Analysis of Gene Expression Data **Spring Semester, 2004**

Lecture 4: March 25, 2004

Lecturer: Rani Elkon *Scribe: Erez Yaffe and Dan Cohen*

4.1 Low level analysis of microarrays

4.1.1 Introduction

This course deals with *High level analysis* of data gathered by microarrays. Different types of High level analysis include:

- (Bi-)Clustering
- Reconstruction of transcriptional networks
- Induction of classification rules

All of these high level analysis methods are based on the same *raw data* - A numerical description of the expression level for a number of genes, along a number of experiments. *Low level analysis of microarrays* is the set of methods used to obtain this so called *raw data* from the physical data gathered from the microarray (see Figure 4.1), i.e. luminance measurements for each probe on the array.

The numerical values for expression levels should be extracted from the luminance levels, normalized, and systematic errors should be removed. All of these tasks are handled by Low level analysis of microarrays

4.1.2 Types of microarrays

Currently, 3 types of microarrays are in widespread use:

- Spotted cDNA microarrays
- Spotted oligonucleotide arrays produced by *Agilent*
- GeneChip arrays produced by *Affymetrix*

Each of these microarrays will be now presented.

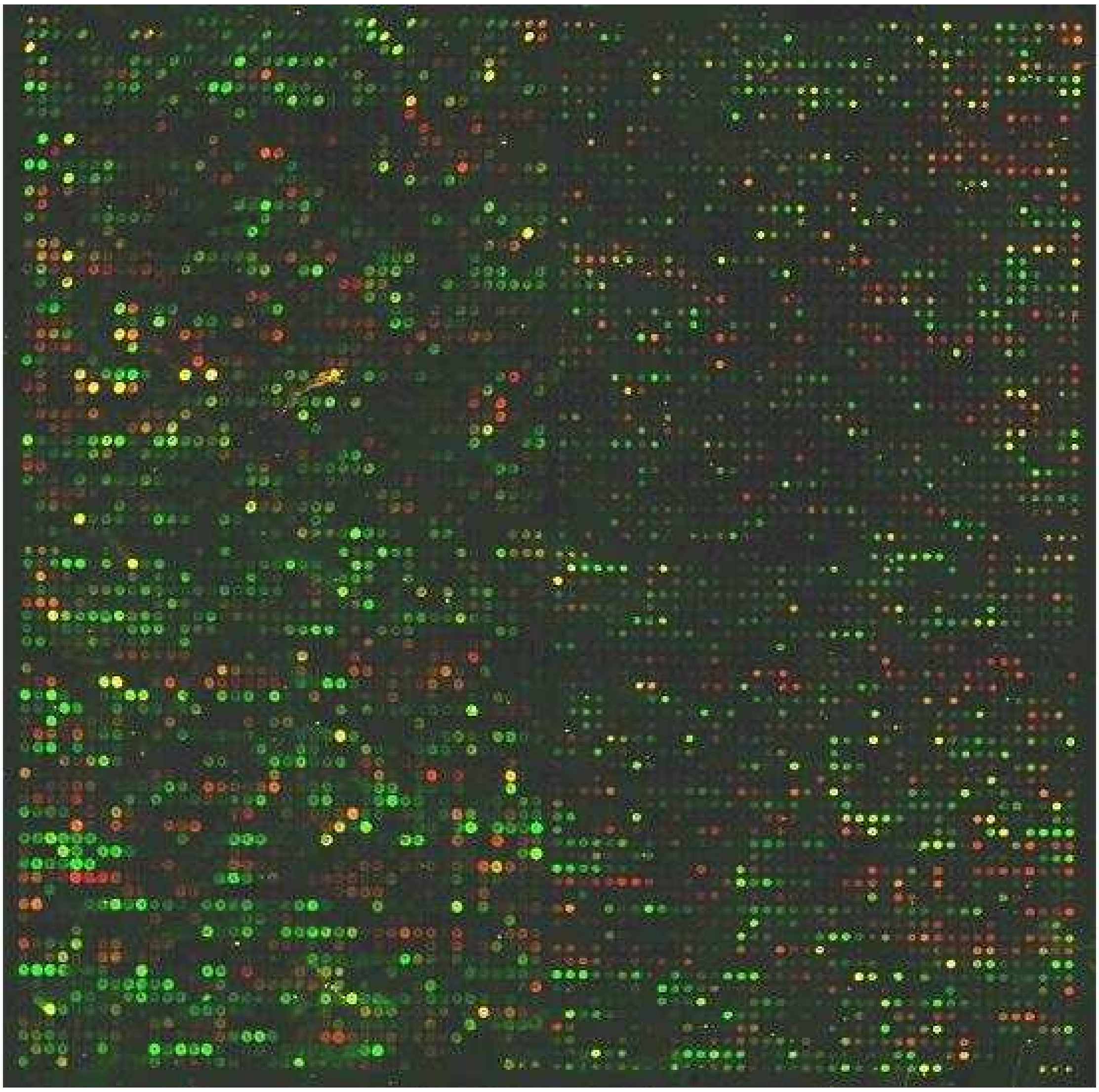


Figure 4.1: Scanned microarray result.

