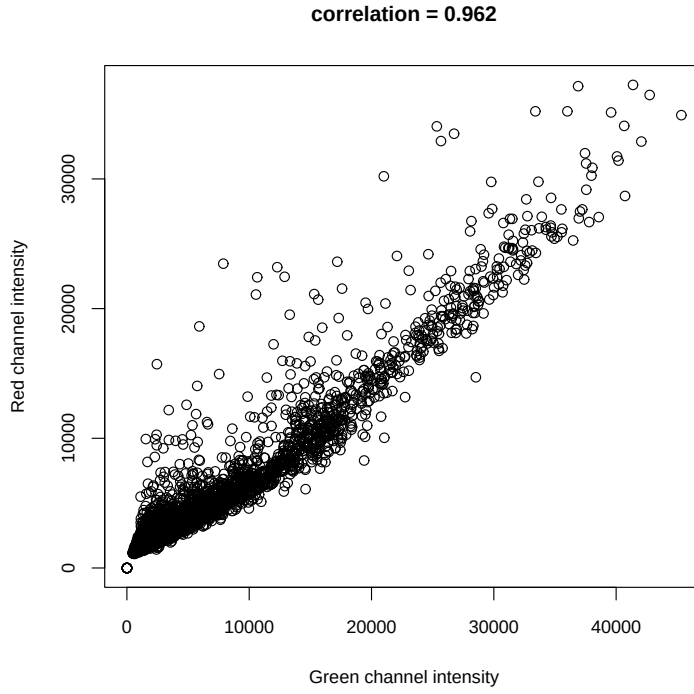

Figure 6.1: *An example of a scatter plot. The red and green channel intensities from a two-color DNA microarray experiment have been depicted as a scatter plot. Variables seem to have a linear relationship, because a straight line can be drawn through the data points. In addition, the linear correlation between these variables is 0.962.*

6.4.3 Linear regression

Linear regression is used for describing how much the dependent variable (the predicted variable) changes if the independent variable (the predictor variable) changes a certain bit. It would be wrong to apply linear regression to non-linear data. Therefore, before applying linear regression, the linearity of the data should be checked from a scatter plot (Figure 6.1). In addition, the normality of the dependent variable should be checked before running the analysis (see the following sections).

Linear regression fits a straight line through the data points so that the square sum deviation of the line from the data points is minimized. The result of the procedure is a regression equation

$$Y = aX + b,$$

where b indicates where the regression line cuts the vertical axis. If only two variables are used in the linear regression model, a is equal to the pearson correlation, and indicates the slope of the regression line. It also indicates by which number change in X (independent variable) must be multiplied to give the corresponding change in Y (dependent variable).

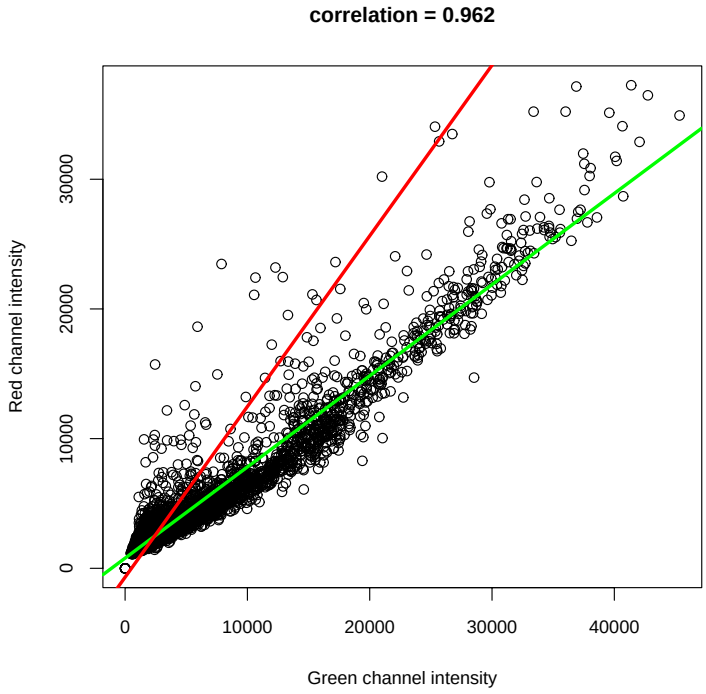

Figure 6.2: *The two linear regression lines fitted into two-color DNA microarray results. The results are different if green channel intensity (green line) is used as a dependent variable instead of red channel intensity (red line).*

Correlation is completely symmetric. The correlation between A and B is the same as correlation between B and A. Linear regression does not have this symmetric property. For linear regression it is absolutely vital to know which is the dependent variable, in which we want to predict the change. This should be decided when the experiment is planned. For example, we could try to predict how many papers we could print with a certain cartridge and machine, but it would not be possible to predict which cartridge and machine we have, if we know that a certain number of papers have been printed. For microarray experiments, linear regression is sometimes used for normalization. This is generally not a good idea, because it is not possible to decide which labeled sample (red or green) should be used as an independent and/or dependent variable (Figure 6.2).

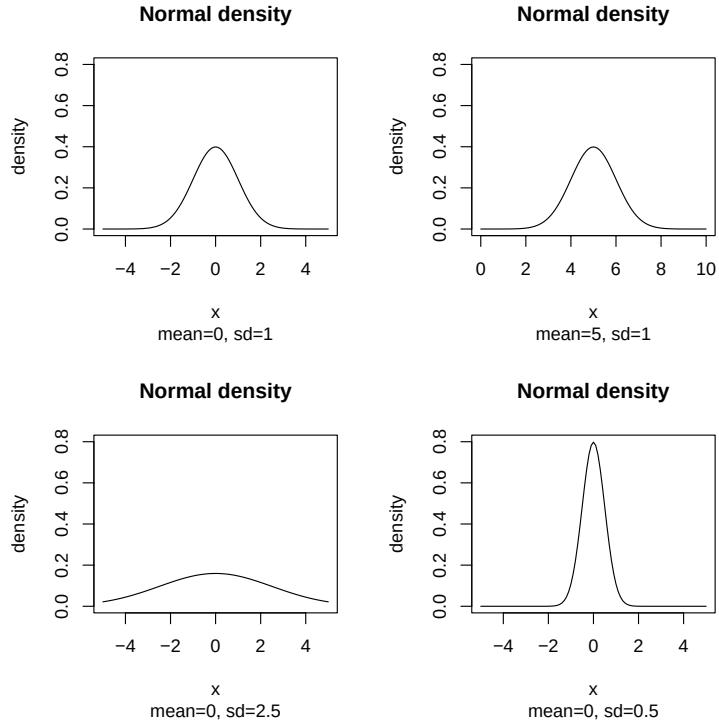

Figure 6.3: *Normal distributions of different shapes. The mean defines the place of the distribution along the horizontal axis, and standard deviation (sd) defines the shape of the curve. Small standard deviation means tight, and large standard deviation a flat distribution.*

6.5 Frequency distributions

6.5.1 Normal distribution

Normal distributions are a family of distributions that have the same general shape. They are symmetric around their mean with more measurements in the middle than

in the tails. Normal distributions are sometimes described as bell shaped. The shape of the normal distribution can be specified mathematically in terms of the mean and standard deviation (Figure 6.3).

A standard normal distribution is a normal distribution with a mean of 0 and a standard deviation of 1. Any normal distribution can be transformed to standard normal distribution by the formula

$$Z = \frac{(X - m)}{s},$$

where X is a score from the original normal distribution, m is the mean of the original normal distribution, and s is the standard deviation of the original normal distribution. This mathematical procedure is also called standardization.

There are biological and historical reasons for the widespread usage of normal distribution: Many biologically relevant variables are distributed approximately normally. Mathematical statisticians also work comfortably with normal distributions, and many kinds of statistical tests can be and are derived from normal distributions. Some of these tests will be described in the next chapters. We have already had a peek at one application of normal distribution, linear regression.

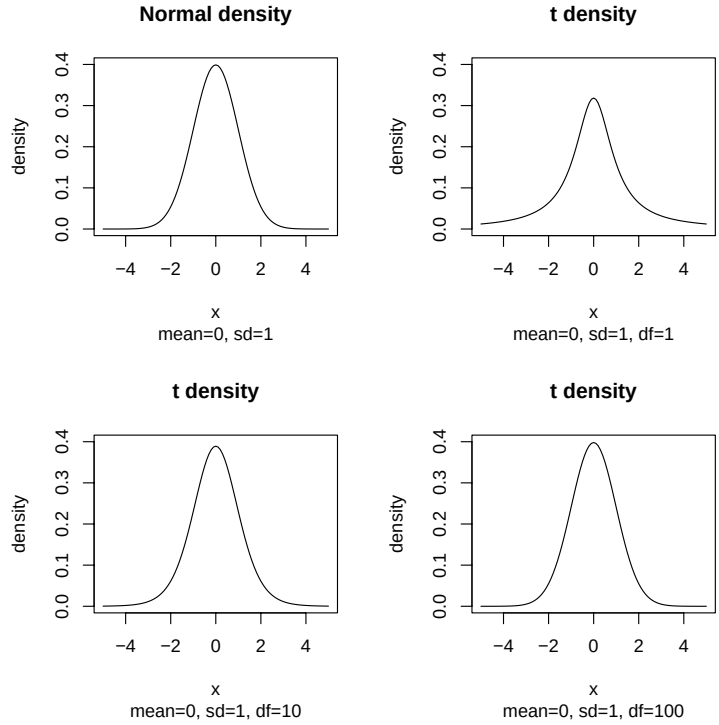

Figure 6.4: *The mean of the t-distribution defines its place on x-axis. The shape of the distribution is defined by standard deviation (s) and degrees of freedom (df). Standard deviation defines the tightness of the distribution, whereas df defines the amount of resemblanse to normal distribution.*

6.5.2 t-distribution

The t-distribution resembles normal distribution, and in mathematical terms, it is an approximation of the normal distribution for small samples sizes. T-distribution differs from the normal distribution, because it has an additional parameter, degrees of freedom (df), which affects its shape. Degrees of freedom reflect the sample size.

Degrees of freedom can take on any real number greater than zero. A t-distribution with a smaller df has more area in the tails than the distribution with a larger df. As the df increases, the t distribution approaches the standard normal distribution. For practical purposes, the t-distribution approaches the standard normal distribution relatively quickly, so that when df = 50, the two are virtually identical (Figure 6.4).

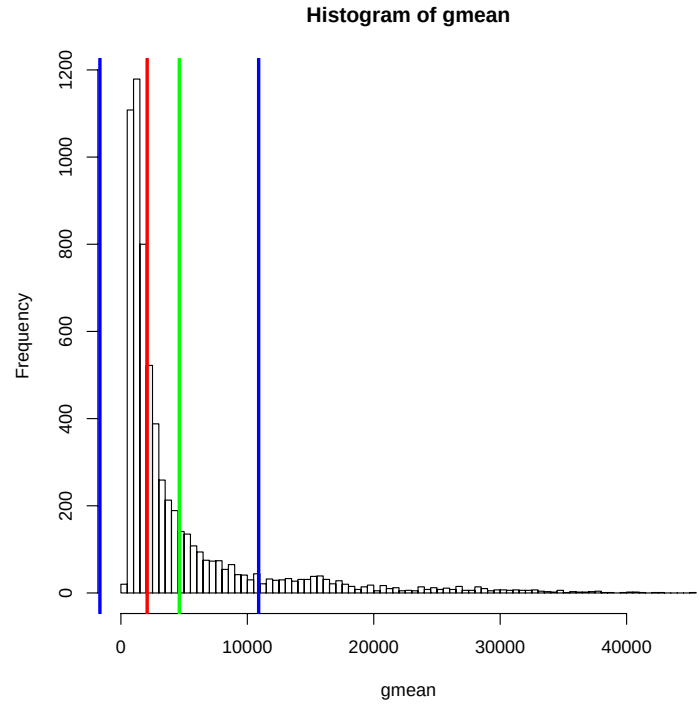

Figure 6.5: *A typical histogram of the DNA microarray intensity values of one channel (gmean). The distribution is highly skewed to the right. Median, mean and 68% standard deviation are depicted as red, green, and blue vertical lines, respectively.*

6.5.3 Skewed distribution

A distribution is skewed if one of its tails is longer than the other. Distributions having a longer tail on the right are positively skewed, and distributions having a longer tail on the left are negatively skewed. The best way to identify a skewed distribution is to draw a histogram of the distribution (Figure 6.5). Numerically,

a skewed distribution can be identified by comparing the mean and median of the distribution. If the mean is larger than median, the distribution is skewed to the right. For left skewed distributions the mean is lower than median.

6.5.4 Checking the distribution of the data

Many statistical tests assume that the data is normally distributed, which means that the histogram drawn from the values of the variable resembles the normal distribution. Even the most basic descriptive statistics can be misleading if the distribution is highly skewed. For example, standard deviation does nor bear a meaningful interpretation if the distribution significantly deviates from normality (Figure 6.5).

Normality of the data is most easily checked from an appropriate histogram . If the distribution is, as judged by eye, approximately symmetric and does not contain more than one peak, it can be assumed to be normally distributed (Figure 6.5 and Figure 7.6). Other graphical possibilities include a density plot, which are basically just a smoothed histogram (Figure 6.6).

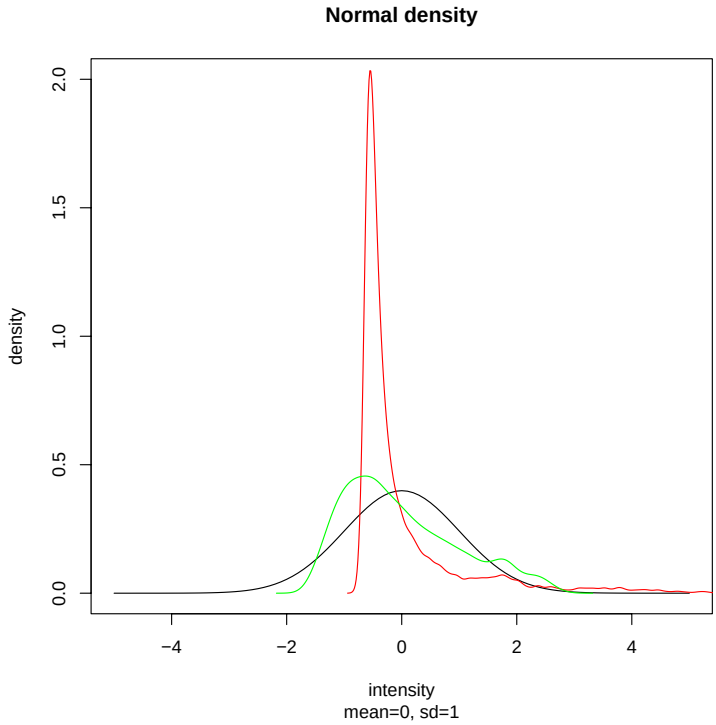

Figure 6.6: *Some real data compared to a normal distribution. Black, normal distribution; red, non-transformed data; green, log-transformed data. The log-transformation was performed, because it often makes highly skewed distribution more normally distributed. The values have been standardized before plotting.*

A more formal way to test for the normality of the data is the normal probability plot. The sample values and the theoretical values assuming normality are plotted against each other in the normal probability plot. If the points fall on the line, the distribution is normal (Figure 6.7).

Both the histogram and normal probability plot are valid methods for checking the normality of the data, but the plot can detect much smaller deviations from normality than histogram.

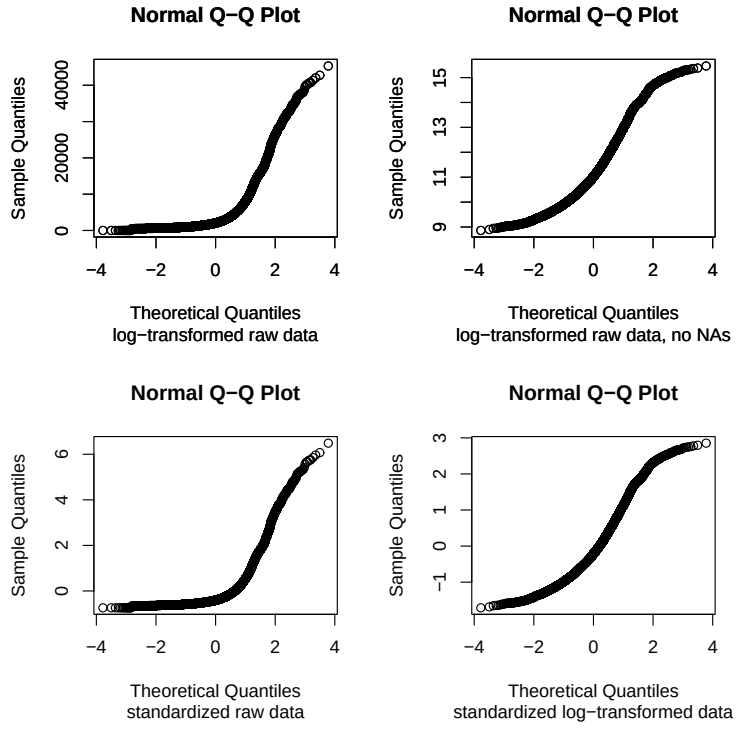

Figure 6.7: *The normal probability plots. The raw data has been pushed through different kinds of transformations, and the results are checked for normality. The standardized log-transformed data was drawn with green in the Figure 6.6. According to the plots, none of the data sets is normally distributed, because the datapoints do not fall on a straight line.*

6.6 Transformation

Transformation means a mathematical procedure, where a new variable is constructed or derived from the original one by applying a certain function or formula. Data is often transformed, because the original data does not fulfill the distributional presumptions. For example, many statistical procedures assume that the data is normally distributed. If this is not the case, transformation can be applied in such a way that the distribution becomes more normal-like.

6.6.1 Log_2-transformation

Log_2-transformation is often used with DNA microarray experiments. Usually, the intensity ratio is log_2-transformed. The resulting new varible is called log ratio. The increase of one in the log ratio means that the actual intensity or expression has doubled.

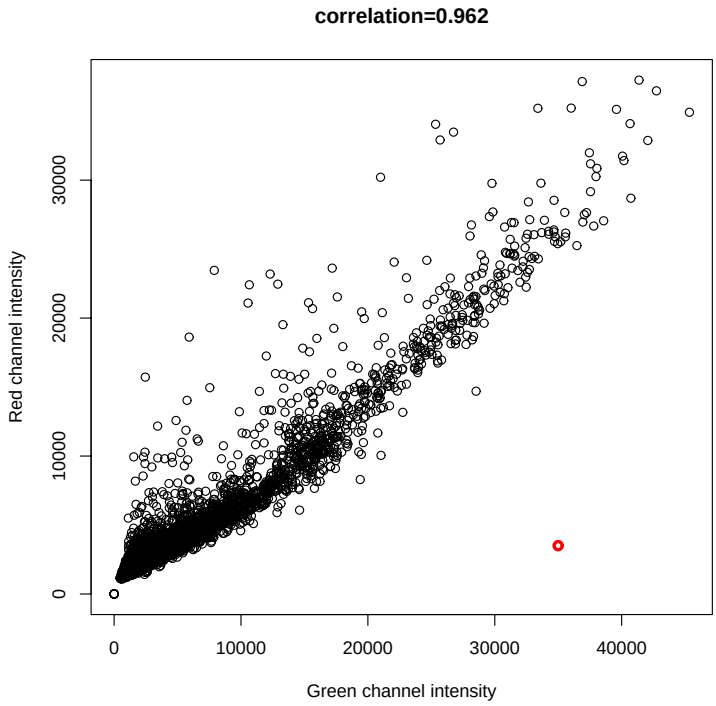

Figure 6.8: *A scatter plot where one probable outlier has been marked with a red dot.*

6.7 Outliers

Outliers are by definition atypical or infrequent observations, data points which do not appear to follow the characteristic distribution of the rest of the data. These may reflect the properties of the variable, or be due to measurement errors or other anomalies which should not be included in the analyses.

Typically, we believe that outliers represent random errors, which we would like to control or get rid of. Outliers can have many undesirable properties: They often attract the linear regression line, which might lead to wrong conclusions. They can also artificially increase the correlation coefficient or decrease the value of the legitimate correlation. The measure of spread, range, is unreliable if the data includes outliers.

Outliers are often excluded from the data after the superficial analysis with quantitative methods. For example, observations that are outside the range of 2

standard deviations from the mean, can be discarded. However, definition of an outlier is subjective, and in principle the decisions should be made on individual basis for every suspicious observation.

There are two graphical methods with which to identify outliers, scatter plot and box plot (Figure 6.8 and Figure 6.9).

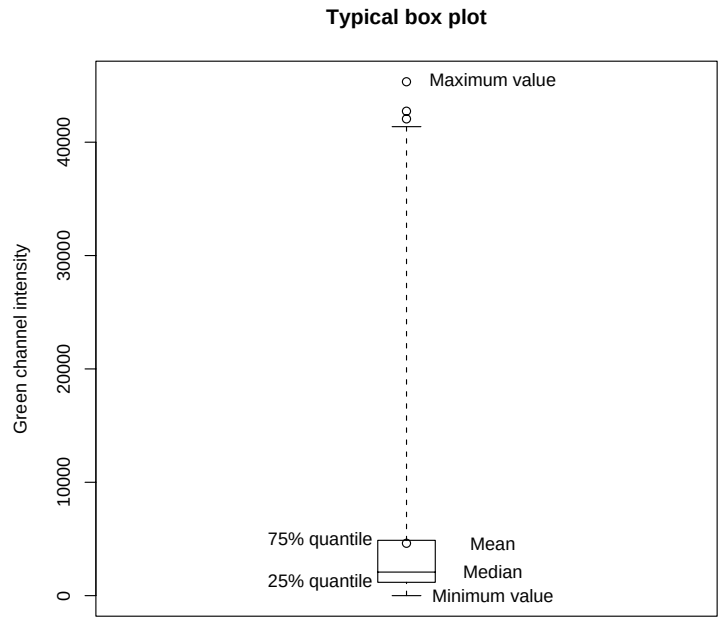

Figure 6.9: *An example of a box plot. Outliers are often indicated with individual marks on the top of the boxplot.*

6.8 Missing values and imputation

The missing values can sometimes markedly change the results, if only mean or median values are inspected. Take, for example, the following experiment. We want to find the genes, which can be used for dividing patients into two cancer subtypes, those responding and not responding to the treatment. We measure the expression values (log ratios) in the these subtypes, and plot the results.

```
Responsive group

No. of obs.       20                          3  4   2   1
                  |------|--------------------------------|
Expression        0.0   mean        5.0                  10.0
```

```
Non-responsive group

No. of obs.       0                  3   4   2   1
                  |----------------------|---------------|
Expression        0.0               5.0 mean            10.0
```

In the example, the missing values influence the mean very much. In a sense, missing values draw the mean towards them. Even the median would not have helped us here, because it would have scored 0 for the responsive group!

Missing values are usually either removed from the analysis or estimated using the rest of the data in a process called imputation. These methods are covered in more detail in the next chapter.

6.9 Statistical testing

Statistical testing tries to answer the question: "Can the difference between these observations be explained by chance alone?" or "How significant is this difference?". Statistical testing can also be viewed as hypothesis testing, where two different hypotheses are compared. For example, we can test whether cancer patients and their healthy controls differ statistically significantly for the expression of the gene X.

6.9.1 Basics of statistical testing

A suitable statistical test can give us an answer to the questions above. In practise, statistical testing is divided into several substeps:

1. Select an appropriate statistical test.
2. Select a threshold for p-value.
3. Form the pair of hypotheses you want to compare.
4. Calculate the test statistic.
5. Calculate degrees of freedom.
6. Compare the test statistic to the critical values table of the test statistic distribution.
7. Find out a p-value, which corresponds to the test statistic.
8. Draw conclusions.

These substeps are covered in more detail in the following subsections.

6.9.2 Choosing a test

There are at least a couple of hundred of different statistical tests available, but only a few of those are covered here in more detail. Usually the means of certain groups are compared in the microarray experiments. For example, we can test whether the expression of a certain gene is higher in the cancer patients than in their healthy controls. The tests introduced here compare the means of two or several groups with each other.

When choosing a test, there are two essential questions, which need to be answered: Is there more than two groups to compare, and should we assume that the data is normally distributed?

If two groups are compared, there are two applicable tests, the t-test and Mann-Whitney U test. If more than two groups are compared, an analysis of variance (ANOVA) or a Kruskal-Wallis test is used. The ANOVA procedure is covered in more detail in section 6.10 of this book. Note, that if the ANOVA procedure is applied to two group means only, it will produce the same results as the t-tests introduced in the next sections.

If the data is normally distributed (see the tests for this in section 6.5.4), the t-test for two groups, or the ANOVA for multiple groups can be used for comparisons. If the data is not normally distributed, and each group has at least five observations, the Mann-Whitney U test or Kruskal-Wallis test can be applied. However, if less than five observations are available per group, it is better to use the t-test or ANOVA.

Tests, which assume that the data is normally distributed, are called parametric tests. Non-parametric tests do not make the normality assumption.

6.9.3 Threshold for p-value

The p-value is usually associated with a statistical test, and it is the risk that we reject the null hypothesis (see section 6.9.4), when it actually is true. Before testing, a threshold for p-value should be decided. This is a cut-off below which the results are statistically significant, and above which the results are not statistically significant. Often a threshold of 0.05 is used. This means that every 20th time we conclude by chance alone that the difference between groups is statistically significant, when it actually isn't.

If the compared groups are large enough, even the tiniest difference can get a significant p-value. In such cases it needs to be carefully weighted whether the statistical significance is just that, statistical significance, or is there a real biological phenomenon acting in the background.

6.9.4 Hypothesis pair

Before applying the test to the data, a hypothesis pair should be formed. A hypothesis pair consists of a null hypothesis (H_0) and an alternative hypothesis (H_1). For the tests described here, the hypotheses are always formulated as follows.

H_0 = There is no difference in means between compared groups

H_1 = There is a difference in means between compared groups.

6.9.5 Calculation of test statistic and degrees of freedom

The test statistic is a standardized numerical description of the differences between the group means. Depending on the test type, the actual mathematical formula for the calculation of the test statistic is different. Here we will present these formulas for a couple of t-tests.

The simplest t-test is the *one-sample t-test* . It is constructed to compare a sample group mean with a certain hypothesis. The general formula of the test is

$$T = \frac{M - \mu}{\frac{s}{\sqrt{n}}},$$

where M is the sample mean, μ is the expected sample mean (hypothesis), s is the standard deviation and n is the number of observations in the sample group. Degrees of freedom are calculated using the following formula:

$$df = n - 1$$

The means of two samples are compared with an *independent samples t-test* . In order to make the test, we need to know whether the variances of the groups are equal. As a rule of thumb, the variances of the groups can be assumed to be equal, if the variance of the first group is not more than three times larger than the variance of the second group. Assuming that the variances of the two-groups are not equal, the formula of the test statistic (Welsh's t-test) is:

$$T = \frac{X_i - X_j}{\sqrt{\frac{s_i^2}{n_i} + \frac{s_j^2}{n_j}}},$$

where X_i and X_j are the means of the compared groups, s_i^2 and s_j^2 are the variances of the compared groups, and n_i and n_j are the numbers of observations in the compared groups.

The degrees of freedom are calculated with a formula:

$$df = \frac{\left(\frac{s_i^2}{n_i} + \frac{s_j^2}{n_j}\right)^2}{\frac{(\frac{s_i^2}{n_i})^2}{n_i - 1} + \frac{(\frac{s_j^2}{n_j})^2}{n_j - 1}},$$

where s_i^2 and s_j^2 are the variances of the compared groups, and n_i and n_j are the numbers of observations in the compared groups.

If the variances of the groups are equal, the test statistic (student's t-test) is calculated with the formula:

$$T = \frac{\bar{x_1} - \bar{x_2}}{s \times \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

where

$$s = \sqrt{\frac{\sum_{i=1}^{n_1}(x_{1i} - \bar{x_1})^2 + \sum_{i=1}^{n_2}(x_{2i} - \bar{x_2})^2}{n_1 + n_2 - 2}}.$$

Degrees of freedom for the equal variances t-test are calculated as:

$$df = n_i + n_j - 2$$

Formulas for the test statistics of non-parametric tests are not presented here, but they can be easily found from any statistical textbook.

6.9.6 Critical values table

A critical values table contains the standardized values of a distribution. The critical value table for t-tests contains the standardized values of the t-distribution. Recall, that the shape of the t-distribution was partly defined by degrees of freedom. Therefore, dfs are also essentially needed when reading the critical values table for the t-distribution. Below (Table 6.1) is a sample of the ten first rows of a t-distribution table

Table 6.1: *Probability of exceeding the critical value.*

df	0.10	0.05	0.025	0.01	0.005	0.001
1.	3.078	6.314	12.706	31.821	63.657	318.313
2.	1.886	2.920	4.303	6.965	9.925	22.327
3.	1.638	2.353	3.182	4.541	5.841	10.215
4.	1.533	2.132	2.776	3.747	4.604	7.173
5.	1.476	2.015	2.571	3.365	4.032	5.893
6.	1.440	1.943	2.447	3.143	3.707	5.208
7.	1.415	1.895	2.365	2.998	3.499	4.782
8.	1.397	1.860	2.306	2.896	3.355	4.499
9.	1.383	1.833	2.262	2.821	3.250	4.296
10.	1.372	1.812	2.228	2.764	3.169	4.143

From the table, read the value at the intersection of the degrees of freedom and the set p-value threshold divided by two. For example, if the we used p-value of 0.05 and df = 9, the critical value would be 2.262.

6.9.7 Drawing conclusions

Compare the test statistic and the critical value from the table. If the calculated test statistic is larger, then the null hypothesis is rejected, and we can conclude that there is a difference between the groups with a certain p-value threshold.

6.9.8 Multiple testing

The more analyses are performed on a data-set, the more the results will meet the conventional significance level by chance alone. Often the p-value is corrected to

account for this problem. One commonly used correction is the Bonferroni correction, where the original p-value is divided by the number of comparisons to create a new corrected p-value against which those comparisons should be tested.

