Roland Pieler

Java Tool for Normalization and Analysis of Microarray Data

Master Thesis

Institute of Biomedical Engineering
University of Technology, Graz, Austria
Inffeldgasse 18, A - 8010 Graz
Head of Institute: Univ.-Prof.Dipl.-Ing.Dr.techn. Gert Pfurtscheller

Supervisors:
Dipl.-Ing. Gerhard Thallinger, Univ.-Prof. Dipl.-Ing. Dr.techn. Zlatko Trajanoski

Evaluator:
Univ.-Prof. Dipl.-Ing. Dr.techn. Zlatko Trajanoski

Graz, September 2003

For my parents

and Barbara

Abstract

Microarray technology has become an essential tool in functional genomics. However, there are many sources of variation which affect the measured gene expression levels. Normalization refers to the process of removing systematic variation. For this purpose, a platform independent Java application for normalization and analysis of microarray experiments has been developed. The experiment data are graphically organized according to the design, scatterplots, histograms and boxplots allow the visualization of the data. Several normalization methods have been implemented: 1) global method, 2) LOWESS-regression, 3) self-normalization for dyeswapped slides and 4) normalization with controls. After normalization, replicated measurements can be combined and averaged to enable statistical analysis. For the detection of genes with significant changes in expression, a module is provided including: 1) fold-change detection and 2) t-test with adjusted p-values. The selected genes can be saved to a text file, readable by other microarray-analysis software. The variety of normalization methods and the ability of dealing with a wide range of experimental designs makes this software a useful and freely available tool to normalize microarray experiments.

Keywords: microarray, normalization, experimental design, statistical analysis, Java

Kurzfassung

Die Microarray Technologie ist ein vielversprechendes Verfahren in der Genomforschung. Die gemessenen Genexpressionswerte werden jedoch durch verschiedenste Fehlerquellen verfälscht. Normalisieren bedeutet Erkennen und Korrigieren von systematischen Fehlern. Für diesen Zweck wurde eine betriebssystemunabhängige Java-Anwendung entwickelt. Abhängig vom Design des Experimentes werden die Daten grafisch organisiert und können mit einer Reihe von Visualisierungsmethoden wie Scatterplots, Histogramme oder Boxplots angezeigt werden. Eine Reihe von Normalisierungsmethoden wie 1) globale Methoden, 2) lineare Regressionsmethode, 3) Selbst-Normalisierung für Dyeswap-Paare und 4) Normalisieren mittels Kontrolspots wurden implementiert. Nach dem Normalisieren können replizierte Messwerte kombiniert und gemittelt werden, um eine statistische Auswertung zu ermöglichen. Gene mit signifikanten Expressionswerten können mittels 1) Foldchange-Detektion oder 2) T-Test mit korrigierten p-Werten identifiziert werden. Diese Gene können in ein Textfile, welches mit weiterführender Analysesoftware kompatibel ist, exportiert werden. Durch die Auswahl verschiedener Normalisierungsmethoden und die Fähigkeit, unterschiedliche Experiment-Designs zu berücksichtigen, wird ein frei verfügbares und benutzerfreundliches Programm für das Normalisieren und Auswerten von Microarray-Experimenten angeboten.

Schlüsselwörter: Mikroarray, Normalisieren, experimentelles Design, statistische Analyse, Java

Contents

List of Figures

Glossary

Bias The word bias refers to all sources of systematic variations: bugs in software, miscalibrated measurements, etc. Biased measurements are systematically wrong.

Biological replicates biological samples from independent sources, representing the same condition, e.g. liver tissue from individual mice of the same sex and strain.

Bonferroni correction Multiple-testing adjustment in which the significance-level is divided by the total number of tests.

cDNA (complementary DNA) DNA synthesized from mRNA or DNA by reverse transcriptase often synthesized from a cellular extract.

Channel A channel is an intensity-based portion of an expression dataset. In some cases, such as Cy3/Cy5 array hybridizations, multiple channels (one for each label used) may be combined to create ratios.

Chromosomes Part of a cell that contains genetic information. A chromosome is a grouping of coiled strands of DNA, containing many genes. Most multicellular organisms have several chromosomes, which together comprise the genome. Sexually reproducing organisms have two copies of each chromosome, one from the each parent.

Class In experimental design, a *class* denotes a subset of the whole experiment. For example one single time-point out of a time-course experiment represents one class, containing all

microarrays belonging to this time-point. An experiment can consist of any number of classes.

Control The reference for comparison when determining the effect of some procedure or treatment.

Cy3, Cy5 Cyanine fluorescent dyes used in microarray experiments for labelling different samples of DNA. Cy3 can be visualized as green, Cy5 as red.

DE Short form for *differentially expressed.*

Distribution A distribution is a graphic representation of the values of a variable. The line formed by connecting data points is called a frequency distribution. An important aspect of the ”description” of a variable is the shape of its distribution. Typically, one is interested in how well the distribution can be approximated by the normal distribution.

DNA (DeoxyriboNucleic Acid) The molecule that encodes genetic information. DNA is a double-stranded polymer of nucleotides. The two strands are held together by hydrogen bonds between base pairs of nucleotides. The four nucleotides in DNA contain the bases: adenine (A), guanine (G), cytosine (C), and thymine (T).

Dye-swap pair Two slides comparing the same samples of RNA, one with normal and one with reversed dye-assignment.

Error In statistics, *error* refers to all kinds of unspecific variability (variability introduced in the measurement). That is different from the everyday-use to mean *mistake*.

Estimation The process of using sample statistics to estimate population parameters.

Expression The conversion of the genetic instructions present in a DNA sequence into a unit of biological function in a living cell. Typically involves the process of transcription of a DNA sequence into an RNA sequence.

Fold change The ratio of RNA quanitites between two samples in a microarray experiment.

Gene DNA which codes for a particular protein or a functional or structural RNA molecule.

Genesis Cluster analysis software for large scale gene expression studies. Developed by Alexander Sturn, *TU-Graz Bioinformatics Group*.

GUI Graphical User Interface

Hybridization A hybridization is the act of treating a microarray with one or more labeled preparations from a specified set of conditions.

MARS Microarray Analysing and retrieval system. A J2EE application for persisting and organizing microarray data, based on Enterprise Java Beans (EJB) and Struts framework. Developed by the *TU-Graz Bioinformatics Group*.

Microarray A microarray (or slide) refers to the physical substrate to which biosequence reporters (cDNA or oligos) are attachted. Microarrays are hybridized with labeled samples and then scanned and analyzed to generate data.

Microarray experiment An experiment studies a system under controlled conditions while some conditions are changed. In gene expression, one varies some parameter such as time, drug, developmental stage, or dosage on a sample. The sample is processed and labeled with a detectable tag (Cy3, Cy5) so that it can be used in hybridization with microarrays.

mRNA (messenger RNA) A specialized form of RNA that serves as a template to direct protein biosynthesis. The amount of any particular type of mRNA in a cell reflects the extent to which a gene has been ”expressed”.

Normal distribution Also called Gaussian distribution, this is one of the most important statistical distributions, since experimental errors are often normally distributed. Further, the normal assumption simplifies many methods of data analysis.

Normalization The process of removing the effect of all sources of non-biological variation from microarray data, making them comparable.

Null hypothesis A hypothesis for which the effects of interest are assumed to be absent. Commonly used as basis for setting up statistical tests.

Oligo (Oligonucleotide) Short sequence of nucleotides ($<$ 80 bp) always single stranded to be used as probes or spots. Oligos are often chemically synthesized.

PCR (Polymerase Chain Reaction) Allows the exponential copying of part of a DNA molecule using a DNA polymerase enzyme.

PMT (Photomultiplier tube) Part of optical scanner for microarrays, which detects photons emitted by fluorescent dyes.

Protein A biological molecule which consists of many amino acids chained together by peptide bonds. Proteins perform most of the enzymatic and structural roles within living cells.

p-Value A measure of evidence against the null hypothesis in a statistical test.

Ratio Also referred to as ”fold change”. A ratio refers to a normalized signal intensity generated from one feature in a given channel divided by a normalized signal intensity generated by the same feature in another channel.

Replication A replicate set refers to repeated experiments where the same type of array is used, and the same probe isolation method is used to get more statistically meaningful interpretation of results. Reproducing an experiment helps to verify its results.

RNA (ribonucleic acid) A class of nucleic acids that consist of nucleotides containing the bases: adenine (A), guanine (G), cytosine (C), and uracil (U). An RNA molecule is typically single-stranded and can pair with DNA or with another RNA molecule.

RT-PCR (Reverse Transcription Polymerase Chain Reaction) The most sensitive technique for mRNA detection and quantitation currently available. It is sensitive enough to enable quantitation of RNA from a single cell.

Sample A subset of a population. Usually, the size of the sample is much less than the size of the population. The primary goal of statistics is to use information collected from a sample to try to characterize a certain population.

Significance level The p-value that is regarded as providing sufficient evidence against a null hypothesis. If the p-value falls below the significance-level, the null hypothesis is rejected.

Statistics A statistic is a number computed from a sample.

Statistical significance A result is statistically significant when it doesn't happen by chance.

Subgrid A subarea of a single microarray. Within one subgrid all spots are printed by the same print-tip.

Technical replicates Multiple hybridisations with RNA samples obtained from the same biological source.

TIFF (Tagged Image File Format) One of the most popular and flexible public domain raster file formats. It's main strengths are a highly flexible and platform-independent format which is supported by numerous image processing applications.

Variable Numerical data are observations which are recorded in the form of numbers. Numbers are variable in nature. E.g, when measuring gene expression levels, the score will vary for reasons such as temperature, cell activity etc. For this reason, the gene expression level is called *variable*.

z-score A statistical measure that quantifies the distance (measured in standard deviations) a data point is from the mean of a data set. The z-score associated with the ith observation of a random variable x is given by

$$z_i = \frac{x_i - \mu}{\sigma} \tag{1}$$

where μ is the mean and σ is the standard deviation of all observations $x_1, ..., x_n$.

Chapter 1

Introduction

With only a few exceptions, every cell on organism contains a full set of chromosomes and identical genes. Only a subset of these genes is active at a given time; these gene define the unique properties of a cell type. *Gene expression* describes the transcription of information contained within the DNA into messenger RNA (mRNA) molecules that are then translated into proteins, performing most of the critical functions of cells. Biologists study the kinds and amounts of mRNA produced by a cell to learn which genes are expressed. Gene expression is a complex regulated process that allows a cell to respond dynamically to environmental stimuli and to its own changing needs. It controls which genes are expressed in a cell and increases or decreases the level of expression of particular genes as necessary. The study of gene expression helps to understand fundamental aspects of growth and development and underlying genetic causes of many human diseases (see [6]). Microarrays help to monitor the expression of many genes in parallel. They consist of glass slides prepared by high density printing of complementary DNAs (see [32]).

1.1 Microarrays

A *microarray* employs the ability of a given mRNA molecule to bind specifically to, or hybridize to, the DNA template from which it is originated. It contains many DNA sequences, and the expression levels of thousands of genes can be determined in a single experiment by

measuring the amount of mRNA bound to each site on the array (see [11]). Microarrays are small glass slides or nylon membranes onto which the gene sequences are printed, spotted or synthesized. The sequences can be DNA, cDNA or oligonucleotides. Arranged systematically, the particular sequences can be identified by the location of the spots on the slide (see [36]).

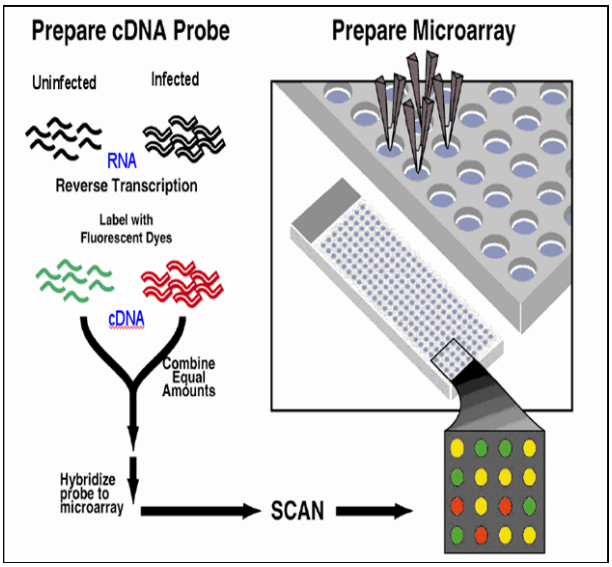

Figure 1.1: *DNA clones are mechanically printed to a glass slide. Then fluorescently labeled cDNA probes are hybridized to the microarray. Afterwards the slide is scanned using two different wavelenghts, resulting in two images including information about the fluorescence intensities (see [5] and [2]).*

The relative abundance of each of the gene-specific sequences in two RNA samples (test and reference) may be estimated by fluorescently labelling the *samples*, mixing them, and *hybridizing* the mixture to the sequences on the glass slide. The two samples of mRNA from cells (target) are reverse transcribed into cDNA, and labelled using two different *dyes* (red Cy5 and green Cy3 in general). Usually, the reference sample is labeled with Cy3 and the test sample with Cy5. The mixture reacts with the spotted cDNA sequences (probes). This, called competitive hybridization, results in cDNA sequences from the targets and the probes base-pairing with one another. After this hybridization step is complete, the microarray is placed in a scanner, consisting of lasers with different wavelengths, a microscope and a camera. The slide is scanned twice, first using one color laser, and then the second. Laser light excites the fluorescent dyes (Cy3 is excited by green laser light such as 532nm, Cy5 is excited by red laser light at

635nm). The dyes emit fluorescent radiation with characteristic spectra, which is measured by the microscope and the camera to create monochrome digital images of the array, one for each wavelength.