Spotted cDNA microarrays

In a spotted cDNA microarray, each probe is an mRNA sequence or an EST[1] created by the method of PCR. The probes' length is 300-1000 bp. The probes created by PCR are *double stranded* and heat or an alkalying agent is used to separate them .The probes are placed on the chip using a *spotter*, which is a mechanic head that touches test tubes containing the probes and then touches the microarray, placing the probes on it.

The Spotted cDNA microarray suffers from the fact that not all of the probes are single stranded (there is no way to know how many of the probes were separated) thus hindering hybridization, and very long, thus causing *cross-hybridization* in which a target binds with a probe, even though it doesn't not match it completely. On the other hand, this is a relatively cheap method to create microarrays and used primarily by research facilities to create their own chips. About 50 percent of the microarrays used nowadays are *Spotted cDNA microarrays.*

Spotted oligonucleotide arrays

Spotted oligonucleotide arrays are created by Agilent and use synthetic oligonucleotides as probes. Each probe is 60-70 bases long and placed on the chip using an inkjet printer (Agilent uses HP for the printing technology). Using synthetic oligonucleotides means that the probes are single stranded, with known sequence, thus allowing better hybridization and less cross-hybridization. On the other hand, this method is much more expensive.

The probes' sequences are chosen according to the purpose of the chip, i.e. the genes it is supposed to detect. There are a number of available types of chips:

- Human
  - **Whole human genome microarray**
  - **19k well characterized genes (1A)**
  - **19k ESTs and predicted genes (1B)**
- Mice - 36K probes representing over 20k genes
- Other organisms - Rat, Arabidopsis, rice, yeast

About 5 percent of the microarrays used nowadays are Spotted oligonucleotide arrays.

[1] ESTs are mRNA sequences that form a fraction of a gene's sequence

Affymetrix GeneChip arrays

Affymetrix microarrays are currently the most commonly used commercial microarrays. In these microarrays, for each gene that a microarray is intended to detect, a number of probes(11-20) called *positive match probes (PM)* are set. These probes are about 25 bp long, matching different positions along the gene. Furthermore, for each such probe, another probe, a *mismatch probe (MM)* is placed on the microarray (see Figure 4.2). This mismatch probe is identical to the correct probe with exception of the base located in the middle. The mismatch probe is used to detect cross-hybridization, in which case the probe and its mismatch probe will both bind to the target. When hybridization occurs only for the correct probe, and not its mismatch probe, we know that this is true hybridization.

Lets us assume for example that the sequence of the gene to detect is

ATGC**TGATC**GATGCAGAATCGATC

one possible (yet short) probe will be TGATC. The chips will contain this probe and also TGTTC. The possible hybridization results will be analyzed as follows:

- Both probes are detected - cross-hybridization or non specific binding has occurred. this probe won't provide any useful information.
- Only the correct probe is detected - the wanted gene is present. Of course, in a real experiment one would require all of the correct probes (or at least most of them) to be detected in order to decide that the gene is present.
- Neither probe was detected - the wanted gene probably isn't present.

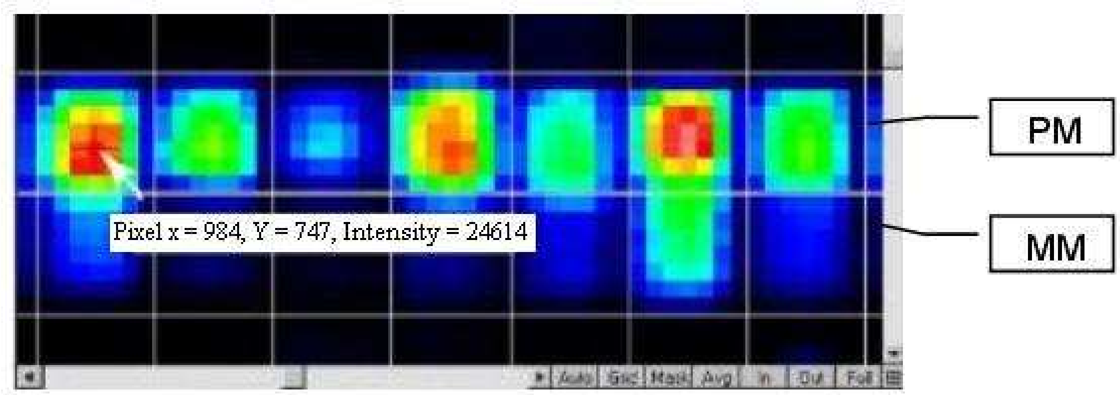


Figure 4.2: probe intensity for Affymetrix

As in Agilent chips, there are tailor made chips for a number of organisms:

- Human
  - **Human Genome U133 plus 2 - 47,000 probe sets for known genes and EST transcripts**
  - **Human Genome Focus Array: 8,500 well annotated genes**
- Other organisms - Rat, Drosophilae, C. elegans, Arabidopsis, Yeast, Zebra Fish, E. Coli

4.1.3 Image analysis

The first step in low level analysis of a microarray is *image analysis*, a process in which the raw visual data of observed illumination intensities is transformed into an estimate for gene expression levels (for each gene). This step is mostly composed of image processing tasks.