6.10 Analysis of variance

The purpose of the analysis of variance (ANOVA) is to test for significant differences between the means of several groups. If we are comparing two means, the ANOVA will give the same result as the t-test. However, unlike the t-test, ANOVA does not specify which of the groups are significantly different from each other, only that there are significant differences.

It may seem odd that a procedure that compares means is called analysis of variance. The name is derived from the fact that in order to test for a statistical significance between means, we actually need to compare or analyze variances.

6.10.1 Basics of ANOVA

At the heart of the ANOVA is a procedure where the variances are divided up or partioned. Variance is computed as the sum of squared deviations from the overall mean, divided by the corrected sample size. Thus, given a certain sample size, the variance is a function of sums of squares (squared deviations from the means), or SS. In the most simple form SS can be divided as SStotal = SStreatment + SSerror. The variance that is not explained by the treatment or group membership is thought be due to experimental errors.

There are some assumptions the data should fulfill in order to be eligible for the ANOVA analysis. First of all, the data should be normally distributed. Furthermore, variances of the compared groups should be similar. These assumption are robust to small deviation from these assumptions. The strictest assumption, for which the test is not robust, is the assumption of independence. The observations should be independent of each other. Therefore,a time series can not usually be analyzed by the ANOVA.

6.10.2 Completely randomized experiment

A completely randomized experiment is the simplest ANOVA design. This procedure allows us to compare the means of different treatments, and is also known as the one-way ANOVA. For microarray experiments, this would mean, for example, comparison of the mean expression of different cell lines or different subarrays. Before the experiment, subjects (here, chips) are randomly selected for the treatment groups. If randomization is not used, the effect of treatment might get masked by the effect of variation of the subjects.

Let us consider the following experiment (Table 6.2). We want to compare the effect of three different bacterial treatments on the same cell line. The nine available cell culture flasks are randomized into treatment groups. The experiment is conducted, and the mean expression of certain immunorelated genes is quantified.

The partition of variance is done in two phases. First, the SS within groups or SSerror (how much individual observations deviate from the group mean) is calcu-

lated using the groupwise means (Table 6.3). This reflects the various errors in the experiment. Then the SS between groups or SStreatment (how much the individual observations deviate from the overall mean) is calculated using the overall mean (Table 6.4). It reflects the effect of treatment on different groups.

SStreatment has k-1 degrees of freedom, where k is the number of groups. SSerror has N-k dfs, where N is the total number of subjects, and k is the number of groups. Mean squares (MS) are calculated by dividing the SSs by the concommitant dfs.

The results are summarized in an ANOVA table (Table 6.5), which reports sum of squares, and the F-statistic with an associated p-value. The F-test statistic is calculated as

$$F = \frac{MS_{treatment}}{MS_{error}}$$

The p-value reported in the table below has been read from the F-distribution table of critical values with the appropriate dfs and F-statistic.

Table 6.2: *Primary data.*

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	2	6	4
Observation 2	3	7	5
Observation 3	1	5	3
Mean	2	6	4
Overall mean		4	

Table 6.3: *Calculation of error terms.*

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	2-2=0	6-6=0	4-4=0
Observation 2	2-3=-1	6-7=-1	4-5=-1
Observation 3	2-1=1	6-5=1	4-3=1
Sum of squares	0+1+1	0+1+1	0+1+1
SS within groups		2+2+2=6	

Table 6.4: *Calculation of treatment effects.*

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	4-2=2	4-6=-2	4-4=0
Observation 2	4-3=1	4-7=-3	4-5=-1
Observation 3	4-1=3	4-5=-1	4-3=1
Sum of squares	4+1+9=14	4+9+1=14	0+1+1=2
SS total		14+14+2=30	

Table 6.5: *ANOVA table presents a summary of the results.*

	SS	df	MS	F	p-value
Effect	24	2	12	12	<0.01
Error	6	6	1		

The interpretation of the ANOVA *p*-value is similar to the t-test. Because the calculated *p*-value is lower than 0.05, we conclude that the null hypothesis is rejected, and the alternative hypothesis comes to power. More specifically, we know that there is a difference between the treatment groups, but we do not know which treatments differ from each other. T-test would give a definite answer, if we need to know specifically which groups are different.

The completely randomized ANOVA design presented above is also known as the one-way ANOVA, because the states of only one variable are studied. If there had been two variables, say the treatment and the cell-line, then the procedure would have been called a two-way ANOVA. There are many more experimental designs for the ANOVA, which are more thoroughly covered elsewhere (Sokal, Biometry).

6.11 Statistics using GeneSpring

A few examples of statistical data manipulation, simple statistics calculation and statistical testing using the DNA microarray data analysis software GeneSpring are given in this section.

6.11.1 Simple statistics

Genewise Simple statistics, like the average, minimum, maximum, standard error of the mean, and standard error can be produced through *Edit->Copy->Copy Annotated Genelist* in GeneSpring. After pasting the information in Excel or other spreadsheet program, the values will become apparent. These simple statistics can be produced for intensities of either channel (raw and control data in GeneSpring), intensity ratio and log-transformed intensity ratio (normalized data in GeneSpring).

6.11.2 Tranformations

In GeneSpring, data transformations are linked to Experiment Interpretation. There are three options to choose from: Non-transformed data (ratio) log_2-transformed data (log of ratio) and fold change . When one transformation is chosen, GeneSpring will automatically recalculate the data values, and use the new values for any subsequent statistical analyses (statistical group comparison, k-means clustering, etc.).

6.11.3 Scatter plot and histogram

A scatter plot can be produced in GeneSpring through *View->Scatter plot*. The shown axes can be modified from *View->Display Options*. A scatter plot can easily

be used for testing the linearity of the data.

A histogram is displayed, if you have selected the *View->View Graph*, and additionally have set up parameters so that the quantification results are shown separately for every hybridized chip. For example, in a simple time series experiment, setting time as a non-continuous parameter would produce a histogram of expression values. You can use histograms for assessing the distribution of the data. After log-transformation the distribution of expression values should be symmetric and one-peaked.

6.11.4 Correlation

The Pearson correlation between chips is automatically calculated. The values of correlation coefficients can be viewed through Condition Inspector. Condition Inspector is invoked when the right-hand mouse button is clicked over one chip in the navigator bar, and Inspect is selected. From the opening window, select the Similar Conditions tab. The correlation coefficients between the selected chip and all the other chips are reported in the column Correlation (Figure 6.10).

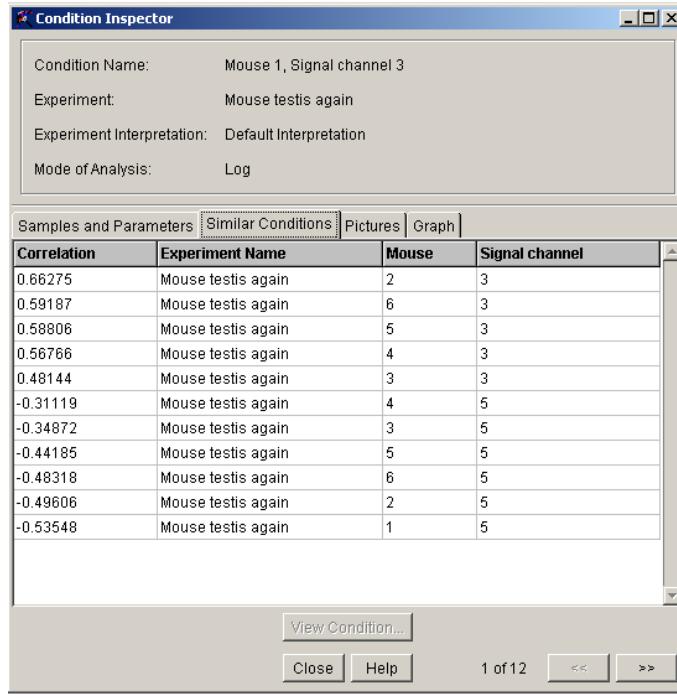

Figure 6.10: *The Pearson correlation in GeneSpring is found under the Similar Condition-tab in Condition Inspector.*

6.11.5 Linear regression

GeneSpring can calculate a linear regression model producing a line of best fit for a 2D scatter plot view. The line of best fit is produced from *View->Display options*.

Select the Lines to Graph tab, and tick Line of Best Fit box. The linear regression line is overlaid with the scatter plot, and the regression equation of form $y = aX + b$ is displayed at the bottom of the scatter plot view. Recall that a in the regression equation equals the correlation coefficient between the two variables plotted along the axes.

6.11.6 One-sample t-test

The one sample t-test in GeneSpring is automatically calculated for all the genes, whenever replicates are available. The t-test p-values can be found from the Gene Inspector, which is opened by double-clicking the left mouse button over a gene (Figure 6.11). The same p-values are also reported in the Spreadsheet, which is invoked from *File->View as spreadsheet*.

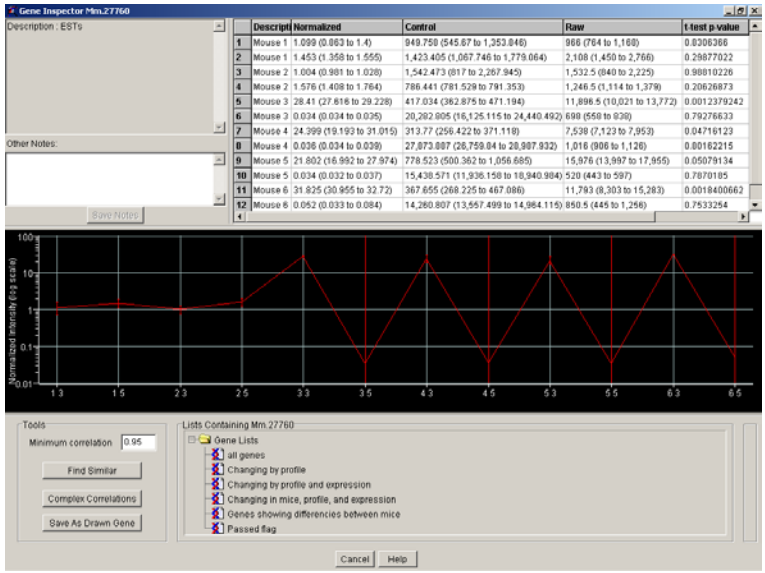

Figure 6.11: *In GeneSpring the p-values for the one-sample t-test are found from Gene Inspector*

6.11.7 Independent samples t-test and ANOVA

The independent samples t-test and ANOVA are located in *Tools->Filtering and statistical analysis*. Clicking the right hand mouse button on an experiment in the opening window enables one to Add Statistical Group Comparison. You can specify the parameters by which the compared groups are defined, select the groups to compare, select the appropriate statistical test and adjust p-value and multiple testing correction (Figure 6.12).

The result of the statistical group comparison is a list of genes, which are statistically differentially expressed between the specified groups. The actual p-values for these genes can be found from the Gene List Inspector, which can be opened by clicking a gene list with the right hand mouse button, and selecting

Inspect (Figure 6.13).

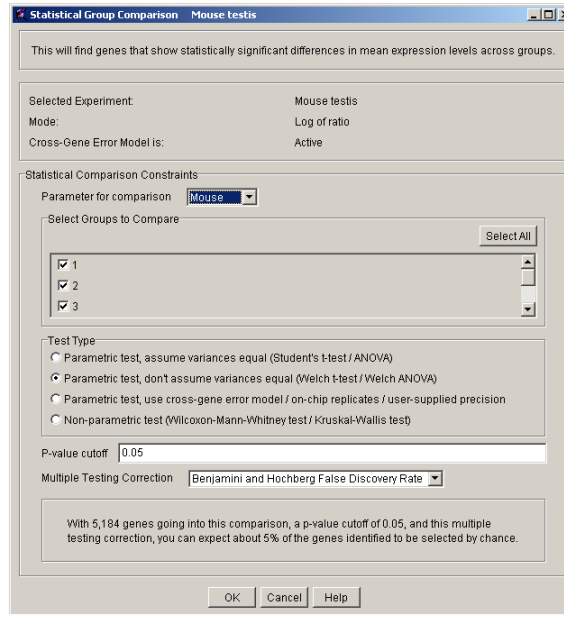

Figure 6.12: *In GeneSpring statistical group comparison is a tool of its own.*

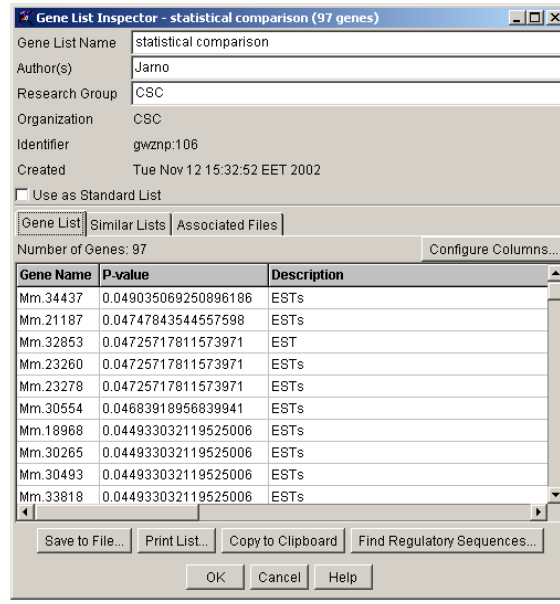

Figure 6.13: *The result of the statistical group comparison is stored into a genelist, which can be viewed with a Genelist Inspector.*

6.12 Suggested reading

1. Ranta, E., Rita, H., Kouki, J. (1997) Biometria, Yliopistopaino, Helsinki.
2. Sokal, R. R., Rohlf, F. J. (1992) Biometry, Freeman and co, New York.
3. Hopkins, W. G. (2003) A new View of Statistics, http://www.sportsci.org/resource/stats/index.html.
4. Pezzullo, J. C. (1999) Interactive Statistical Calculation Pages, http://members.aol.com/johnp71/javastat.html.
5. Statsoft Inc. (2002) Electronic Statistics Textbook, http://www.statsoft.com/textbook/stathome.html.

This chapter was written by Jarno Tuimala.

Part II

Analysis

7 Preprocessing of data

7.1 Rationale for preprocessing

Preprocessing includes analytical or transformational procedures that need to be applied to the data before it is suitable for a detailed analysis.

The black-box thinking, where data is fed into a program, and the result pops out, is gaining ground rapidly. This kind of approach for statistical analysis in simply erraneous, because the results coming out from the program can be statistically erraneously derived. In such cases, also the biological conclusions can be wrong. Statistical tests have often strick assumptions, which need to be fulfilled. Violation of assumptions can lead to grossly wrong results.

We strongly recommend that the researcher, even if he/she is not himself performing the data analysis, gets basic knowledge of the data, because the results presented by the bioinformatician or statistician are more easily interpretable, if one is at least somewhat familiar with the data.

Here we will introduce some methods for checking the data for violation of statistical test assumptions. We also present some things to consider before and during the data analysis. The methods are introduced in the order of applicability, although some methods are needed in several steps.

7.2 Missing values

There are often many missing values in microarray data. As you might recall, missing values are observations (intensities of spots), where the quantification results are missing. In the context of microarrays, we define missing values as

- Missing because the spot is empty (intensity = 0).
- Missing because background intensity is higher than the spot intensity (background corrected intensity < 0).

Missing values can lead to problems in the data analysis, because they easily interfere with computation of statistical tests and clustering. Therefore, it is worth giving a thought to the treatment of missing values. There are a couple of options for the treatment of missing values: They may be replaced with estimated values in a process called imputation, or they can be deleted from the further analyses.

The default way of deleting missing data (in most of the software packages), for example while calculating a correlation matrix, is to exclude all cases that have missing data in at least one of the selected variables; that is, by casewise deletion of

missing data. However, if missing data are randomly distributed across cases, you could easily end up with no "valid" cases in the data set, because each of the genes will highly likely have at least one missing observation on some chip. The most common solution used in such instances is to use the so-called pairwise deletion, where a statistic between each pair of variables is calculated from all cases that have valid data on those two variables.

Another common method is the so-called mean substitution of missing data (mean imputation, replacing all missing data in a variable by the mean of that variable). Its main advantage is that it produces internally consistent sets of results. Mean substitution artificially decreases the variation of scores, and this decrease in individual variables is proportional to the number of missing data. Because it substitutes missing data with artificially created average data points, mean substitution may considerably change the values of correlations. Imputation is commonly carried out for intensity ratios, but can also be done for raw data.

Different computer programs manipulate missing values very differently, and drawing any consensus would be futile. At least statistical programs often offer a possibility to define whether to use imputation, pairwise deletion or casewise deletion.

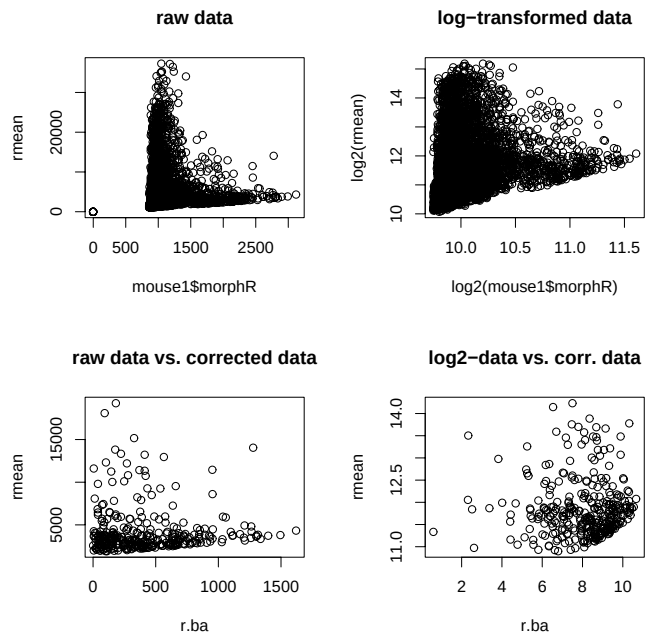

Figure 7.1: *Raw and log-transformed intensity values of the red channel plotted against its background. Upper row contains uncorrected scatter plots, lower row background corrected scatter plots.*

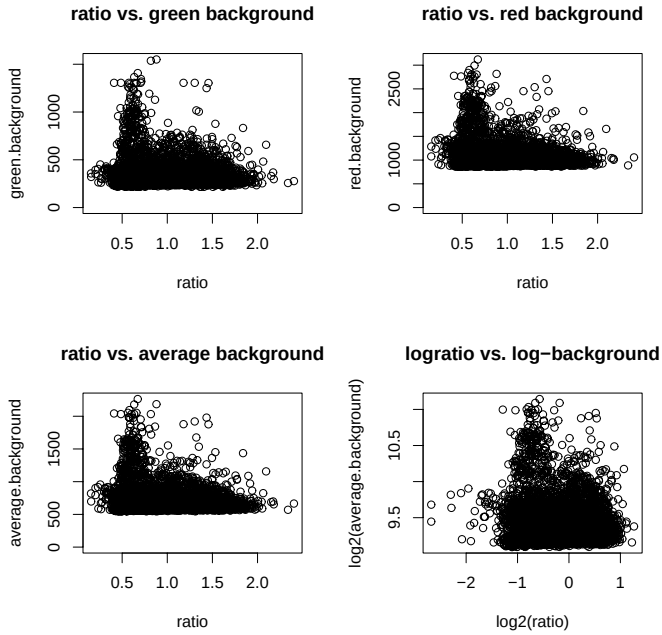

Figure 7.2: *Intensity ratios (expression) plotted against the background of the channels.*

7.3 Checking the background reading

There is some dispute on whether the background intensities should be subtracted from the spot intensities. At the moment, the background adjustion seems to be commonly used, but strictly speaking, it's applicability should be checked.

The background intensities should not correlate highly with the spot intensities, if the spot intensities are truly independent of the background intensities. This can be assessed by a scatter plot of background intensities against spot intensities. If the spot intensities are dependent on the background intensities, it is possible either not to apply any background correction to the data (Figure 7.1 and Figure 7.2) or discard the deviating observations from further analyses.

Another common problem with the background correction is that it might produce "pheasant tail" images on the scatter plot (Figure 7.3). Pheasant tails are formed by observations, which have exactly the same intensity value on one channel, but the other channels' intensities can vary. In such cases long vertical or horizontal straights of observations are produced, and the resulting data cloud resembles a pheasant tail in a scatter plot. This is especially common in the lower end of the intensity distribution, and can be a sign that the scanner is not very reliable below a certain cut-off intensity or that the image analysis software calculates background intensities in a peculiar way. When pheasant tails are observed, background *un*corrected intensity values can be used the analyses, or the deviating observation can be excluded from any further analyses.

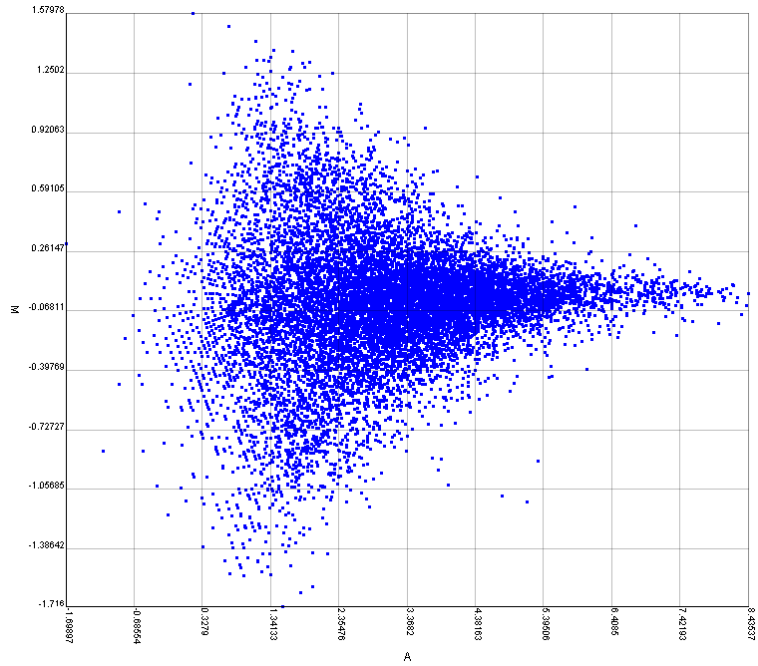

Figure 7.3: *An A versus M plot, where the pheasant tail is visible in the lower intensity values.*

The background corrected intensity values are calculated spotwise by subtracting the background intensity from the spot (foreground) intensity. The formulas for Affymetrix and two-color data are

$$Cy3' = Cy3_{spot} - Cy3_{background}$$

$$Cy5' = Cy5_{spot} - Cy5_{background}$$

Note, that Affymetrix arrays do not quantify background around the spots, but instead use a certain mathematical formula (check the Affymetrix manual).

7.4 Calculation of expression change

There are three commonly used measures of expression change . Intensity ratio is the raw expression value, and log ratio and fold change are transformationally derived from it.

7.4.1 Intensity ratio

After background correction, expression change is calculated. The simplest approach is to divide the intensity of a gene in the sample by the intensity level of the same gene in the control. Intensity ratio can be calculated from background corrected or uncorrected data (here we use corrected data) The formula for two-color data is

$$\text{Intensity ratio} = \frac{Cy3'}{Cy5'}$$

For Affymetrix data, substitute Cy3' and Cy5' with the appropriate intensities from the sample and control chips.

This intensity ratio is one for an unchanged expression, less than one for down-regulated genes and larger than one for up-regulated genes. The problem with intensity ratio is that its distribution is highly asymmetric or skewed. Up-regulated genes can take any values from one to infinity whereas all downregulated genes are squeezed between zero and one. Such distributions are not very useful for statistical testing.

7.4.2 Log ratio

To make the distribution more symmetric (normal-like) log-transformation can be applied. The most commonly used log-transformation is 2-based (log_2), but it does not matter whether you use natural logarithm (log_e) or base 10 logarithm (log_{10}) as long as it is applied consistently to all the samples.

The formula for the calculation of log_2-transformation is:

$$\text{Log ratio} = log_2(\text{Intensity ratio}) = log_2\left(\frac{Cy3'}{Cy5'}\right)$$

After the log-transformation, unchanged expression is zero, and both up-regulated and down-regulated genes can take values from zero to infinity. Logratio has some nice properties compared with other measures of expression. It makes skewed distributions more symmetrical, so that the picture of variation becomes more realistic.

In addition, normalization procedures become additive. For example, we have the following intensity ratio results for two replicates (A and B) after normalization:

$$GeneA = \frac{120}{60} = 2.0$$

$$GeneB = \frac{30}{60} = 0.5$$

The mean of these replicates is 1.25 instead of 1, which would have been expected. If the 2-based logarithmic transformation is applied, the log ratios are:

$$GeneA = log_2\left(\frac{120}{60}\right) = 1.0$$

$$GeneB = log_2\left(\frac{30}{60}\right) = -1.0$$

The mean of these log ratios is 0, which corresponds to the mean intensity ratio of 1. Although log-transformation is not always the best choice for microarray data, it is used because the other transformations lack this handy additive property. One downside of the log-transformation is that it introduces systematic errors in the lower end of the expression change distribution.

7.4.3 Fold change

Another means to make the distribution of intensity ratios more symmetrical is to calculate the fold change. The fold change is equal to the intensity ratio, when the expression is higher then one. Below one, the fold change is equal to the inversed intensity ratio.

$$\text{For values } >1\text{, fold change} = \frac{Cy3'}{Cy5'}$$
$$\text{For values } <1\text{, fold change} = \frac{1}{(Cy3'/Cy5')}$$

The fold change makes the distribution of the expression values more symmetric, and both under and over-expressed genes can take values between zero and infinity. Note, that the fold change makes the expression values additive in a similar fashion as the log-transformation.

7.5 Handling of replicates

Replicates are a very powerful way to reduce noisiness of the data. Noisiness (random errors) is an undesirable feature of the data, because it potentially abolishes interesting information. It can results from many sources, some of which are hard to deal with.

7.5.1 Types of replicates

There are at least three different kinds of replicates that are potentially useful in the context of DNA microarray analysis. For example, if we treat a certain cell line with a cancer drug, we can set up several culture flasks, and then harvest them as biological replicates. For every culture flask, the isolated and labeled mRNA population can be hybridized to several chips making a number of technical replicates . In addition, every hybridized chip can be quantified several times or using different image analysis software (software replicates) .

These are all valuable sources of information about the variability in the data. In practise, it might be a good idea to make some biological replicates and some technical replicates. Then the biological and technical variation can be taken into account in the same experiment.

7.5.2 Time series

In a time series experiment expression changes are monitored with samples taken between certain time intervals. Although several replicates can be made per every time point, it should be considered that these replicate chips can possible be made a better use of, if they are added to the time series as sampling points. That is, it should be weighted whether a high precision in every time point is more valuable than the additional information of expression changes new sampling points (time points) produce.

7.5.3 Case-control studies

For case-control studies, where for instance cancer patients are compared with their non-diseased controls, the individuals in both groups can be considered replicates of that disease state. This is assuming that we are interested in the differencies of those two disease states instead of inter-individual variation.

7.5.4 Power analysis

Using power analysis we can easily compute the number of replicates that are needed to reliably discover the genes that are expressed. For example, we would need 11 replicates to reliably pick up the genes, which are at least 1.41-fold over-expressed at a p-value of 0.05. Similarly, 6 replicates are needed if genes with 2-fold over-expression need to be quantified using a p-value cutoff of 0.01.

7.5.5 Averaging replicates

The goal of DNA microarray experiments is to semi-quantitatively analyze the gene expression level in a certain material. Often the precision of the expression level estimate is important. The estimate of the true expression necessarily becomes more precise when more replicates are used for that purpose.

Replicates also produce data on variability between the measured states, for example cancer patients and healthy controls. More importantly, replicates potentially reduce noise in the data. However, if replicates are averaged, we lose information about the inter-individual variability in the studied groups.

Assume that we have an experiment, where there are two biological replicates and three technical replicates of both biological replicates, making a total of 6 replicates altogether. The correlation between any of the technical replicates from the two biological replicates is, at first, quite low (<0.5). Taking average of the technical replicates makes the correlation between biological replicates much higher (>0.8). This also hints that the noisiness of the data has been reduced. Of course, bad replicates (chips or spots) should ideally be removed before calculation of averages, even though averaging diminishes the influence of outliers on the results.

In practise, replicates are often averaged after normalization, but sometimes even the raw intensity values are averaged. We feel that it would be better to average replicates after normalization, because then it is possible to take chipwise variation into account in the normalizations.

7.6 Checking the quality of replicates

If the experiment includes replicates, their quality can be checked with simple methods, using scatter plots and pairwise correlations or hierarchical clustering techniques, which are explained in the cluster analysis chapter in more detail. Often the quality of replicates is checked before and after the normalization. If the correlation between replicates drops dramatically after certain normalization, the applicability of the normalization method should be reconsidered.

7.6.1 Quality check of replicate chips

The first task is often to find out, how well two replicate chips resemble each other. In such cases the intensity or log ratios can be plotted along the axes of a scatter plot. Three replicates can be plotted using a three dimentional scatter plot. If the replicates are completely similar, the data points in the scatter plot fall onto a perfectly straight line running through the origin of the plot. If the replicates are not exactly similar, the observations form a data cloud rather than a straight line. The absolute deviation from a line can be better visualized, if a linear regression line is fitted to the data.

Correlation measures the linear similarity quite well. It might give misleading information, if the data cloud is not linear. Correlation coupled with a scatter plot gives much more information. In practise, the Pearson's correlation coefficients between two replicate cDNA chips produced from the same mRNA pool (technical replicates) are often in the range of 0.6–0.9, and the correlation between similarly hybridized Affymetrix chips is typically over 0.95. When biological replicates are performed, the correlation between replicate chips usually drops from these values. For cDNA chips, a correlation of 0.7 between replicates may be considered good, whereas for Affymetrix chips a correlation over 0.9 may be considered a good result.