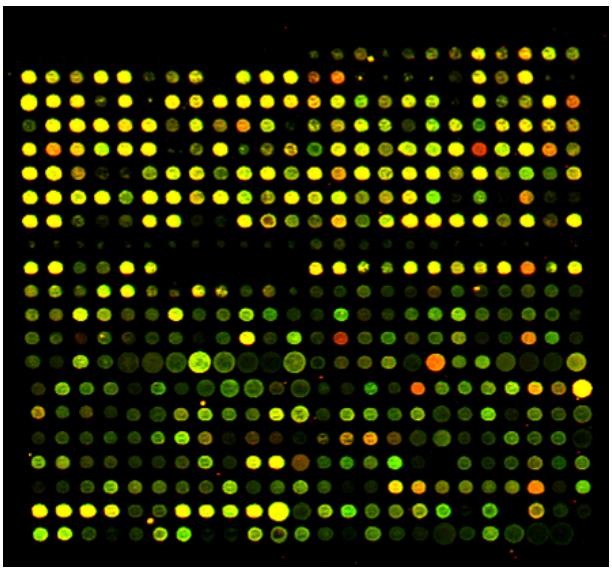

Figure 1.2: *Green spots indicate that the test substance has lower activity than the reference substance, red spots indicate that the test substance is more abundant than the reference substance, yellow spots mean that there is no change in the activity level between the two populations of test and reference substance. Black represents areas where neither test nor control substance has bound to the target DNA (taken from [10]).*

Usually, the raw wavelength images are collected at 16-bit resolution, giving fluorescence intensity measurements for each sample for each spot. The channel intensities for any spot should be proportional to the amount of mRNA from the corresponding gene in the respective sample. In practice the absolute channel intensities are usually less reproducible, and for the most part, the *ratios* (Cy5 / Cy3) are used for further analysis. Thus, the spotted arrays provide information only on the relative gene expression between specific cells or tissue samples only. For display purposes only, the two images are pseudo-colored and merged, to create a ratio image of the microarray.

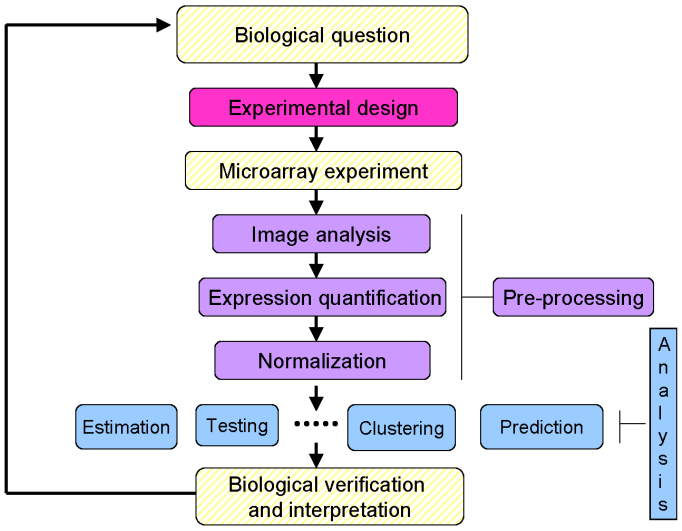

Figure 1.3: *The workflow of microarray experiments (taken from [4]).*

1.2 Design of microarray experiments

Microarray experiments are large-scale experiments and can be costly in terms of consumables and time. Careful design is important if the results should be maximally informative, given the effort and the resources. Which issues need to be adressed when planning, which features have impact on the resulting measurements? This section gives a short overview about important points for planning micro array experiments. (For further information see [48], [27], [33], [42], [12].)

1.2.1 Main requirements

Before planning an experiment, following issues need to be considered:

- Aim of the experiment. What are the questions to be answered? (search for *differentially expressed* (DE) genes,search for patterns between different samples, etc.)
- Types of mRNA samples. What mRNA will be used?
- The amount of available RNA extractable from the biological samples.

- The number of slides available.
- Experimental process before hybridization (i.e. RNA isolation, labelling)
- Type of controls: positive, negative, etc.
- Verification method: northern/western blot, RT-PCR, knock-out studies etc.

An experiment should be planned to fulfil requests like:

- It should be simple in design and analysis.
- Proper statistical analysis should be possible.
- The experiment's results should allow valid conclusions.
- Using as few experimental units as possible, random errors should be properly small.
- Different treatments of experimental units should not lead to systematical dependance.

1.3 Graphical design representation

In many papers graphical elements are used to illustrate microarray designs (see [48], [12], [33]). One way is to use directed graphs, containing *nodes* and *edges*. The nodes correspond to target mRNA samples, and the edges correspond to hybridizations between two samples. By convention, the green-labelled sample is placed at the tail and the red-labelled sample at the head of the arrow. The structure of the graph determines which samples can be compared (there must be a path between the two samples) and the precision which can be achieved (the shorter the path, the higher the precision will be). Some examples should show how microarray designs can look like (see [48], [42]).

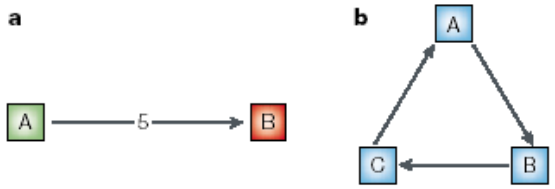

Figure 1.4: *Graphical representation: a) shows replicate hybridisations. Each slide involves sample A (labelled red) and sample B (labelled green). The number 5 indicates five replicated hybridisations. b) describes the simplest loop design. Three samples - A,B,C - are hybridized together in pairs, each sample labelled once in red and once in green (taken from [43]).*

1.3.1 Basic example

1.3.2 Direct versus indirect comparisons

The first issue in design is to decide whether to use *direct* or *indirect comparisons*: whether to make the comparison within or between slides(see [43]).

1.3.2.1 Example with two hybridizations

Direct comparison means that both samples of RNA, T and C, are hybridized together on both slides: T-C. *Indirect comparison* makes use of a common reference R. Two hybridizations will be needed, T-R and C-R. The key difference between direct and indirect design are the variances of the log-ratios. Direct design provides less variance compared to indirect design, in this case.

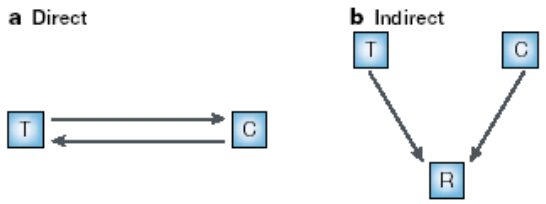

Figure 1.5: *Comparing samples T and C. a) describes direct comparison, the expression of the genes is measured on the same slide. b) using indirect comparison, T and C are measured separately on two slides. R is the reference sample (taken from [43]).*

1.3.3 Dye-swap experiments

Systematic bias[1] can be reduced by doing each hybridization twice with *reversed dye assignment* in the second hybridization. Microarray experiments show systematic differences (due to dye-effects, see section 1.5.1 and 1.6)in the red and green intensities, which require *normalization*. For this reason, dye-swap pairs are recommended wherever possible. Importantly, direct comparisons of replicated slides with the same labelling should be avoided because unadjusted color bias might accumulate.

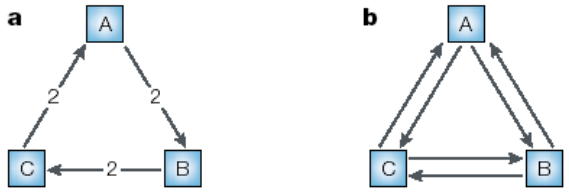

Figure 1.6: *Design choices: a) shows a loop design with each hybridisation done twice (indicated by the number 2). b) shows dye-swap replication, which involves two hybridisations for two samples (indicated by antiparallel arrows). One arrow describes normal dye-assignment, its reversed partner indicates reversed dye assignment (By convention, the green-labelled sample is placed at the tail, the red-labelled sample at the head of the arrow). Both experiments a) and b) consist of 6 hybridisations (taken from [43]).*

1.3.4 Time-course experiments

In *time-course experiments*, the design depends on the comparisons of interest. Additionally, practical constraints (e.g. restricted number of hybridizations, number of time points) determine the design choices. Some examples will illustrate the wide range of design possibilities (see [43]).

The design A) in figure 1.7 uses T1 (timepoint 1) as common reference. When the main focus of the experiment is on the relative changes between T2, T3, T4 and T1, this is the best choice. Design B) has one extra hybridisations between T2 and T3. Design A) in figure 1.8 shows direct hybridisations between neighboring time points. If variations frome one timepoint to

[1] see glossary

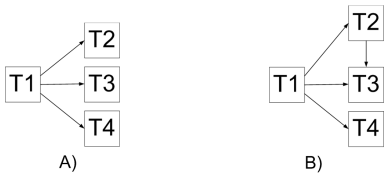

Figure 1.7: *Common reference: A) T1 is used as reference, B) has one extra hybridisation (see [43]).*

another are of greater interest, then this design is preferable. Design B) describes loop-design. At present, reference designs are mostly used, giving the advantage of easy interpretation and analysis.

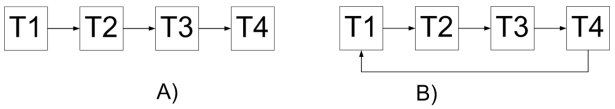

Figure 1.8: *A) Direct hybridisations and B) Loop design (see [43]).*

1.4 Random Errors and Replication

Unreplicated microarray experiments seem to be still the most common although the do not provide any statistical significance. One reason may be that researchers do not want to waste hybridizations for replication, when they could do a different one. *Replicates* reduce variability in summary statistics. In addition, the data obtained from replicate slides can be analyzed using statistical methods and tests. It is difficult to say how many replicates are essential (see [30], [44]), however it is proposed that three replicates are sufficient. Replication allows averaging, and averages are less variable than their component terms (see [23], [20]).

1.4.1 Replicated spots on one slide

This common form of replication is valuable for monitoring the overall slide quality. It is advisable to have them well spaced and not sticked together, as this gives a better reflection of the variability across the slide. But they should not be used as replicates in terms of statistical analysis (e.g. t-tests). All processing steps of the slide (printing, hybridisation, scanning) will

be shared by spot replicates, therefore any systematic effects on the measurement will also be shared.

1.4.2 Replicate slides

Assessing the significance of log-ratios using data from just one single slide fails to take into account an important source of variation - between-slide variability. Replication is essential to estimate the variance of intensity-values between slides and allows the application of statistical methods (e.g. t-tests, nonparametric tests). There are two forms: *Technical* and *biological* replicates (see [48]).

- **Technical replicates** Technical slide replicates are multiple hybridizations with RNA from the same pool (from the same extraction). These replicates generally involve a smaller degree of variation in measurements than biological replicates. Therefore they do not provide the independence of data, and shared systematic effects of the replicates will remain after averaging.

- **Biological replicates** This term refers to hybridisations using RNA extracted from different biological sources (mice, cell cultures, etc.). Carrying out sample labelling separately for RNA from different extractions will lead to more independent experiment results. It is recommended to use biological replicates.

The type of replication can affect the precision of the experimental results. Optimal result will be achieved by using biological replicates to provide independency in data and technical replicates to assist in reducing the variability.

1.5 Sources of Variation in Microarray Experiments

In order to accurately measure gene expression changes, it is important to take into account the *random* and *systematic variations* that occur in every microarray experiment (see [20]).

1.5.1 Systematic errors

From the sources of systematic variation the most important *bias* is associated with the different fluorescent dyes. Even such biases may be small, they may be confounding when searching for subtle biological differences. Dye biases can stem from a variety of factors, including physical properties (heat, light sensitivity, half life), efficiency of dye incorporation, or scanner settings. Other artifacts result from the robotic printing process, hybridization including nonuniform background signal intensities and any other spatial effects that are introduced during the production and use of the microarrays. All these factors make distinctions between differentially and constantly expressed genes difficult. The purpose of normalization is to get rid of systematic errors.