Grid alignment

Before extracting the intensity for each probe, one has to locate it. In order to locate the probes one has to superimpose a grid on the scanned intensity levels picture, thus finding the border of each probe. Unfortunately there are many error factors (e.g. movement of the scanner during the scan) which make it hard to align a grid with the entire picture. This can be solved by segmenting the picture and aligning each segment to its own grid (see Figure 4.3).

Affymetrix microarrays are always created with E-coli probes along their border. By adding E-coli to the tested sample, one can make sure that these probes will be lighted (i.e. detected) and will help determine the border of the chip, and its grid alignment (see Figure 4.4).

Target detection

Target detection is the process of deciding which pixels in the scanned picture will be used to calculate the intensity of a probe. This task is especially important in Spotted cDNA microarrays in which the spotter creates an uneven spread of each probe's copies, thus creating an uneven intensity measurement for each probe type (see Figure 4.5).

Target intensity extraction

Given all of the relevant pixels for a probe, one needs to compute a numerical value representing the expression level for that probe. This could be the mean intensity value, the median, etc.

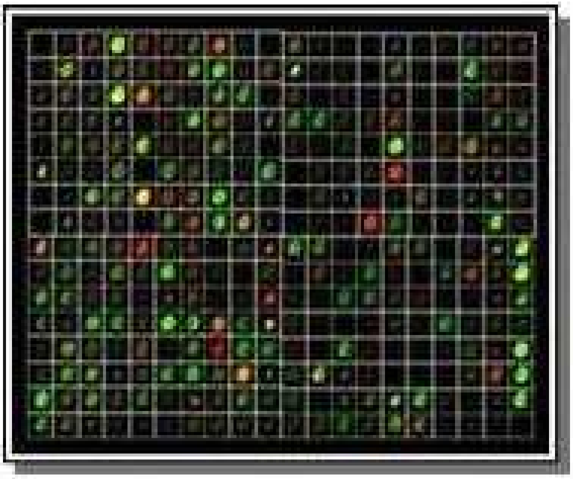


Figure 4.3: general grid alignment. [?]

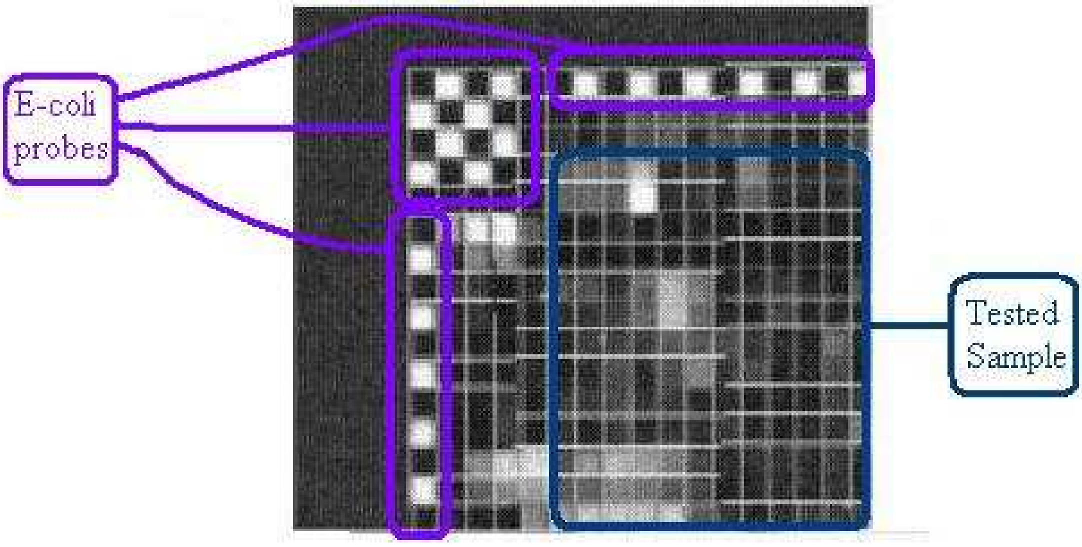


Figure 4.4: Affimetrix chip grid alignment - An example of illuminating the corner and borders of the array.

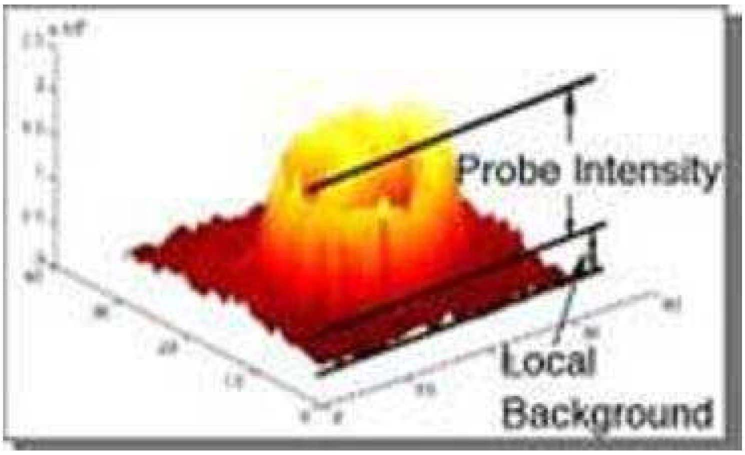


Figure 4.5: Intensity picture for cDNA micro. Notice the crater like distribution of probes. [?]