In addition, the scaling factor should be checked for the Affymetrix chips. If the factor is larger than 50, the hybridization is probably bad.

Quality of several replicate chips is most easily checked by hierarchical clustering. For example, if the two replicate chips are placed closest to each other in the dendrogram, they can be expected to be good replicates. Depending on the similarity measure, the hierarchical clustering can be applied either after (Euclidian distance) or before the normalization (Pearson's correlation).

7.6.2 Quality check of replicate spots

The quality of replicate measurements of one gene (one spot) can be assessed in a similar fashion to replicate chips, using correlation measures. However, this is more laborous than with replicate chips, but it can sometimes be essential that bad replicate spots are found and removed from the data.

7.6.3 Excluding bad replicates

It is quite common to exclude bad replicates from further analyses. For example, if there are four replicate chips available for the same treatment, and one seems to deviate from every other replicate, then this most deviating one can be removed from any further analyses. This often results in a massive loss of data, because whole chips are removed from the analysis. Thus, it would be better to study the quality of replicates on a gene-wise basis, after initially studying the quality of whole chips using the tools described above.

In practise, there often are replicates which deviate from the others. These can be results of modified experimental conditions (different mRNA batch, etc.) or just important biological variation, if different animals of human subjects are studied.

Because of this, some caution should be exercised when removing bad replicates.

7.7 Outliers

Outliers in chip experiments can occur at several levels. There can be entire chips, which deviate from all the other replicates. Or there can be an individual gene, which deviates from the other replicates of the same gene. These deviations can be caused by chip artifacts like hairs, scratches, and precipitation. Precipitation, among other causes, can also perturb one single spot or probe.

Outliers should consist mainly of quantification errors. In practise, it is often not very easy to distinguish quantification errors from true data, especially if there are no replicate measurements. If the expression ratio is very low (quantification errors) or very high (spot intensity saturation), the result can be assumed to be an artifact, and should be removed. Most of the actual outliers should be removed at the filtering step (those that have too low intensity values), and some ways to identify deviating observations were presented in the section on checking replicates.

In the absence of replicate, the highest and lowest 0.3% of the data (gene expression values on one chip) is often removed, because assuming normality, such data resemble outliers. These values are outside the range of three standard deviations from the distribution mean. This is often equivalent to a filtering, where observations with too low or high intensity values are excluded from further analyses.

Formally, a statistical model of the data is needed for the reliable removal of outliers. The simplest model is equality between replicates. If one replicate deviates several standard deviations from the mean of the other replicates, it can be considered an outlier and removed. The t-test measures standard deviation and gives genes, where outliers are present among replicates, a low significance.

In practise, it is easiest and best to couple the removal of outliers with filtering , for example, using the following procedure: First, genes outside the range of three standard deviations from the chip mean are removed. In the succeeding filtering steps, the genes whose intensity is too low are removed, if needed. Next, genes which do not show any expression changes during the experiment are eliminated (this can be based on either log ratio or standard deviation of log ratios). What is left in the end, is the good quality data. There are also some advanced statistical methods developed that also allow for outlier detection and removal, but they are outside the scope of this book.

7.8 Filtering bad data

Filtering is a process where observations which do not fulfill a preformulated presumption are excluded from the data. For example, there is a certain limit of the scanner below which the intensity values can not be trusted anymore. Typically, the lowest intensity value of the reliable data is about 200 for Affymetrix data and 1000 for cDNA microarray data. These cut-offs are likely to change as the scanners get more precise. The values below the cut-off point are usually removed (filtered) from the data, because they are likely to be artifacts.

The idea of filtering is simple. We want to remove all the data we do not have confidence in, and proceed with the rest of the data. The results based on the trustworthy data are often biologically more meaningful than the results based on very confusing or noisy data.

The first step of filtering is flagging. Flagging is performed at the image analysis step. It's idea is to mark the spots which are, by eye, judged to be "bad". Then, in the data analysis phase, the spots, which were flagged as "bad" are removed from the data. For example, the spots overlapped with a sizable dust particle should be flagged as bad and removed at the data filtering step.

There are a couple of statistical measures that can be used for filtering. Some image analysis programs give signal-to-noise or signal-to-background measurements for every spot on the array. These quality measurements can be used for filtering out bad data. Often a cut-off point of 90–95% for signal-to-noise ratio is used, at least on control channel.

Signal-to-background can be calculated after the image analysis, from the intensity values:

$$\frac{\text{spot signal on green channel}}{\text{background signal on green channel}}$$

The spots can then be filtered on the basis of this quality statistic. Signal to background plots of both channels appear often very much the same as the original data (Figure 7.4).

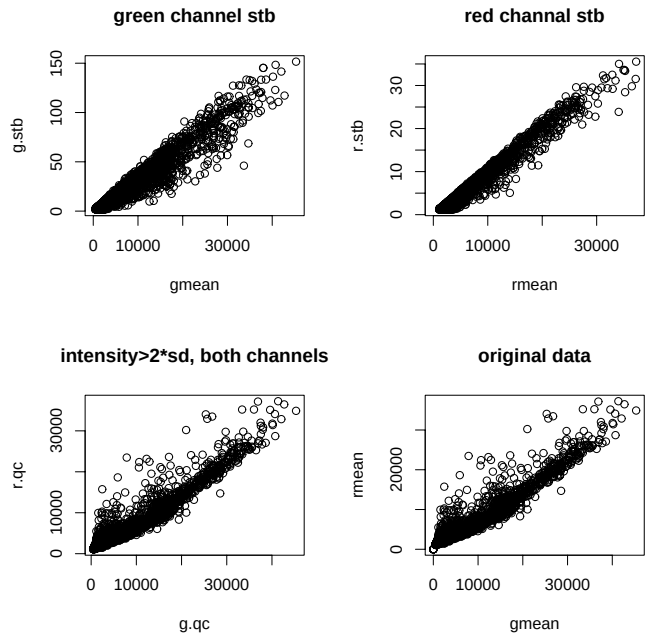

Figure 7.4: *Raw intensities and log-transformed intensities of different channels plotted against their signal-to-background ratios (stb).*

It would also be expected that the signal to background ratio would increase as the intensity of the channel increases, if the background is approximately the same in all areas of the chip (Figure 7.5). This is one hallmark of nice-quality data.

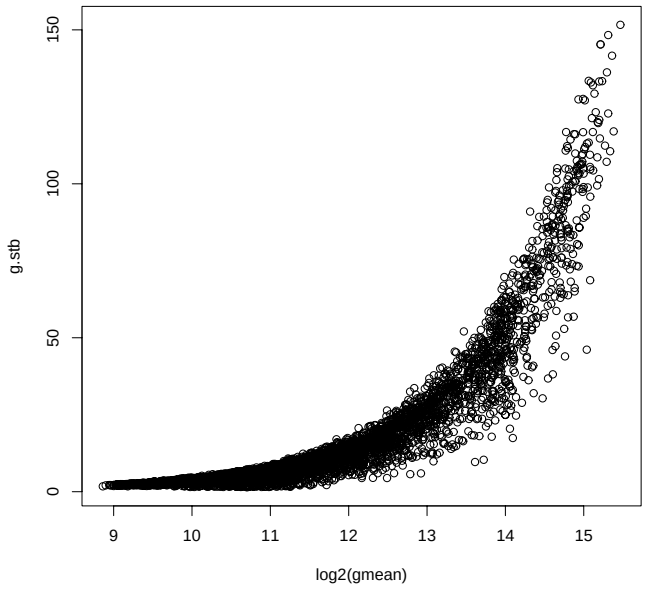

Figure 7.5: *The signal-to-background ratios (stb) increase as the intensity of the spot increases.*

What is also often used, is the filtering by the standard deviation of background intensities. The observations, which have an intensity value lower than background + 2–6 times standard deviation of the background intensities, are often removed from the data.

7.9 Filtering uninteresting data

After the removal of bad data, we are left with the good quality data, of which most is probably uninteresting for us. If the goal of the study is to find a couple of dozens of genes for further studies of the biologically interesting phenomenon, it is a good idea to remove the uninteresting part of the data before clustering or classification analyses. Uninteresting data comprises of the genes that do not show any expression changes during the experiment.

For example, assume that we have a time series, and we want to find the genes that have a most upregulated expression at the end of the sampling. In such cases, genes that remain unchanging during the study, are removed from further analyses. This is because the clustering methods generally work better, if the uninteresting data is not swamping the changes in the interesting part of the data.

Similarly, if we are interested in finding the genes that differentiate between two groups of patient, the genes that do not show any change between these groups, can be removed from the analysis.

Most often the intensity ratio cut-offs for uninteresting data (not-changing genes) are set at 0.5 and 2.0. These cut-offs also roughly correspond to the thresholds of reliable data: All the variation between these cut-offs can be assumed to be due to random quantification errors of the non-changing genes. The modern chip technology is able to produce better chips, and in practise these cut-offs can often be set on a lower level, for example to 0.7 and 1.5. The cut-offs can be better characterized using a Volcano plot (see chapter on genelists).

7.10 Simple statistics

There are some simple statistical measures of the data, which can be performed before or after normalization. They will give hints as to what statistical tests to apply or what is the distribution of the observations. These statistics should be checked from the distribution of log-transformed data.

7.10.1 Mean and median

When the distribution of the intensity values is skewed, the median characterizes the central tendency better than the mean. The median and mean can also be used to check the skewness of the distribution. For symmetrical distributions mean and median are approximately equal.

7.10.2 Standard deviation

The standard deviation gives you an idea of how spread your data are. It is also good to check the coefficient of variation (standard deviation / mean). This should give you a better idea of the real spread of the data, because the value of CV is independent of the absolute values of the variable. For example, replicates should have a small CV, although the value of standard deviation can vary a bit.

7.10.3 Variance

Sometimes you need to make a decision of the appropriate statistical test. For example, there are different kinds of t-tests for variables with equal or unequal variances. As a rule of thumb, the variances can be assumed to be equal if the variance of the first variable if less than three times the variance of the second variable.

7.11 Skewness and normality

Skewness and normality should be checked after normalization, but it is usually worthwhile, if they are checked before normalization, too. Then the distribution of the data (log-transformed intensity ratio) can be compared before and after normalization. This can be partly used for checking the success of the normalization.

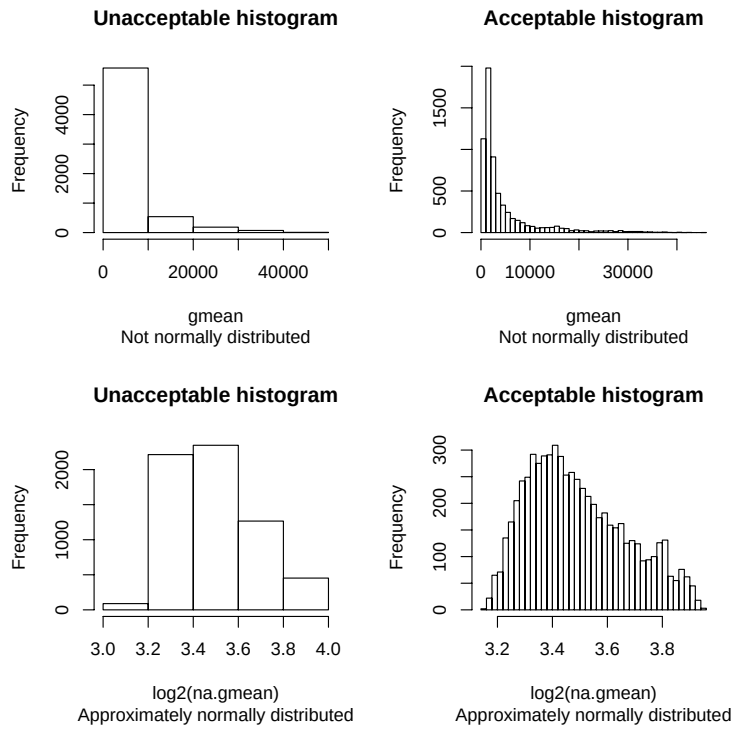

Figure 7.6: *Four histograms of the same data. The two upper histograms contain the raw data, and the two lower histograms the log-tranformed data. The log-transformed data is clearly more normal-like than the non-transformed data.*

Normality means how well your data fits to the normal distribution. This should be checked, because most of the statistical procedures assume that the data is normally distributed. Even the most basic descriptive statistics can be misleading if the distribution is highly skewed; standard deviation does not bear a meaningful interpretation if the distribution significantly deviates from normality.

The easiest method for checking normality and skewness of the distribution is to draw a histogram of the intensities (Figure 7.6). For checking the skewness, comparison of the mean and median is complementary to this graphical method. Note that there should be enough columns in the histogram in order to make the results reliable.

There is also a more informative way to check for the normality, quantile-quantile plot. This should be used with the histograms as a more diagnostic method for observing the deviations from normality.

7.11.1 Linearity

It is also important to check the linearity of the data (log ratio). Linearity means that in the scatter plot of channel 1 (red colour) versus channel 2 (green colour), the relationship between the channels is linear. It is often more informative to produce

a scatter plot of the log-transformed intensities, because then the lowest intensities are better represented in the plot (Figure 7.7). In this kind of a plot, the data points fit a straight line, if the data is linear.

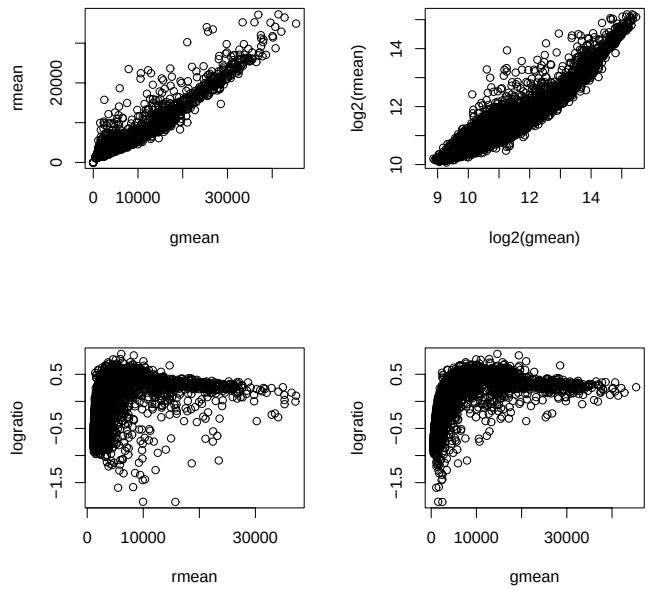

Figure 7.7: *Checking the linearity of the data. Top row, red channel intensities versus green channel intensities; bottom row, log ratio versus different channel intensities.*

Another way to test linearity is to plot the log ratio versus the intensity of one channel (Figure 7.7). This should be done for both channels independently. In this plot the data cloud has been tilted 45% to right, and it should be easier to identify non-linearity. Again, the data points should fit an approximately straight line, which in this case is horizontal.

Checking the linearity of the data helps to pick the right normalization method. It also provides information about the reliability of the data, especially in the lower intensity range.

7.12 Spatial effects

Often the intensity values (especially background) vary a lot on the same chip depending on the location of the spot on the chip. This effect is known as spatial effect (Figure 7.8).

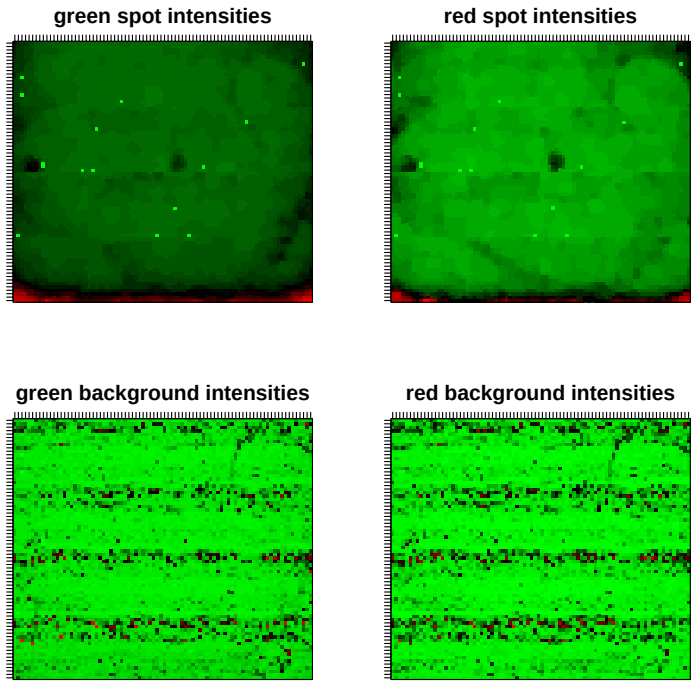

Figure 7.8: *Intensities of spots and their backgrounds spotted in the form of the original chip in order to check for spatial biases. Background intensities seem to be highly variable in different areas of the chip. Especially the corners have an abnormally high background. These areas might be removed from further analyses.*

Spatial effects are often most pronounced in the corners of the chip, where the hybridization solution has possibly been squeezed thinner than in the other areas of the chip. Furthermore, corners of the chip can also dry up more easily than other areas of the chip.

Another source of spatial effects is the regional clustering of genes with a similar expression profile. This is often caused by bad chip design, where the probes have not been randomized onto the chip. For example, if genes acting in a cell cycle regulation are spotted into one corner of the chip, the expression values in the corner are probably dependent on the cell cycle stage of the sample material. In such cases, the spatial effect should not be removed by normalization procedures, because it contains real biological information instead of random noise.

The similar kind of spatial effect can also be checked for from the calculated expression values (Figure 7.9). There should not be any suspicious areas on the subarrays (assuming good chip design), where the intensity values are much higher than everywhere else on the same chip.

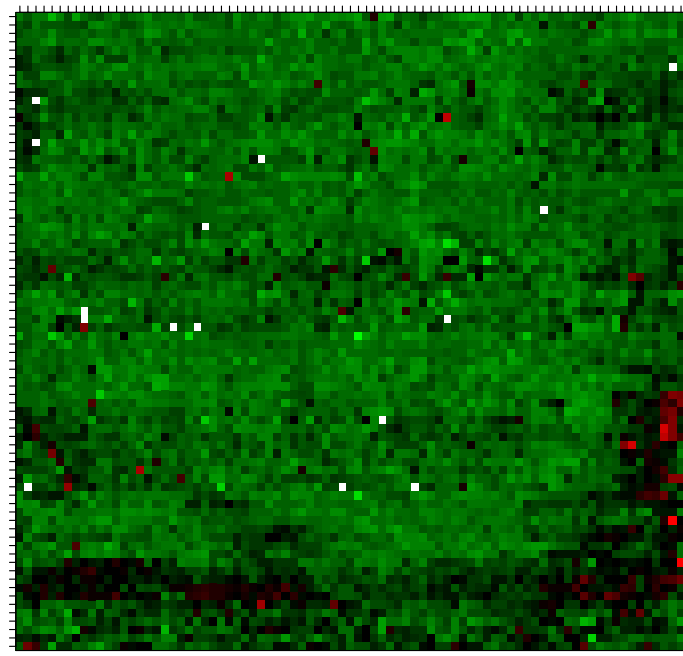

Figure 7.9: *The measured gene expression spotted in the form of the original chip in order to check for spatial biases.*

If the spatial effect is identified as random noise, bad hybridization conditions or other undesirable sources of errors, it can be compensated for by a suitable normalization method (subarraywise methods).

7.13 Normalization

Normalization can be regarded as a preprocessing step, but it is so central a method for the microarray experimentation that it warrants a chapter of its own. Thus we will not discuss normalization in more detail here, but instead chapter 8 is dedicated for the issue.

7.14 Similarity of dynamic range, mean and variance

Many statistical procedures assume that across the experiment, the dynamic range (minimum and maximum), mean and variance of the chips are equal. These should be checked after normalization. If they are not, the results of the statistical tests should be interpreted with caucion.

7.15 Examples using GeneSpring

This section describes some preprocessing examples using the GeneSpring program.

7.15.1 Importing data

Before any preprocessing can be done for the data using GeneSpring, it should be imported into the program. GeneSpring read virtually any file format if it is a tab-delimited text file. In GeneSpring you need to specify what kind of information some key columns contain. Basically, the minimal datafile contains just two columns: one for the gene name (it can also ne Genbank accession number) and another one for the intensity (Affymetrix) or intensityratio (cDNA chips). Because importing data is covered very thoroughly in the online manual accessible from *Help->Online documentation*, we do not go into details here.

7.15.2 Background subtraction

GeneSpring subtracts the background from the foreground or spot intensities automatically, if the background intensities are present in the datafile.

7.15.3 Calculation of expression change

For cDNA microarrays this is one normalization option, Per Spot: Divide by control channel, but the option is not used with lowess-normalization. There are two possible transformations, log-transformation (log of ratio) and fold change. The transformation options can be accessed from *Experiments -> Experiment interpretation*.

For the Affymetrix chips, a calculational control channel is created. The expression is then calculated as with cDNA microarrays. It is also possible to calculate the expression change using a certain control chip. This is one of the normalization options, Per Spot: Normalize to control samples.

Calculation of the expression change is coupled to normalization. Normalization only affects the control channel, which is adjusted so that the median intensity ratio of the chip is one when using Per chip: Normalize to percentile normalization. Therefore, the control channel is created for the Affymetrix chips, also. Note, that the original raw intensity values are always retained.

7.15.4 Replicates

When importing the data from a chip where the same spot is present in multiple copies, GeneSpring automatically calculates an average of those replicates. This is assuming that the replicate spots have exactly the same Gene identifier in the data file. Replicate chips are also averaged, after defining the replicates in the *Experiment -> Experiment parameters* window and setting up the parameters in *Experiment -> Experiment interpretation*.

For example, if we have a time series experiment (Table 7.1) with three time points and two replicates per every time point, the parameters should be set up

Table 7.1: *Time series experiment*

	Time point	Replicate
chip 1	1 hour	1
chip 2	1 hour	2
chip 3	2 hours	1
chip 4	2 hours	2
chip 5	3 hours	1
chip 6	3 hours	2

as follows. Two parameters, a time point and a replicate are created. In the time point, mark the time points suitably. The replicates are then set up using the other parameter. Inside one time point, the replicates are marked with a running number. Last, set up the value order for the time points. This tells GeneSpring, in which order the time points should be displayed on the screen.

If there is missing data for some genes either when importing the data or when setting up the replicate chips, GeneSpring only uses the existing data for the calculation of means.

7.15.5 Checking linearity

Linearity is easily checked in GeneSpring using the M versus A plot. Go to scatter plot (*View -> Scatter plot*). Change the display options (*View -> Display options*) so that the horizontal axis is the average of raw and control (A), and the vertical axis is normalized (M). To help plot the non-linearity, from the Lines to Graph tab tick the Line of best fit box, which draws the linear regression line to the plot. Note, that the M versus A plot should be initially produced for unnormalized data.

7.15.6 Normality

Normality can be checked using the histogram. A histogram can be investigated in (*View -> Graph*) window by checking *All Samples* - interpretation mode, that is automatically produced for each experiment.

7.15.7 Filtering

A filtering tool can be invoked from the menu *Tools -> Filtering and statistical analysis*. A new window opens, where one genelist and one experiment should be selected. The actual filtering tool is accessed by right-clicking on the selected experiment and selecting Add expression percentage restriction from the opening list. Often the bad quality data is first filtered out. After that, the not-changing genes are removed from the dataset.

Using the scatter plot tool , define the intensity value, under which you can't trust your data anymore on either raw or signal channel. For cDNA chips, this is often around 200-1000 and for Affymetrix chips around 200. In the expression percentage filtering, use this signal value as a minimum cut-off. Also select that it

applies to all the conditions. Create one such filter for both channels. Create a new genelist of the results (Make list button).

In the next filtering phase we will try to find the genes that are changing and we can trust in. Using the genelist created above, set up a new filter, where you select genes, which have expression values between 0.5 (minimum) and 2 (maximum). These genes are not showing any expression changes and are uninteresting to us. This filtering should also apply to the whole dataset (all conditions). After saving the new genelist, the filtering tool can be closed.

Go to the navigator bar and right-click on the good quality gene list. From the list, pick *Venn diagram -> Left (red).* Similarly, add the not-changing and all genes genelist to the Venn diagram. Then select the All genes genelist from the navigator. From the Venn diagram identify the region that contains the genes included in the reliable genelist, but not in the not-changing genelist. Right-click on that area of the diagram, and make a list of these genes.

Now you have a list of genes, which are reliable and also changing. You're ready to proceed with the analysis, for example to clustering or classification analyses.

7.16 Suggested reading

1. Kohane, I. S., Kho, A., Butte, A. J., (2002) Microarrays for an intergrative genomics, MIT press, Massachusetts.

2. Baldi, P., Hatfield, G. W. (2002) DNA microarrays and gene expression, Cambridge University Press, United Kingdom.

3. Knudsen, S. (2002) A biologist guide to analysis of dna microarray data, John Wiley and Sons Inc., New York.

This chapter was written by Jarno Tuimala.

8 Normalization

8.1 What is normalization?

There are many sources of systematic variation in microarray experiments that affect the measured gene expression levels. Normalization is the term used to describe the process of removing such variation [12]. Normalization can also be thought of as an attempt to remove the non-biological influences on biological data.

Sources of systematic variation will affect different microarray experiments to different extents. Thus, to compare microarrays from one array to another, we need to try to remove the systematic variation, to bring the data from the different experiments onto a level playing field, so to speak. One goal of normalization is also to make possible the comparison from one microarray platform to another, but this is clearly a much more complicated problem and is not covered in this section.

The biggest problem with the normalization process is the recognition of the source of systematic bias. There is a strong possibility that some or even most of the biological information will also be removed when normalizing the data. Thus it is good to keep in mind that the amount of normalization should be minimized to avoid losing the real biological information. In the next chapter some of the most common sources of systematic bias will be introduced and also some methods for recognizing the sources and dealing with the bias will be discussed.

8.2 Sources of systematic bias

8.2.1 Dye effect

Differences in dye (labeling) efficiencies are the most common source of bias and also easily identifiable. This can be seen when the intensity of one channel on the array is much higher than the other (see Figure 8.1, B and C). Dye effect can be corrected for by balancing the dyes using the assumption that both channels should be equally "bright". Extra information for dye balancing can be received from dye-swap experiments. Problems will occur when dye also has interaction effects; *i.e.*, the labeling efficiency may depend on the gene sequence.

8.2.2 Scanner malfunction

Scanners might show many different failures. When laser or PMT intensity values are wrongly adjusted, the scanner can cause the dye effects to show up. But most of the scanner malfunctions are hard to deal with *in silico* and the best solution is to

fix the scanner and repeat the scanning. For example, when lasers are misaligned the two channels are slightly out of register. This can cause big problems if image analysis software does not allow the user to align images manually.

8.2.3 Uneven hybridization

Sometimes patterns are seen on the slide that can be caused by many different reasons. The most common ones of these are described in the next three sections.

When spatial bias is seen on the slide, the first impression is usually that it is caused by uneven hybridization. In most cases this is also true. Uneven hybridization can be recognized, for example, as lighter areas in the middle of the slide or on the edges of the slide. This is usually very hard to fix numerically and it is recommended to aim for developing more consistent techniques for hybridization. For single cases, spots that are not hybridized properly can be excluded from further analyses. Background difference can also cause bias (see Figure 8.1, E).

8.2.4 Printing tip

Slides are usually printed using more than one pen (2, 4, 8, 16, ..). If any of these pens work differently from others, for example a pen gets infected by hair or is defective in some other way, the corresponding subarray could differ from other subarrays. Quite commonly printing pens also wear out at a different rate. One way to see if one pen performs differently is to visualize the data by subarrays using colors or regression lines so that the faulty subarray stands out. In some cases, printing tip error can be corrected for by applying different normalization parameters to the subarrays.

Printing can also cause problems that are not categorized as spatial bias. Sometimes some or all spots are irregular in shape or they might express as rings instead of circles. It is likely that the detection of these spots will be inaccurate. This should be taken into account during image analysis if possible.

8.2.5 Plate and reporter effects

These often ignored but very common effects are not artificial but are caused by biology itself. Sometimes concentration of reporters (probes) on the slide might differ and this can cause patterns on a slide. Usually this is easy to notice if the position of the plates (assuming that concentration is constant on a plate) is known. Other quality control methods should also be used. Plate effects can be corrected for by using methods as in the case of different printing tips.

More difficult to recognize and especially to distinguish from other sources of bias is the bias caused by the biological role of reporters. If reporters are arranged according to their biological function (as is done in many cases) this can also be seen as patterns on a slide. It is very important to note that this effect should *not* be normalized! The reporter effect needs to be taken into account when slides are being designed. Related reporters should never be grouped together.

8.2.6 Batch effect and array design

When large amounts of slides have been studied, it has been noted that slides from the same print run (or batch) often cluster together and also slides from an identical print design but different print run. Some of these effects might be due to mistakes in printing. This kind of bias is very hard to notice and the only way to prevent it is to keep track of printing (LIMS) and use of good pre- and post-printing quality control methods.

8.2.7 Experimenter issues

Equal to batch effect is the systematic bias caused by the experimenter. Experiments done by the same experimenter often cluster together more tightly than warranted by biology. A survey made at Stanford University showed that the experimenter was one of the largest sources of systematic bias. The best but not a very practical solution for the experimenter issue would be to let the same experimenter do everything. Because this naturally isn't possible, consistent hybridization techniques are needed as well as methods to recognize bias caused by the experimenter.

8.2.8 What might help to track the sources of bias?

If you have the possibility to hybridize one or several arrays where the same sample has been used both as target and control *i.e.*, the same sample labeled with both dyes on the same array), or to study two replicate arrays as sample and reference, the results of both channels should be equal, a scatter plot should show a straight line, and optimally, your data should follow natural distribution. Studying self-self hybridized arrays shows you clearly the sources of error, such as uneven incorporation of the dyes, that are not dependent on your sample. It can also help you in deciding which normalization steps would minimize these errors.