1.5.2 Random errors

Random errors occur when a sample of a variable population is taken, and by chance the sample does not perfectly represent the real population. This will always happen to some degree, because populations are naturally variable. Random errors have larger effects, if the sample size is small. *Replication* plays an important role in dealing with random errors. Replication allows averaging, which reduces variability.

1.6 Normalization

The goal of the normalization step is to identify and remove any systematic bias in the measured fluorescence intensities, arising from variation in the micro array process rather than from biological differences between the RNA samples or the printed probes. This can be:

- different labelling efficiencies of the dyes
- different amounts of Cy3- and Cy5 labeled mRNA
- different scanning parameters
- spatial or plate effects, print tip effects, etc.

The need for normalization can be seen clearly in *self-self experiments*: Two identical mRNA samples are labelled with different dyes and hybridized to the same slide. One would expect the red and green intensities to be equal, but the green intensities often tend to be higher than the red ones. In addition, this imbalance is not constant across the spots. It can vary according to spot intensity, slide origin, location on the slide and other parameters (see [47]). Normalization

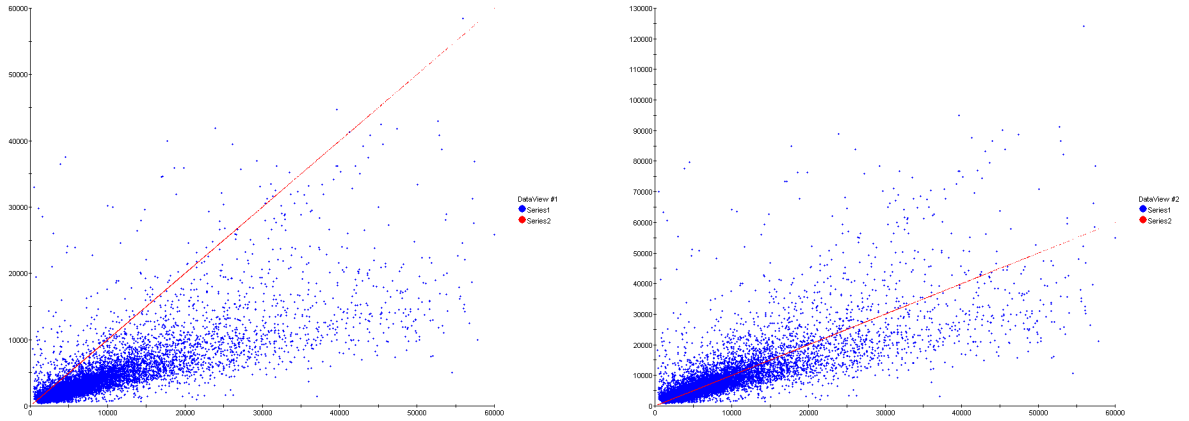

Figure 1.9: *Imbalance of red and green intensities: The left figure shows a scatterplot before (global) normalization. One dye tends to have higher overall intensities. Right figure: after normalization the dye-imbalance is corrected.*

is carried out *within* and *between* arrays. To be able to compare across slides, a *scaling* between arrays is applied. After normalization, the data for each gene are typically reported as an *expression ratio* or as the logarithm of the expression ratio. Note that normalization cannot rescue bad quality data in any steps before: For example, any missing values resulting from scanning or hybridization steps are lost information, so the associated values cannot be reproduced (see [26], [22]).

1.7 Analyzing experimental results

After carrying out experiments, it is time to answer the biological questions posed when designing the experiment. In most cases, the core question is which genes are DE[2] (*differentially expressed*) and therefore of interest for further studies or verification methods. The possibilities of

[2]see glossary

analyzing the results mainly depend on the experiment's design. Since unreplicated microarray experiments are still quite common, statistical methods are unapplicable for those experiments and the conclusions made have no statistical validation. For microarray experiments using replication, parametrical and non-parametrical tests as the t-test or the Mann-Whitney-test can be applied (see [28], [25], [45]).

Chapter 2

Objectives

The aim of this thesis was to develop a freely available, platform independent application for visualization, normalization and analysis of microarray experiments. It should provide a wide range of possible experimental designs and normalization methods. Users should be guided through the steps of normalization and data analysis.

2.1 Uploading Experiment Data

Since microarray experiments include multiple slides, the data should be treated according to the experimental design. The user should be able to organize the uploaded slides into experiment classes (biological conditions, i.e. different time-points) and to provide all the information needed for normalization (e.g. reverse labelled slides, groups of dye-swap pairs,...).

2.2 Supported Image Analysis Software

Starting point of all normalization steps are result files from image analysis software, containing all essential data. The program should get by without extra data (e.g. image files, GAL files). Result files from different software vendors (e.g. GenePix, Agilent) should be supported, it should be easy to add other result file formats.

2.3 Normalization Methods

A wide choice of common normalization methods should be offered to the user to remove the systematic errors within and between arrays (e.g. Lowess fit, using control spots, scaling and averaging within and between slides). It should be possible to add newly developed algorithms at a later stage. The normalization methods should be available as a library to be used in other analysis tools.

2.4 Data Visualization

Because visualization of microarray data is the best help for choosing the way to normalize experiments, common used plots (e.g. scatterplot, histogram, MA plot, boxplot) should be implemented.

2.5 Identifying Genes of interest

One aim of microarray technology is the search for new target genes. A module should be provided to detect DE genes with help of various methods (e.g. fold change detection, statistical tests).

2.6 Export Files to other Software

As normalization is just a preprocessing step, the results should be easily exportable to other analysis software (e.g. for clustering Genesis, see [13]). Output files should be printed, containing normalized data and results of statistic tests (e.g. significance level), respectively.

2.7 Using JCCharts

All plots should use the Sitraka JCChart base components.

2.8 Adding new normalization methods

It should be easy to add new developed normalization methods to the software. The available algorithms should be defined by a XML configuration-file and be loaded at runtime.

2.9 Different source-file formats

Different source-file formats (e.g. GenePix, Agilent) and their file-loaders should also be declared by a XML file and be loaded at runtime.

Chapter 3

Methods

This chapter gives a survey of the programming tools and methods used for visualization, normalization, analysis and organization of microarray data.

3.1 Image acquisition and analysis

The goal of image analysis is to provide foreground and background intensity values for the red and green channels for each spot on a microarray (see [14]). Secondly, quality measures for each spot are collected for marking weak or unreliable spots. Some issues of image acquisition highly affect further analysis steps like normalization. These should be illustrated here (see [29]).

3.1.1 Scanning and scanner settings

An optical scanner scans the array, recording the fluorescence emissions at each point on the slide. One scan for each channel (Cy5 and Cy3) is performed and the data is stored into two 16-bit TIFF (Tagged Image File Format) images. To avoid *saturated pixels* (pixels emitting more photons than the photomultiplier tube (PMT[1]) can process), the PMT-voltage can be adjusted so that the brightest pixels are below the scanner saturation. However, PMT settings are sometimes

[1] see glossary

a compromise between avoiding saturated pixels and getting weak sensitivity for less intense pixels.

3.1.2 Varying PMT settings

In common, the Cy3-channel shows higher intensities. To balance this effect, the PMT voltage for the Cy5-channel can be increased. The ideal scan is one in which the same amount of signal is acquired in each channel, because most normalization methods assume that the majority of spots on a slide show equal expression and therefore equal intensities in both channels. One should keep in mind that varying PMT settings between the two channels has already a kind of normalization effect, but it will have just little impact on the log_2-ratios, provided that an appropriate normalization method is applied. Especially intensity dependent methods (e.g. lowess) mitigate such effects (see [24])

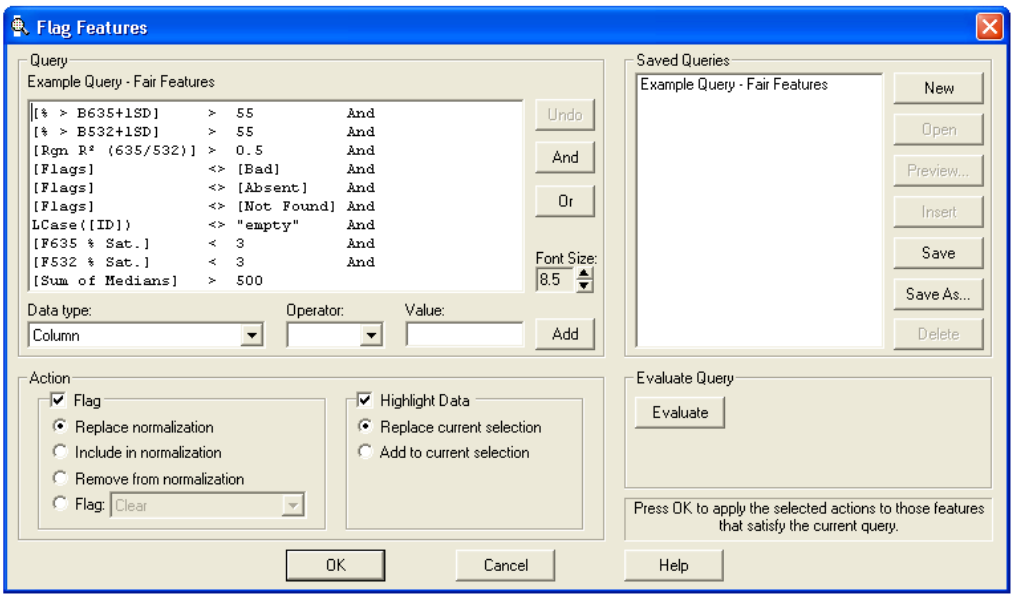

Figure 3.1: *Editing 'flag features' in GenePix Pro. The user can define quality criteria and link them logically. Spots (features) which do not meet these criteria are marked as 'bad' and can be removed from normalization if required (see [24]).*

3.1.3 Marking spots, flagging

Certain spots should be ignored during later analysis because of defects on the slide, saturation, small signal intensities etc. Most microarray acquistition programs have the ability to *flag* such spots. For example, GenePix Pro lets the user define special *flagging criteria* which can be saved to be applied to every microarray. Boolean queries consisting of several conditions (e.g. non-uniformity of spot, signal-intensity near background, etc.) joined by logical AND or OR can be used for quality control. Spots not meeting the conditions would be marked as *bad* and therefore excluded for later steps (see [24]).

3.2 Notation of microarray data

3.2.1 Intensities

With 2-color microarrays, the data acquisition process (scanning of the slide) provides at least four parameters for each spot, the red and green foreground and background intensities. The foreground red and green values are written as Rf and Gf, the background values are Rb and Gb. After background correction, the intensities are simply R and G. Intensity values are the origin for normalization techniques and data visualization.

3.2.2 Ratios

After normalization, the data for each gene is typically described as an *expression ratio*. The ratio X for the ith gene is simply:

$$X_i = \frac{R_i}{G_i} \quad \text{where} \quad i = 1, ...N_{genes} \tag{3.1}$$

Ratios have the advantage of providing an intuitive measure of expression change, but they remove information about the gene's absolute expression level. And they treat up- and downregulated genes differently: Genes upregulated by a factor 2 have an expression ratio of 2, whereas genes downregulated by factor 2 have an expression ratio of 0.5.

3.2.3 Log Ratios

A logarithmic transformation produces a continuous spectrum of values and treats up- and downregulated genes equal.

$$X_i = \log_2 \frac{R_i}{G_i} \quad \text{where} \quad i = 1, ...N_{genes} \tag{3.2}$$

On this log_2-scale, $X = 0$ represents equal expression, $X = 1$ represents upregulation by a factor 2, $X = -1$ downregulation by a factor 2, $X = 2$ upregulation by factor 4, and so on. Additionally, calculating log_2-values spreads the values more evenly across the intensity range and provides better visualization of the data. And it tends to make the variability of data more constant over the intensity range (see [46] for more information about data-transformations).

3.3 Graphical Representation of Microarray Data

Visualization can help to assess the success of the experiment and can guide the choice of normalization method or analysis tool. Thus, it is useful to have a variety of graphical displays for microarray data (see [37]).

3.3.1 Scatter Plot

Single slide expression data are typically displayed by plotting the log-intensity of the red dye against the log-intensity of the green dye: log_2R versus log_2G. Scatterplots may help to identify relationships between the two dyes, Cy5 and Cy3. The high correlation between the channel intensities always dominates the plot, so it may be difficult to discern the interesting features of the plot.

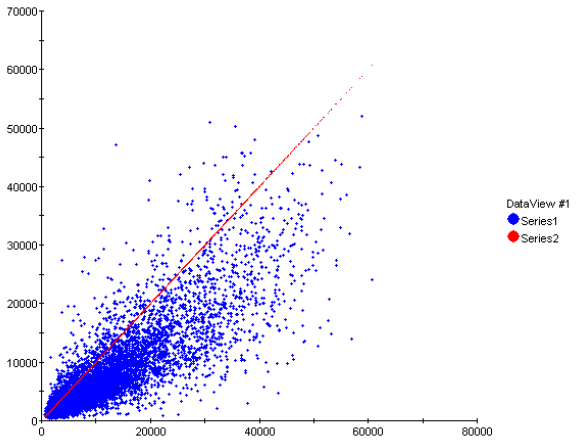

Figure 3.2: *Scatterplot of* $log_2 R$ *vs.* $log_2 G$*. The line marks* $log_2 \frac{R}{G} = 1$.