Local background correction

The intensities measured may be severely biased due to dust, glare and non specific binding. Local background correction is used to crudely correct these biases.

4.1.4 Summation of probe set signals for Affymetrix chips

As was explained before, an Affymetrix chip has a number of probes for each gene it intends to detect. In this step, needed only for Affymetrix chips, one computes a numeric expression estimate for the gene, based on the expressions values for each correct probe (PM) and mismatch probe (MM). There are a number of methods to do this calculation ([?], [?], [?]).

In general we will mark the expression level of probe j for gene i by index $_{ij}$. The expression level for positive probe j will be marked as PM_{ij} The expression level for mismatch probe j will be marked as MM_{ij} The *true* expression level for gene i will be marked θ_i The calculated expression level for gene i will be marked E_i

Average Difference (MAS 4)

This method is based on the idea that the gene expression level is estimated by the difference between the PM and the MM value, with the exception of completely random error :

$$\theta_i + \epsilon_{ij} = PM_{ij} - MM_{ij}$$

Thus, to cancel the noise we should take the mean value for all of the probes :

$$E_i = \frac{\sum(PM_{ij} - MM_{ij})}{T}$$

(T is the number of MM-PM probe couples).

A possible improvement is to ignore outliers - probes with intensities very different from the rest and treat them as measurement errors.

MAS 5

The problem with the MAS4 model is that it assumes ϵ_{ij} has equal distribution so it could be cancelled by a simple mean. It appears that the distribution of errors depends on the general intensity of the probe, and its mean value increases with the probes' expression levels.

One way to reduce this dependency is to use log transformation on the expression values. Thus one should calculate

$$log(E_i) = \frac{\sum(log(PM_{ij} - MM_{ij}))}{T}$$

Again, in order to handle obvious measurement errors, one could give a smaller weight to values far from the mean (in comparison to the values' variance), i.e. use

$$log(E_i) = \frac{\sum(w_j \cdot log(PM_{ij} - MM_{ij}))}{T}$$

When w_j is bigger when PM_{ij}, MM_{ij} are closer to their mean.

dCHIP

the dCHIP method, devised by Li and Wong ([?]), is based on a model in which in addition to random errors each probe has a different affinity to hybridization, i.e. some of the probes for the same gene have stronger affinities and will be more expressed.

$$\alpha_j \cdot \theta_i + \epsilon_{ij} = PM_{ij} - MM_{ij}$$

when α_j is the affinity of probe j to hybridization.

By using a number of gene chips with the same probes, one can use ML (maximum likelihood) estimation to estimate the values of α_j and θ_i

Robust Multi-array Average(RMA)

This method is based on the idea that the MM values contain no additional information about the gene expression level. The reason is that it appears that MM values are very strongly dependant on PM values (see Figure 4.6), and cannot be used to improve results based solely on PM values. Thus the model used is the same as in MAS 5 and dCHIP combined but ignores MM :

$$log(\alpha_j \cdot \theta_i) + \epsilon_{ij} = log(PM_{ij})$$

Again θ_i is estimated by using linear fitting.