Scatter plot (MA plot, see 8.5.5.) may be used to illustrate the different types of effects due to intensity-dependent variation as shown in Figure 8.1.

8.3 Normalization terminology

There are three broad groups of normalization techniques, each of which can contain multiple methods. Classically, normalization means to make the data more normally distributed. In microarray analyses, this is still the goal, but the methodology has concealed the original idea.

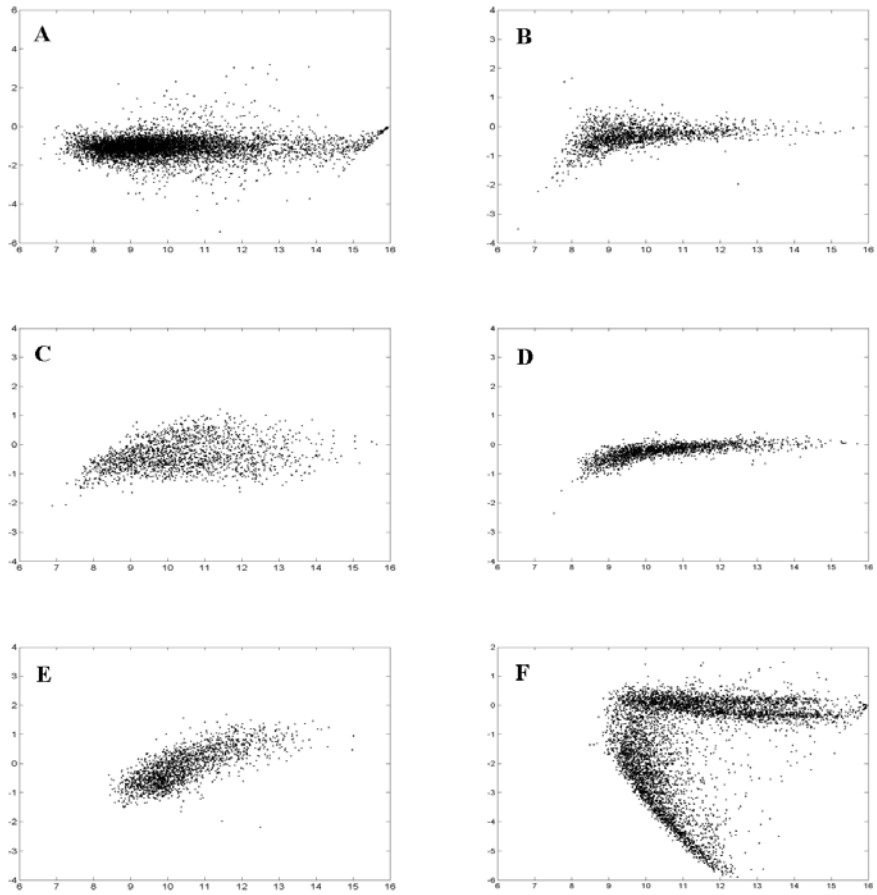

Figure 8.1: *Truncation at the high intensity end (A) is caused by scanner saturation. High variation at the low intensity end (B) is from larger channel- specific additive error. High variation at the high intensity end (C) is from larger channel- specific multiplicative error. The curvature in (D) is from channel mean background difference. The curvature in (E) is from slope difference. The split RI plots in (F) come from heterogeneity. Figure adopted and modified from a paper entitled "Data transformation for cDNA Microarray Data" by Cui, X., Kerr, M. K., and Churchill, G. A. (Submitted to Statistical Applications in Genetics and Molecular Biology. The manuscript web site is at* `http://www.jax.org/staff/churchill/labsite/pubs/index.html`*)*

8.3.1 Normalization, standardization and centralization

In general, *normalization* means the process of transforming a statistic (*e.g.*, intensity ratio) so that it will approximate normal distribution, or be more normal-like (see 7.9). In the microarray field, normalization means also the standardization and centralization methods.

Usually a log-transformation works quite well as a normalization method for microarray data. Log-transformation of intensity-ratios is good for several reasons. The simple ratio flattens all the underexpressed genes between 1 and 0. Log-

transformation removes this bias. In statistical terms, log-transformed data gives a more realistic sense of variation, and makes the variation of intensities and ratios of intensities more independent of absolute magnitudes. Log-transformation also stabilizes the variance of high intensity spots. It evens out highly skewed distributions (*e.g.*, makes more normal-like), and makes normalization (centralization) additive. After log-transformation the expected mean of the dyes is 0 as opposed to 1 in the case of plain intensity ratios.

Standardization is the process of expanding or contracting the distribution of a statistic so that the experimental values can be compared with those from another experiment. In statistical terms standardization means converting the intensity ratios to Z-scores (see 6.5.1.), which are distributed as a standard normal distribution.

Most of the methods meant by normalization in the microarray field are covered by the term *centralization*. It is a process of moving a distribution so that it is centered over the expected mean (balancing the two channels). For the log-transformed intensity ratios, an intensity dependent centralization (*e.g.*, lowess) might help to correct the dye bias.

8.3.2 Per-chip and per-gene normalization

Per-chip normalization compensates for the experimental biases introduced on the individual chips. It assumes that the median intensity stays relatively constant during the experiment. If there are large differences due to the biological phenomenon, you can accidentally remove the relevant variation in your data by normalization.

Per-gene normalization accounts for the difference in the detection efficiency between different spots. It also enables us compare the relative gene expression levels, as already pointed out. For two-color data, the per-gene normalization is usually not used by default, because the calculation of intensity ratios corrects for the same bias.

8.3.3 Global and local normalization

When normalizing the data, we estimate some descriptors of the data (*e.g.*, the mean). If only one descriptor is used for normalizing the whole data for one chip, then a global normalization is performed.

Sometimes we have different estimates of means for different subarrays or for different intensity ranges on the same chip. If these different estimates or a function (nonlinear regression, *i.e.* lowess) is used for the normalization of the concomitant subarrays or intensity ranges, the local normalization method is applied.

8.4 Performing normalization

8.4.1 Choice of the method

There are several normalization methods and little consensus about which one to use, but here we will try to clarify the choice of methods and their applicability with a few simple rules. The choice of the normalization method is coupled to experimental design. Optimally, the method should be chosen before the actual

experiment is conducted. However, the selection of the best normalization method depends also very much on how the results turn out, *e.g.*, the outcome of quality, success of different steps in printing, hybridization, and scanning can be evaluated only after the experiment has actually been done. Therefore, after inspecting the quality of your data in general (see previous Chapter), it is also advisable to try several different methods and see which one gives the most realible results for your experiment. In any case, inside one experiment the same normalization scheme should be applied uniformly to all chips.

8.4.2 Basic idea

The basic idea behind all the normalization methods is that the expected mean intensity ratio between the two channels (two-color data) or two chips (one-color data) is one, because only about 10–20% of all the genes in a cell are expressed at any one time. If the observed mean intensity ratio deviates from one, the data is mathematically processed in such a way that the final observed mean intensity ratio becomes one. When the mean intensity ratio is adjusted to one, the distribution of the gene expression is centered so that different chips (arrays) can be compared.

Often, when a big enough number of genes are plotted on a chip, it can be assumed that most of them actually aren't changing, and the mean intensity ratio can be assumed to be one. If only a couple of dozen genes are plotted on the slide, this assumption does not necessarily hold, and some other means of normalization should be undertaken. This is especially true, if all the genes plotted on the slide are relevant to the studied hypothesis.

8.4.3 Control genes

If the whole genome of the organism is printed on the microarray, we can assume that most of the genes are not expressed, or are expressed in very low quantities. In such cases all the genes on the chip can be used for normalization. If only a few dozen or hundred of genes are present on the chip, some normalization standards need to be added on the chip. Suitable controls are housekeeping genes and spiked controls.

When using spiked controls, a known labeled DNA target is added to the labeled cDNA samples. This spiked DNA control hybridizes to its probe on the DNA microarray chip. The spiked control should optimally be RNA or DNA from another species than the studied one that do not cross-hybridize, but behaves similarly to your sample RNA or DNA. The expected intensity ratio of the spiked controls is one. If the observed ratio deviates from expected, the observed ratio is used for normalization.

Housekeeping genes are now known not to be very reliable controls, because their expression often changes during the experiment. However, it is often possible to find a set of genes that do not change during a certain experiment. Using such a set of non-changing genes may mean that you need to modify your arrays and re-do your hybridizations after you have found the working set of genes you wish to use as controls.

Controls are needed in microarrays as in any quality-checked laboratory exper-

iment. Ideally, you would have several types of controls in different concentrations scattered throughout the printed array. If you use Affymetrix products, chips contain controls for the experiment and for the hybridization of the correct target (perfect match and mismatch, Chapter 3). More about controls can be read in Chapters 2 (Experimental design) and 7 (Preprocessing of data).

8.4.4 Linearity of data matters

There are several mathematical procedures and different algorithms (median centering, standardization, lowess smoothing) for normalization from which to choose from. Before deciding on the method, the linearity of data should be checked (see Chapter 7). If the data is linear, such procedures as median centering can be applied. And the other way around, median centering can only be applied to linear data. If the data is non-linear, lowess smoothing or an other local method should rather be applied. Moreover, if the data is normally distributed, it can be standardized. Also for certain purposes, for example clustering, the spread of the data should be standardized.

8.4.5 Basic normalization schemes for linear data

In order to make comparisons between chips of the same experiment possible, every single chip has to be normalized. In it's simplest form this per-chip normalization means median centering as described in section 8.5.2 and Table 8.1. In other words, the median intensity (or expression) of every chip is brought to the same level. Median centering does not change the spread of the data, which also means that the original information content of the data is not altered. Median centering is often a suitable choice for one-color datasets.

In addition to per-chip normalization, per-gene normalization can be applied. This controls the hybridization efficiency differences between the individual probes. For two-color experiments, the calculation of the intensity ratio corrects for differences between spots, and other normalization may not be needed. However, for one-color experiments, per-gene normalization is usually applied.

One-color experiment can also be normalized against another one-color chip (a reference chip), and handled thereafter as two-color experiment. With regard to Affymetrix data, it is good to remember that the data has already been normalized and scaled to some extent using specific algorithms (to calculate results for perfect match and mismatches) (see Chapter 3).

If per-chip and per-gene normalizations are applied, both individual chips and genes can directly be compared with each other. This is often the desired situation.

8.4.6 Special situations

Sometimes basic normalization schemes don't perform well enough. If most of the genes on a chip are likely to change, or there are spatial biases on the chip, more sophisticated methods should be used. Again, global methods should not be used, if the data is nonlinear, if there are spatial bias on the chip, or if the number of expressed genes varies a lot between individual chips. In such cases, local methods

should be used. As already mentioned, lowess smoothing is a a recipe for the normalization of nonlinear data.

If the expression of most of the genes on the chip are expected to change during the experiment, a median centering using all the genes is not a viable option. In such cases spiked controls or housekeeping genes can be used for the normalization (see 8.5.8). These methods correct effectively for the differences between the channels, but if the amount of mRNA hybridized to individual chips varies a lot, errors are likely to occur.

Spatial biases can be corrected in multiple ways. Probably the easiest method is to use median centering subarraywise. In other words, a new median is calculated for every subarray (or a block of about 500 genes on a chip). This subarray specific median is then used for the normalization of that subarray. The same principle can be applied if the printing tip groups deviate from each other. Sometimes the left side of the chip has lower expression values than the right side of the chip. Such spatial biases can also be corrected using the median centering inside one probe column or row.

8.5 Mathematical calculations

There are several ways to calculate the normalized intensity ratios. Here, we present a few of the most commonly used ones. If you want to play around with these, please check the logarithmic rules of calculation before proceeding. For example, the mean centering (subtract the mean or median to get a mean or median of zero) with log-transformed data is equivalent to mean scaling (divide with a certain number to get a mean or median of one) with untransformed data. The next formulas are presented for the log ratios. Usually the normalization is calculated using only a control channel or a reference chip.

8.5.1 Mean centering

Calculate the mean of the log ratios for one microarray. Produce the centered data by subtracting this mean from the log ratio of every gene.

8.5.2 Median centering

Calculate the median of the log ratios for one microarray. Produce the centered data by subtracting this median from the log ratio of every gene.

8.5.3 Trimmed mean centering

Remove the most deviant observations (5%) from the data for one microarray. Calculate the median of the log ratios of the remaining genes. For centered data, subtract this median from the log ratios of every gene.

8.5.4 Standardization

Equivalent to the Z-score calculation. See section 6.5.1.

8.5.5 Lowess smoothing

Lowess-regression is often performed in a space of A versus M. This MA plot, also called RI plot, is equivalent to the usual scatter plot, but the data cloud has been tilted 45 degrees to the right. In other words, M is the log-transformed intensity ratio, and A is the average intensity of the two channels or chips. Under and over-expressed genes are much easier to find from this scatter plot, and the linearity of the data is also easier to check. Thus, MA plots can reveal the intensity dependency of log ratios, which appears as curvature, as well as extra variation at low intensity spots (see Fig. 8.1.).

$$M = log_2(R) - log_2(G)$$

$$A = \frac{log_2(G) + log_2(R)}{2}$$

After normalization the M and A can be transformed back to the dye intensities as

$$R = \left(2^{2A+M}\right)^{\frac{1}{2}}$$

$$G = \left(2^{2A-M}\right)^{\frac{1}{2}}$$

Lowess-normalization works as follows. A lowess-function smoothing is applied to the data and a curve is estimated by the sliding window method. The normalized log-transformed intensity ratio is calculated by subtracting the estimated curve from the original values. (Note: If a linear function is used for the local regression, it is called lowess with a "w". If a quadratic function is used for the local regression, it is called loess without the "w".)

Figure 8.2 gives an example of how the lowess-regression treats real life data.

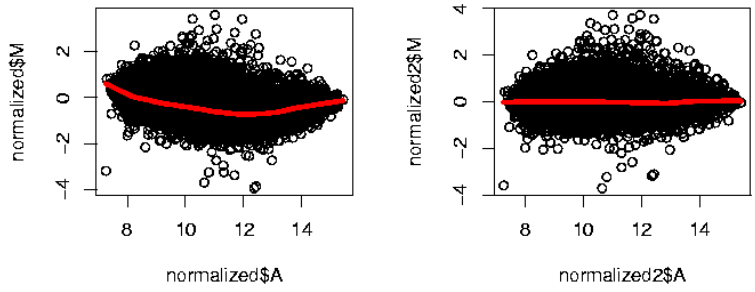

Figure 8.2: *Centralization with a lowess-regression. On the left, the log_2-normalized data (centered around zero) in an A versus M plot. On the right, lowess-normalized data.*

Linlog transformation can also be used to remove intensity-dependent variation. The linlog transformation combines linear and logarithmic transformations through a smooth transition to take advantage of both methods, because the linear transformation of raw data is more appropriate for low intensity spots and the

logarithmic transformation is more appropriate for high intensity spots (Cui *et al.*, submitted). The linlog transformation can precede lowess smoothing and in that way stabilize the variation due to the additive error dominant at low intensity, and the multiplicative error dominant at high intensity in microarray data.

8.5.6 Ratio statistics

Data is assumed to follow a certain distribution. This distribution can then be used to normalize the current data. The ratio statistics method is closely related to estimating an error model from the data.

8.5.7 Analysis of variance

If there are data from several replications of the same gene, the ANOVA method can be applied to centralize the data. With the ANOVA, many sources of error can be taken into account at the same time.

8.5.8 Spiked controls

We assume that the intensity ratio from the spiked controls is one, if there is no dye-incorporation bias. If the ratio deviates from one, this information can be used for the calculation of the centered intensity values for the genes:

$$\text{green intensity} = \frac{\text{red intensity}}{\text{spiked control intensity ratio}}$$

The housekeeping genes are used for normalization in a similar manner.

8.5.9 Dye-swap experiments

Dye-swap experiments can be normalized with the lowess-regression, when the average of M and the average of A are plotted against each other as in the usual lowess procedure. New labeling methods should remove the dye-incorporation bias, but meanwhile this normalization method can be handy, especially in the cases that indicate that the mRNA sequence may influence the labeling efficiency.

If we assume that the expression of any one gene in the original and the dye-swap experiment is of equal magnitude but opposite signs, the normalization of dye-swap experiments can be performed in a similar way as the normalization of non-dye-swap experiments.

The normalized values for the dye-swap experiment can be then calculated as:

$$\frac{1}{2}\left(log_2\frac{RG'}{GR'}\right) - c,$$

where R and G are the original and R' and G' are the dye-swap experiment intensities of the red and green channels, respectively. The normalization constant can be estimated as

$$c = \frac{1}{2}\left(log_2\frac{R}{G} + log_2\frac{R'}{G'}\right)$$

In other words, the average of the original and dye-swap chip is calculated for every gene. Normalized values are calculated by subtracting the average of the chips from the averaged genewise intensities.

8.6 Some caution is needed

Most of the aforementioned methods are used for per-chip normalization. Per-gene normalization is not worthwhile if you only have a few chips, because then you can potentially introduce errors to your data. This is because the mean, standard deviation, and regression curves can not be effectively estimated from a very small number of observations. As a guideline, the experiment should consist of at least five chips, if you want to perform the per-gene normalization using mean or median centering. Even more observations (at least 10-25) are needed in order to use standardization, regression or other advanced normalization method.

Furthermore, using very sophisticated normalization methods can lead to a phenomenon called overfitting, in which case the model (*e.g.*, a linear regression) describes the variability of the data too well. This effectively removes biologically relevant variation from the data, but it also introduces biases to the analysis. It is common that the correlation of two chips (especially replicates) slightly decreases after normalization. Whether this means that the biologically relevant information has been removed, and some noise added, is currently unknown.

The mean and median centering do not usually influence the standard deviation very much, but the regression and other more sophisticated methods do. Usually they move towards a higher standard deviation, which might mean more noisy data. Therefore, the normalization procedure should be as simple as possible, yet taking the systematic errors into account.

8.7 Graphical example

A couple of examples how the normalization procedure affects the graphical representation of the data are presented in Figure 8.3. These examples showes, how the number of genes affects the results. In the image a diminishing series of housekeeping genes is used for normalization.

8.8 Example of calculations

A short example of mean centering procedures is presented in Table 8.1. Mean centering can be applied either as per-chip normalization or per-gene normalization. With Affymetrix data, both procedures are often applied. If both procedures are applied, the per-chip centering should be performed before the per gene centering. Note that the standard deviation is not affected by mean centering.

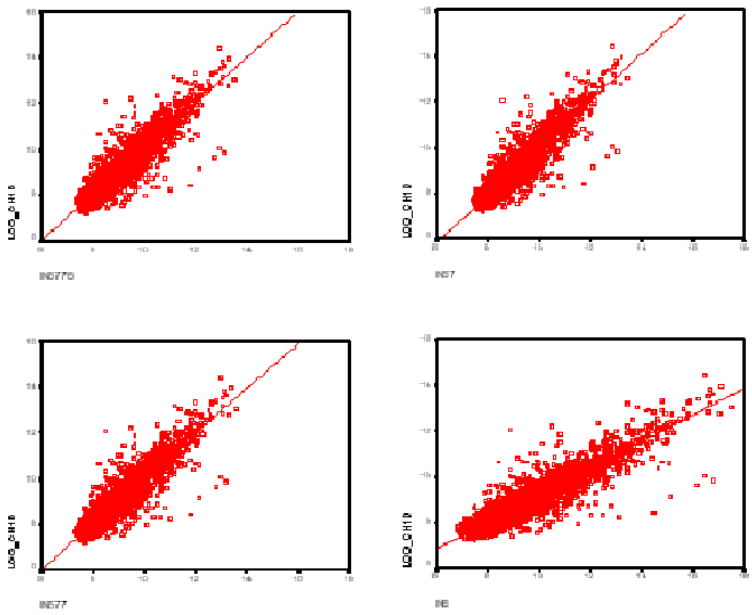

Figure 8.3: *The four images represent the same data processed with per-chip mean centering. The number of genes used for the mean calculation has been varied. From top left to bottom right: 5776 genes, 57 genes, 577 genes, and 6 genes. Note the varying slope of the linear regression line fitted to the normalized data.*

8.9 Using GeneSpring for normalization

GeneSpring offers the basic repertoire of normalization methods, including both per-chip and per-gene methods. GeneSpring includes calculation of intensity ratios and dye-swap calculations as normalization methods. Dye-swap normalization in GeneSpring simply calculates the intensity ratio as (Cy5 / Cy3) instead of the usual (Cy3 / Cy5) for the specified samples.

Normalization methods in GeneSpring 5.x are

- lowess-smoothing (per-chip and per-gene)
- median polishing (per-chip and per-gene)
- median or percentile scaling (per-chip or per-gene)

Also housekeeping genes or spiked controls can be used for normalization (Per-chip: normalize to positive control genes) . If chips have been scaled in the Affymetrix MAS program to a certain mean intensity value, the transformational effect can be removed in GeneSpring using Per-chip: Normalize to a constant value option. If one-color data is analyzed, and the experiment includes one control chip that is used for the calculation of intensity ratio, this can be done in GeneSpring

Table 8.1: *An example of per-chip and per-gene mean centering. Because of rounding errors, the results presented in the table are not necessarily products of individual observations and their means.*

Uncentered data

	chip1	chip2	chip3	chip4	mean	standard dev.
gene1	2.12	2.01	4.37	2.01	2.63	1.16
gene2	2.20	2.06	4.32	2.03	2.65	1.11
gene3	2.18	1.90	4.37	1.90	2.59	1.20
gene4	2.15	1.92	4.38	1.89	2.59	1.20
gene5	2.14	2.00	4.52	1.99	2.66	1.24
gene6	1.93	2.02	4.18	2.01	2.54	1.09
gene7	2.26	1.96	4.19	1.98	2.60	1.07
gene8	2.07	2.00	4.39	2.01	2.62	1.18
gene9	2.25	2.06	4.34	2.04	2.67	1.12
gene10	1.95	1.76	3.97	1.82	2.37	1.07
mean	2.13	1.97	4.30	1.97		
standard dev.	0.11	0.09	0.15	0.07		

Per-chip mean centering

	chip1	chip2	chip3	chip4	mean	standard dev.
gene1	0.00	0.05	0.07	0.04	0.04	0.03
gene2	0.08	0.09	0.02	0.07	0.06	0.03
gene3	0.06	-0.07	0.07	-0.07	-0.01	0.08
gene4	0.03	-0.05	0.08	-0.08	0.00	0.07
gene5	0.01	0.03	0.21	0.02	0.07	0.10
gene6	-0.19	0.05	-0.13	0.04	-0.06	0.12
gene7	0.13	-0.01	-0.11	0.01	0.00	0.10
gene8	-0.05	0.03	0.09	0.05	0.03	0.06
gene9	0.13	0.09	0.04	0.07	0.08	0.04
gene10	-0.18	-0.21	-0.33	-0.15	-0.22	0.08
mean	0.00	0.00	0.00	0.00		
standard dev.	0.11	0.09	0.15	0.07		

Per-gene mean centering

	chip1	chip2	chip3	chip4	mean	standard dev.
gene1	-0.51	-0.61	1.74	-0.62	0.00	1.16
gene2	-0.45	-0.59	1.66	-0.62	0.00	1.11
gene3	-0.40	-0.69	1.78	-0.69	0.00	1.20
gene4	-0.43	-0.67	1.80	-0.70	0.00	1.20
gene5	-0.52	-0.66	1.86	-0.67	0.00	1.24
gene6	-0.60	-0.52	1.64	-0.52	0.00	1.09
gene7	-0.34	-0.64	1.60	-0.62	0.00	1.07
gene8	-0.54	-0.62	1.77	-0.61	0.00	1.18
gene9	-0.42	-0.61	1.67	-0.63	0.00	1.12
gene10	-0.43	-0.61	1.60	-0.56	0.00	1.07
mean	-0.47	-0.62	1.71	-0.62		
standard dev.	0.08	0.05	0.09	0.06		

using Per-gene: Normalize to specific samples option. Furthermore, normalization can be applied only to certain samples, if needed.

GeneSpring automatically detects the imported data type (one- or two-color), and suggests a normalization scheme for the experiment. These schemes are following

For one-color data (Affymetrix or nylon filter):

- Data transformation: Set measurements less than 0.0 to 0.0
- Per-chip: Normalize to 50th percentile
- Per-gene: Normalize to median

For two-color data (cDNA microarrays) either:

- Per-chip and per-gene: Intensity dependent (lowess) normalization

or

- Per spot: divide by control channel
- Per-chip: Normalize to 50th percentile

Normalization methods are accessed through *Experiments -> Experiment normalizations* in GeneSpring. GeneSpring gives a warning if the normalization you are trying to perform does not make any sense (red text), or that the calculation can be performed, but the results might be unreliable (orange text). Note that the normalization methods are applied in the same order as they appear in the list.

8.10 Suggested reading

1. Anonymous. Presentations from Normalization working group, MGED4 meeting, Boston, February 2002. See latest information from the MGED Data Transformation and Normalization Working Group's home page at `http://www.dnachip.org/mged/normalization.html`.

2. Beißbarth, T., Fellenberg, K., Brors, B., Arribas-Prat, R., Boer, J., Hauser, N. C., Scheideler, M., Hoheisel, J. D., Schutz, G., Poustka, A., and Vingron, M. (2000) Processing and quality control of DNA array hybridization data. Bioinformatics 16, 1014-22.

3. Bolstad, B. M., Irizarry R. A., Astrand, M., and Speed, T. P. (2003), A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance. Bioinformatics 19, 185-193.

4. Brazma, A., and Vilo, J. (2000) Gene expression data analysis. FEBS Lett. 480, 17-24.

5. Cui, X, Kerr, M. K., and Churchill, G. A.. Data Transformations for cDNA Microarray Data. Statistical Applications in Genetics and Molecular Biology, submitted.

6. Dudoit, S., Y. H. Yang, M. J. Callow, and Speed, T. P.. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments (Statistics, UC Berkeley, Tech Report # 578).

7. Eickhoff, B., Korn, B., Schick, M., Poustka, A., and van der Bosch, J. (1999) Normalization of array hybridization experiments in differential gene expression analysis. Nucleic Acids Res. 27, e33.

8. Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., Speed, T.P. (2002) Exploration, Normalization, and Summaries of High Density Oligonucleotide Array Probe Level Data, submitted. http://www.stat.berkeley.edu/users/terry/zarray/Affy/affy_index.html

9. Quackenbush J. (2001) Computational analysis of microarray data. Nat Rev Genet. 2, 418-427.

10. Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H., and Hanspeter, H. (2000) Normalization strategies for cDNA microarrays. Nucleic Acids Research 28, e47.

11. Sherlock, G. (2001) Analysis of large-scale gene expression data. Briefings in Bioinformatics 2, 350-362.

12. Yang, Y. H., Dudoit, S., Luu, P. and Speed, T. P. (2001) Normalization for cDNA Microarray Data. Proceedings of Photonics West 2001: Biomedical Optics Symposium (BiOS),The International Society of Optical Engineering (SPIE), San Jose, California. http://www.stat.berkeley.edu/users/terry/zarray/Html/normspie.html

13. Yang, Y. H., Dudoit, P., Luu, P., and Speed, T.P. (2001) Normalization for cDNA microarray data, submitted.

This chapter was written by Jarno Tuimala, Ilana Saarikko, and M. Minna Laine.

9 Finding differentially expressed genes

After removing the bad quality data we are left with reliable data. As we have already seen, good quality data can be further filtered so that only the genes that show some changes in the expression during the experiment are preserved in the dataset. Often the differentially expressed or otherwise interesting genes are stored as simple lists of gene names, *i.e.*, genelists. Most typically genelists are created by driving the data through a certain filter, but they can also include lists of genes, which have been produced by clustering analyses or other more complex methods.

9.1 Identifying over- and underexpressed genes

Sometimes it is sufficient to get information about genes that are either under- or overexpressed during the experiment. For example, we might be interested in genes that have an elevated expression because of a drug treatment. Such genes are most easily found by simple filtering. Simple filtering (by absolute expression change) can even be used for experiments, where there are no replicates.

9.1.1 Filtering by absolute expression change

If the log-transformed data is used for analysis, the overexpressed genes have an expression above zero and underexpressed genes have an expression below zero. However, often the experimental errors are of the order of two, and for finding the over- and underexpressed genes the cutoffs of -1 and 1 are used. All the genes which fulfill the filter criteria are saved in a new genelist.

The method can also be used for filtering by expression duration. If the expression change has been at least x for a duration of y, the gene names are saved into a new genelist.

This kind of genelist can be created in any spreadsheet program, and the actual data-analysis can be very simple to perform.

9.1.2 Statistical single chip methods

The crude filtering by absolute expression change is not always the optimal method for finding the differentially expressed genes, because the information about the reliablity of the expression change is lacking. In the absence of replicates, there are still a few statistical methods that can be used, if we want to have statistical

significance values for expression changes. However, using such methods carries the risk that the most unstable probes or mRNAs are identified as differentially expressed.

In the next four sections we will go through some of these methods.

9.1.3 Noise envelope

Noise envelope means simply a method where the standard deviation (calculated with sliding window) is used as a cut-off for under and over-expressed genes.

The variance in the reported gene expression level is a function of the expression level so that the variance is higher in the lower expression values and lower when the expression is high. Thus, the identification of the differentially expressed genes calls for a statistical model of the variance or noise. The simplest model could be constructed on the basis of the standard deviation of expression values.

Such a model can be constructed relatively easily using statistical programs. The method relies on a segmental (sliding window) calculation of standard deviation. A datapoint refers to an (x, y) paring, where x is the absolute intensity value of a gene from the hybridization, and y is the corresponding intensity ratio value. Using all data points in a given sliding window of expression values, the standard deviation of the intensity ratios is calculated. The average of expression values within the window are then paired with the average intensity ratio value within the same window plus the number of standard deviations (usually 2 or 3) specified by the experimenter. This new pair becomes the candidate upper noise envelope point. Similarly a candidate lower noise envelope points are determined.