3.3.2 MA Plot

A *MA-plot* is a scatterplot with transformed axes. The X-axis conforms with the loged total intensity value of the spot, the Y-axis shows exactly the log-ratio of the two dyes.

$$M = \log_2 R - \log_2 G \quad \text{and} \quad A = \frac{1}{2}(\log_2 R + \log_2 G) \tag{3.3}$$

With it information about the intensity is introduced.

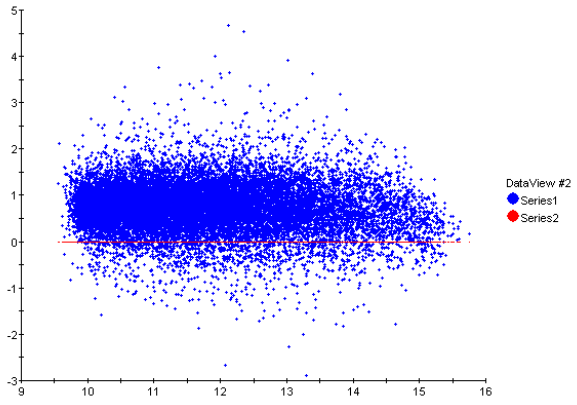

Figure 3.3: *MA plot of the same data set.*

3.3.3 Box and Whiskers Plot

A *boxplot* roughly displays the central tendency and variability of a dataset. A box in the middle encloses 50 percent of the data (Interquartile range IQR). The median is marked as a line in the box. The whiskers present the data values' total spread. In microarray experiments boxplots are useful for comparing log-ratios between different subsets of the data. For example, comparing multiple boxplots, where one single plot contains the log-ratios of one single subgrid.

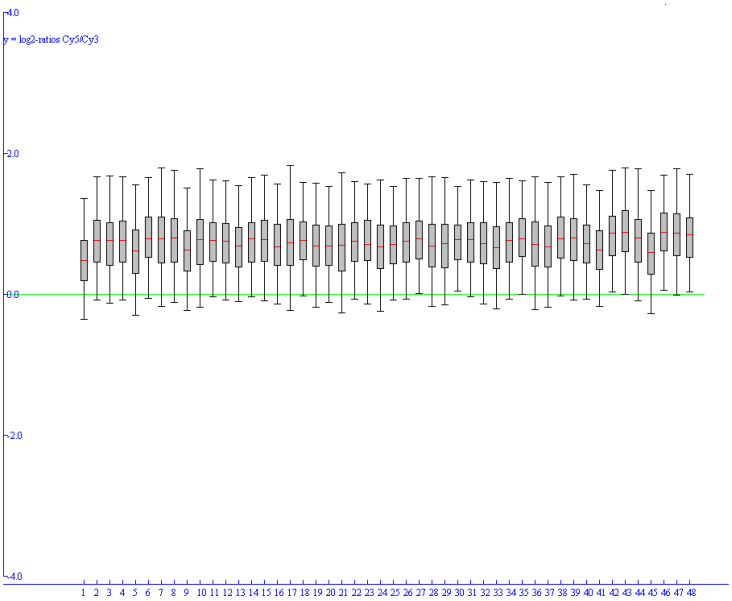

Figure 3.4: *Boxplot of slide with 48 subgrids, not normalized.*

3.3.4 Histogram

A *Histogram* plots the distribution of values in one sample. It can be used to answer following questions:

- How is the data distributed?
- How spread is the data?
- Are there outliers?

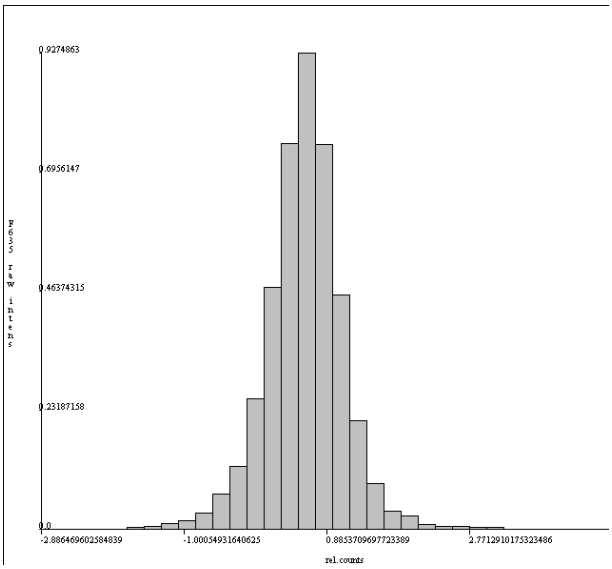

Figure 3.5: *Frequency histogram*

A histogram showing the distribution of log-ratios for a single slide can give an overview about how the data is distributed and therefore serves for deciding which normalization method to use or which statistical analysis is appropriate.

3.4 Background Correction

Having information about background intensities, it is recommended to correct the foreground intensities by subtracting the background, $R = Rf - Rb$ and $G = Gf - Gb$. The disadvantage of this subtraction is that negative values for R and G may be produced for some spots. Negative values cannot be log-transformed, therefore these spots would show missing values and must be excluded from analysis.

Improved methods of background adjustment are in development, because experience suggests that the background intensities often overestimate the true background (see [17]). In any case, spots with negative values for R or G are usually to weak and therefore of less interest. The adjusted intensities R and G are the origin for further processing steps, like normalization.

3.5 Normalization Methods

Depending on the experiment, normalization is used in different ways (see [22]). One has to distinguish between

- **within-slide normalization**
- **paired-slides normalization for dye-swap pairs**
- **multiple-slides normalization** (scaling between slides)

In each case the set of genes used for normalization has to be defined. That can be

- **All genes on a slide:** Assuming that most of the genes on the array have constant expression and all expression values are normally distributed (this is rarely the case!), all genes on the slide can be used for normalization. Using all genes offers the most stability for estimating spatial and intensity-dependent effects (see [22]).
- **Constantly expressed genes:** In biological samples with high divergence, normalization based on all genes may not be accurate. Often a set of housekeeping genes that show no change across any conditions are used to normalize other genes. Unfortunately, housekeeping genes often show sample specific bias and are typically highly expressed, so they will not allow the estimation of dye-bias for less-expressed genes (see [21]).
- **Set of control genes:** Alternatively to housekeeping genes, a set of spiked controls (clones from other organisms, e.g. Arabidopsis-clones spotted on human chips) or a titration series of control sequences can be used. Control sequences should have equal red and green intensities. Spots from titration series should show equal red and green intensities across the whole intensity range. Typically, titration series are done with a specially designed MSP (microarray sample pool) or genomic DNA (see [31]).
- **Rank-invariant genes:** An alternative method is to find an invariant set of genes per slide and to use this set for the normalization of all genes: All genes are sorted in ascending order according to their expression ratio. The position of a particular gene in this sorted

list is its rank. A set of genes is called rank-invariant if their ranks are the same for red and green intensities (see [42]).

3.5.1 Within slide normalization

If an experiment contains multiple slides, each of the slides must be normalized separately (except dye-swap paired slides) before scaling between slides can be done.

3.5.1.1 Global normalization

This simplest normalization method assumes that the red-green bias is constant across the array and the red and green intensities are related by a constant factor , i.e. $R = kG$. The goal is to estimate a constant factor c and correct the ratios by simply substracting c that the mean (or median) of the resulting intensity ratios is 1. This is equivalent to shifting the mean of the log-ratios to zero.

$$\log_2 \frac{R_i}{G_i} \rightarrow \log_2 \frac{R_i}{G_i} - c = \log_2 \frac{R_i}{kG_i} \tag{3.4}$$

A widely used choice for parameter $c = \log_2 k$ is the mean or median intensity log ratio of the particular slide.

3.5.1.2 Intensity dependent normalization

Several reports have shown that ratio values can have a systematic dependence on overall spot intensity. The global normalization approaches does not account for this bias. *Locally weighted linear regression* (lowess) or other robust linear regression methods can be used to remove such intensity-dependent effects. An easy way to visualize intensity-dependent effects is to generate a MA-plot for each slide to normalize. It can be seen that the majority of points lie on a curve, showing that the red-green bias depends on the intensity of the spot. Lowess estimates this curvature. It smoothes the MA-plot by subtracting the values of the estimated function from the original M-values (see [8]).

$$\log_2 \frac{R_i}{G_i} \rightarrow \log_2 \frac{R_i}{G_i} - c(A_i) = \log_2 \frac{R_i}{k(A_i)G_i} \quad \text{where} \quad A_i = \frac{1}{2}(\log_2 R_i + \log_2 G_i) \tag{3.5}$$

Here $c(A_i)$ is the lowess-fit to the MA-plot for the ith spot,$i = 1, ..., N$, and N is the number of spots.

3.5.1.3 Normalization using control spots

In both of the above mentioned methods, all or most of the genes are used for normalization. If a suitable set of *control spots*, arrayed on the same slide, is available (e.g. housekeeping genes, MSP titration series, rank-invariant genes), these elements can be used to normalize all genes on the slide. The set of control spots should cover the whole intensity range. Normalization is done by fitting a curve to the control spots and correcting all genes with this function. This is similar to intensity dependent normalization, but just using the control set to estimate the curvature.

3.5.1.4 Composite normalization

This procedure combines normalization methods based on all genes and those based on only MSP titration spots. At low spot intensities, normalization is done using all genes. For higher intensities, normalization is based on the MSP control spots. This principle may yet be applied to experiments without MSP series, using housekeeping genes for normalizing very high intensities and all genes for low intensities (see [21]).

3.5.1.5 Print tip group dependent normalization

Every *subgrid* (or block) is printed with the same print-tip. There may exist systematic differences between the tips, like differences in length or tip-opening and abrasion. These variations can cause spatial effects on the slide. Previously explained methods (global and intensity dependent) can be adapted to account for this problem, simply applying them to every single subgrid of one slide.

$$\log_2 \frac{R_i}{G_i} \rightarrow \log_2 \frac{R_i}{G_i} - c_j(A_i) = \log_2 \frac{R_i}{k_j(A_i)G_i} \tag{3.6}$$

where $c_j(A_i)$ is the lowess-fit to the MA-plot for the jth subgrid and the ith spot. $i = 1, ..., N$, and N is the number of spots. $j = 1, ..., M$, and M is the number of subgrids.

After print-tip group normalization some scale adjustment between the subgrids may be required to adjust differences in the spread of the log-ratios.

3.5.2 Paired Slides Normalization, Dye-Swap

A dye-swap pair consists of two slides. Every hybridization is done twice, with reverse dye assignment in the second hybridization.

$$\text{Slide 1} \quad M_i = \log_2 \frac{R_i}{G_i} = \mu_i + c_i \quad \text{and} \qquad \text{Slide 2} \quad M_i' = \log_2 \frac{R_i'}{G_i'} = \mu_i' + c_i' \tag{3.7}$$

where μ_i and μ_i' are the true log-ratios, c_i and c_i' the dye-effects. Because of reversed dye assignments one can expect:

$$\rightarrow \mu_i = -\mu_i' \tag{3.8}$$

Assuming that the dye biases in the two slides are similar, the log_2-ratios for the two slides are combined:

$$\frac{1}{2}(M_i - M_i') = \frac{1}{2}(\mu_i - \mu_i' + c_i - c_i') = \mu_i \quad \text{if} \quad c_i = c_i' \tag{3.9}$$

The normalized log_2-ratios will then be

$$M_i = \hat{\mu}_i = \frac{1}{2}(\log_2 \frac{R_i}{G_i} + \log_2 \frac{G_i'}{R_i'}) = \sqrt{\frac{R_i G_i'}{G_i R_i'}} \tag{3.10}$$

Another possibility is to correct the single intensity values. Calculating

$$k = \sqrt{\frac{R_i R_i'}{G_i G_i'}} \tag{3.11}$$

and correcting the intensity values with this factor k

$$R_{i,corr} \rightarrow R_i \quad \text{and} \quad G_{i,corr} \rightarrow kG_i \tag{3.12}$$

will lead to the same results as correcting the log_2-ratios directly, but gives the opportunity to visualize the effects of normalization (e.g. with scatterplot, MA plot). This step is called *self-normalization*.

To verify the assumtion of $c = c'$, the lowess-fits from both slides could be compared. If both fits show similar trends, self-normalization should provide reasonable results.

3.5.3 Multiple slides normalization, scaling between microarrays

After within-slide normalization, the normalized log-ratios will be centered around zero. However, there are often substantial scale differences (different spreads in the log-ratios) between microarrays, due to changes in PMT settings or other influences. To allow comparisons between microarrays, it is useful to scale the log_2-ratios of a series of slides.