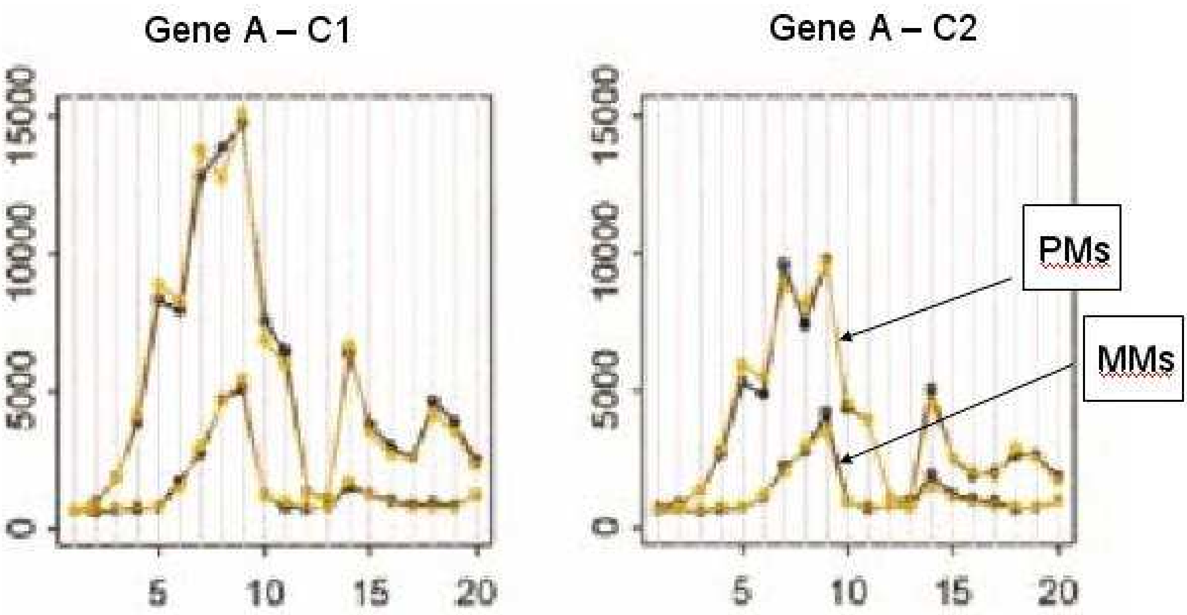


Figure 4.6: Affymetrix probes affinity effect

To compare the effect of different methods, a controled test was performed. 11 known RNAs were added to a test sample in known concentrations (which were much higher than the concentrations of native sequences in the sample). The expression levels of these RNA samples were calculated using each of the methods and the results were compared to the correct values. Based on these tests, it appears that RMA is the best among the presented methods.

4.1.5 Normalization

The normalization step is intended to deal with the fact that the results from identical experiments on two identical microarrays will never be exactly the same. In addition to unavoidable random errors (see Figure 4.7) there are also systematic differences (see Figure 4.8) caused by:

- Different efficiencies of dyes. for example, green colored markers are stronger then red ones (measured as stronger illumination) thus creating a bias between experiments done with green and red markers.
- experimental differences (whether by mistake or because of differing experimental protocols) will lead to different amounts of mRNA in the tested sample, causing different expression levels. This problem is especially important when comparing data gathered in different laboratories.
- Different scanning parameters
- Differences between chips created in different production batches.

these differences can be corrected by the use of *normalization* methods which are the process of removing systematic errors (biases) from the data. Without correcting these differences, it is be impossible to compare the results of two experiments.

In the following graphs, the gene expression levels will be presented as a histogram of $log(intensity)$ values. The results from two chips (or two tests of the same sample with differing markers) will be colored in red and green. for example see Figure 4.9

Notice that even though a comparison of identical samples is used in the examples, normalization is important when comparing different samples in order to detect differential genes. In such cases it is harder to normalize the results because one cannot know whether the different expression levels are caused due to actual differences or a normalization problem.

A normalization scheme should answer two questions:

- Which genes (probes) are used for the normalization process
- How is the normalization performed, i.e. what is the mathematical algorithm used to normalize the values.

There are a number of methods for choosing the normalization genes, i.e. those genes on which the normalization scheme will be based.

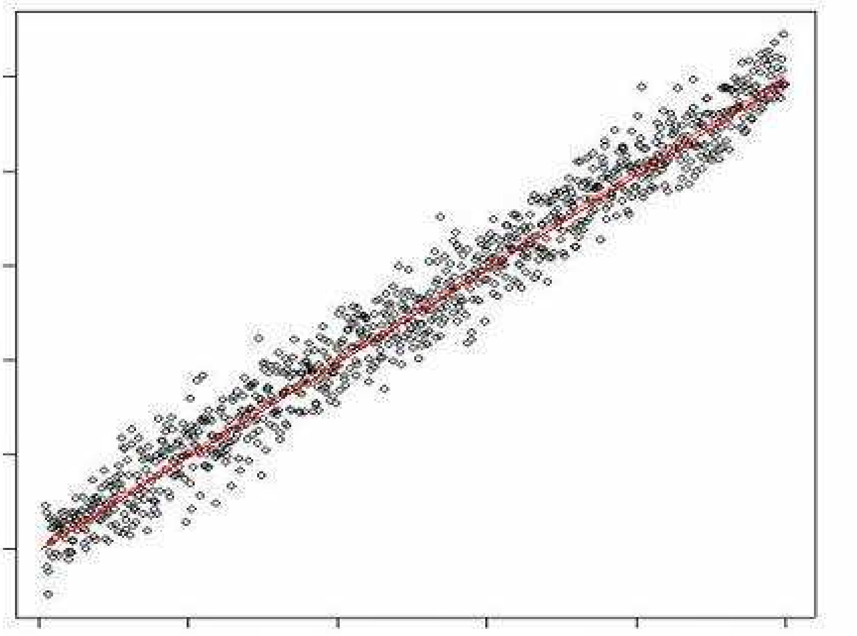


Figure 4.7: expected results with noise.

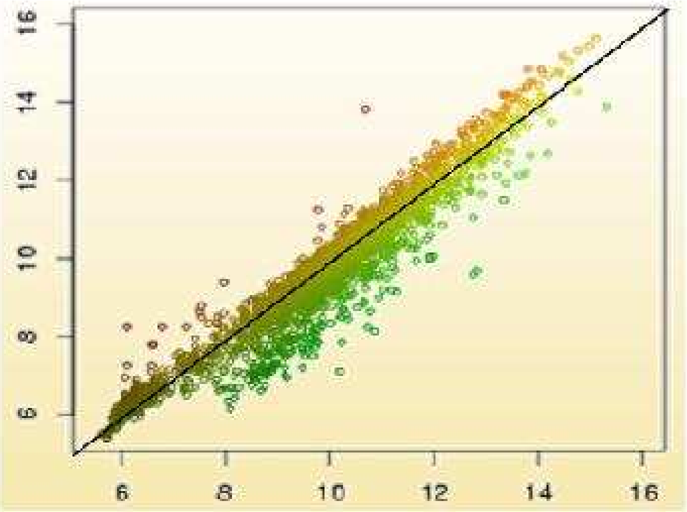


Figure 4.8: expected results with systematic bias.