Using these noise envelopes, the insignificant expression changes are removed: The region between the upper and lower noise envelopes contains genes with insignificant changes.

9.1.4 Sapir and Churchill's single slide method

After logarithmic transformation and normalization, Sapir and Churchill's method fits a linear regression function to a scatter plot of sample intensity versus control intensity to yield orthogonal residuals. These residues represent the gene expression changes in the experiment. The expectation-maximization (EM) algorithm applying the Bayes' rule is used to obtain estimates of the proportion of differentially expressed genes. Using the Bayes' rule and information on the residuals, the (posterior) probabilities of differential expression are calculated.

The method is implemeted in the R language package sma. The results can be obtained either as a genelist or in a form of a scatter plot. The scatter plot example is given here (Figure 9.1). The residuals are modeled as a uniform distribution, so the cut offs are constant for all intensity values.

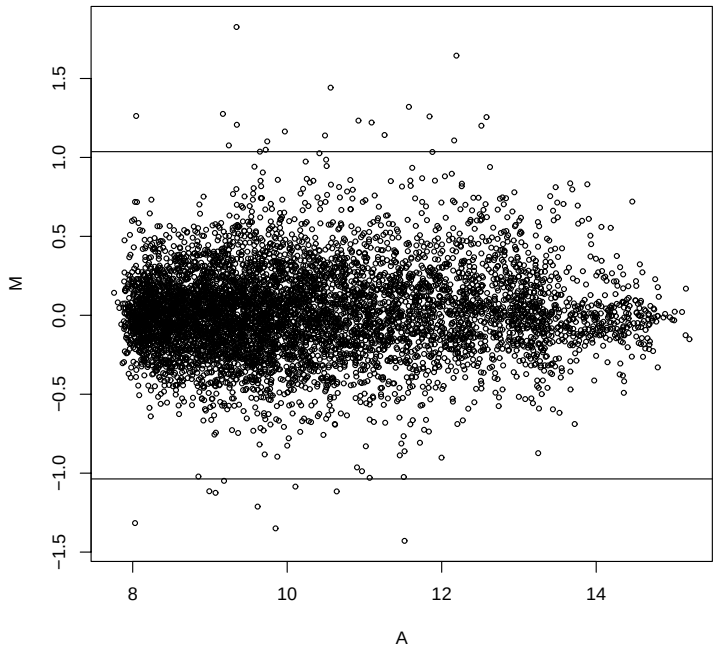

Figure 9.1: *A scatter plot with Sapir and Churchill's method with the cut off lines marked. The vertical lines represent the cut offs for up- and downregulated genes. Note that the axes of the plot are normalized log ratio (M) and average intensity (A).*

9.1.5 Chen's single slide method

The idea central to Chen's method is that the gene expression levels are determined by the intrinsic properties of each gene, which means that the expression levels vary widely among genes. Therefore it is inappropriate to pool statistics on gene expression differences across the microarray. Assuming that the green and red channel intensities are normally distributed and have similar coefficients of variation and a similar mean, the confidence intervals for actual differential expression are calculated using the data derived from the maximum likelihood estimate of the coefficient of variation and a set of genes, which do not show any expression change on the chip.

The method is implemeted in the R language package sma. The results can be obtained either as a genelist or in a form of a scatter plot. The scatter plot example is given here (Figure 9.2).

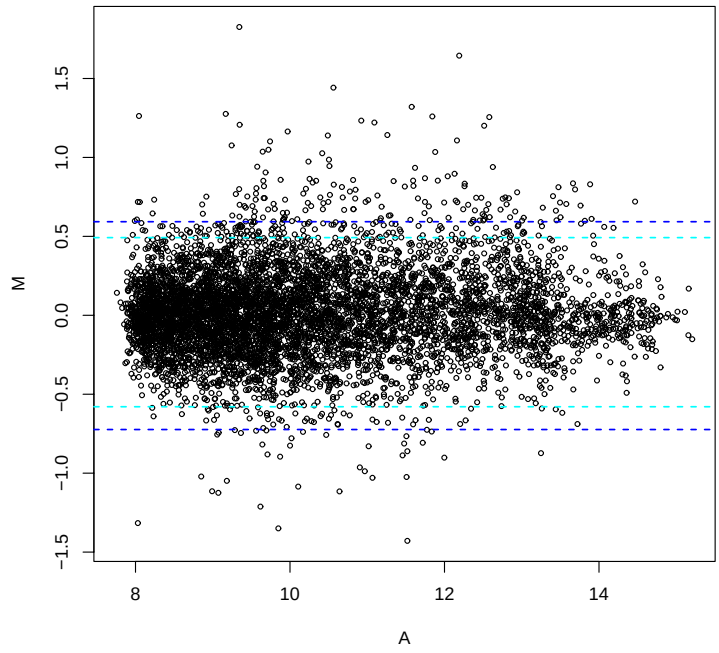

Figure 9.2: *A scatter plot with Chen's method with the 95% (inner, light blue) and 99% (outer, dark blue) cut off lines marked. The vertical lines represent the cut offs for up- and downregulated genes. Note that the axes of the plot are normalized log ratio (M) and average intensity (A).*

9.1.6 Newton's single slide method

Newton's method models the measured expression levels using terms that account for the sources of variation. The first obvious source is measurement error. The second source of variation is due to the different genes spotted onto the microarray. Newton's method can be viewed as a hybrid of Chen's and Sapir and Churchill's methods, because it models the variability on the slide very similarly to Chen's method, but the mathematical calculations are done with the EM-algorithm similar to the one Sapir and Churchill used. Results are presented as log-odds ratios that the gene is differentially expressed.

The method is implemeted in the R language package sma. The results can be obtained either as a genelist or in the form of a scatter plot. The scatter plot example is given here (Figure 9.3).

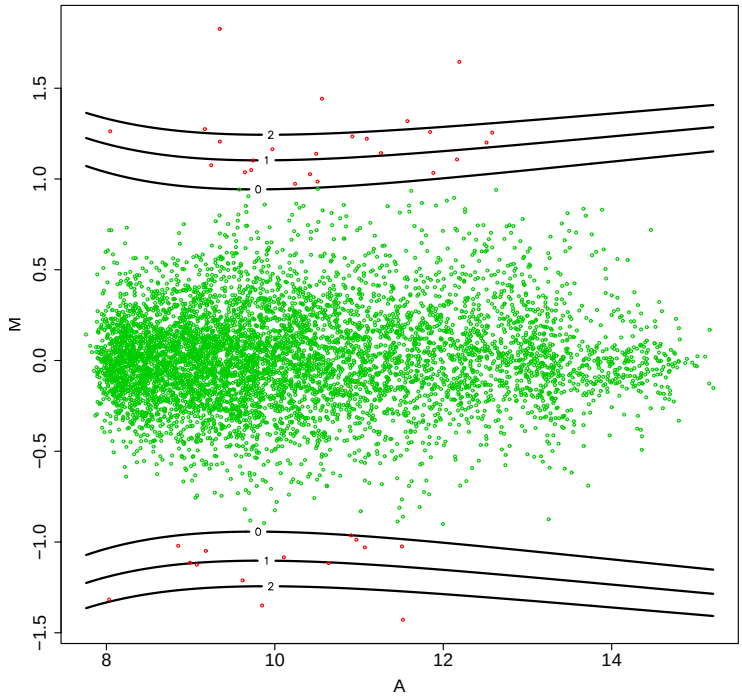

Figure 9.3: *A scatter plot with Newton's method with the log-odds ratios 0, 1 and 2 cut off lines marked. The vertical lines represent the cut offs for up- and downregulated genes. Note that the axes of the plot are normalized log ratio (M) and average intensity (A).*

9.2 What about the confidence?

Say, we have found a gene that is threefold upregulated after the drug treatment. How do we know that the result is not just an experimental error? We need to determine the statistical significance of the upregulation of the gene. Significance can only be assessed, if replicate measures of the same gene were performed during the experiment. The chips (or expression of invidual genes on them) can be compared by the t-test (two conditions) or ANOVA (more than two conditions).

9.2.1 Only some treatments have replicates

If the gene expression has been measured for control and treatment, but only treatment has been replicated, the experimental error can be crudely estimated from the variation between the replicates. The standard deviation is determined for every gene, and the expression change is compared with the standard deviation. The more the change exceeds the standard deviation between replicates, the more significant the gene is. Note that this method does not allow for the calculation of p-values.

9.2.2 All the treatments have replicates: two-sample t-test

The statistical basis for the t-test has been covered in detail elsewhere (see 6.9), but briefly, the two-sample t-test looks at the mean and variance of the two distributions (say, control and treatment chip log ratios), and calculates the probability that they were sampled from the same distribution. The t-test can be applied successfully in situations, where both the control and treatment have been repeated.

Here (Figure 9.4), we present an example of a graphical t-test result, which has been produced using R language supplemented with package sma.

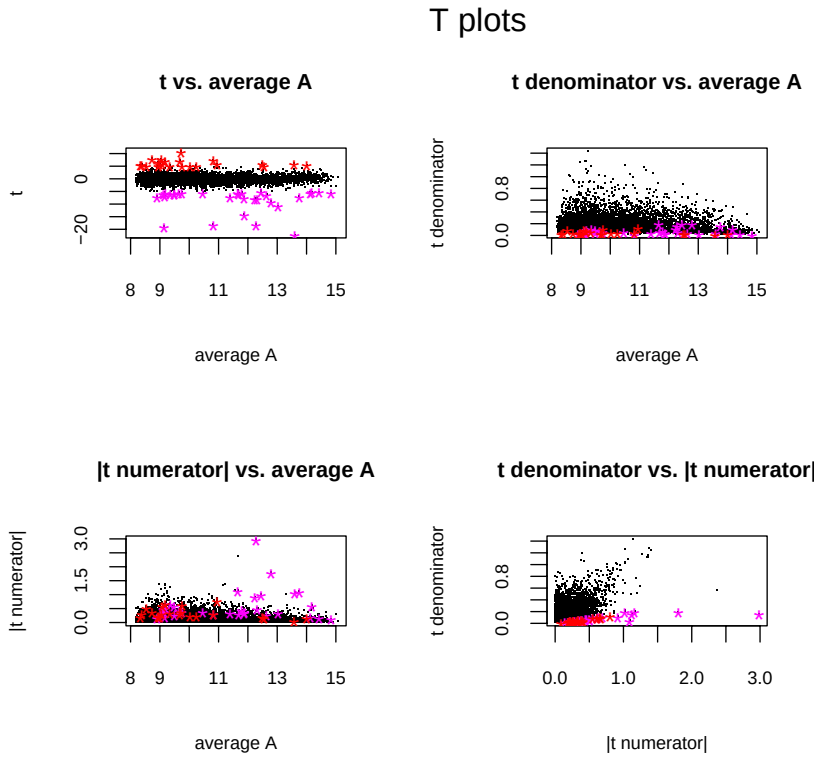

Figure 9.4: *Package sma automatically produces these four plots for t-test results. The genes that have an expression pattern most significantly deviating from zero are highlighted with a colored star.*

After calculating the t-test p-values for the replicated genes, the ones with the lowest p-value (marked with a star in the images in Figure 9.4) can be saved into a new genelist and used in further analyses, for example cluster analysis. These are the genes that most significantly differ between two conditions, say control and

treatment mice.

9.2.3 All the treatments have replicates: one-sample t-test

Up- and downregulated genes can be more effectively found, if the chips are replicated. In such cases, the one-sample t-test for the deviation from zero (expected mean) can be performed. Finding the differentially expressed genes is even more easier, if the so called Volcano plot is produced (Figure 9.5). This plot of log ratio versus statistical significance shows also which is the lowest expression change we can be statistically confident in. In our example, 2-fold down-regulated genes have a p-value of 0.01. In contrast, 1.4-fold upregulated genes have a similar p-value. This indicates that new chip technology coupled with an appropriate number of replicates can produce reliable data even for very small expression changes.

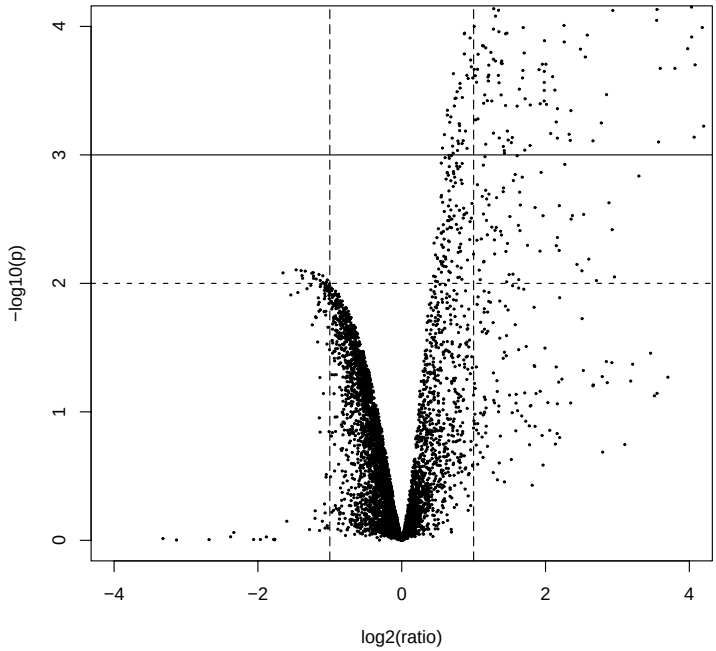

Figure 9.5: *Volcano plot, where the statistical significance by one-sample t-tests [-log10(p)] is plotted against normalized log ratio [log2(ratio)]. Vertical dashed lines represent a 2-fold difference between the control and the sample. Horizontal dashed line represents the t-test p-value of 0.01, and the solid horizontal line represent p=0.001.*

9.3 GeneSpring examples

An example on how to generate genelists with GeneSpring filtering tools is presented in section 7.15.6.

Examples of statistical testing are presented in sections 6.11.6 and 6.11.7.

9.4 Suggested reading

1. Chen, Y., Dougherty, E. R., and Bittner, M. L. (1997) Ratio-based decisions and the quantitative analysis of cDNA microarray images. Journal of Biomedical Optics 2, 364-374.

2. Sapir and Churchill (2000), Estimating the posterior probability of differential gene expression from microarray data, http://www.jax.org/research/churchill/.

3. Newton, M. N., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (1999) On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. Technical Report 139, Department of Biostatistics and Medical Informatics, UW Madison, http://www.stat.wisc.edu/~newton/papers/abstracts/btr139a.html.

4. SMA-package for R, http://www.stat.berkeley.edu/users/terry/zarray/Software/smacode.html.

This chapter was written by Jarno Tuimala.

10 Cluster analysis of microarray information

10.1 Basic concept of clustering

Microarray experiments generate mountains of data, which has to be stored and analysed. For humans it is difficult to handle very large numeric data sets. Therefore a general concept is to try to reduce the dimensionality of the data. A number of clustering techniques have been used to group genes based on their expression patterns. The user has to choose the appropriate method for each task. The basic concept in clustering is to try to identify and group together similarly expressed genes and then try to correlate the observations to biology. The idea is that coregulated and functionally related genes are grouped into clusters. Clustering provides the framework for this analysis. The hard part is to analyze the biological processes and consequences. However, clustering can be a very useful tool. The other side of the coin is the visualization of the information.

Before clustering a large number of requirements have to be met. The data has to be of good quality, the chip design should be correct, one should have interesting genes contained in the chip, sample preparation has to be flawless, data has to be properly treated (outliers, normalization etc). Clustering will not improve the quality of the data. If poor-quality data is used then also the outcome is useless: garbage in, garbage out.

10.2 Principles of clustering

Clustering organizes the data into a small number of (relatively) homogeneous groups. Usually normalization of the expression values is used. At this stage, it is of interest to look at the changes in expression patterns, not to follow the actual numeric changes. Thus, the methods are used to find similar expression motifs irrespective of the expression level. Therefore, both low and high expression level genes can end up in the same cluster if the expression profiles are correlated by shape.

The majority of clustering methods has been available already for long. During the last few years, some new microarray analysis dedicated methods have been developed, too. The methods can be described and classified in different ways. It is not possible to go into the details of all the methods here, so only the most commonly used methods are described.

Clustering methods can be grouped as supervised and unsupervised. Supervised methods assign some predefined classes to a data set, whereas in unsupervised methods no prior assumptions are applied.

Hierarchical clustering, K-means, self-organizing maps (SOMs), and principal component analysis (PCA) have been commonly used. There are also other methods, such as multidimensional scaling (MDS), minimum description length (MDS), gene shaving (GS), decision trees, and support vector machines (SVMs).

10.3 Hierarchical clustering

Hierarchical clustering is a statistical method for finding relatively homogeneous clusters. The hierarchical clustering algorithm either iteratively joins the two closest clusters starting from single clusters (agglomerative, bottom-up approach) or iteratively partitions clusters starting from the complete set (divisive, top-down approach). After each step, a new distance matrix between the newly formed clusters and the other clusters is recalculated. If there are N cases, N-1 clustering steps are needed. There are several methods of hierarchical cluster analysis including

- single linkage (minimum method, nearest neighbor)
- complete linkage (maximum method, furthest neighbor)
- average linkage (UPGMA).

For a set of N genes to be clustered, and an NxN distance (or similarity) matrix, the hierarchical clustering is performed as follows:

1. Assign each gene to a cluster of its own.
2. Find the closest pair of clusters and merge them into a single cluster.
3. Compute the distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all genes are clustered.

Step 3 can be performed in different ways depending on the chosen approach. In single linkage clustering , the distance between one cluster and another is considered to be equal to the shortest distance from any member of one cluster to any member of the other cluster. In complete linkage clustering, the distance between one cluster and another cluster is considered to be equal to the longest distance from any member of one cluster to any member of the other cluster. In average linkage, the distance between one cluster and another cluster is considered to be equal to the average distance from any member of one cluster to any member of the other cluster. The hierarchical clustering can be represented as a tree, or a dendrogram. Branch lengths represent the degree of similarity between the genes. The method does not provide clusters as such. Conceptually, different clusters and sizes of clusters can be obtained by moving along the trunk or branches of the tree and deciding on which level to put forth branches (cut the tree).

Hierarchical clustering is often applied in the analysis of patient samples to organize the data based on the cases. Indeed, for patient samples the hierarchical clustering method is most often the best option. Usually, in addition to patient-based organization, genes are also clustered by applying two-way clustering. On one axis are the samples (patients) and on the other axis the genes (Figure 10.1).

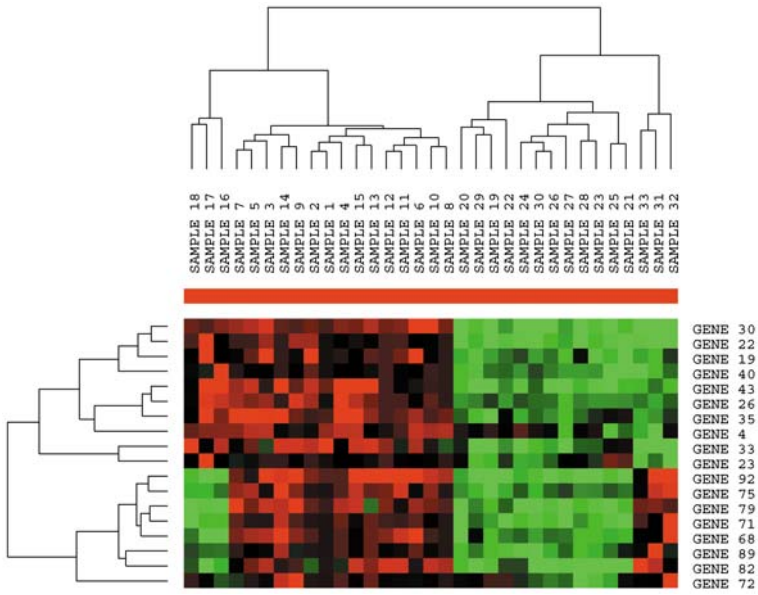

Figure 10.1: *An example of hierarchical clustering.*

10.4 Self-organizing map

Kohonen's self-organizing map (SOM) is a neural net that uses unsupervised learning for which no prior knowledge of classes is required. SOMs are usually used to visualize and interpret large high-dimensional data sets. In SOM, every input is connected to every output via connections with variable weights. Also, the output nodes are highly interconnected. SOM tries to learn to map similar input vectors (gene expression profiles) to similar regions of the output array of nodes (Figure 10.2). The method maps the multidimensional distances of the feature space to two-dimensional distances in the output map. The SOM algorithm is iterative.

The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At the same time, the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other. So the order and organization of the nodes (tentative clusters) contain more information than just the actual partition of genes to clusters.

In SOMs, the number of clusters has to be predetermined. The value is given by the dimensions of the two-dimensional grid or array. One has to experiment with the actual number of clusters. Most programs facilitate the analysis of all the genes within a cluster or clusters. One should find such array dimensions that there

is a minimum number of poorly fitting genes in the clusters. The actual number of clusters is also difficult to assign, but the square root of the number of genes is a good initial estimate. However, it depends on the data set.

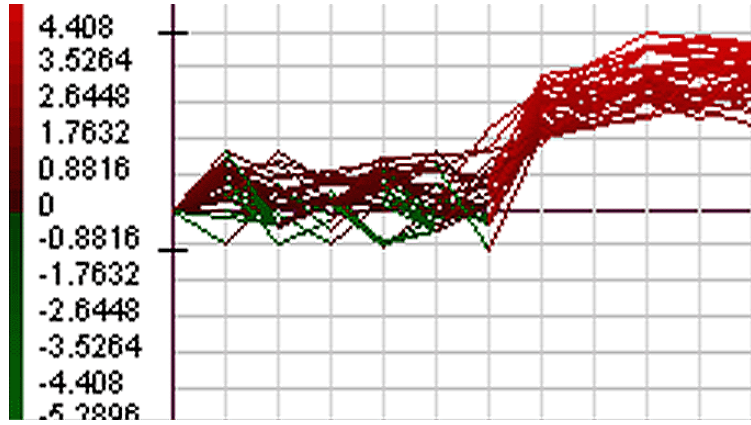

Figure 10.2: *An example of self-organizing map.*

10.5 K-means clustering

K-means is a least-squares partitioning method for which the number of groups, K, has to be provided. The algorithm computes cluster centroids and uses them as new cluster seeds, and assigns each object to the nearest seed (Figure 10.3). However, it is also possible to estimate K from the data, taking the approach of a mixture density estimation problem.

1. The genes are arbitrarily divided into K centroids. The reference vector *i.e.*, location of each centroid, is calculated.
2. Each gene is examined and assigned to one of the clusters depending on the minimum distance.
3. The centroid's position is recalculated.
4. Steps 2 and 3 are repeated until all the genes are grouped into the final required number of clusters.

During the course of iterations, the program tries to minimize the sum, over all groups, of the squared within-group residuals, which are the distances of the objects to the respective group centroids. Convergence is reached when the objective function (*i.e.* the residual sum-of-squares) cannot be lowered any more. The obtained groups are geometrically as compact as possible around their respective centroids (Figure 10.4).

K-means partitioning is a so-called NP-hard problem (there is no known algorithm that would be able to solve the problem in polynomial time), thus there is no guarantee that the absolute minimum of the objective function has been reached.

Therefore, it is good practise to repeat the analysis several times using randomly selected initial group centroids, and check whether these analyses produce comparable results.

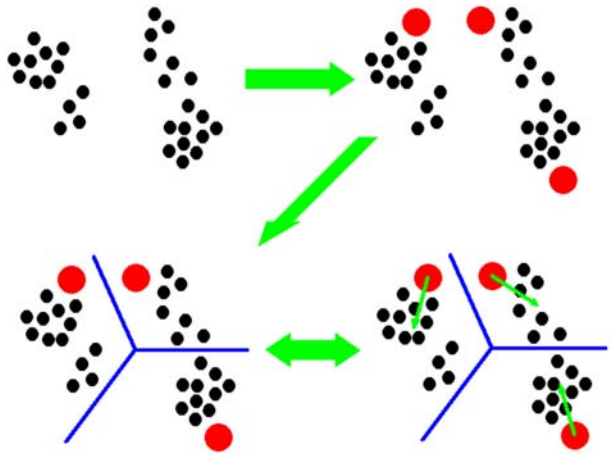

Figure 10.3: *K-means algorithm. Genes are initially divided into K clusters, and the final clusters are iterated from these.*

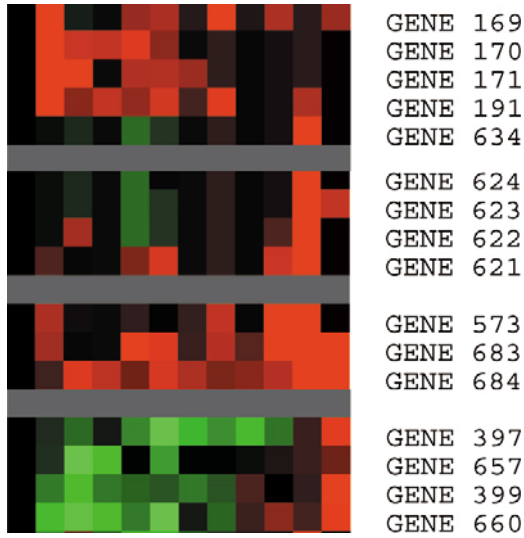

Figure 10.4: *An example of k-means clustering with four clusters.*

10.6 Principal component analysis

Objectives of principal component analysis (PCA) are to discover or to reduce the dimensionality of the data set and to identify new meaningful underlying variables. PCA transforms a number of (possibly) correlated variables into a (smaller) num-

ber of uncorrelated variables called principal components (Figure 10.5). The basic idea in PCA is to find the components that explain the maximum amount of variance possible by n linearly transformed components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

PCA can be also applied when other information in addition to the actual expression levels is available (this applies to SOM and K-means methods as well). The method provides insight into relationships, but no matter how interesting the results may be, they can be very difficult if not impossible to interpret in biological terms.

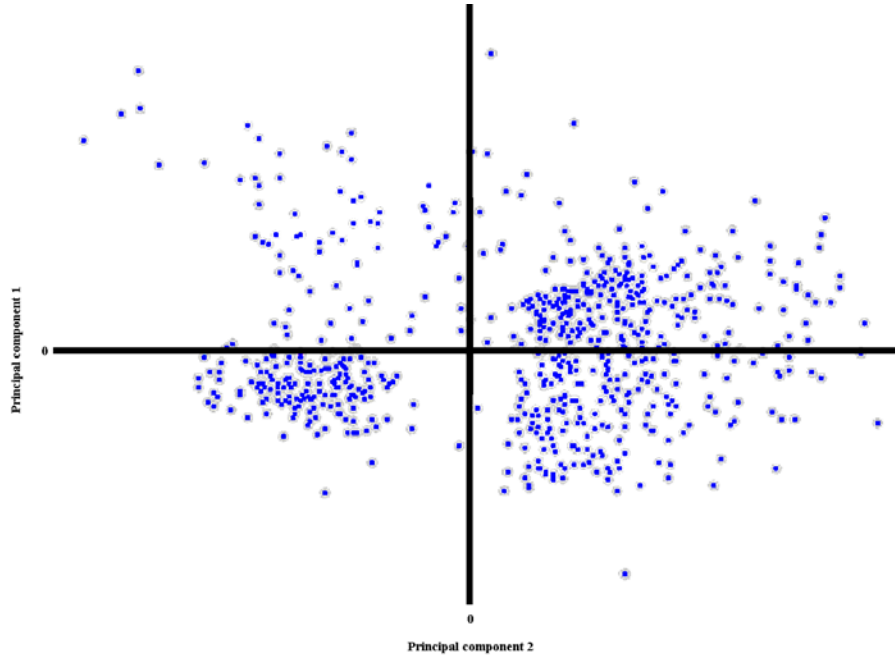

Figure 10.5: *An example of principal component analysis. The two most significant principal components have been selected as the axes of the plot.*

10.7 Pros and cons of clustering

There are no objective rules for determining the correct number of gene expression clusters. Therefore, an extensive manual analysis is required to find out the optimal number of clusters. The expression patterns can be visually inspected, and the distribution of every gene from the centroid analyzed along with the variation from the other genes within clusters. In suboptimal clusters, some genes do not follow the general expression pattern within a cluster due to clearly different expression profiles. It is important to find out a working solution for the number of clusters, because the subsequent analyses and data mining rely on the partition obtained by clustering. The only exceptions are methods using supervised learning, *i.e.*, in

addition to clustering information other data is used to organize the genes. The problem with supervised methods is that they may force the data to behave in a certain way. After all, we cannot know what is the correct partitioning for a given data set. The usefulness of the data for interpretation of biological processes is in fact the true test for the correctness of clustering.

Clustering methods provide a relatively easy way to organize the cluster information. Together with visualization methods they allow for the user an intuitive way of looking at, understanding and analyzing the data. Different clustering methods and the same method with different premises produce different end results, so the user has to try to find out a useful result. In many articles the analysis is finished at clustering. This is a mistake, since clustering is actually only a starting point for more detailed and interesting studies.

For the clustering methods to work properly, the data should be well separated meaning that clusters are easily separable. However, this is usually not the case in microarray studies. Diffuse or interpenetrating groups cause problems in the determination of cluster boundaries during the clustering, leading to different results when using different methods and even the same method with different parameters. Therefore, the user has to consider clusters as a working partitioning, not as the final truth.

Single linkage is not the best approach for hierarchical clustering, because the result can be remarkably skewed. The method is good for picking outliers that are connected in the very last steps of the process. Complete linkage tends to produce very tightly packed clusters. The method is very sensitive for the quality of the data. K-means is one of the simplest and fastest clustering methods. The result is dependent on the initial location of centroids. This problem can be overcome by applying a number of intialization approaches.

10.8 Visualization

Gene expression experiments may include millions of individual data items. It is not possible to comprehend such an overwhelming amount of information without proper visualization methods. Scatter plots are very useful for the initial analysis and comparison of data sets (Figure 10.6).