3.6 Identifying differentially expressed genes

One main goal of microarray experiments is to identify DE genes. Usually, it will be practical to follow-up only a limited number of genes, a hundred say, so it is most important to identify the 100 most likely candidates. This reduced data set of candidates can then be used for further analysis (e.g. clustering techniques, see [13]). The complete list of all genes considered DE may be too large to be followed-up and therefore of less interest (see [37]).

3.6.1 Simple detection methods

Not every microarray experiment provides replicate measures (replicated spots on slides, replicated slides). In this case, it is impossible to apply statistics (e.g. t-test) for finding DE genes. Simpler methods have to be used.

3.6.1.1 Fold change detection

In this simple approach a fixed *fold-change*-cutoff is used to find the genes most differentially expressed. If a gene's log-ratio exhibits the cutoff (e.g. two-fold), it is marked as significant.

3.6.1.2 Setting confidence limits

Slightly more sophisticated is to calculate the *mean* and *standard deviation* of the distribution of log-ratios. Then a *confidence limit* (e.g. +/- two standard deviations) is defined to select genes with significant log-ratio. This is equivalent using *Z-scores* for the data set (see [26]). Such Z-score criterium would be illustrated by a MA plot as two horizontal lines, on both side of the zero-line. Points outside of the two lines would represent differentially expressed genes. For example, using a Z-score of +/- 1.96 would find exactly 5 % significant genes per data set. This is called 95 percent confidence level (see [1], [9]).

3.6.2 Statistical tests

Assuming that a series of n replicate arrays is available, statistics can be applied to find genes differentially expressed. First, an appropriate statistic ranks the genes in order of evidence for differential expression. Second, a critical value is choosen for the ranking statistic above which values are considered to be significant (see [3], [9]).

3.6.2.1 Student's t statistic, t-test

Simply sorting the genes according to their mean log-ratio level (for every particular gene across the replicate arrays) does not take account of the variability of the expression levels for each gene. The variability of log-ratios over replicates is not constant for all genes, so genes with larger variance may be detected as differentially expressed even if they are not. Ranking genes according to the t-statistic incorporates these different variabilities.

For every gene i in every particular replicated array j, the ratio-value can be written as

$$M_{ij} = \log_2 \frac{R_{ij}}{G_{ij}} \tag{3.13}$$

With the average over replicates

$$\overline{M}_i = \frac{1}{n_r} \sum_j \log_2 \frac{R_{ij}}{G_{ij}} = \frac{1}{n_r} \sum_j M_{ij} \tag{3.14}$$

where n_r is the number of replicated slides, the t-statistic for every gene i can be calculated:

$$t_i = \frac{\overline{M}_i}{s_i/\sqrt{n_r}} \;\; \forall \; i = 1...n_{genes} \tag{3.15}$$

s_i is the estimator for the standard deviation for a particular gene i:

$$s_i = \frac{\sum_j (M_{ij} - \overline{M}_i)}{n_r - 1} \;\; \forall \; i = 1...n_{genes} \tag{3.16}$$

This form of t-statistic might be used comparing two samples, A and B, spotted on the same array.

Comparing two samples with different sample-sizes (e.g. comparison of two experiment classes), the t-statistic turns to

$$t_{i12} = \frac{\bar{M}_i^1 - \bar{M}_i^2}{\sqrt{\frac{(s_i^1)^2}{n_1^2} + \frac{(s_i^2)^2}{n_2^2}}}. \tag{3.17}$$

where n_1 and n_2 are the number of replicates provided for the first and second biological condition to be compared.

Large absolute t-statistics suggest that the corresponding genes have different expression levels in sample A and sample B. Note that replication is essential for such tests.

3.6.2.2 Mann-Whitney U-test

T-statistics assume that the data has a *gaussian* or normal distribution. In practice, this is rarely the case. Most distributions of microarray log-ratios tend to have heavier tails than a normal distribution. So the number of differentially expressed genes might be over-estimated. Non-parametric tests, like the Mann Whitney test, incorporate this fact and do not lead to false estimations.

3.6.2.3 Assigning significance, p-values

After computing the t-statistics, the genes of interest are found by testing the *null hypothesis* of equal mean expression levels in the two samples, A and B. This can be obtained by calculating *p-values* for each gene, using t-value and sample-size. (See appendix 2 for *statistical testing*)

The null hypothesis is rejected, if a gene's p-value falls below a certain threshold (alpha level). Then the gene is marked as significantly DE.

A common alpha-value is 0.05, which denotes a 5 % type I error (false positive rate) rate: 5 genes out of 100 are found significant, even if they are not.

3.6.2.4 Adjusted p-values, Bonferroni step down

One concern in applying hypothesis testing to microarrays is *multiple testing*. Testing many hypothesis (e.g. 30000), the probability of getting false positive hits can increase sharply. One method for controlling the type I error rate are *adjusted p-values*. Among many others, the *Bonferroni method* is the best known (see [34]). It simply divides the alpha-level by the number of hypothesis tests. This single step adjustment assumes normality of the data and therefore tends to be very conservative. Other more complex methods (e.g. Westfall and Young step down) exceed the scope of this thesis.

3.7 Usability testing

3.7.1 Background

The basic idea behind usability testing is simple. If one wants to know wheter a software or a web site is easy enough to use, some people are told to use it and note where they run into trouble. This information can be used for fixing bugs and improvements. Usability test should be perfomed on the one hand by the targeted users and on the other hand by users not familiar with the field. But it is much more important to test early and often. Even the worst test with

the wrong user early in the project will show always more valuable results than a sophisticated test near the end of the development (see [38]).

3.7.2 How many users

In most cases, the ideal number of users is three, at most four (see [38]). The first three users are very likely to detect the most significant problems, and with only three users it is simple to do further test rounds. After each round one can improve and fix bugs.

3.8 Software development methods

3.8.1 Java

One precondition was the use of a platform-independent programming language. Java (see [41]) offers this attribute. For program development Borland's JBuilder 7.0 has been used (see [15]). Other IDEs would be Forte4Java (see [40]) or netBeans (see [7]).

3.8.1.1 Java Language

Java is a simple, platform-independent, object-oriented, distributed, interpreted, robust, architectureneutral, multithreaded high-level programming language. The Java programming language is unusual in that a program is both compiled and interpreted. A Java compiler generates an architecture-neutral object file executable on any processor supporting the Java run-time system. The object code consists of bytecode instructions designed to be both easy to interpret on any machine and easily translated into native machine code at load time. So compilation happens just once, interpretation occurs each time the program is executed.

3.8.1.2 The Java Platform

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other

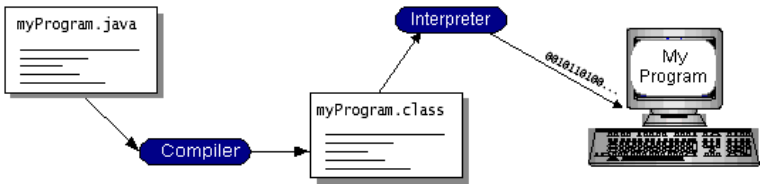

Figure 3.6: *The compiler translates a program into an intermediate language called Java bytecode. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens just once, interpretation occurs each time the program is executed.*

hardware-based platforms (although there are special processors which execute Java bytecode directly). The Java platform has two components:

- **Java Virtual Machine** (Java VM). It's the base for the Java platform and is ported onto various hardware-based platforms.
- **Java Application Programming Interface** (Java API) The Java API is a large collection of software components that provide many useful capabilities, such as *graphical user interface* (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

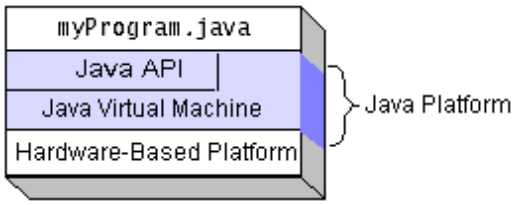

Figure 3.7: *A program running on the Java platform. The Java API and the virtual machine insulate the program from the hardware.*

3.8.2 JClass Libraries

JClass helps to build Java applications by offering Java developers versions of the components required by many standard applications, such as charts, tables, and reporting/printing (see [39])

3.8.2.1 JClass Field

JClass Field is a set of Java components (e.g. combo boxes, spin fields, text fields) that permits the collection, validation, and display of textual, calendar, and numeric data. Built-in validation methods can be applied for various consistency checks on the information and to give the end-user visual and audible feedback when the validator detects an incorrect entry.

3.8.2.2 JClass Elements

JClass Elements is a broad collection of GUI components and utility classes designed to augment Java Swing's basic offerings. And it's easy to adapt them to the programmer's custom needs. E.g. the Wizard Creator lets one create and manage a wizard-style group of user-dialogs.

3.8.2.3 JClass Chart

JClass Chart is a charting/graphing component written entirely in Java. The chart component displays data graphically in a window and can interact with a user. Different chart types (e.g. Scatter plot, Pie charts, Bar charts) are available.

3.8.3 TUGUtilities Library

The TUGUtilities library (developed by the Bioinformatics Group, TU Graz) provides several useful software utilities. The library makes it easy for other developers to access common used functions (e.g. mathematics, statistics, logging).

Chapter 4

Results

This chapter presents results of organizing, normalizing and analyzing microarray experiments using the software tool, which has been developed in the course of this work.

4.1 ArrayNorm

ArrayNorm is an application which provides tools for visualizing, normalizing and analyzing data from a wide range of possible microarray experiment designs. The features were tested, using data from former experiments.

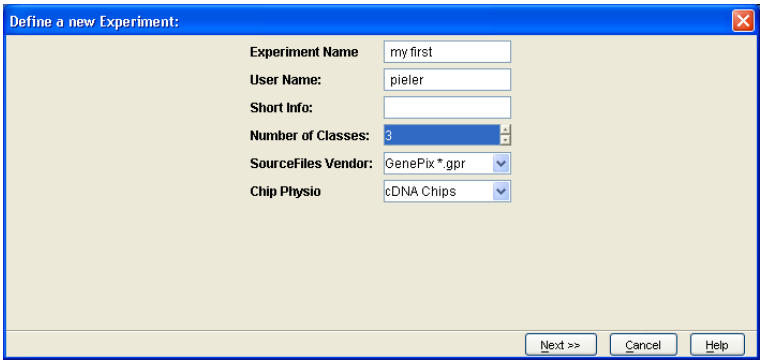

Figure 4.1: *First step in setting up a new experiment. Defining general parameters, like name and sourcefile vendor.*

4.1.1 Loading data, defining the experimental design

Starting a new experiment requires information about general information, sources of data and experimental design. A wizard leads through these steps.

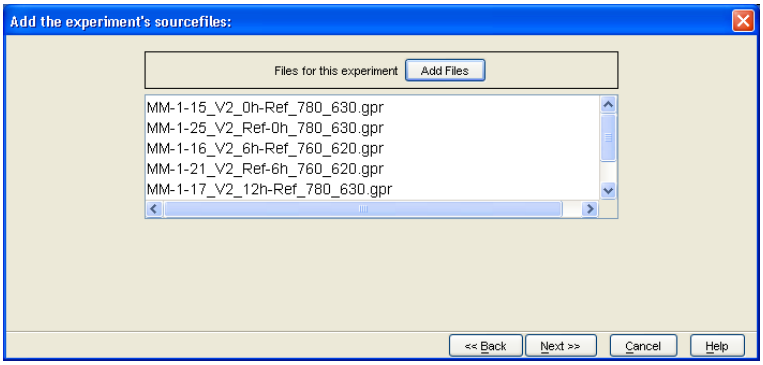

Figure 4.2: *Selecting all source-files*

4.1.1.1 Experimental setup

Every new experiment can be attached with general informations, like name, number of *experimental classes*[1], sourcefile vendor, etc. Defining the number of classes is not definite, it may be changed afterwards.

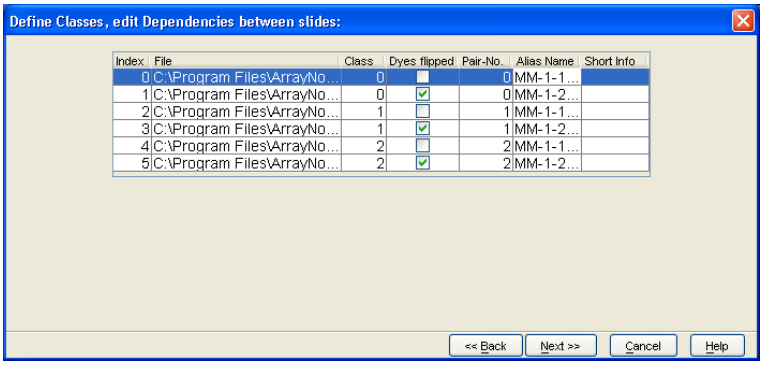

Figure 4.3: *Experimental Design: defining relationships between slides, marking dye-assignments,...*

[1] see glossary

4.1.1.2 Selecting source files

Since the number of possible microarrays is not limited, the wizard provides a file list, to which multiple files can be added. After selection, all files are numerated in the list, without information about class-affiliation or dye-assignment. Note that all files in one experiment must have the same number of spots.