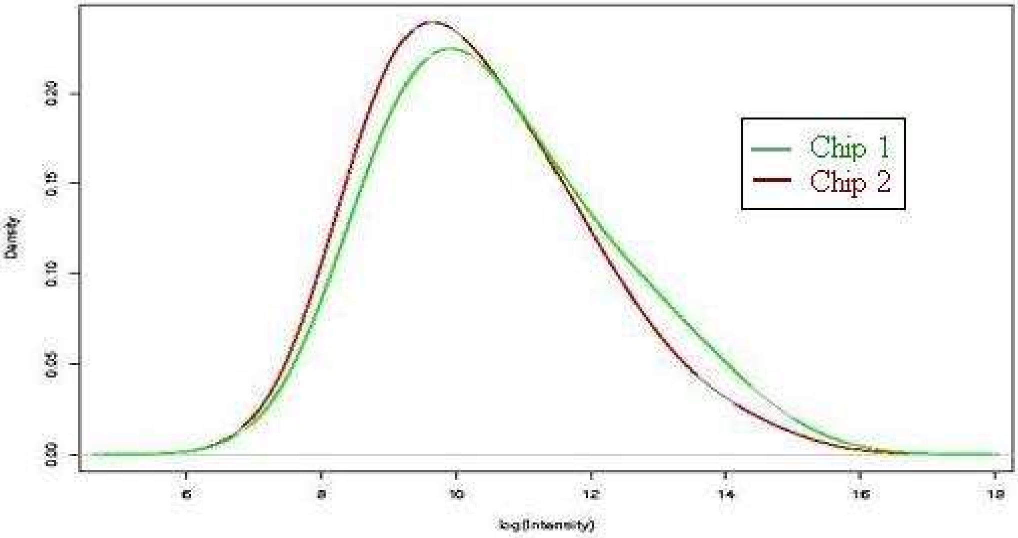


Figure 4.9: intensity histograms with bias.

All gene normalization

Using all of the genes on the chips for normalization is based on the assumption that most of the genes have the same expression levels in the two (different) samples which are compared. The proportion of the differential genes is low (less than 20 percent). Thus we can assume that by using all of the genes, we will have a large number of equally expressed genes for the normalization and achieve good results.

This method cannot be used when the previous assumption is wrong. for example, when the samples are highly heterogeneous (e.g. samples from completely different tissues).

Housekeeping genes only

The idea is to use a small set of genes that, based on prior knowledge, are known to have equal expression levels in the compared samples. two currently used normalization schemes are based on housekeeping genes:

- Affymetrix chips have a set of 100 housekeeping genes used for normalization
- NHGRI's cDNA microarrays have a set of 70 housekeeping genes

One problem with using housekeeping genes is that they are usually expressed at high levels, so they are not informative for the normalization of the low intensities range.

Spiked in controls

In the *spiked in controls* method, a number of control mRNAs are added to each sample. These mRNAs are taken from another organism (as to make sure that they do not exist in the sample itself). The microarrays are designed to have probes that detect these mRNAs. The controls are added in a range of concentrations thus providing normalization data for different expression levels.

This method's main limitation is that due to the fact that the controls are added only to the final sample, they cannot compensate for differences caused during its preparation. Only differences in the scanning and image analysis steps can be compensated. Imagine two samples that were produced with different amounts of genes due to some experimental error. later, the controls are added in equal amounts, so they can provide no clue on the initial difference. One should remember that sample preparation is probably the most common cause for biases, rendering this method much less effective. Furthermore, spikes normalization is based on small (70-100) number of probes so it isn't as robust as the other methods.

Invariant set

Contrary to the other methods, in the *Invariant set* method, one decides on the normalization genes only after the results are analyzed. The idea is to detect genes with similar expression levels in all of the chips, assume they should have identical expression level and base the normalization scheme on them. One way to detect these genes is by ranking the expression levels for all of the genes and choose genes with the same rank (global biases should have less effect on the comparative rank of each gene).

Once the normalization genes were chosen, there are a number of methods for the normalization itself. One should remember that all of these methods are always computed based on the expression levels of the normalization genes, and later the transformation is applied to the entire data set.

Global normalization

this normalization scheme is intended to equalize the mean value of expression levels. all of the values are multiplied by a constant which is the ratio between the mean expression level of the normalization genes in the two samples. The normalization factor k is

$$k = \frac{\sum(E_i^1)}{\sum(E_i^2)}$$

when the summation is other normalization genes. (E_i^j is the expression level for gene i in sample j). Normalization of E_i^2 values is done by multiplication by k.