It is customary to present the clustering results by grouping genes of clusters next to each other (Figure 10.7). In SOM based analysis, also the order of the clusters contains information about the relationships between clusters. For the manual analysis of the goodness of clustering, one has to look at the expression patterns within the generated clusters. Genes within a cluster should follow the average expression pattern of the cluster. Because every gene has to be assigned to a cluster, genes with unique expression patterns do not fit well in any group. The reason for this behavior may be truly different expression profile or it may originate from experimental errors or difficulties. It is impossible to find the reason without further experimental tests unless the behavior is consistent in independent experiments.

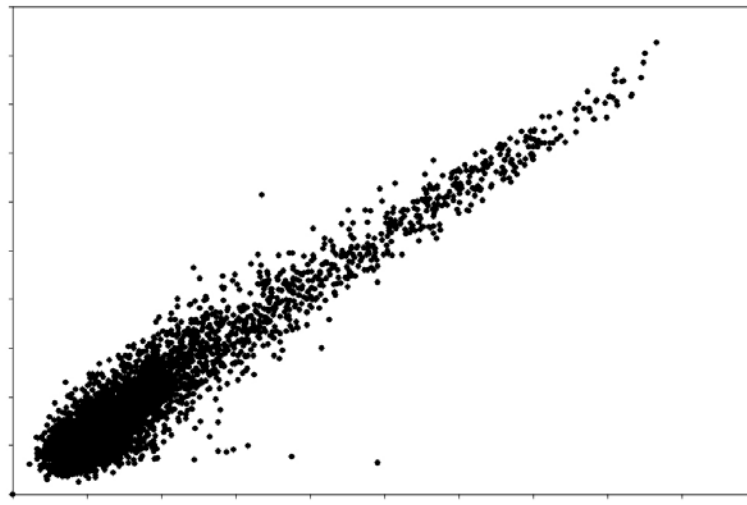

Figure 10.6: *An example of scatter plot.*

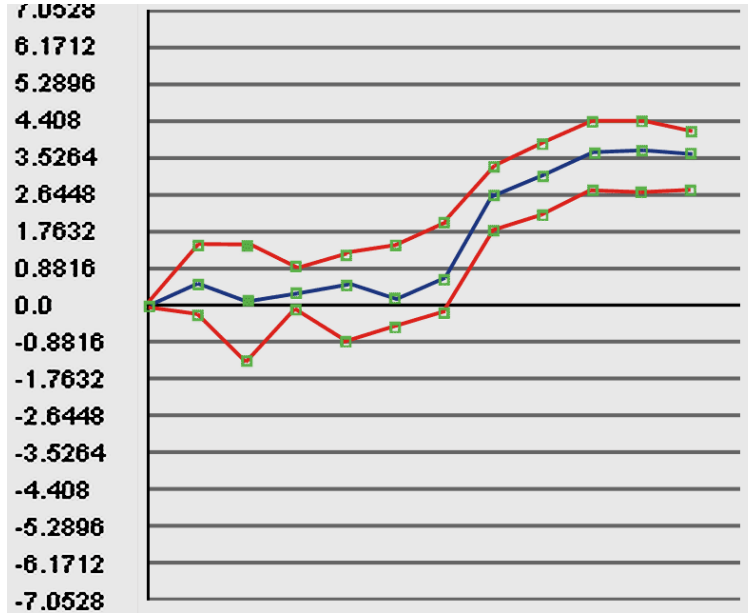

Figure 10.7: *An example of expression pattern within one cluster.*

In microarray studies, usually a red/green color scheme is applied. The data are shown in a matrix format, with each row representing all the hybridization results for a single gene of the array, and each column representing the measured expression level for all genes at a single time point. To visualize the results, the expression level of each gene is represented by a color, red representing overex-

pression and green underexpression within each row. The color intensity represents the magnitude of deviation. It is naturally possible to use other colors, too, and it has been recommended to use a blue/yellow scheme instead, and avoid the problems with color blind persons. Most visulization packages allow free choice of the colors.

Red/green figures can be found from most microarray analysis articles. These figures provide a general overview of the data. With large data sets, it is not possible to identify or even mark individual genes into such figures. Often either more detailed figures of certain types or groups of genes are provided or the whole data is in a (web accessible) appendix. From the beginning of 2003, major journals have required the data to be submitted to data repositories (see chapter on MIAME/MGED system).

10.9 Programs for clustering and visualization

Several programs are available for clustering and visualization of gene expression patterns (Table 10.1). Here we present an incomplete list of programs that we have found useful. Many programs are freely available on the Internet. Both the GeneSpring and Kensington package available at CSC contain several options for clustering. Despite the initial time required for learning the use of these programs, they provide some benefits, mainly due to allowing several analyses to be done within a single package. These programs are not as such any better than freely available programs. They, however, might have a more user-friendly interface.

Table 10.1: *Freely available software for cluster analysis and visualization.*

Program	Description
Cluster	Performs hierarchical clustering, self-organizing maps
SAM	Significance Analysis of Microarrays: Supervised learning
ScanAlyze	Processes fluorescent images of microarrays
TreeView	Graphical results of analyses from Cluster
Expression Profiler	Analysis and clustering of gene expression data
GeneCluster	Self-organizing maps
J-Express	Clustering and visualization

Program	Author	Platform
Cluster	Michael Eisen	Windows
SAM	Rob Tibshirani	Excel Add-in
ScanAlyze	Michael Eisen	Windows
TreeView	Michael Eisen	Windows
Expression Profiler	EBI	Web
GeneCluster	Whitehead Institute / MIT	Java
J-Express	MolMine	Java

10.10 Function prediction

Genes that fall into the same cluster might have a similar transcription response to a certain treatment. It is likely that some common biological function or role is acting in the background. If the cluster has a bunch of genes with known and similar functions, the characteristics of unknown genes in the cluster can be inferred. Function prediction can be assisted with additional data on sequence similarity, phylogenetic inference based on sequences, posttranslational modifications, cellular destination signals, and so on. Taken together, a sufficiently large number of common features can be used to make fairly accurate predictions of gene functions.

10.11 GeneSpring and clustering

GeneSpring offers five different clustering and classification algorithms, one of which is a supervised method. These are hierarchical clustering (gene and experiment trees), K-means clustering, self-organizing maps, principal component analysis, and parameter value prediction. Clustering tools are invoked from the menu *Tools->Clustering*, the supervised classification method can be find from *Tools->Predict parameter values*, and the principal component analysis is located in *Tools->Principal components analysis*.

10.11.1 Clustering tool

The clustering tool has four fields, which need to be filled in before running the analysis (Figure 10.8). From top to down, the first box indicates the gene list to be clustered. Initially the list is the same that was highlighted when the clustering tool was invoked, but it can be changed from the navigator bar on the left. Next box contains the information about the experiment to be clustered. This can also be easily changed from the navigator bar on the left. The pull-down menu offers the choice of three clustering algorithms (hierarchical, K-means and SOM). The setting for the current analysis is in the box below the pull-down menu. For K-means, the number of clusters needs to specified, as well as the number of iterations and the desired measure of similarity.

GeneSpring has several different similarity measures, which fall into the following categories: correlation, confidence, and distance. The selection of the similarity measure should be given some thought, because it significantly affects the generated results. Pearson's correlation emphasizes both over- and underexpressed genes, and the Standard correlation finds especially overexpressed genes. Spearman's correlation is highly similar to Pearson's correlation except it uses ranks for the calculation of the correlation coefficient (and is thus a nonparametric measure of correlation). Distance measures the euclidian distance between two gene expression profiles. It is calculated as the square root of averaged squared deviation of the profiles. Spearman's confidence measures the probability of getting a correlation of S or higher by chance alone, if the true correlation is zero.

If there is only one measurement per gene (*i.e.*, one chip), only the distance measure can be used for the clustering of the genes. If there are two replicates of

every gene, the Standard correlation can also be used. If there are three replicates, Pearson's correlation becomes available, and with five replicates the confidence measures can be used.

In other words, the decision about the applied similarity measure depends on the biological question you are interested in, and the amount of replicates in your dataset.

After specifying the aforementioned settings, the run can be started by clicking the Start-button on the bottom of the window. When the run has ended, you can name and save the clustering result. The result appears on the main screen of GeneSpring. Thereafter, all the results can be found from the navigator bar under the folder classification. Note that the hierarchical clustering results are stored under two different folders, Gene Trees and Experiment Trees.

After viewing the clustering results, you can get back to the original view by selecting *View->Unsplit window*. Hierarchical clustering results can be dismissed, for example, by selecting *View->blocks*.

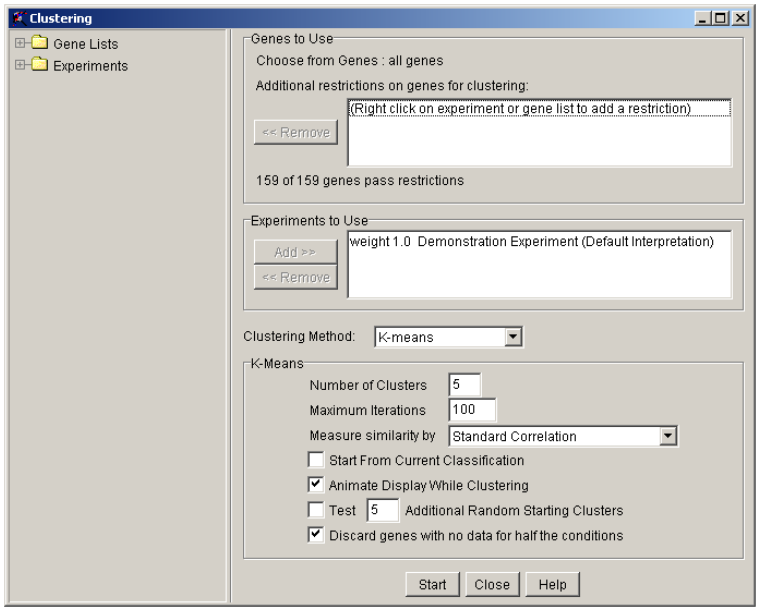

Figure 10.8: *The clustering tool in GeneSpring.*

10.11.2 Principal components analysis tool

Principal component analysis can be used for creating a set of the most significant expression patterns from the data. It can also be used for checking the results of the clustering methods. After selecting *Tools->Principal components method* the analysis is run, and the results are displayed. The opening window contains the most significant profiles. Double-clicking one profile transfers the view to the Gene Inspector, where genes with a similar expression profile can be searched.

The results are displayed in a scatter plot format in the main window. If you

want to compare PCA with some clustering results, you can right-click on one clustering result in the navigator bar, and select Set as coloring scheme from the appearing menu. A good clustering result often creates clusters that do not overlap with each other in the PCA scatter plot.

10.11.3 Predict parameter value tool

The predict parameter value tool (Figure 10.9) is used in situations, where we have a certain set of known samples, and based on these, we want to predict in which group the new, unknown samples fall. For example, if we have information about the leukemia type of 60 samples, we can find the genes, which differentiate these leukemia types from each other. After finding the suitable set of genes, the identity of the unknown samples can be predicted. GeneSpring uses the K-means algorithm described by Golub *et al*.

Training and test sets (experiments) need to be specified. GeneSpring also needs to know which parameter contains the information about the groups to be compared (parameter to predict). After cross-validating the test set, the test set can be predicted. The result of the analysis is a prediction of the test set sample identities and a set of genes differentiating the selected groups.

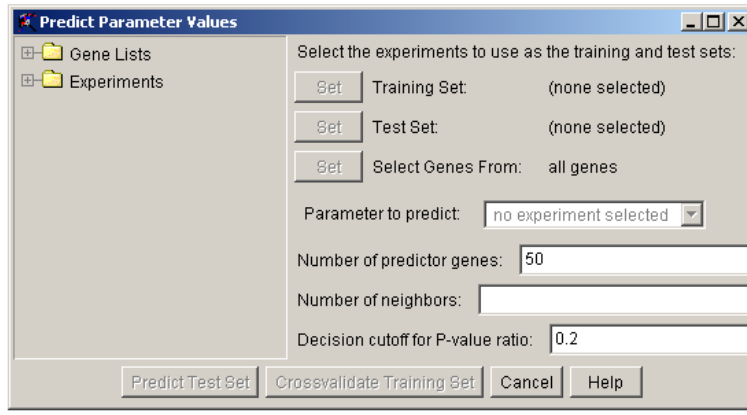

Figure 10.9: *The predict parameter value -classification tool in GeneSpring.*

10.12 Suggested reading

1. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., Lander, E. S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286, 531-7.

2. Brazma, A., Vilo, J. (2000) Gene expression data analysis, FEBS Lett. 480, 17-24.

This chapter was written by Mauno Vihinen, and Jarno Tuimala.

Part III

Data mining

11 Gene regulatory networks

11.1 What are gene regulatory networks?

By combining gene expression analysis, perturbations or treatments, and mutations of genes we can study processes like signal transduction and metabolism yielding information on molecular effects or functions of specific genes. However, gene expression data permits us to go beyond the traditional line of research and to study finer structures of molecular pathways exposing causal regulation relations between genes. The finer structures do not only describe the nature of interactions between genes, inhibition or activation, but also exemplify direct and indirect effects of genes. For example, is gene A regulating gene B or vice versa, is the regulation direct or indirect where there is a mediating gene C (or maybe many) so that A regulates C and then C regulates B. Inspecting the finer structures, which are called regulatory networks, gives us a more intricate view of molecular interactions offering further possibilities for medical interventions.

Inferring regulatory networks from expression data, a process which is called reverse-engineering, is not a computationally simple problem because an enormous amount of time is needed even when a trivial approach is applied. Therefore, we sketch principles and methods with an example approach applicable for the inference process in next sections.

11.2 Fundamentals

It is customary to visualize regulatory dependencies between genes by a graph $G(V,E)$ with a vertex set V consisting of the genes examined and an edge set E, see Figure 11.1. A directed edge $E_{(i,j)}$ between vertices V_i and V_j represents regulation between genes attached to vertices V_i and V_j, and the direction of the edge, denoted by $V_i \rightarrow V_j$, reflects the order of regulation where gene V_i regulates gene V_j (up- or downregulation). Node V_i is called a *parent* of V_j and node V_j is called a *child* of V_i. To present logical structures of group regulations, which appear extensively in nature, hyper edges should be used, where an edge is adjacent to several genes, but these are used quite rarely because of the convenience of simple drawings. A graph like G is called *gene regulatory network*.

By inspecting a putative regulatory graph, like G, build from earlier experiments, we can pose several questions. First of all, is the structure right, or if there

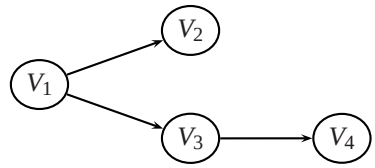

Figure 11.1: *Graph G consisting of four nodes V_1, V_2, V_3, V_4, and tree edges $E_{(1,2)}$, $E_{(1,3)}$, and $E_{(3,4)}$.*

are several possible structures with respect to the experiments done so far, which one is right? So a graph represents hypotheses based on current knowledge. By locating central nodes or nodes of interest we can conduct additional experiments to confirm the dependencies around nodes or get an idea which of the graphs is the correct one. Choosing the best node to inspect more carefully is not trivial, at least not, when we try to make distinctions between putative graphs and we try to minimize experiment costs at the same time. To this category belongs also the question: is the order of regulation correct? Manual perturbation, like the deletion of a gene, might give a clear clue on the order of regulation. Finally, if we stick to the obtained structure there is still questions on the strength of each regulation, whether they are they right.

But before all the above questions, we need to know how to construct such a graph directly from data, because, in theory, any kind of an edge is possible between any set of vertices, and so the problem is to infer the true connections based on the data produced by the experiments. This approach is called *reverse-engineering of gene regulatory networks*. It is interesting to note that similar to biochemical reactions, sequences, and three dimensional structures having their own motifs, it is clear that regulatory networks have to have their own regulatory motifs or circuit motifs based on the lower level motifs.

To infer a regulatory network, several distinct expression samples of the genes of discourse are needed. In a time series analysis, the expression levels are recorded along fixed time points, while in a perturbation analysis, the expression levels are recorded separately for each manual perturbation (over/under, deletion) of specific genes. A fundamental difference between perturbation data and time series data is that perturbation allows a firm inferring order of regulation while time series data can only reveal the probable regulation direction. The problem can be overcome by combining a few manual perturbation experiments with time series data. In both cases, the genes are monitored at fixed points inducing an expression matrix

$$\{X_i[j] \mid 1 \leq i \leq n, 1 \leq j \leq m\},$$

where $X_i[j]$ denotes the expression level X_i of gene or node V_i at point j, see Table 11.1. Based on the data, we can try to infer the most prominent regulatory subnetworks, or, if we know the structure of some subnetwork beforehand, we can try to model mathematically the regulations between nodes in the subnetwork.

Because the number of possible networks increases super-exponentially on the number of genes, prominent regulatory networks are usually searched for by using

Table 11.1: *Top, part of the table for time series expression data of genes SWI5, CLN2, CLN3, and CLB1 [1]. The mean of whole data was* 0*. Bottom, the expression levels are discretized so that values higher than or equal to the mean get a value of* 1 *and values lower than the mean get a value of* 0*.*

gene	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
1 (SWI5)	-0.41	-0.97	-1.46	0.16	0.74	0.72	1	0.77	0.3
2 (CLN3)	0.49	0.62	0.05	-0.13	0.02	0.04	-0.14	0.24	0.91
3 (CLB1)	0.6	-0.53	-1.37	1.03	1.13	1.27	1.04	1	0.07
4 (CLN2)	-1.26	1.6	1.54	0.31	-0.14	-0.88	-1.7	-1.88	-1.7
1 (SWI5)	0	0	0	1	1	1	1	1	1
2 (CLN3)	1	1	1	0	1	1	0	1	1
3 (CLB1)	1	0	0	1	1	1	1	1	1
4 (CLN2)	0	1	1	1	0	0	0	0	0

Table 11.2: *Conditional probabilities based on the structure of G and the bottom values of Table 11.1.*

$\Pr(X_1 = 1)$
6/9

X_1	$\Pr(X_2 = 1)$
1	4/6
0	1

X_1	$\Pr(X_3 = 1)$
1	1
0	1/3

X_3	$\Pr(X_4 = 1)$
1	1/7
0	1

supervised approaches, where some intrinsic partial knowledge about regulations between genes are formulated by implicit or explicit rules. For the mathematical modeling of regulation inside a network, there are many approaches like Bayesian network, Boolean network and its generalization, ordinary and partial differential equations, qualitative differential equations, stochastic master equations, Petri nets, transform grammars, process algebra, and rule-based formalisms, to mention some.

We will use Bayesian network as a sample tool for several reasons: its framework offers a clear separation of structure and parameter optimization, and adding predefined rules and information is easy. Moreover, it is widely used for microarray data.

11.3 Bayesian network

Bayesian network modelling consists of two parts: the qualitative part, where direct (causal) influences between nodes V are depicted as directed edges E in graph G, and the quantitative part, where local conditional probability distributions of expression levels are attached to nodes V of G. Thus, expression levels X_i of nodes V_i are considered as random variables and the edges represent conditional dependencies between distributions of the random variables.

Let us consider a binary case where a gene can be “off”, denoted by 0, or “on”, denoted by 1, but not both (see the bottom values of Table 11.1). Having a fixed structure like graph G in Figure 11.1, the conditional probabilities are easy to calculate, see Table 11.2.

The benefit of this modeling is that we need to store only $O(2^k n)$ parameters

when each node has at most k parents, *i.e.*, one "success" probability for each row in each table just calculated.

This is clear advantage over $O(2^n)$ parameters when the expression value of a node depends on the expression values of all other nodes. Thus, parameters pertain only to local interaction.

Underlying the above discussion there are some basic probability rules that are examined next. The joint probability for random variables X and Y (not necessarily independent) is calculated using the conditional probability:

$$\Pr(X \cap Y) = \Pr(X,Y) = \Pr(Y \mid X)\Pr(X) = \Pr(X \mid Y)\Pr(Y),$$

where notation $\Pr(X \mid Y)$ denotes the probability of observation X after we have seen evidence Y, see Figure 11.2. Reorganizing the equation induces Bayes' rule

$$\Pr(X \mid Y) = \frac{\Pr(Y \mid X)\Pr(X)}{\Pr(Y)}$$

while generalization of it forms the chain rule

$$\begin{aligned}\Pr(K,X,Y,Z) &= \Pr(Z \mid K,X,Y)\Pr(K,X,Y)\\ &= \Pr(Z \mid K,X,Y)\Pr(Y \mid K,X)\Pr(K,X)\\ &= \Pr(Z \mid K,X,Y)\Pr(Y \mid K,X)\Pr(X \mid K)\Pr(K).\end{aligned}$$

If variables X and Y are independent, then

$$\Pr(X \cap Y) = \Pr(X)\Pr(Y),$$

and if variables X and Y are independent for given a value of Z, then

$$\Pr(X \mid Z,Y) = \Pr(X \mid Z).$$

Based on our dependency graph G, probability $\Pr(X_1, X_2, X_3, X_4)$ can be simplified as follows

$$\Pr(X_1,X_2,X_3,X_4) = \Pr(X_1)\Pr(X_2 \mid X_1)\Pr(X_3 \mid X_1)\Pr(X_4 \mid X_3).$$

For example, $\Pr(X_4 \mid X_1,X_2,X_3) = \Pr(X_4 \mid X_3)$ because for given the value of X_3, X_4 is independent of the values of X_1 and X_2. So, it is enough to consider only direct local dependencies from parents when evaluating the probabilities (for X_4 the only parent is X_3). We see that in Bayesian networks the probability calculations follow in a natural way the structure of a graph.

11.4 Calculating Bayesian network parameters

Having graph G and the expression matrix $D = \{X_i[j]\}$, our aim is to obtain distribution dependency parameters $\theta = \{\theta_1, \theta_2, \ldots\}$ that are fitted best to the structure of G and data D. This is done by using the maximal likelihood principle where we maximize the likelihood function $\mathrm{L}(\theta : D)$:

$$\mathrm{L}(\theta : D) = \Pr(D : \theta) = \prod_{j=1}^{m} \Pr(X_1[j], X_2[j], \ldots X_n[j] : \theta).$$

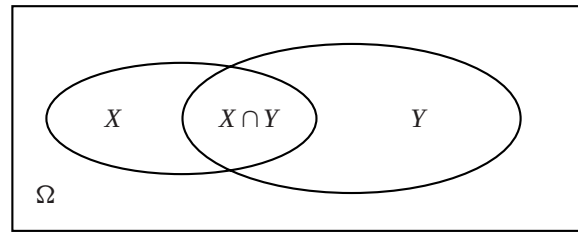

Figure 11.2: *A Venn diagram of the probability space* Ω*, where the areas denote probabilities, for example,* $\Pr(X)$*, is the left circle area divided by the total area of rectangle* Ω*. It is easy to see that* $\Pr(X \cap Y) = \Pr(Y \mid X)\Pr(X)$*.*

By using the structure of G (see again Figure 11.1) we can decompose our optimization problem to independent subproblems:

$$\begin{aligned} \mathrm{L}(\theta : D) \;=\; & \prod_{j=1}^{m} \Pr(X_1[j] : \theta_1) \cdot \prod_{j=1}^{m} \Pr(X_2[j] \mid X_1[j] : \theta_1) \\ & \cdot \prod_{j=1}^{m} \Pr(X_3[j] \mid X_1[j] : \theta_3) \cdot \prod_{j=1}^{m} \Pr(X_4[j] \mid X_3[j] : \theta_4), \end{aligned}$$

or, in general,

$$\prod_i \mathrm{L}_i(\theta_i : D).$$

The network structure decomposes the parameter set θ to independent parameters θ_1, θ_2, θ_3 and θ_4, which are computationally fast to estimate.

For simplicity, we consider again the binary case, see bottom of Table 11.1. Since V_1 has no parents, *i.e.*, it is independent of other nodes, its likelihood function $\mathrm{L}_1(\theta_1 : D)$ corresponds to the binomial probability where the probability of getting value 1 is θ_1 while $1 - \theta_1$ is the probability of getting value 0. Thus,

$$\mathrm{L}_1(\theta_1 : D) = \Pr(D : \theta_1) = \prod_{j=1}^{m} \Pr(X_i[j] : \theta_1) = (\theta_1)^{N(X_1=1)}(1-\theta_1)^{m-N(X_1=1)},$$

where $N(\cdot)$ denotes the number of each occurrence in data with respect to the requirements inside parentheses. The estimator $\hat{\theta}_1$ for θ_1 maximizing the product is simply the already calculated probability for $\Pr(X_1 = 1)$, *i.e.*,

$$\hat{\theta}_1 = \frac{N(X_1 = 1)}{N(X_1 = 1) + N(X_1 = 0)} = 6/9.$$

Similarly, the conditional estimators for parameters $\theta_{(2|X_1)}$, $\theta_{(3|X_1)}$ and $\theta_{(4|X_3)}$ are the probabilities already calculated:

$$\begin{aligned} \hat{\theta}_{(2|X_1=1)} &= 4/6, & \hat{\theta}_{(2|X_1=0)} &= 1, \\ \hat{\theta}_{(3|X_1=1)} &= 1, & \hat{\theta}_{(3|X_1=0)} &= 1/3, \\ \hat{\theta}_{(4|X_3=1)} &= 1/7, \text{ and} & \hat{\theta}_{(4|X_3=0)} &= 1. \end{aligned}$$

Together these estimators form $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_{(2|X_1)}, \hat{\theta}_{(3|X_1)}, \hat{\theta}_{(4|X_3)}\}$, which is the best explanation for the observed data D restricted to the structure of G, and we can compute

$$\begin{aligned} \mathrm{L}(\theta : D) &= (\hat{\theta}_1)^6 \cdot (1-\hat{\theta}_1)^3 \cdot (\hat{\theta}_{(2|X_1=1)})^4 \cdot (1-\hat{\theta}_{(2|X_1=1)})^2 \\ &\quad \cdot (\hat{\theta}_{(3|X_1=0)})^1 \cdot (1-\hat{\theta}_{(3|X_1=0)})^2 \cdot (\hat{\theta}_{(4|X_3=1)})^1 \cdot (1-\hat{\theta}_{(4|X_3=1)})^6 \\ &= (6/9)^6 \cdot (3/9)^3 \cdot (4/6)^4 \cdot (2/6)^2 \cdot (1/3)^1 \cdot (2/3)^2 \cdot (1/7)^1 \cdot (6/7)^6 \end{aligned}$$

as the maximal likelihood value for graph G. Note that we have dropped out terms $1^x = 1$ and $0^0 = 1$.

Usage of more states for genes like "low", "medium", and "high" follows the multinomial theory conveniently generalizing the binomial theory. Using the parameters, we can formulate hypothetical behaviors of genes in different situations fixing some expression levels and observing others; moreover, updating the parameters is easy as new data arrives.

11.5 Searching Bayesian network structure

Given a graph G, we know now how to calculate the parameter set θ^G maximizing the likelihood score $\mathrm{L}(\theta^G : D)$. To restrict the search space of possible networks, it is practical to have constrains on which kind of networks are allowed, that is, which dependencies are possible between genes. This helps a lot because of the super-exponential number of possible networks; for example, the number of possible graphs consisting of four nodes is 64 since there are $\binom{4}{2}$ possible (undirected) edges, each of which can be taken to form a graph making $2^{\binom{4}{2}}$ different graph in total. For evaluation of the resulting candidate networks, we use again the likelihood score calculation of a network structure using maximum likelihood estimates of θ^G when the structure is searched

$$\begin{aligned} \mathrm{L}(G, \theta^G : D) &= \prod_m \Pr(X_1[m], X_2[m], \ldots, X_n[m] : G, \theta^G) \\ &\quad \prod_m \prod_n \Pr(X_n[m] \mid \mathrm{Parents}(X_n)[m] : G, \theta_n^G). \end{aligned}$$

Here, $\mathrm{Parents}(X_n)$ denotes the expression values of all parents of node V_n and θ_n^G denotes the parameter of node V_n in graph G. Instead of direct calculation it is more convenient to use a logarithm of $\mathrm{L}(G, \theta^G : D)$ for comparing different structures. After simplifying we have a surprisingly simple formula

$$\log(\mathrm{L}(G, \theta^G : D)) = m \sum_{i=1}^{n} (\mathrm{I}(V_i, \mathrm{Parents}(V_i)) - \mathrm{H}(V_i)),$$

where $\mathrm{H}(V_i)$ is the entropy of expression values X_i of node V_i

$$\mathrm{H}(V_i) = \sum_{x=0,1} \Pr(X_1 = x) \lg \frac{1}{\Pr(X_1 = x)}$$

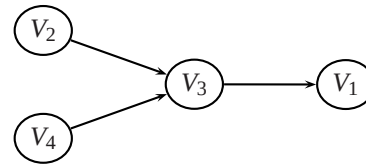

Figure 11.3: *A different structure explaining conditional dependencies between the expression values of nodes V_1, V_2, V_3, V_4.*

which can be ignored because it is the same for all network structures. The function $\mathrm{I}(V_i, \mathrm{Parents}(V_i))$ is the mutual expression information between node V_i and its parent nodes $\mathrm{Parents}(V_i) = \{P_1, P_2, \ldots\}$ defined by

$$\sum_{\substack{x=0,1 \\ (\forall i): x_i = 0,1}} \Pr(X_i = x \bigcap_i P_i = x_i) \lg \frac{\Pr(X_i = x \bigcap_i P_i = x_i)}{\Pr(X_i = x) \prod_i \Pr(P_i = x_i)}.$$

Function $\mathrm{I}(V_i, \mathrm{Parents}(V_i)) \geq 0$ measures how much information the expression values of nodes $\mathrm{Parents}(V_i)$ provide about V_i. If V_i is independent of parent nodes, then $\mathrm{I}(V_i, \mathrm{Parents}(V_i))$ has the value of zero. On the other hand, when V_i is totally predictable for given values of $\mathrm{Parents}(V_i)$, then $\mathrm{I}(V_i, \mathrm{Parents}(V_i))$ reduces into the entropy function $\mathrm{H}(V_i)$. It should be noted that in general $\mathrm{I}(X, Y) \neq \mathrm{I}(Y, X)$ so the direction of edges matters.