4.1.1.3 Experimental design

For normalization and analysis tasks, it is necessary to define relationships between microarrays. For example, which slides belong to the same class, which are reverse-labelled and if there are dye-swap pairs available. All these informations will be used for normalization, scaling between slides and replicate handling. For every hybridisation the user can edit

- the assignment to a class.
- whether the particular slide is reverse-labelled or not.
- the assignment of a dye-swap partner, if available.
- an alias-name to appear in the data-tree.

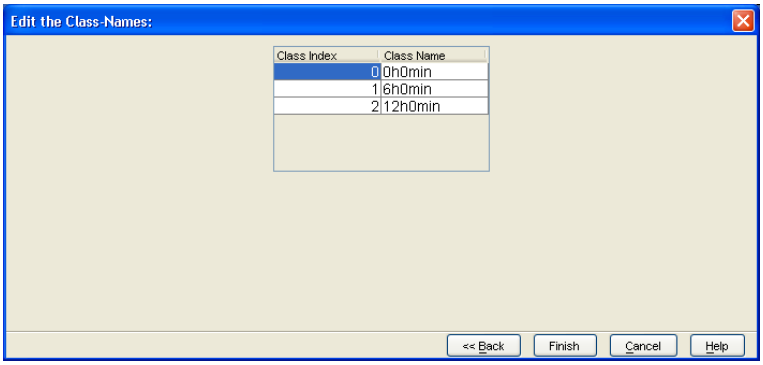

Figure 4.4: *Editing class names.*

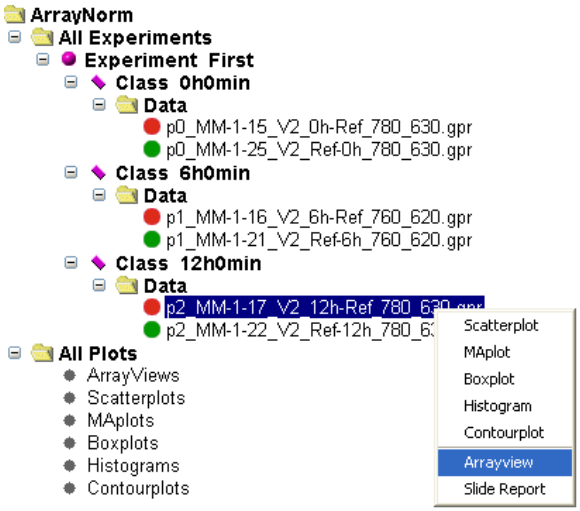

Figure 4.5: *Data organization tree. After uploading, the slides are arranged according to their classes. This experiment features three classes, each holding one dye-swap pair (indicated by the colored points and prefixes). The microarray popup-menu is opened by a rightclick on the slide.*

4.1.1.4 Data organization tree

After setting up the experiment, all data are organized by a graphical tree, reflecting the structure of the experimental design.

- The experiment is splitted into *classes*, each class represents a biological condition (e.g. a timepoint of a timecourse experiment). The classes are named as specified in the wizard.
- Every class holds its associated slides, marked with colors and prefixes. A red point indicates normal dye-assignment, a green point indicates a reverse-labelled microarray. The prefix pX states that the particular slide belongs to the Xth dye-swap pair. A dye-swap pair always contains one normal and one reverse labelled slide. Naturally, a dye-swap pair can only include slides from the same class. Multiple dye-swap pairs within a single class are possible.
- The 'All Plots'-folder holds plots created by the user for fast navigation between all opened plots.

4.1.1.5 Accessing methods

In principle, all actions or methods can be accessed by

- **Rightclicking a tree's component or folder.** Depending which kind of folder (experiment, class, slide, results,...), a menu pops up with possible actions for the particular component.
- **Buttons.** Especially for Mac users. For some functions (e.g. normalization), the wanted tree component must be preselected by a mouseclick. Warning dialogs will inform the user about incorrect or impossible selections.

4.1.2 Visualization

To get an idea about the condition of the data sets or the effects of different normalization methods, means of graphical display can help assessing the success of the experiment and choosing the analysis tools.

4.1.2.1 Array view

The Array Viewer features false-colored images for the red and green channel per slide. It does not represent the scanned microarray output image (e.g. provided by GenePix Pro), but a diagnostic plot showing

- the arrangement of print-tip groups.
- rough information on spatial artifacts (e.g. scratches)
- highlighted control-spots, orange
- bad quality spots filtered out by GenePix-criterium, grey

The coloring is automatically adapted to the maximum intensity occuring in the particular channel.

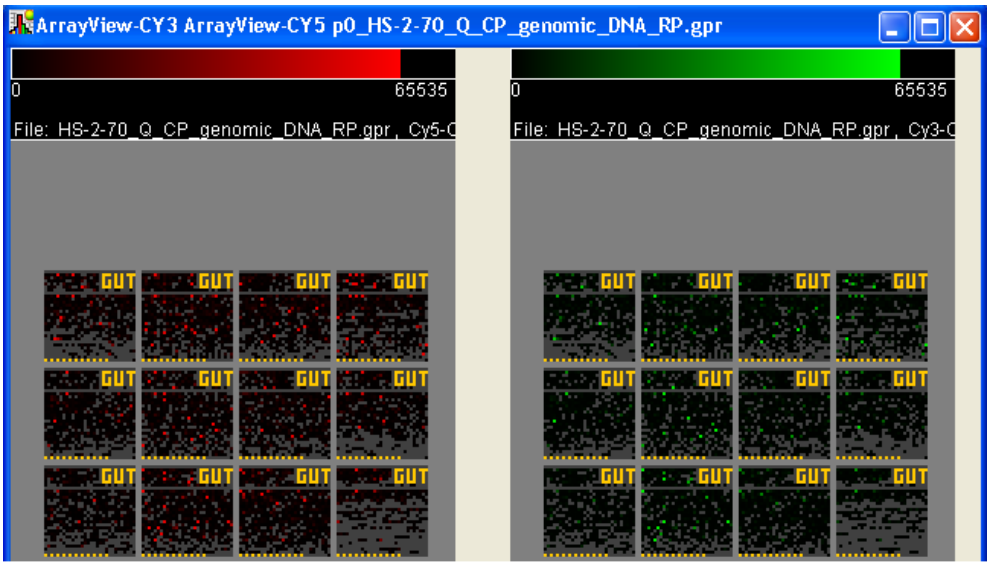

Figure 4.6: *ArrayView. This cutout from a 43.000-spots slide shows the arrangement of sub-grids, how the controls are situated (orange dots) and the distribution of bad spots (grey colored spots).*

4.1.2.2 Scatterplot and MA-plot

Plotting the $log_2 R$ intensities versus the $log_2 G$ intensities is a common way to display single slide expression data. An alternative is to transform the axes to introduce intensity information (see figures 3.2 and 3.3).

4.1.2.3 Ratio histogram

Frequency histograms counts the number of ratios for every intensity value and provides information about the distribution of the ratios (see figure 3.5).

4.1.2.4 Boxplot

Boxplots are useful for comparing ratio-values between different groups of data. That can be

- different print-tip groups on a single slide.
- all slides contained in one experimental class.

Especially, they give a good display of normalization effects (see figure 3.4).

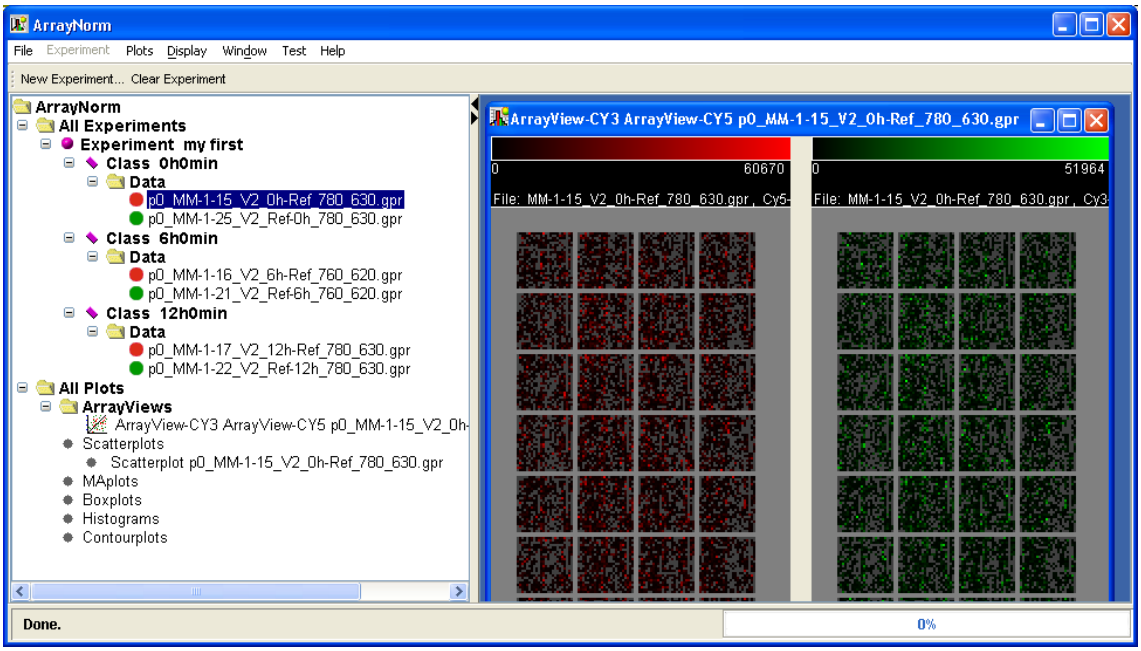

Figure 4.7: *ArrayNorm GUI. All information is shown by an experiment tree, representing the experiment design. Red points mark normally labelled slides, green points reverse labelled slides. The prefix p0 indicates that this slide belongs to dye-swap pair 0.*

4.1.2.5 Capturing plots

Every open plot can be exported to a file. Possible encoding formats are PNG and JPEG. To capture a plot, select its frame and press the ”Capture” button. A filechooser will open for editing filename and encoding-format.

4.1.3 Background correction

Background subtraction can be done separately for each class or for the whole experiment at once. If a negative intensity value (background $>$ foreground intensity) occurs, the particular spot will be marked bad and therefore ignored. Generally, this case will not arise, because pre-filtering (e.g. flagging in GenePix) checks for bad quality spots (see [24]).

4.1.4 Normalization methods

The user can choose among several normalization methods, depending on the experimental design.

4.1.4.1 Available methods

- Global-median normalization
- Global-median print-tip group dependent normalization
- Intensity-dependent normalization (Lowess fit)
- Normalization using control-spots
- Paired-slides normalization (dye-swap experiment)

4.1.4.2 Applying normalization

All methods can be applied for

- **one single class.** The chosen method will be performed on every slide belonging to the particular class. Dye-swap normalization is only carried out, if dye-swap pairs exist.
- **the whole experiment.** All classes will obtain the same method. This is the recommended way to keep classes comparable for analysis.

Using different normalization methods within a class is not possible. It would introduce additional errors and inhibit reasonable analysis. In some cases it is necessary to use different normalization methods for different classes (e.g. one class includes dye-swap pairs, other classes just replicated slides). But this should always be an exception. Different methods treat data in different ways and that makes comparisons less meaningful.

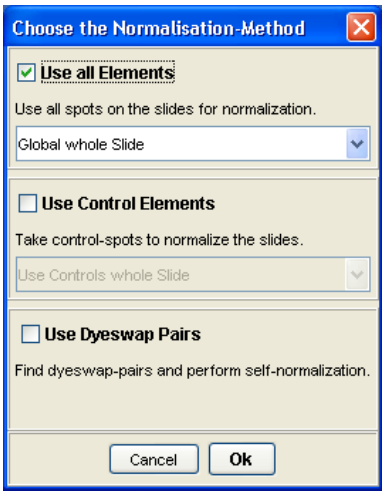

Figure 4.8: *The user can choose the wanted method out of three groups: a) using all spots, b) using control spots, c) using dyeswap pairs.*

4.1.4.3 Normalization examples

Two examples illustrate the normalization step.