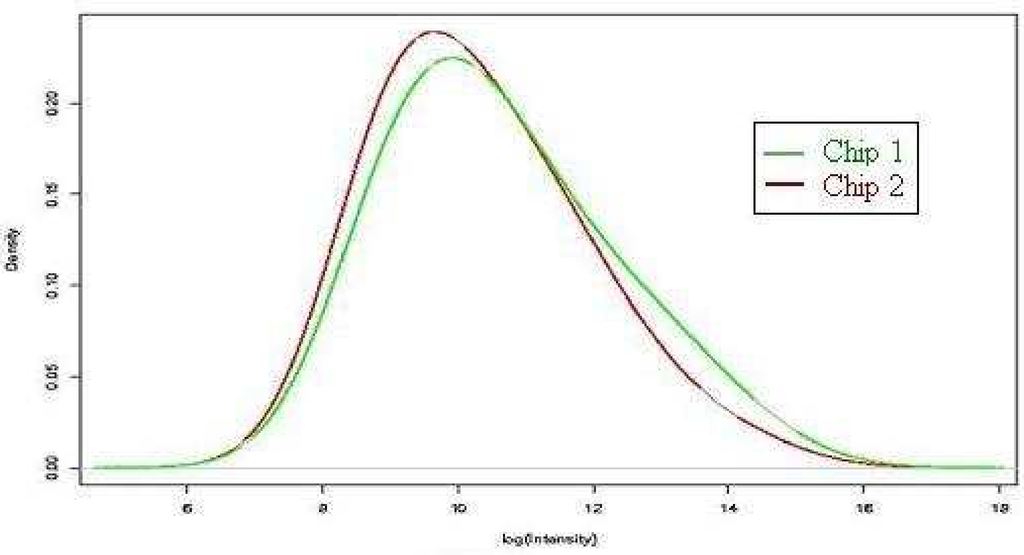


Figure 4.10: Histogram before normalization.

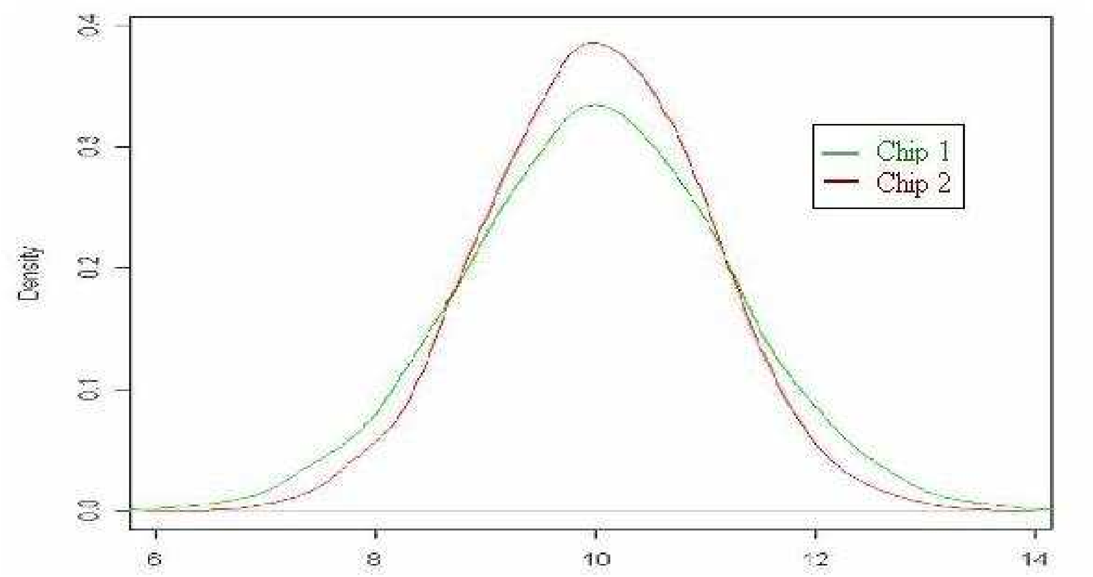


Figure 4.11: Histogram after normalization.