Is the structure presented in Figure 11.1 optimal with respect to the data? What about the structure in Figure 11.3. Because our data set is small we do not resort to the above calculations but we calculate the parameter set $\hat{\theta}$:

$$\begin{array}{llll} \hat{\theta}_2 = 7/9, & & \hat{\theta}_4 = 3/9, \\ \hat{\theta}_{(3|X_2=0,X_4=0)} = 1, & & \hat{\theta}_{(3|X_2=0,X_4=1)} = 1, \\ \hat{\theta}_{(3|X_2=1,X_4=0)} = 1, & & \hat{\theta}_{(3|X_2=1,X_4=1)} = 0, \\ \hat{\theta}_{(1|X_3=0)} = 1/3, & \text{and} & \hat{\theta}_{(1|X_3=1)} = 1. \end{array}$$

The comparison of these figures with the previous ones shows that the new graph is more probable with respect to data.

11.6 Conclusion

Our example data was complete without any missing observations of expression values, which happens quite rarely in reality. Two of the most familiar conventions to deal with the problem of missing expression values is to omit the data rows missing some values or substituting the most common values for the missing values. Of course, what is the best policy, is an everlasting topic to debate on.

Even a more difficult problem is to decide how the discretization from expression values to logical "on/off" values should be done. Can we really use a global step value like in our example or should we use use gene-specific values? On the other hand, can we use a single limit because then a small variation in measurements can toss a gene between the "on" and "off" state without any real reason?

Table 11.3: *Bayesian network softwares (some are commercial).*

Bayesia	www.bayesia.com
JavaBayes	www-2.cs.cmu.edu/ javabayes
PowerConstructor family	www.cs.ualberta.ca/ jcheng/bnsoft.htm
Bayesian Knowledge Discoverer	kmi.open.ac.uk/projects/bkd
GeNIe	www.sis.pitt.edu/ genie
Prevision	www.prevision.com
Hugin	www.hugin.com
Norsys	www.norsys.com
Knowledge Industries Company	www.kic.com
Microsoft MSBN system	www.research.microsoft.com/msbn

One cure for the last problem might be to use fuzzy logic, where a gene has an increasing grade of being "on" and after a fixed point it is completely or fully "on"; similarly for "off" but reversed.

Today, there are some Internet-based servers set up for inspection of reverse-engineering results based on researchers' own data and few software packages, see Table 11.3. Our better graph was calculated by using a server hosted by Technical High School (`B-course`). In future, some of these tools will be integrated to common gene expression software tools like GeneSpring and Kensington.

11.7 Suggested reading

1. Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9, 3273-97.

2. Pe'er, D., Regev, A., Elidan, G., Friedman N. (2001) Inferring subnetworks from perturbed expression profiles, Bioinformatics 17, 215S-224.

3. D'haeseleer P., Shoudan Liang S., Somogyi R. (2000) Genetic network inference: from co-expression clustering to reverse engineering, Bioinformatics 16, 707-726.

This chapter was written by Tomi Pasanen, and Mauno Vihinen.

12 Data mining for promoter sequences

12.1 Introduction

In all gene expression studies there is an underlying element of gene regulation. In order to be expressed, the gene has to be transcribed to RNA in a regulated process. Therefore, many expression experiments can be used to probe the regulation patterns. This is most obviously the case when you observe gene expression in different conditions or gene expression over time in some process, but comparisons of different cells or tissues or diseased v. healthy tissues may benefit from analysis of regulation, too. Whatever your study, there will be changes in the expression levels of some genes, either upregulation or downregulation. How the expression levels can be correlated to mechanisms of regulation is the topic of this chapter.

We start with an introduction about gene regulation and promoters and then turn to the practical issues of bioinformatics in promoter data mining . We deal with finding promoter sequences, searching known regulatory sequence patterns and searching potential regulatory patterns that are not known a priori.

12.2 Introduction

Changes in the expression levels of genes can be due to a number of factors. First, there are such large-scale factors as chromatin accessibility: some regions of DNA are packed too tightly to allow strand separation or even binding of various factors. One well-known mechanism of chromatin packing is found in methylation of cytidines in CpG islands to effect gene silencing via heterochromatin formation, and another one is related to histone acetylation/methylation. Second, there are specific transcription factors that may inhibit or activate transcription. The balance between activating and inhibiting transcription factors can determine the expression level in loosely packed euchromatin. Third, there are some genes that seem to be on "by default" without much apparent regulation, the so-called maintenance or household genes which are needed in all cell types. "Household gene", however, does not imply constant expression; in any gene the expression level will fluctuate over time due to the general metabolic activity of the cell, especially the activity and amount of parts in the transcription machinery itself. If maintenance genes are used for normalization of array data it is necessary to verify the constant expression, *e.g.*, by RT-PCR in the system and conditions of the experiment. (See also the chapter on

normalization.)

Data mining for gene regulation is currently feasible only in the second case above, *i.e.*, in search and study of transcription factor binding sites.

Most known transcription factor binding sites are located close to the transcription start site (TSS), particularly in the 500 bp directly upstream (5') of TSS. This upstream region is often referred to as the promoter region (but sometimes the binding sites themselves are called promoters). Other regions that activate transcription (the enhancers) may occur almost anywhere downstream (3') or upstream from the gene, even 20 kB away from the promoter region, or in the introns. The methods for the search of regulatory elements that we describe could be used for potential enhancer regions as well as for the promoter regions, but commonly only promoters are analyzed. This because a larger search space weakens the signal-to-noise ratio. Therefore, we limit our discussion to data mining in promoter regions only.

Coexpressed genes are often grouped by different clustering methods. It is reasonable to present the hypothesis that some of the similar changes of expression are due to the effect of the same or similar transcription factors – therefore it makes sense to search for shared sequences that could be transcription factor binding sites. Of course, depending on your clustering (cluster sizes and number), you may have several mechanisms working within one cluster, or you may have the same mechanism affecting members of several clusters, but in any case, at least some clusters may show enrichment of similar transcription binding sites.

In brief, it is presumed that coexpression implies coregulation, and looking for shared patterns in promoter regions allows you to formulate hypotheses of regulation mechanisms for focused experimental testing. However, before verification, your results will be only sets of potential promoter sites of potentially coregulated genes.

12.3 Finding promoter region sequences

Promoter analyses need the promoter region sequences, but getting them may not be trivial! In this section, we introduce some tools and data sources that we have found useful for retrieving promoter regions for human genes .

First of all, it is instrumental to know precisely which genes are contained in the microarray. The data from the manufacturer is not always systematically presented or complete. There may be a mix of references to GenBank accession numbers, UniGene clusters, gene names, protein accession numbers, RefSeq mRNA or protein accession numbers, and LocusLink ID codes, and if you are lucky, even actual sequences that are contained on the array.

Even if we skip the concern over the correctness of the data provided by the manufacturer, there are some issues that you should be aware of before you can trust your gene assignments:

- gene names have been subject to many changes, so your data may not contain the current official names,
- the UniGene clusters of ESTs are dynamic entities that undergo fusions and

fissions in every UniGene build as more data becomes available, so your ESTs may be assigned to other genes, or may lack a UniGene association in the current build,

- the data regarding your filter or chip may have been updated after it left the factory, so the shipped gene lists may not be as complete and up-to-date as you can find at the manufacturer's web site. You should definitely work with the most recent data available as long as it refers to the same manufacturing batch that you actually used,
- you may find several RefSeq mRNA and protein codes for a single gene locus, corresponding to alternatively spliced forms,
- the provided EST code is not always the actual sequence on the array, because the manufacturer may have used a longer and more correct version of the 5'-end of the corresponding mRNA.

The ultimate source for the promoter regions is in the genome data. The annotations of gene locations are still incomplete, and there are no ready-made tools for pulling out the regions preceding each gene. Currently (early 2003) we have found that human promoters are easiest to retrieve in large scale from three data sources. They are multi-species resources, so when more genomes become available, the same approaches work.

- University of California in Santa Cruz (UCSC) provides ready-made collections of upstream sequences of as many genes as they can localize based on RefSeq mRNA sequences – different sets contain different lengths of 5'-sequences preceding the TSS, called upstream1000, upsteam2000 etc. The human data from the most recent freeze of November 2002 is at `http://genome.ucsc.edu/goldenPath/14nov2002/bigZips/`, but you will always find most recent data at `http://genome.ucsc.edu/downloads.html` under Full data set.
- LocusLink gives pointers to NCBI RefSeq Contigs, allowing retrieval of promoters by a little programming of your own. LocusLink is essential for using the UCSC data, too, if you do not have RefSeq mRNA codes.
- EnsMART (http://www.ensembl.org/EnsMart/) allows direct retrieval of upstream sequences of any length if you can map your genes to Ensembl gene codes or RefSeq codes (more details follow).

For a smaller number of genes the most reliable data (regarding the placement of TSS) is available in the Eukaryotic Promoter Database (EPD), which contains promoter regions of 500 bp, only from genes with an experimentally verified TSS. Organism-specific promoter databases may exist to fit your needs, consult the genome site of your favourite organism or Michael Zhang's lab at `http://rulai.cshl.org/software/index1.htm`.

If these sources fail, or if you feel you cannot trust all of your findings, you can always resort to using your own similarity searches to place the gene in the genome

assembly, and then pick the region preceding the gene, but this is far from straightforward. It may be advisable to drop data for which you cannot get unambiguous gene mapping. Currently full genome information is in most cases not available for all genes contained in the microarrays.

If you aim at finding your promoter sequences from the UCSC upstream data sets, you need the corresponding RefSeq mRNA accession codes, because these codes are used as identifiers for the promoter sequences. You can also retrieve Ensembl genes with RefSeq codes. Figure 12.1 shows you the paths and tools for arriving at RefSeq codes from other data.

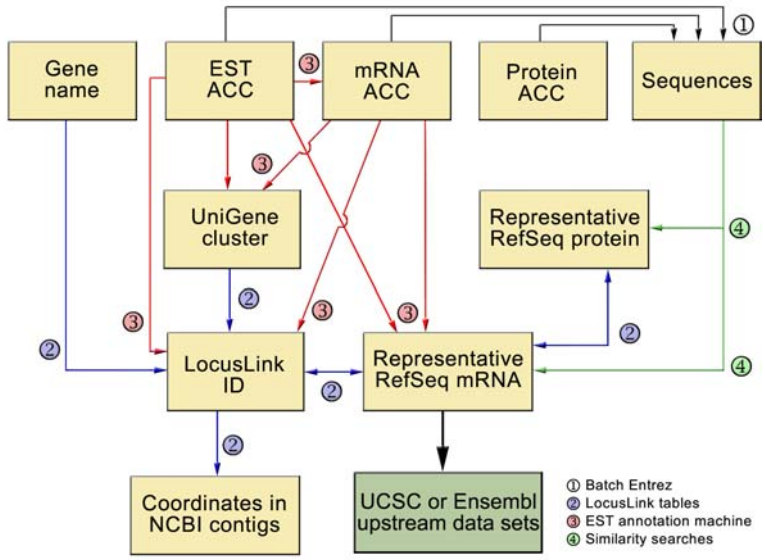

Figure 12.1: *Different data items that you may receive from your microarray manufacturer and how you can find missing data items and the actual sequences.*

The tools featured in Figure 12.1 include:

- similarity searches, such as Blast or BLAT (`http://genome.ucsc.edu/cgi-bin/hgBlat?command=start`). For large-scale searches you probably want to install them locally and use text processing tools such as Perl for extracting relevant information from the result files. BLAT has the advantage of giving directly the coordinates in the genome, and on-line BLAT accepts modest batches of sequence (a few hundred) if you just want to patch missing data,
- EST annotation machine – a nice service at `http://bio.ifom?firc.it/EST_MACHINE/index.html` for obtaining UniGene, RefSeq, and/or LocusLink codes and functional data starting from accession codes of ESTs or other sequences. Batch runs of 1000 codes at once are possible,
- Batch Entrez – mass retrieval of sequences from NCBI (`http://www.ncbi.nlm.nih.gov/entrez/batchentrez.cgi?db=Nucleotide`),

- utilization of LocusLink data (http://www.ncbi.nlm.nih.gov/LocusLink/ – use Download to get the data) – very useful for finding alternative gene names, or RefSeqs corresponding to UniGene clusters etc. In large scale you need programming or setting up a local database for lookups of the information,

- the HUGO nomenclature service (http://www.gene.ucl.ac.uk/nomenclature/) can be used for verifying gene names, too, even though LocusLink files make it easier in large scale, and they aim to conform to official HUGO names,

- CSC also offers tools for fast BLAST searches (gepardi.csc.fi) and mass retrieval of sequencies in FastA format (Seqret in the EMBOSS package).

If you want to be really certain that you know which genes you are dealing with, you should follow several paths in the diagram to see if they all lead you to same Locus IDs. Remember that there may be several RefSeq mRNAs per gene, whereas Locus IDs are unique.

The data files from GeneLynx (http://www.genelynx.org/cgi-bin/a?page=info) are likely to allow many shortcuts to obtain RefSeq mRNA codes, but we have not tried this yet. Likewise, querying of various UCSC genome data tables seems a promising option.

If you work with the Affymetrix Arabidopsis chips, you may want to look at VIZARD (http://www.anm.f2s.com/research/vizard/), which includes annotation and upstream sequence databases for the majority of genes represented on the Affymetrix Arabidopsis GeneChip® array. Whitehead Institute at MIT provides upstream sequence data for *Neurospora crassa* and several other organisms, see e.g. http://www?genome.wi.mit.edu/cgi?bin/annotation/neurospora/

download_license.cgi

12.4 Using EnsMart to retrieve promoter regions

Yet another option for retrieving upstream sequences is found at the EnsMart service (http://www.ensembl.org/EnsMart/) of the Ensembl project. This is limited only by the number of Ensembl genes that are annotated (currently almost 25,000), and it is very easy to use, as illustrated in Figure 12.2. The performance of EnsMart v. UCSC upstream data searches is compared later in this chapter. They have several species to select from:

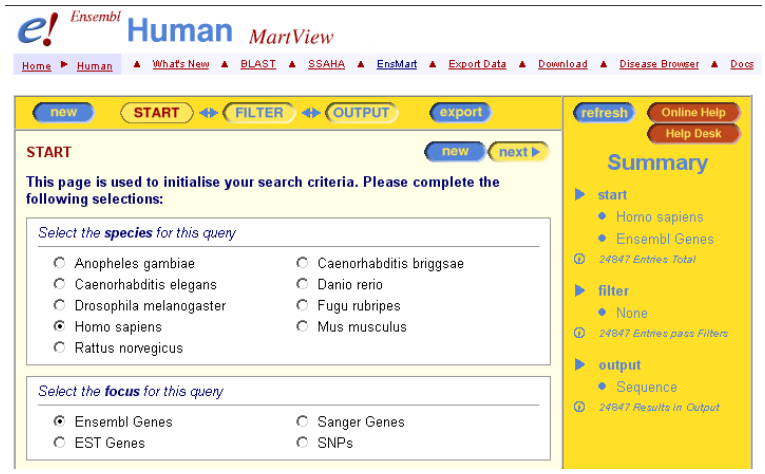

Figure 12.2: *EnsMart start screen.*

Next, enter your list of gene identifiers in the Filter step (Figure 12.3):

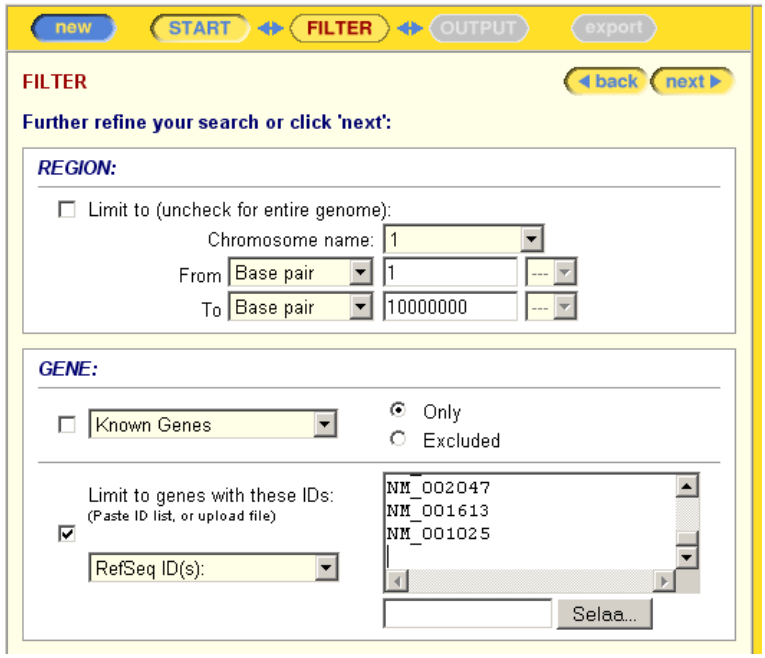

Figure 12.3: *EnsMart filtering. In addition to RefSeq, there are other options, too, but you should be aware of that some mappings are more complete than others. Next to internal Ensembl references, RefSeq is your best choice. A long list of other filtering options is omitted from the figure.*

Then, in the next phase, you should choose output as Sequences, and select your options as below to retrieve your upstream sequences in Fasta format (Fig-

ure 12.4). Additional options for data compression, saving locally etc. are not shown.

For a few microarrays (currently only two human Affymetrix chips), EnsMart provides direct mappings, so you can skip all the previous steps for finding a consistent set of gene identifiers. Instead, you can choose the genes in these chips directly in the Filter step (Figure 12.5).

We can expect such mappings with microarray contents to become more common in the genome sites, both at Ensembl and at UCSC. Therefore, some day retrieving the upstream sequences will be less of a problem.

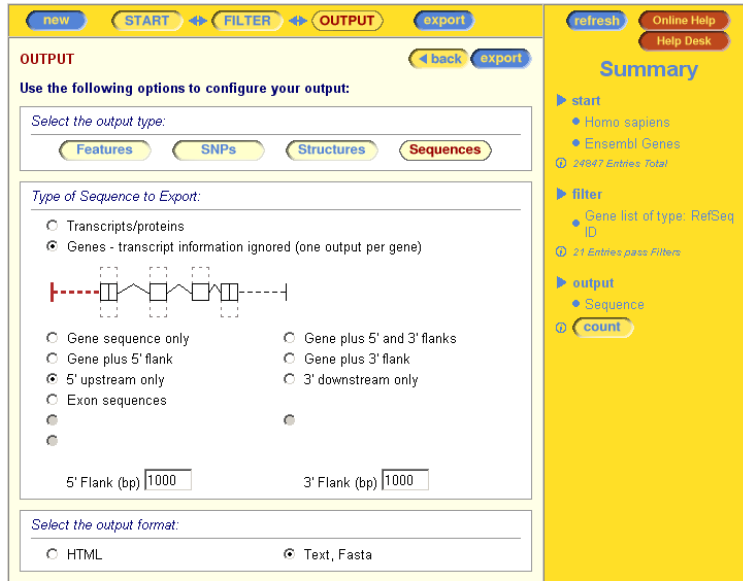

Figure 12.4: *Sequence output options in EnsMart.*

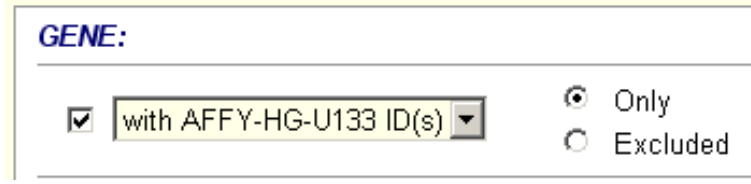

Figure 12.5: *Selecting genes that are represented on AFFY-HG-U133. Compare with Fig. 12.3.*

12.5 Comparison of EnsMart and UCSC searches

In order to evaluate how the most extensive two search systems perform, we took a set of 1187 RefSeq codes (from our own microarray results) and retrieved the upstream sequences from both services. USCS upstream 1000 data set was from

Table 12.1: *Search of upstream sequences from the precomputed upstream data set of UCSC (*`http://genome.ucsc.edu/`*) and an interactive search from Ensembl data via EnsMart (*`http://www.ensembl.org/EnsMart/`*)*

Entries retrieved	Unique mRNA	codes
UCSC	1145	1087
EnsMart	1171	1131
Found in both	1054	
UCSC not Ensmart	33	
EnsMart not UCSC	77	
Not found in either	23	

the November 2002 freeze of the human genome, and Ensembl data was as of 31st March, 2003. Table 12.1 summarizes our findings.

There seem to be two lessons to be learned. First, there is some duplication, so finding the real gene and promoter sequence is not unambiguous even if you know a code in a curated collection and use the most authoritative data sources. Second, currently it is useful to use both services, even though both data sources are expected to become more complete in the near future as the human genome data becomes more fully assembled and annotated. The situation may be different for your favourite organism.

We did not make any comparison whether the sequences or genome locations matched between the two data sets, but we would guess there are some differences.

Two final warnings regarding upstream data:

- in some cases the genome assembly may be broken close to the start of your gene, so that you do not get a full 1000 (or whatever) nucleotides of sequence. In Ensembl this shows as a long string of Ns in the sequence (this may be another reason why Ensembl gives more sequences)

- the start of a RefSeq mRNA may not always correspond to a true transcription start site. There are mRNAs or ESTs which give evidence of longer transcript variants than some of the current RefSeqs. UCSC genome browser only includes entries with a certain TSS in their upstream1000 data set Searching known patterns

Once you have found the genes and their 5' regions you can proceed to the analysis of patterns in the regions.

If you want to run a check of known transcription factor binding sites , Transcription Regulatory Regions Database (TRRD, `http://wwwmgs.bionet.nsc.ru/mgs/dbases/trrd4/trrdintro.html`) is a site with lots of information, but only for on-line searching. The complete database is not available. The Transcription Factor Database (TFD) and some tools that utilize the data are available at `http://www.ifti.org/`.

Transfac database ver. 5.0 (from 2001) is said to be publicly available for academic use (`http://transfac.gbf.de/TRANSFAC/`), but currently most of the

data and services seem to be closed. The EMBOSS program tfscan (`http://www.csc.fi/molbio/progs/emboss/Apps/tfscan.html`) can be used to find Transfac sequence patterns in your sequences once Transfac is set up in the EMBOSS environment.

The most recent versions of Transfac database and its search tools have moved into a commercial environment at `http://www.gene-regulation.de/`. They offer PC software for promoter analysis, Transplorer (Figure 12.6). Their "Transfac Professional" package contains search tools, too.

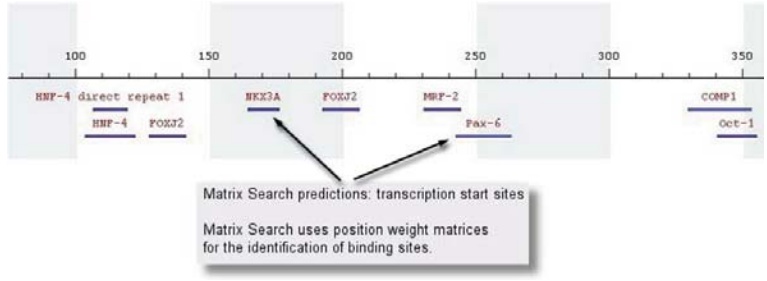

Figure 12.6: *Screenshot from Transplorer.*

Whatever method you choose to search for TF binding sites, you have to bear in mind that they are short patterns that will be found very frequently by chance alone, so you need to confirm the statistical significance of your findings. A search strategy which aims at finding clusters of several sites might work better in this respect.

12.6 Pattern search without prior knowledge of what you are searching for

Several programs and different approaches are available for automated pattern discovery in your promoter region sequences. We will mention here a few examples of different programs and strategies, all of which are capable of finding patterns in sequences without their previous alignment. A more extensive review can be found in Vilo and Kivinen, 2001.

Expression Profiler is a software suite for many tasks in microarray data analysis (`http://ep.ebi.ac.uk/EP/`). It contains the program SPEXS, Sequence Pattern EXhaustive Search, for pattern discovery, which finds regular-expression-like discrete patterns. As an example, this program was used by Vilo *et al.* (2000) to find regulatory patterns from publicly available yeast expression data. To overcome the fuzziness created by the incertainties in clustering, they generated a large number of different clusters (52,000, with lots of overlapping) and analyzed enriched patterns from all of them. After filtering and grouping the patterns, they had 62 groups, 48 of which contained patterns matching some sites in the *Saccharomyces cerevisiae* Promoter Database.

MEME (`http://meme.sdsc.edu/meme/website/intro.html`) is based on

position-dependent letter-probability matrices, which describe the probability of each possible letter at each position in the pattern. The WWW interface analyzes your set of sequences that you believe to contain common patterns, and gives you the strongest patterns that MEME finds. In addition, you get automatically results from a companion program MAST, which locates the occurences of the patterns in your set of sequences.

AlignACE (`http://arep.med.harvard.edu/mrnadata/mrnasoft.html`) provides a Gibbs sampling algorithm for finding patterns in DNA sequences. The program is optimized for finding multiple motifs and it automatically considers both strands in the sequence. The authors used the program to identify successfully several known transcription factor binding sites and to propose some previously unknown regulatory mechanisms based on the earliest public yeast gene expression data sets [2].

All of the above methods depend on a clustering that you have made before pattern discovery, and it is presumed that the clustering is correct. Kimono (`http://www.fruitfly.org/~ihh/kimono/`) takes a unified approach that optimizes clustering and pattern discovery together. The approaches described above first find genes with similar expression patterns, and then see if they have similar promoters. Kimono finds clusters of genes that have similar expression patterns and similar promoters as described by Holmes and Bruno (2000).

12.7 Summary

In summary, we identified a similar expression profile of certain genes using DNA microarrays and clustering tools. We hypothesized that the similar expression was due to common regulatory elements situated in the promoter regions of the genes. We retrieved the promoter sequencies from the databanks, analyzed the promoter regions, and successfully identified a common element on their promoter region.

12.8 GeneSpring and promoter analysis

GeneSpring includes a promoter analysis tool, which can be used for finding novel common regulatory sequencies in a gene list, or to search for a known sequence. The tool can be invoked from *Tools->Find potential regulatory sequencies*. In order to search for potential regulatory sequencies, you need to have a whole genomic sequence of the organism under study. In principle, if only a partial genome of the organism is known, it is not possible to search for regulatory elements (GeneSpring forbids the use of the tool), because the statistical support and frequency values of the elements would be erraneous. However, there is a trick, which allows the analysis of partial genomes. For more information, see the tech note at `http://www.silicongenetics.com/cgi/TNgen.cgi/GeneSpring/GSnotes/Notes/how_contig`.

The tool opens a new window (Figure 12.7). First, you need to select a genelist you want to study, but do not use the “all genes” or “all genomic elements” list, because then you would compare the whole genome against itself, which is not a viable analysis. From the pull-down menu, select whether you want to search for

new sequencies or for a specific sequence. You can also select the lenght of the sequence to be considered a promoter region, how long a regulatory element is being searched, and how many unknown bases are allowed. The longer the sequence, and the larger the number of unknown bases, the longer the analysis time. You have control over the probability statistics: The p-value cut-off for a significant pattern can be modified. Whether the sequence is relative to the sequence upstream of other genes or relative to the whole genomic sequence can also be modified. The first option is far more common.

After the analysis have completed, or you stop the search, the results are reported. They appear on right side of the toolbox. Potential regulatory sequencies, the number of genes they were detected in, and the detection p-value are reported. The best findings are reported first.

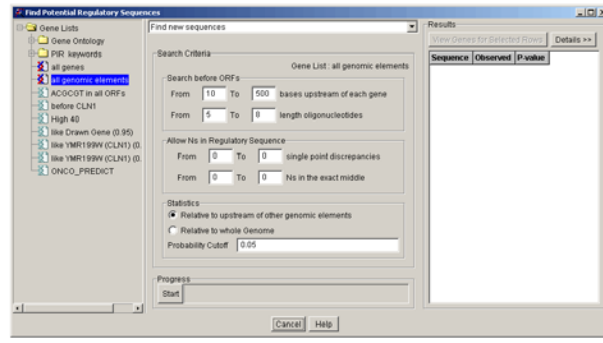

Figure 12.7: *Find potential regulatory elements -tool in GeneSpring.*

12.9 Suggested reading

1. Holmes, I., Bruno, W. J. (2000) Finding regulatory elements using joint likelihoods for sequence and expression profile data. Proc. Int. Conf. Intell. Syst. Mol. Biol. 2000 8, 202-10.

2. Roth, F. P., Hughes, J. D., Estep, P. W., Church, G. M. (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat. Biotechnol. 16,939-45.

3. Vilo, J., Brazma, A., Jonassen, I., Robinson, A., Ukkonen, E. (2000) Mining for Putative Regulatory Elements in the Yeast Genome Using Gene Expression Data. Proc. Int. Conf. Intell. Syst. Mol. Biol. 2000 8,384-94.

4. Vilo, J., Kivinen, K. (2001) Regulatory sequence analysis: application to interpretation of gene expression Eur. Neuropsychopharmacol. 11, 399-411.

This chapter was written by Martti Tolvanen, Mauno Vihinen and Jarno Tuimala (GeneSpring examples).

13 Annotations and article mining

Annotation is a "comment" attached to a gene. The comments can, for example, describe the gene's biological function, it's interactions between other genes, and the metabolic pathways the gene is acting in. After the interesting gene has been identified using statistical tools, it's annotation should also be acquired, in order to make inferences about the validity of the findings, and to generate new hypotheses.