- **Paired-slides normalization:** If the currently loaded experiment contains dye-swap pairs, the use of self-normalization would be recommended. Each pair will be corrected individually. If no pairs were predefined, the reverse-labelled slides will be averaged for

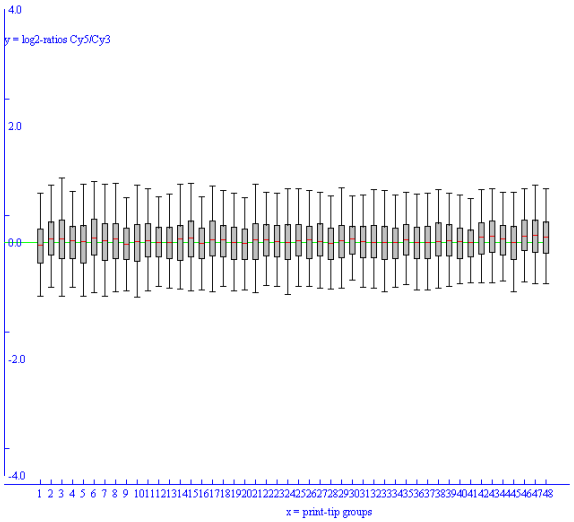

Figure 4.9: *Boxplot after self-normalization. The boxes have been shifted to the zero-line.*

building a template. This template is used as *dye-swap partner* for every normal-labelled

slide. Thus, the number of self-normalizations within a class equals the number of normal labelled slides. An informative way to illustrate the effect of self-normalization are boxplots. After normalization, the medians of log-ratios should be shifted to zero. Additionally, the plot of the normal-labelled slide should be exactly the reversed to its reverse-labelled partner.

- **Normalization with control-spots:** Providing control-spots, slides can be normalized by this subset of genes. This assumes that the gene-names of the controls are marked with a specific prefix (see Appendix A for 'Marking control genes'). Using the Levenberg-

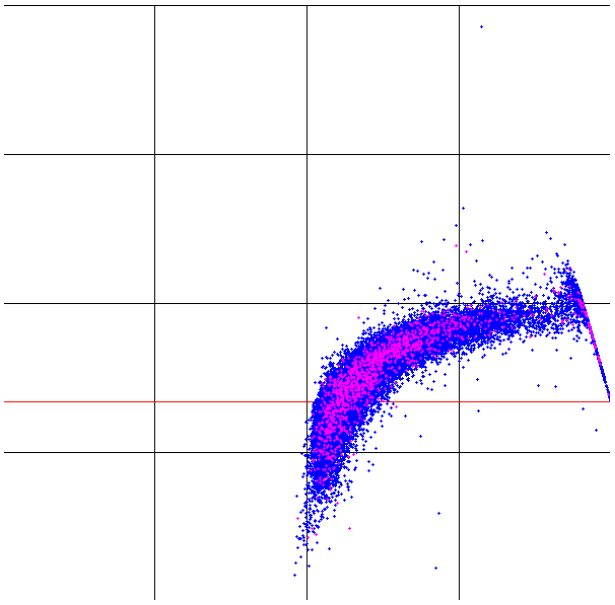

Figure 4.10: *MA plot: The highlighted spots are controls. The sharp edge on the right end is due to scanner saturation.*

Marquardt algorithm, a polynomial function is fitted to the control spots (see [18]). This function is used to correct all other genes on the slide. The idea is similar to intensity-dependent normalization, just using another set of genes to fit the function. A common way is to apply Lowess to the set of controls. But this can be critical with a small amount of control spots. The example-slide (43000 spots, 1900 controls) just has 150 good-quality controls, after serious quality-filtering in GenePix Pro. That would be too little for Lowess.

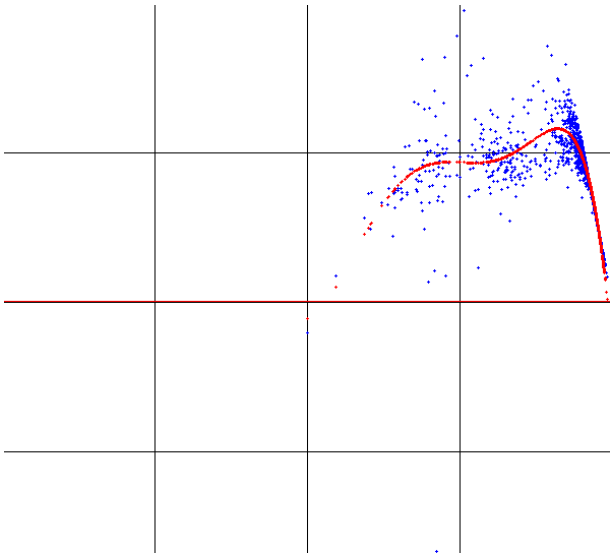

Figure 4.11: *MA plot: The curve is the Levenberg-fit to the control spots. It follows a polynomial function of* $3rd$ *order.*

4.1.4.4 Resetting the data

Having applied any changes to the data (like background subraction, normalization), one might undo all steps. By clicking 'Reset Experiment' on the experiment's popup-menu the origin will be reloaded, all changes will be cancelled. Note that already opened plots will keep unaffected.

4.1.5 Finalize, replicate handling and generating results

Before analysis can be carried out, some steps have to be done:

- **Replicate handling.** If there are replicated spots on a single slide, find and average them.
- **Scaling between slides.** All slides within a class will be rescaled to ensure comparability.
- **Merging slides.** If replicated slides within a class are available, average them. The results are ratio-values for each gene on the slide.
- **Data transformation.** The ratios can be log_2-transformed or not.
- **Export to file.** The resulting values (i.e. ratios or log_2-ratios) can be exported to a file, which is suitable for further software (e.g. Genesis, see [13]).

These steps are applied to each class. For each gene, the results contain values for ratios, standard-deviations and sample-size. All these values will be needed for statistical testings. Guided by a wizard, the user can define the replicate treatment, the transformation of ratios and if control spots should be printed to the file or not. A 'Results'-icon added to the data-tree indicates that results are available and analysis can be performed.

4.1.6 Determining differentially expressed genes

4.1.6.1 Simple methods

The simple methods address experiments with no or insufficient use of replication (i.e. replicated slides).

- **Defining a fixed fold-change cut-off:** Every gene in every class will be marked as DE, if its absolute intensity-ratio exceeds this threshold. In experiments with more than one class, a gene will be exported to the output file if it is DE in any of the classes. The user can edit the log_2-scaled cut-off. For example, a value of 1.0 denotes a two-fold change in expression.

- **Setting a confidence limit:** For every gene in every class, z-values are computed and compared to a user-defined cut-off score. Those genes with higher z-scores will be marked and treated as above.

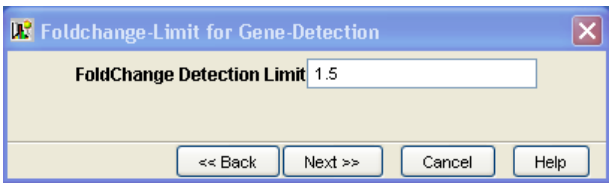

Figure 4.12: *If a gene's log2-ratio exceeds 1.5, it will be marked as DE. A short dialog will tell the user how many genes are detected.*

4.1.6.2 Statistical tests

Providing a sufficient number of replicates for each gene, reliable t-values can be computed. This can be done

- **within a single class:** Every log-ratio within this class is tested for significance.
- **comparing two particular classes:** The log-ratios of two classes are compared. If a log-ratio shows significant differences between the two classes, the particular gene will be marked as DE.
- **comparing all classes, loop design:** Each class is compared with its neighbor class: The first with the second, the second with the third, and so on. Having n classes, the last comparison is class n vs. class 1. The number of t-tests per gene will be n. A gene will be marked as DE, if at least one t-test shows a significant difference in log-ratios.
- **comparing all classes, reference design:** By default, the first class is the reference. The log-ratios of all other classes are compared to the reference's log-ratios. Having n classes, the number of t-tests per gene will be $n - 1$. Again, at least one t-test has to be significant to mark a gene as DE.

With aid of t-value and sample-size, p-values are computed for every gene. If a particular gene's p-value falls below a certain user-defined alpha-value, it is marked DE. To account for multiple testing, it is possible (and recommended) to adjust the p-values according to Bonferroni's step-down method.

4.1.6.3 Output files

Having figured out DE genes, the gene-names, gene-IDs and log_2-ratio values (or optional the p-values) for every class can be saved to a text-file. The file format is compatible with the "Stanford Flat File" format (see [35]) and can be loaded with the Genesis clustering software.

4.1.7 Extensibility

4.1.7.1 Adding new normalization methods

It is easy for developers to implement new normalization algorithms (e.g. local-mean methods) and add them to the existing software. A XML configuration-file defines the names and the according Java class-names of all available methods. After choosing the normalization method by the user, the particular class is loaded at runtime.

4.1.7.2 New source-file formats

Different sourcefile formats (e.g. GenePix, Agilent) need different file parsers. All file formats and their parser-classes are also declared within a XML file and can easily be extended.

4.2 Usability test

Four people (two biologists, two software-developers) took part in the usability test. The test contained a list of predifined tasks (loading and defining an experiment, creating plots, normalization and analysis) and the verbal and written evaluation of the user.

4.2.1 Results of the test

- The overall design of the GUI was assessed well, due to its similarity to Genesis. Data loading and defining the experiment's design caused some problems for users without knowledge about microarray design (defining dyeswap-pairs). It was intuitive for the rest of the test persons. The data tree was understandable for all testers.

- Creating plots and showing reports was easy to use for all persons, however the handling (re-scaling, closing, labelling of the axes) of the plots was not perfect due to slow repaint-methods. Accessing the functions by right-clicking the tree-elements was much more easier than pressing the particular buttons at the toolbar.

- Why background correction should be performed was not intuitive. Experienced users assumed that this is done by GenePix. Choosing the Normalization method was very easy for all users, which are familiar with the process. Finalizing and handling of the replicates caused some misunderstanding in one case, but it was rated well by the people who helped to define the specifications of the software.

- Some users proposed to provide a kind of 'guideline' or 'workflow display' which should help to find through the normalization process. Also a 'nice to have' feature would be to mark already normalized classes (or slides) with different folder-icons.

Description of task	**Rating**
First impression of the GUI	1.5
Loading of GPR-files	1.75
Definition of experimental design	1.75
Data Tree: understanding?	1.5
Opening plots and reports	2.5
Background-Correction of all slides	1.5
Normalizing using dyeswap-pairs	1.5
Finalize and Replicate Handling, log2-transformation	2.0
Analysis (find DE genes) using foldchange detection	1.75
Saving the list of DE genes	1.5
Overall impression: usability and functions	1.5

Table 4.1: Usability-test: tasks and their ratings.

Concluding, the use of ArrayNorm and its features was very intuitive for all persons with knowledge about the sense of normalization and data analysis. The test gave important suggestions for improvement, in the meantime most issues have been solved, but some are still in progress.

Chapter 5

Discussion

In this thesis, a platform-independent and versatile software-tool for normalizing and analyzing microarray experiment data was developed. The presented program handles a wide range of possible experiment designs, from simple single-slide experiments to time-course experiments with replicated dye-swap pairs. Any number of microarray-resultfiles can be uploaded, special attention was given to manage data from GenePix gpr-files.

- Since experimental design is an important concern, ArrayNorm lets the user define relationships between single microarrays or divide the experiment into classes of microarrays. It can deal with replicated spots on single slides, replicated slides, control-spots and reverse-labelled slides.

- Besides background-subtraction, ArrayNorm provides several means of normalization. The possibilities are global median, lowess function for intensity dependent normalization, self-normalization for dye-swap pairs and normalization with control spots using a Levenberg-Marquardt fitting-algorithm. Global and lowess methods can optionally be performed for single subgrids to get rid of spatial effects.

- The effects of normalization can be observed by capable graphical plots like scatterplots, MA-plots, boxplots and histograms. All plots can be done before and after normalization. The ArrayView helps to visualize spatial effects, bad spots and control elements within a

single slide. Simple slide-reports give an idea about the quality of the slides (percentage of good spots, overall intensities, etc.). Dealing with bad spots and missing values can be crucial when looking for marginal differences in data. Provided that proper quality-prefiltering was applied (e.g. flagging bad spots in GenePix), ArrayNorm excludes bad spots from all normalization, replicate handling and analysis.

- Although clustering methods are widely used for analysis, it is often useful to reduce the amount of data to a subset of genes, usually to those which are most variable between samples. To detect differentially expressed genes, the software provides simple fold-change detection and additionally statistical tests (t-test, Mann-Whitney). Detected target genes can be printed to a file, compatible to other analysis-software (e.g. Genesis).

5.1 Potential improvements

Since currently new methods for normalization and analysis are developed, it may be useful to adapt the ArrayNorm software to these needs. Possible improvements would be:

- Compatibility with other sourcefile-vendors (Affymetrix,...)
- Handling of multiple experiments at once
- Further normalization methods, e.g. *Local mean* (see [16])
- Improved methods for scaling between slides
- Automatic detection and disabling of bad-quality slides
- More sophisticated methods for detection of DE genes
- Correlation plots for testing replicated slides
- Additional diagnostic plots (e.g. QQ-plot)
- Database-connectivity to MARS[1]

[1] see glossary

- ANOVA
- User management
- Web-interface for uploading and normalizing of data

5.2 Conclusions

5.2.1 Experimental design

To give experimenters a short guideline about planning an experiment, some items should be discussed, although it is not possible to state universal recommendations for all situations at once. In cases where two mRNA samples, A and B, should be compared, it is always more accurate to hybridize A and B together on the same slide rather than comparing them indirectly by a reference sample. Additionally, in such *direct comparisons* it is recommended to plan **dye-swap pairs**. For **time-course experiments** a loop-design is a good choice. In experiments where the goal is to compare different mutant-types with a wild-type, it is suggested to use reference design with the wild-type RNA as common reference (see [33], [27], [43]).

5.2.1.1 Replication

Replication provides the ability to use formal statistical tests to decide whether log-ratios are significantly different to zero. For example, t-tests are applicable to analyzing data from replicate slides. The type of replication used in the experiment affects the precision of the results:

- To achieve averages of independent data and to generalize conclusions, *biological replicates* would be the right choice.
- *Technical replicates* help to reduce variability in measurement.

A combination of both types would be best.

5.2.2 Preprocessing issues

Testing different normalization-methods, one would recognize the major role of datapreprocessing (e.g. using GenePix Pro). Attention should be given to quality-filters for finding and flagging bad spots, because well filtered data enables better normalization-performance. During saturated spots will corrupt the fitted function and as well the results. Using different PMT-settings for Cy3- and Cy5-channels is less crucial and can be balanced by normalization. Altering the PMT settings within an experiment would render the slides incomparable (see [24]).

5.2.3 Choice of normalization method

The choice of normalization depends not only on the experimental design but also on the data quality. For instance self-normalization using two slides with a large number of bad spots can cause even much more missing values in the results. Although global normalization is not as accurate as self-normalization, it will be more stable in such cases. Blind trust in lowess-fitting can also be misleading. It detects any curvature in the MA-plot as systematic bias even if the shape results from biological conditions.
Anyway, once a normalization method is choosen, it is recommended to keep it up for all classes and all slides.

Acknowledgements

I want to thank my supervisors Gerhard Thallinger and Zlatko Trajanoski. Gerhard helped me in all matters of programming and informatic problems. Zlatko gave me the opportunity and motivation to work on this thesis.

Thanks to all the members of the bioinformatics group for their efforts to answer my questions. Especially Hubert Hackl, Marcel Scheideler and Andreas Prokesch always were competent opponents in many discussions and shared their knowlegde with me. Susi Prattes helped testing my software.

Further I want to thank my student-fellows Tom Truskaller, Johannes Rainer, Christoph Thumser and Bernhard Mlecnik for their every-day help and the nice working-atmosphere.

Special thanks to Fatima Sanchez-Cabo for her friendship, support, ideas and visions about this thesis. Grácias!

Last not least, I want to thank my parents for giving me the opportunity to study. They always supported me with faith, energy, and encouragement.

Bibliography

[1] Definition of statistical terms. WWW, September 4th, 2003. http://www.niwa.cri.nz/rc/prog/stats/intro/def.

[2] DNA Microarray Methodology - Flash Animation. WWW, September 4th, 2003. http://www.bio.davidson.edu/courses/genomics/chip/chip.html.

[3] Hyperstat Online Textbook. WWW, September 4th, 2003. http://davidmlane.com/hyperstat/index.html.

[4] Microarray Pipeline. WWW, September 4th, 2003. http://brc.mcw.edu/microarray/overview/workflow.html.

[5] Microarrays Factsheet. WWW, September 4th, 2003. http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html.

[6] Microarrays.org. WWW, September 4th, 2003. http://www.microarrays.org/.

[7] NetBeans. WWW, September 4th, 2003. http://www.netbeans.org/.

[8] Numerical Recipes. WWW, September 4th, 2003. http://www.nr.com/.

[9] Online Statistics Tutorials. WWW, September 4th, 2003. http://www.texasoft.com/tutindex.html.

[10] Ornl Array Images. WWW, September 4th, 2003. http://homer.hsr.ornl.gov/CBPS/Arraytechnology/images.html.

[11] TIGR, The Institute for Genomic Research. WWW, September 4th, 2003. http://www.tigr.org/.

[12] Churchill G A. Fundamentals of experimental design for cdna microarrays. *Nature Genetics Supplement*, 32:490–495, December 2002.

[13] Sturn A, Quackenbush J, and Trajanoski Z. Genesis: cluster analysis of microarray data. *Bioinformatics*, 18:207–208, September 2002.

[14] Axon. Axon Instruments Microarray Analysis. WWW, September 4th, 2003. http://www.axon.com/gn_Genomics.html.

[15] Borland. JBuilder. WWW, September 4th, 2003. http://www.borland.com/jbuilder/.

[16] Colantuoni C, Henry G, Zeger S, and Pevsner J. Local mean normalization of microarray element signal intensities across an array surface: Quality control and correction of spatially systematic artifacts. *BioTechniques,* 32:1316–1320, June 2002.

[17] Kooperberg C, Fazzio T G, Delrow J J, and Tsukiyama T. Improved background correction for spotted dna microarrays. *Comput Biol.*, 9:55–66, 2002. PM:11911795.

[18] Marquardt D. An algorithm for least squaresestimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441.

[19] Hackl H. SOP: Analysis of microarray images with GenePix Pro 4.1 by Axon Instruments. 2003. https://gold.tugraz.at.

[20] Motulsky H. *Intuitive Biostatistics*. Oxford University Press, 1th edition, 1995.

[21] Yang Y H, Dudoit S, Luu P, Lin D, Peng V, Ngai J, and Speed T. Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research, 2002 Oxford University Press*, 30:1–10, December 2002.

[22] Yang Y H, Dudoit S, Luu P, and Speed T. Normalization for cdna microarray data. *Speed Group Microarray Page*, 2002. http://www.stat.berkeley.edu/users/terry/zarray/Html/papersindex.html.

[23] Loennstedt I and Speed T. Replicated microarray data. *Statistical Sinica, accepted*, 12:31–46, 2002.

[24] Axon Instruments. *GenePix Pro 4.1 User's Manual*.

[25] Quackenbush J. Computional analysis of microarray data. *Nature Reviews, Genetics*, 2:418–427, June 2002.

[26] Quackenbush J. Microarray data normalization and transformation. *Nature genetics supplement*, 32:496–501, December 2002.

[27] Kerr M K and Churchill G A. Experimental design for gene expression microarrays. *Jackson Laboratory*, 2002. http://www.biostat.wisc.edu/geda/LITERATURE/CHURCHILL/churchill2.pdf.

[28] Kerr M K and Churchill G A. Statistical design and the analysis of gene expression microarray data. *Jackson Laboratory*, 2002.

[29] Smyth G K, Yang Y H, and Speed T. Statistical issues in cdna microarray data analysis. *Functional Genomics*, June 2002.

[30] Lee M L, Kuo F C, Whitmore G A, and Sklar J. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cdna hybridizations. *Proc Natl Acad Sci U S A*, 97:9834–9839, 2000. PM:10963655.

[31] Wilson D L, Buckley M., Helliwell C A, and Wilson I W. New normalization methods for cdna microarray data. *Bioinformatics*, 19:1325–1332, December 2002.

[32] Schena M, Shalon D, Davis RW, and Brown PO. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, 1995. PM:7569999.

[33] Simon R M and Dobbin K. Experimental design of cdna microarray experiments. *BioTechniques*, 34:16–21, March 2003.

[34] MathWorld. Bonferroni Correction. WWW, September 8th, 2003. http://mathworld.wolfram.com/BonferroniCorrection.html.

[35] Eisen MB, Spellman PT, Brown PO, and Botstein D. Cluster analysis and display of genome-wide expression. *Proc Natl Acad Sci U S A*, 95(25):467–470, Dec 1998.

[36] Hedge P, Qui R, Abernathy K, Gay C, Dharap S, Gaspard R, Hughes J E, Snesrud E, Lee N, and Quackenbush J. A concise guide to cdna microarray analysis. *BioTechniques*, 29:548–562, September 2000.

[37] Dudoit S, Yang Y H, Callow M J, and Speed T. Statistical methods for identifying differential expressed genes in replicated cdna microarray experiments. *Speed Group Microarray page*, 2002. http://stat-www.berkeley.edu/users/terry/zarray/Html/matt.html.

[38] Krug S. *Don't make me think*. New Riders, 1th edition, 2000.

[39] Sitraka. JClass. WWW, September 4th, 2003. http://java.quest.com/jclass/jclass.shtml.

[40] Sun. Forte4Java. WWW, September 4th, 2003. http://www.sun.com.

[41] Sun. Java. WWW, September 4th, 2003. http://java.sun.com.

[42] Speed T. *Statistical Analysis of Gene Expression Microarray Data*. Chapman and Hall/CRC, 1th edition, 2003.

[43] Speed T and Yang Y H. Direct versus indirect designs for cdna microarray experiments. 2002. http://biostats.snu.ac.kr/seminar/microarray/Speed.pdf.

[44] Pan W, Lin J, and Le CT. How many replicates of arrays are required to detect gene expression changes in microarray experiments? a mixture model approach. *Genome Biology*, 3:research0022–research0022, 2002. PM:12049663.

[45] Cui X and Churchill G A. Statistical tests for differential expression in cdna microarray experiments. *Genome Biology*, 4, March 2003.

[46] Cui X, Kerr M K, and Churchill G A. Data transformations for cdna microarray data. *Jackson Laboratory*, 2002. http://www.jax.org/staff/churchill/labsite/research/expression/Cui-Transform.pdf.

[47] Wang X, Hessner M, Wu Y, Pati N, and Gosh S. Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction. *Bioinformatics*, 19:1341–1347, January 2003.

[48] Yang Y and Speed T. Design issues for cdna microarray experiments. *Nature Reviews, Genetics*, 3:579–588, August 2002.

Appendix A

Labelling Control Genes

A simple strategy to mark control-elements is to add a well defined prefix to the gene-name. Special software-tools can use these information to identify control-spots and make use of it. To distinguish between different types, a list of possible control-elements and appropriate prefixes was considered. This list should be binding if any control-elements are spotted to microarray-slides.

Prefix	**Description**
C_	Control
CN_	negative control
CPG_	pos. control - Genomic DNA
CPH_	pos. control - housekeeping genes
CPS_	pos. control - spike control
CPM_	pos. control - microarray sample pool (MSP)

Table A.1: Table of possible prefixes. A prefix is added to a gene-name.

Appendix B

Submitted Paper

Pieler R., Hackl H., Sanchez-Cabo F., Thallinger G. and Trajanoski Z., ArrayNorm: Comprehensive normalization and analysis of microarray data. Application Note. Submitted to Bioinformatics.

Abstract ArrayNorm is a user-friendly, versatile and platform independent Java application that comprises tools for the normalization and analysis of microarray data. A variety of normalization options were implemented to remove the systematic and random errors in the data, taking into account the design and the particularities of every slide. In addition, ArrayNorm provides a module to statistically identify with significant changes in expression.

Appendix C

GenePix Flagging Criteria

Bad-quality spots on a microarray heavily affects all data analysis steps and the experiment's results. Most important is to mark bad spots to remove them from further analysis.

The GenePix Pro software features a 'Flag feature' dialog box, where multiple criterias can be linked to a boolean query. Here is one example for such a query.

```
[Flags]            =  [Bad]       Or
[Flags]            = [Absent]    Or [Flags]            = [Not
Found] Or
[F635 % Sat.]      >  10           Or
[F532 % Sat.]      >  10           Or
[Sum of Medians]   <  1000  Or [Sum of Means] < 1000 Or
([% > B635+1SD] < 55  Or
([% > B532+1SD] < 55  Or
(([F532 Mean]-[F532 Median])/[F532 Mean])> 0.2 Or (([F635
Mean]-[F635 Median])/[F635 Mean])> 0.2
```

Every single spot on a microarray has to pass these criteria, otherwise it would be marked as ’bad’.

This reference-query was developed by Hubert Hackl and is used for the majority of microarray-experiments in the *TU-Graz Bioinformatics Group* (see [19]).

Appendix D

Usability test

D.1 ArrayNorm - Usability Test

D.1.1 Task

A complete microarray experiment

- 3 experiment classes
- each containing one dyeswap-pair (6 slides overall)

should be loaded into the program. All possible plots should be opened, one will be captured as picture. After background-correction, normalization with dyeswap-pairs should be applied to the experiment. The effects of normalization can be illustrated by rerunning of the plots, especially boxplots. After finalizing and replicate handling, DE genes should be found by simple foldchange detection. Export of the generated gene list to a textfile.
This usability test is not for testing the user (you). It just helps the developer to detect and fix bugs or insufficiency.

D.1.2 The test-run

Please try to do all following steps and rate them.

- First impression of the GUI
- Loading of GPR-files
- Definition of experimental design
- Data Tree: understanding?
- Opening plots and reports
- Background-Correction of all slides
- Normalizing using dyeswap-pairs
- Finalize and Replicate Handling, log2-transformation
- Exporting of the results-file
- Analysis (find DE genes) using foldchange detection
- Saving the list of DE genes
- Overall impression: usability and functions

D.1.3 Reclamations and suggestions