Stanford Knowledge Systems lab defines *ontology* as "a formal and declarative representation which includes the vocabulary (or names) for referring to the terms in that subject area and the logical statements that describe what the terms are, how they are related to each other, and how they can or cannot be related to each other. Ontologies therefore provide a vocabulary for representing and communicating knowledge about some topic and a set of relationships that hold among the terms in that vocabulary".

Gene Ontology Consortium (`http://www.geneontology.org/`) aims to produce "controlled vocabularies for the description of the molecular function, biological process and cellular component of gene products". The Consortium has produced a hierarchical structure of Gene Ontologies (GO), which is independent of any organism. Genes of different organisms may be annotated using GO descriptions: Three GO-numbers, which correspond to a gene function, process or localization, are assigned to every gene.

13.1 Retrieving annotations from public databases

Sequences are usually annotated when they are submitted to databanks, *e.g.*, Genbank or EMBL. Sequences in Genbank and EMBL are computationally annotated and might contain errors. Some sources of annotations, like SWISS-PROT and RefSeq, contain curated annotations for the sequences, and the information is probably more reliable than in Genbank. Annotations for individual genes can easily be retrieved from the databanks using the sequence accession numbers.

Also Locuslink and Unigene, which are available at NCBI (`http://www.ncbi.nlm.nih.gov/`), contain valuable information about gene functions. Locuslink presents information on official nomenclature, aliases, sequence accessions, phenotypes, EC numbers, MIM numbers, UniGene clusters, homology, map locations, and related web sites. This information can be retrieved using common gene names or Locuslink accession numbers. In addition, GeneCards (`http:`

//bioinfo.weizmann.ac.il/cards/) contain information about gene functions and their associations with certain diseases. GeneCards can be queried using gene name or accession number.

Genome databases, such as Ensembl, also include detailed and curated information about the gene functions. Ensembl also has very good links to remote databases. The Ensembl data can be queried using any sequence database accession number. The Ensembl server can also be used for converting from one numbering system to another. Uses of the Ensebml database are described in more detail in Chapter 12.

The databases can be queried one gene at a time using public www-interfaces. With a little programming, multiple genes can be retrieved from the databases using the public www-interfases. It is also possible to install the databases locally, and make searches using the local query tools.

13.2 Retrieving annotations using BLAST

If sequences, or at least Genbank accession numbers for the genes of interest are available, their annotations can be updated using BLAST. After performing a (standard) BLAST search, you can pick three to five best scoring hits, and check what annotations they have. Likely, your query sequence has functions similar to the best hits you find from the database.

13.3 Article mining

Articles and abstracts contain masses of valuable information about the functions of the genes, but the data as such is difficult to approach. Of course, individual searches with gene names can be made from PubMed (http://www.ncbi.nlm.nih.gov/Pubmed/), but this is not feasible, if the number of genes or published articles is high. Article mining makes the retrieval of gene functions easier. Using a suitable query term, *e.g.*, a gene name, a list of functions or common keywords identified from the articles and abstracts can be produced.

MedMiner http://discover.nci.nih.gov/textmining/main.jsp is an article mining tool for biomedical information. Currently, MedMiner combines the information of Whitehead Institute's GeneCards and Pubmed. Searches about gene, gene-gene or gene-drug functions and interactions can be made. MedMiner does not necessarily contain information about all human genes, because the genes in the current version of the program have been selected from a certain DNA microarray.

National Cancer Institute also offers many other text mining tools, like GOMiner (http://discover.nci.nih.gov/gominer/), which helps to interpret the human -omic-data (including DNA microarrays) by classifying the genes into biologically coherent categories based on GO ontologies. Classifications can then be assessed using the tool.

Note that article mining is not a substitute for getting familiar with the relevant articles, but it enables one to easily infer probable gene functions from the established literature.

13.4 Annotation and gene ontologies using GeneSpring

13.4.1 Annotations

GeneSpring contains a built-in annotation tool (Figure 13.1), which can be used if Genbank accession numbers for the genes on the chip are known. If you do not have Genbank accession numbers, don't panic, because RefSeq and SWISS-PROT (and many other) accession numbers can be converted to Genbank IDs using the Ensembl database or other servers available on the Internet.

Moreover, when importing the data to GeneSpring, you have to specify which column in your datafile contains the Genbank accession numbers. This column is marked in GeneSpring as GenBankID.

The annotation tool GeneSpider can be accessed from the menu *Annotation->GeneSpider*. GeneSpider has four different options: you can annotate your chip using either Genbank, Unigene or Locuslink database or Silicon Genetics' mirror. It is recommended to use the Silicon Genetics' mirror, because it contains all the combined information of three other databases. It is also faster to update annotations using the mirror server.

GeneSpider contains several settings. First of all, you need to select the column containing Genbank IDs. Additionally you need to select the sources for the annotations. Annotations from all the selected sources can be combined. Alternatively, only the highest priority annotations are used. Of course, the priority of the databases can be mofified. Usually, the existing annotations are overwritten, but you can also decide not to do so. In addition to annotations, it is possible to retrieve sequences, but it can be very time-consuming, especially if you are analyzing human data.

It is also possible to request GeneSpring genomes for some commercial chips, *e.g.*, Affymetrix, directly from Silicon Genetics. This often cuts down the hazzle with annotations and Genbank IDs tremendously.

13.4.2 Ontologies

After retrieving annotations for your genes, you can build a simplified ontology. The simplified ontology creates a number of new gene lists, which fall into three categories; molecular function, biological process and cellular component of gene products, as specified in the GO ontology. The tool is located in *Annotations->Build simplified ontology*. The simplified ontology tool provided by GeneSpring is, as the name indicates, a simplified version of the actual GO ontology, and it is not updated as often as the official GO ontology is, so you should consider the formed genelist groupings suggestive.

Build simplified ontology tool is handy if you are interested in a certain gene type. For example, if you are interested in transcription factors, you can use the automatically created gene list (transcription factors) as a basis for further analyses. You can create a similar gene list using text filtering tools, but it is much more convenient to let GeneSpring do the work for you.

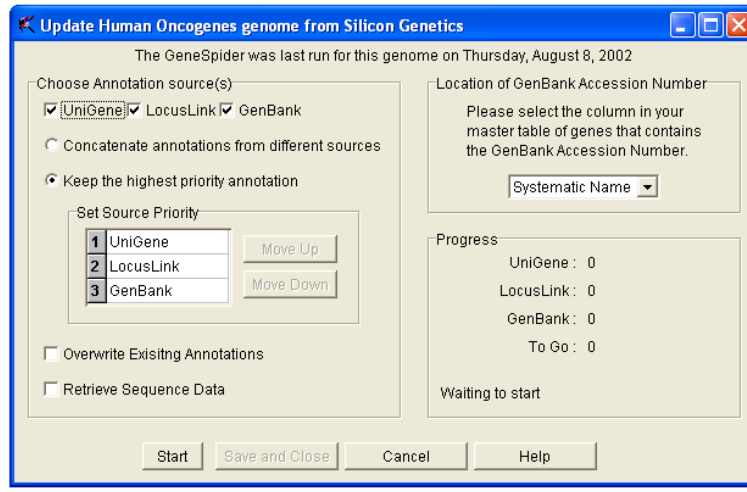

Figure 13.1: *GeneSpring annotation tool GeneSpider. Using the current settings, data from GenBank, Unigene and Locuslink will be searched, but only the Unigene annotations will be saved. Moreover, old annotations will not be overwritten, and no sequences will be retrieved.*

This chapter was written by Jarno Tuimala.

Part IV

Tools and data management

14 Reporting results

14.1 Why the results should be reported

Although microarrays work best as a screening tool, sometimes the final results that you wish to publish lean largely on the microarray data.

There are several good reasons why you should make your microarray data publicly available. It will be beneficial for you and others. Analyzing several good-quality, well-documented experiments together will ease the task of finding actual correlations between the gene expression. Since the microarray experiments are expensive, adding publicly available information from other microarrays into your own findings will save money.

You should think of publishing your microarray results already from the very beginning of the experimental work. Later on, it may be very difficult to "mine your notebook" for all the details on how you performed your experiments and how you handled your data.

Microarray data without detailed information about the experimental conditions and analysis steps is useless, and it should always be accompanied with that supportive information. To be able to compare your data with other researchers' experiments, all this information has to be reported in a common way, using a widely accepted form. Such a form is called Minimum Information About a Microarray Experiment, MIAME.

Furthermore, two major scientific journals, Nature and Cell, as well as EMBO Journal, have already set a condition that the microarray data in the submitted paper have to be MIAME-compliant and publicly available, before they accept the paper for publication (see instructions for authors at `http://www.cell.com` and `http://www.nature.com`, and Nature opinion article "Microarray standards at last", Nature 419, 323 (26 Sep 2002)). In a similar way, The Lancet has adopted MIAME quidelines (`http://www.mged.org/Workgroups/MIAME/miame_checklist.html`).

Probably this demand will become a prerequisite also for other journals, in a same way as publishing your sequence data in public databases, GenBank and EMBL.

14.2 What details should be reported: the MIAME standard

The MIAME standard outlines the minimum information that should be reported about a microarray experiment to enable its unambiguous interpretation and reproduction. The latest version of the MIAME standard can be found from the MGED

(Microarray Gene Expression Data) group's web page, see http://www.mged.org/miame. Simply, "MIAME is what one scientist should tell another scientist about his/her microarray experiment so that the other scientist could reproduce it and verify the results".

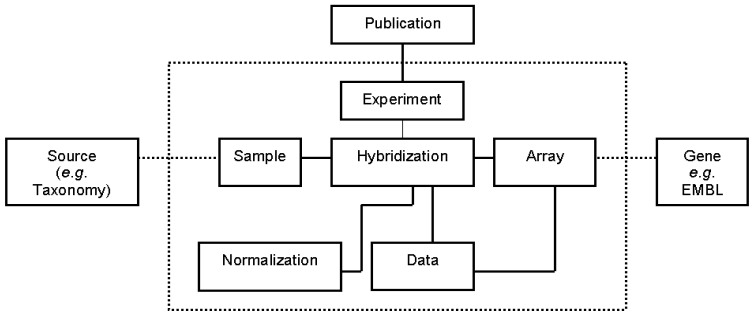

Figure 14.1: *The MIAME organization. A schematic representation of the six components of a microarray experiment (A. Brazma et al. 2001)*

The MIAME includes a description of the following six sections (Figure 14.1):

- Experimental design: the set of hybridization experiments as a whole, which may contain several hybridizations
- Array design: each array used and each element (feature, reporter, compositegroup) on the array and protocols used
- Samples: samples used, extract preparation and labeling, protocols used
- Hybridization: procedures and parameters
- Measurements: image quantification and specifications
- Normalization controls: types, values and specifications
- See link http://www.mged.org/Workgroups/MIAME/miame_glossary.html.

Microarray Gene Expression Data (MGED) Society is an international group of people from different disciplines who has a common task to standardize the methods for presenting and sharing microarray data. In addition to MIAME build-up, MGED Society has several working groups developing microarray standards, such as The Ontology Working Group (http://mged.sourceforge.net/ontologies/index.php). The ontology is a defined vocabulary to describe and name different terms (events, data, methods, material) related to microarray research so that the microarray data

can be annotated based on this ontology. MGED ontology is used as a basis for building the MAGE object model and markup language. MGED ontology is an integral part of the MIAME standard, and they are being developed together.

14.3 How the data should be presented: the MAGE standard

MAGE (MicroArray and Gene Expression) is a standard for the representation of microarray expression data and it is able to capture information specified by MIAME. MAGE has been developed by the European Bioinformatics Institute (EBI) and Rosetta Biosoftware and recently became an adopted specification of the OMG (Object Management Group) standards group (`http://www.omg.com`), but it's still an ongoing project. MAGE consists of three parts: An object model (MAGE-OM), a document exchange format (MAGE-ML), which is derived directly from the object model, and software toolkits (MAGE-STK), which seek to enable users to create MAGE-ML.

14.3.1 MAGE-OM

MAGE-OM (microarray gene expression object model) is an Object Model for the representation of microarray expression data that facilitates the exchange of microarray information between different data systems and organizations.

MAGE-OM is a data-centric model that contains 132 classes grouped into 17 packages containing, in total, 123 attributes and 223 associations between classes. Entities defined by the MIAME standard are organized as MAGE-OM components into packages, such as Experiment, BioMaterial, ArrayDesign, BioSequence, Array, and BioAssay packages, and the gene-expression data into a BioAssayData package. The packages are used to organize classes that share a common purpose, and the attributes and associations define further the classes and their relations.

A database created using MAGE-OM is able to store experimental data from different types of DNA technologies such as cDNA, oligonucleotides or Affymetrix. It is also capable of storing experiment working processes, protocols, array designs and analyzing results. MAGE-OM has been translated into an XML-based data format, MAGE-ML, to facilitate the exchange of data.

14.3.2 MAGE-ML; an XML-translation of MAGE-OM

MicroArray Gene Expression Mark-up language – MAGE-ML – is an XML-based file format able to capture all MIAME required information, and its derived directly from MAGE-OM. For example, the manufacturer of a chip may deliver to a user a MAGE-ML file containing information about the chip design and production process. Then the user can import the file directly to a local database. Because of the complexity of the model it's not quite human readable (though it's XML), but a software is needed to support importing and exporting MAGE-ML. Implementation of MAGE-ML include Rosetta Biosoftware, which supports it, and MIAMExpress that produces files in that format. The EBI has developed a freely available software tool kit (MAGE-STK) that eases the integration of MAGE-ML into end users'

systems.

14.3.3 MAGE-STK

The MAGE Software Toolkit is a collection of Open Source packages that implement the MAGE Object Model in various programming languages. The suite currently supports three implementations: MAGE-Perl, MAGE-Java, and MAGE-C++. The idea is to be able to have an intermediate object layer that can then be used to export data to MAGE-ML, to store data in a persistent data store such as a relational database, or as input to software-analysis tools.

14.4 Where and how to submit your data

14.4.1 ArrayExpress and GEO

There are two main public repositories for microarray data, ArrayExpress (`http://www.ebi.ac.uk/arrayexpress`) at the EBI, and Gene Expression Omnibus (GEO) (`http://www.ncbi.nlm.nih.gov/geo/`) at NCBI. The former supports MIAME format. Both of them provide you with an accession number that can be used in the publication analogous to GenBank ID of a sequence. These are accepted databases to dispose your data by, *e.g.*, Nature and The Lancet. Since GEO currently isn't MIAME-compliant we will here concentrate more on ArrayExpress.

ArrayExpress supports all the requirements specified by the MIAME standard, as developed by the MGED group and aims to store well-annotated data. In addition to sending your data into ArrayExpress, you can search, browse and retrieve (in MAGE-ML format) other microarray data from it. Currently (spring 2003) ArrayExpress contains 19 publicly available experiments. The amount of submitted data is and will be naturally larger, but there's a lag period before new data have been curated and published.

In addition to ArrayExpress and GEO, there are several species- or site-specific public data repositories available, such as the Stanford Microarray Database (SMD, `http://genome-www5.stanford.edu/MicroArray/SMD/`). For the review and a www page listing of these public repositories, please see a paper by Gardiner-Garden and Littlejohn (2001), and visit at `http://ihome.cuhk.edu.hk/~b400559/arraysoft_public.html`. However, the way data are presented in many of these special gene expression databases is quite diverse, or sometimes only the analyzed data have been made available, which makes it difficult to retrieve and evaluate the data, and impossible to co-analyze them together with your experiments.

Besides direct submission using MAGE-ML (to ArrayExpress) or SOFT (to GEO), microarray data can be submitted to the public databases via web interfaces. Next we will descibe some basic aspects of such submission tools.

14.4.2 MIAMExpress

MIAMExpress (`http://www.ebi.ac.uk/miamexpress/`) is a MIAME-compliant microarray data submission tool. It has a web interface that allows you in a step-by-step manner to submit your microarray data to ArrayExpress database. First you

will create an account, log in to that account, and then select a submission type: Protocol, Array design or Experiment. Protocol and Array design can be submitted individually, but the Experiment always need to be accompanied by details about Protocol and Array Design to complete the submission.

In addition to MIAME attributes, you can add other qualifiers. If the microarray data is being submitted to a journal, the journal name and status (submitted/in press/accepted) should be indicated. After filling the information about a certain protocol and array design, you next link that information to your experiment. The actual experiment results are submitted as scanned files (raw data), that can be either CEL files for Affymetrix data, or .gpr files, which contain two wavelengths in a same file, for others. If you wish, you can also add a data file corresponding to the transformation of all the results generated from the raw datafiles in a given experiment. In that case, the submitted protocol should describe all steps necessary to allow a third party to recreate the transformed data file.

14.4.3 GEO

GEO web interface works in a similar manner, although the vocabulary differs; first you log in, identify yourself, and then define your platform (array type and specifications, annotated gene list as tab-delimited text file, organism used, etc.) using pop-up menues. Next you bring in sample data (scanned file) in either of two standard sample data table formats, one for one- and another for two-channel data. These tables contain columns with standard headings, which can be either required or optional, and non-standard headings, which are user-defined and always optional. User-defined columns may contain useful information for the submitter and other users, but queries on these columns are not supported within GEO. Samples can currently be of four types: single channel, dual channel, comparative genomic, or SAGE. Samples (yours and others in GEO database) can be grouped in series such as dose-response or time course series, or as repeat samples group.

14.4.4 Other options and aspects

There are several other MIAME-compliant data management systems or software. To see a current list, please visit `http://www.mged.org/Workgroups/MIAME/miame_software.html`. One of these is an open-source application called BASE (BioArray Software Environment, `http://base.thep.lu.se/`). BASE consists of array production LIMS (see below), a relational database, and a web interface for management and analysis of microarray data. Additional analysis tools can be added as plug-ins. Although promising, BASE is still in a fairly early developmental stage.

In the future, CSC may provide a centralized place and tools to submit your microarray data to, *e.g.*, ArrayExpress.

Since microarray data is connected to much more information than just that required by MIAME standards, it may be reasonable to set up a LIMS (Laboratory Information Management System) for your laboratory for a storage place to all the information related to the microarrays, especially if you use custom-made arrays instead of commercial ones.

14.5 MIAME-compliant sample attributes in GeneSpring

GeneSpring has a sample manager, a small database in XML-format, where you can, when loading the data in, define numerous MIAME-compliant attributes. GeneSpring contains a reduced list of attributes by default, but a complete, up-to-date list can be downloaded from Silicon Genetics' web site. It is advisable to use predefined attributes to describe your experiments and samples. A sample information can be retrieved in XML-format from the genome's data-folder.

14.6 Suggested reading

1. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat. Genet. 29, 365-71.

2. Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Garcia, L. G., Oezcimen, A., Rocca-Serra, P., and Sansone, S-A. (2003) ArrayExpress-a public repository for microarray gene expression data at the EBI, Nucleic Acids Research 31, 68-71.

3. Edgar, R., Domrachev, M., and Lash, A. E. (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository, Nucleic Acids Research 30, 207-10.

4. Gardiner-Garden, M. and Littlejohn, T. G. (2001) A comparison of microarray databases, Briefings in Bioinformatics 2, 143-158.

5. Gollub, J., Ball, C.A., Binkley, G., Demeter, J., Finkelstein, D. B., Hebert, J. M., Hernandez-Boussard, T., Jin, H., Kaloper, M., Matese, J. C., Schroeder, M., Brown, P. O., Botstein, P., and Sherlock, G. (2003) The Stanford Microarray Database: data access and quality assessment tools, Nucleic Acids Research 31, 94-96. `http://genomebiology.com/2002/3/9/research/0046.1`

6. Saal, L. H., Troein, C., Vallon-Christersson, J., Gruvberger, S., Borg, Å., and Peterson, C. (2002) BioArray Software Environment: A Platform for Comprehensive Management and Analysis of Microarray Data, Genome Biology 3, software0003.1-0003.6.

7. Spellman, P. T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, W. L., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, D., Senger, M., Aronow, B. J., Robinson, A., Bassett,

D., Stoeckert, C. J. Jr, and Brazma, A. (2002) Design and implementation of microarray gene expression markup language (MAGE-ML), Genome Biology 3, research0046.1-0046.9.

8. Stoeckert, C. J., Causton, H. C., and Ball, C. A. (2002) Microarray databases: standards and ontologies, Nature Genetics 32, 469-473.

This chapter was written by M. Minna Laine, and Teemu Toivanen.

15 Software issues

Traditionally, biological data, including DNA and amino acid sequencies and DNA microarray results, have been stored in a flat file format. Flat files are basically just text files, which have been formatted in such a way that programs can read their contents. Flat files are not the ideal solution for long term storage of large data sets. They also cause problems when the same data needs to be analyzed using several programs that use a different kind of file format.

The data standardization and storage in central databases (see chapter 14 for more details) is one solution to the data format issues. However, this solution has not yet fully matured, and most of the programs used for the DNA microarray data analyses are not able to read the data directly from these databases or even import the XML-format the databases commonly produce. Databases and programs will surely learn to cooperate better over time. The standardization and storage of the data is currently under heavy development, and many software developers are implementing their own solution to the these problems. However, the solutions are not always MIAME compliant.

15.1 Data format conversions problems

Especially, if freeware tools are used for the analysis, there is often a need to export the partly analyzed data from one program to another. More often than not, the datafile formats are not compatible, which means additional and tedious conversion work. Data can quite easily be converted between formats using Microsoft Excel or any other spreadsheet program. However, the work load becomes quickly unbearable when the amount of data increases. Excel has some macro programming capacities, but there exists also other, and possibly better tools, if one is interested in programming (see section on programming). In addition, Excel does not tolerate huge data files (more than 65 536 rows or 256 columns). Sometimes it becomes necessary to transpose (switch the rows and columns) the data file, and if there are more than 256 genes, Excel can't be used.

15.2 A standard file format

If the data needs to be imported into several programs at the same time, it is often easier if a "standard file" is first generated from the files written by the chip scanner. Standard files can be produced in Excel or another spreadsheet program easily. Briefly, the first column of the standard file should contain the gene identifiers, for

example their common names. The second column should consist of the sequence accession numbers, Genbank accession numbers are probably the most usable ones. The next four columns should contain the spot and background intensities of the used colors. Depending on your analysis needs, it is also possible to use ratios, for example normalized log ratios, instead of intensity values. The standard file should be saved in a tab-delimited text format.

15.3 Programming

When the capacities of the spreadsheet program are not sufficient, some programming tools can be used instead for the datafile management purposes or for data analyses. There are actually several programming languages that are especially suited for the data file formatting and other text file manipulations.

15.3.1 Perl

One of the most common languages is Perl, which has very powerful and easy to use text manipulation tools. Perl is available for free for UNIX, Linux `http://www.perl.org` and PC machines `http://www.activeperl.com`. Perl is a programming language, which means that getting to know it takes some time and effort. However, if data conversion tools are needed everyday, it would definitely be worthwhile to befriend Perl.

To illustrate how easily text can be manipulated using Perl, we present a short example. The next code produces a complementary DNA sequence from the original sequence, which has been stored into the text string $sequence. Both the original sequence and its complement are printed on the computer screen.

The program starts with a line that tells the computer where The Perl software can be found. The next four lines contain the actual commands that manipulate the DNA sequences. Note that every command line has to end with a semicolon. Function `tr` makes a complementary sequence from the original one, and function `print` outputs the result to the screen.

```
#!/usr/bin/perl
$sequence="aaattcgagtaggtcaggcat";
print "Original:            $sequence\n";
$sequence=~ tr/acgtACGT/tgcaTGCA/;
print "Complementary:       $sequence\n";
```

You can use pico editor on CSC's Cedar server to create a similar file and test the example yourself. Perl programs are started in Cedar with a command `perl filename`.

15.3.2 Awk

Awk is a standard UNIX and Linux tool, which is available on CSC's servers. With Awk, individual columns can be easily extracted from tab-delimited text files. Using other standard UNIX tools, these individual columns can be saved into a new

text file. For example, the next script takes the first column from a specified datafile and saves it into a new file.

```
Awk '{print$1}' datafile > newfile
```

Two columns can be "awked" into a new file next to each other separated with a space:

```
awk '{print\$1, \$2}' datafile > newfile
```

or one after another:

```
awk '{print\$1}{print\$2}' datafile > newfile
```

15.3.3 R

R is a free statistical analysis tool and a self-sufficient programming language. It is available for UNIX, Linux, Macintosh and PC platforms. In R, scripts for analyses and data file manipulations can be easily constructed. R has many add-on packages for cluster analysis, self-organizing maps, and neural networks, among others. There are also some packages available that have been specifically tailored for DNA microarray data analysis.

Here is an example on how to read the tab-delimited datafile to R and how to process it into a new table, which is then written out to a new file. The function `read.table` reads in the specified file with the headers. The table is then saved in a variable data. Two columns are extracted from the data, and saved into new variable (x and y). The new variables are used for the creation of a new table (dataout), which is then written to a new text file (filenameout.txt). Such a script can easily be automated using R, and the analyses can simultaneously be intergrated with the datafile conversions.

```
data<-read.table("filename.txt", header=T)
x<-data$greenintensity
y<-data$redintensity
dataout<-cbind(x,y)
sink("filenameout.txt")
dataout
sink()
```

Many images included in chapters 5–8 have been produced by R using real DNA microarray datasets.

15.4 Freeware software packages

There are several software packages that are either totally free or free for academic researches. We have experience with quite a few software packages, a couple of which we introduce here.

15.4.1 Cluster and treeview

Cluster and Treeview are among the first free software tools for DNA microarray data analysis. They are still heavily used for clustering, *i.e.* production of expression heatmaps. These are small and easy to use tools for beginners.

More information: `http://rana.lbl.gov/EisenSoftware.htm`

15.4.2 Expression profiler

A web-based clustering tool Expression Profiler has been developed at the EBI. Expression profiler performs clustering by various algorithms and distance or dissimilarity methods. It also has nice links to sequence, metabolome, and pathway databases. At the moment the system works best with yeast data, because the links to the outside databases are more extensively implemented for yeast.

More information: `http://www.ebi.ac.uk/microarray/ExpressionProfiler/ep.html`

15.4.3 ArrayViewer

The Institute for Genomic Research (TIGR) has released a couple of software packages for DNA microarray data analysis. ArrayViewer comes in two versions. One is suited for viewing the results from one chip only, the other can cope with multiple slides at the same time. Analysis options cover some basic filtering and normalization methods and various clustering algorithms.

More information: `http://www.tigr.org/software/`

15.4.4 MAExplorer

MAExplorer can be used as a web-based tool or a desktop program. It performs clustering, but also normalization, and has the best statistical tools among the free programs. The program contains a possibility to use a tailored database. MAExplorer is also under constant development, and new features are frequently added.

More information: `http://maexplorer.sourceforge.net/`

15.4.5 Bioconductor

Bioconductor is an attempt to build a DNA microarray data analysis environment on top of R (see section 14.3.3). The broad goals of the projects are to provide access to a wide range of powerful statistical and graphical methods for the analysis of genomic data; facilitate the integration of biological metadata in the analysis of experimental data: *e.g.*, literature data from PubMed, annotation data from LocusLink; allow the rapid development of extensible, scalable, and interoperable software; and promote high-quality documentation and reproducible research.

More information: `http://www.bioconductor.org` and `www.r-project.org`

15.5 Commercial software packages

Many commercial software packages have been developed for DNA microarray analysis during the last couple of years. The ones we have experienced with are briefly presented here.

15.5.1 VisualGene

VisualGene is a powerful data exploration tool. Algorithms are based on self-organizing maps, Summon's mapping, and Fuzzy modeling. In addition, some filtering tools are implemented to pretreat the data before cluster analyses.

More information: `http://www.visipoint.fi/`

15.5.2 GeneSpring

GeneSpring is a general analysis tool specifically tailored for DNA microarray data analysis. The intented group of users are the biologists, who actually perform the experiments. The program contains various data preprocessing (filtering, normalization) and clustering tools. Gene annotation can also be retrieved directly from web-based databases. GeneSpring can be expanded with user-made scripts or programs (Java APIs).

More information: `http://www.sigenetics.com/`

15.5.3 Kensington

Kensington Discovery Edition offers a heavy data mining tool for DNA microarray analyses. Kensington has a visual programming language that describes the dataflow through different analysis steps. Kensington has more or less the same analysis tools as GeneSpring, but it implements the database connections in a more sensible fashion, for example, data can be retrieved from any field of the Genbank report. Kensington can be expanded with user-made modules containing the new algorithms.

More information: `http://www.inforsense.com/`

15.5.4 J-Express

J-Express is another appealing commercial tool. It has some filtering and other preprocessing tools, but the main emphasis is on various clustering algorithms and the visual examination of normalization results. It has nice links to web-based databases. J-Express is also user customizable, because plug-ins, which allow custom algorithm implementation, can be constructed with Java-programming language. There is also a free version of the program available.

More information: `http://www.ii.uib.no/~bjarted/jexpress/`

15.5.5 Expression Nti

Expression Nti is a newcomer in the field of DNA microarray data analysis tools. It has welded together some analysis tools and a tailored database. The analysis tools contain filtering and normalization, and various clustering algorithms. The database system can read directly from any user-made GEML-based database.

More information: `http://www.informaxinc.com/solutions/xpression/`

15.5.6 Rosetta Resolver

Rosetta Resolver is a huge analysis tool, which combines analysis tools and a database under the same program. The novelties of Rosetta Resolver are the built-in error models, which are made for each DNA microarray format separately. Error models for home-made microarrays can also be constructed. Because of error models, Rosetta Resolver also has advanced tools for the detection of outliers.

More information: `http://www.rosettabio.com/products/resolver/default.htm`

15.5.7 Spotfire

Spotfire is a data mining tool that has been modified to suit DNA microarray analyses. Spotfire runs in a web-browser, and different analysis tools are loaded on the user's machine on as-needed basis. Spotfire has very nice tools for visualization of the microarray data. Because it is based on general statistical tools, it also has some well-implemented functions for outlier detection and the assessment of the statistical significance of the results.

More information: `http://www.spotfire.com/`

This chapter was written by Jarno Tuimala.

Index