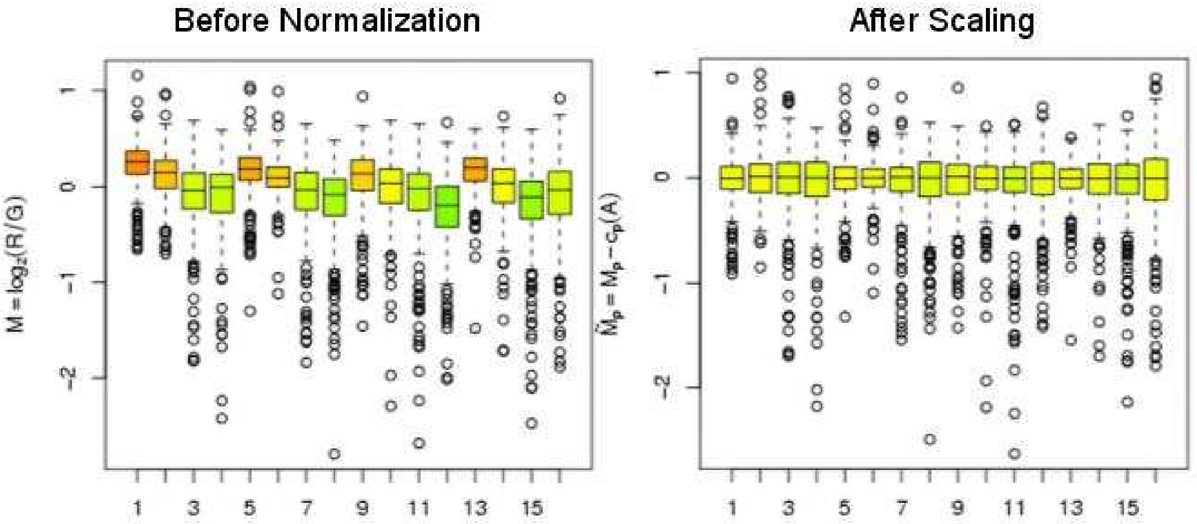


Figure 4.12: boxplots (see appendix 1) before and after global normalization.

Loess normalization - local linear fit

Loess normalization([?], [?]) is based on the (true) assumption that the biases are intensity dependent, thus there is no one normalization factor that can remove the biases for higher and lower expression genes. One should normalize different expression level genes with different factors.

Before tackling the Loess normalization, it is important to be familiar with the $MvsA$ plots which help detect intensity dependent biases. The X axis is the average intensity of a gene in both samples(chips):

$$\frac{E_i^1 \cdot E_i^2}{2}$$

The Y axis is the log ratio of these intensities:

$$log(\frac{E_i^1}{E_i^2})$$

for example, figure 4.12 shows a situation in which there is no intensity dependent bias (the ratio between expression values(Y axis) does not change according to the expression levels themselves(X axis))

On the other hand, figure 4.13 shows a situation in which the ratio between expression levels changes completely for different expression levels. For lower expression levels one of the chips' values are measured to be higher than the other's, and this situation is reversed

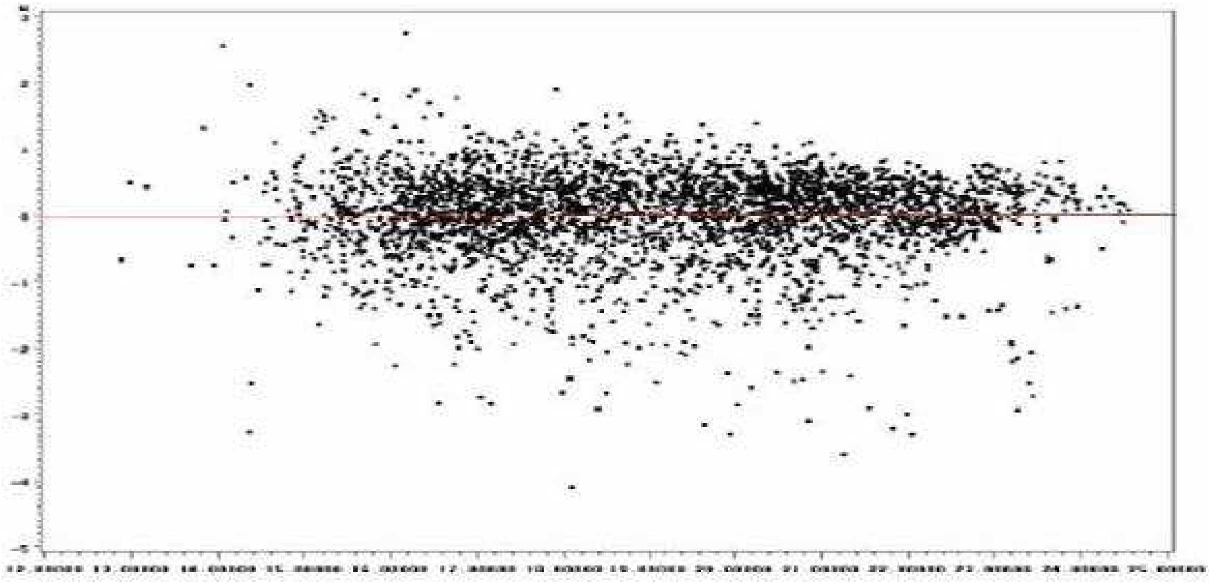


Figure 4.13: M vs A with no bias.

for higher expression values. It is obvious that this situation cannot be corrected by global normalization.

Loess normalization fits a local regression curve to the M vs A graph and uses it to calculate a normalization factor that depends on the mean intensity. the normalization is performed by multiplying each gene expression level by the factor fitting its expression level, thus normalizing figure 4.13 to figure 4.12

Quantile normalization

Contrary to the other normalization methods which tried to equalize the mean expression level between chips (global or per expression level), *Quantile normalization* forces the chips to have identical intensity distributions. (see Figure 4.14 and Figure 4.15)

The idea is to make sure that both chips will have the same intensity distribution histogram (Of course, it doesn't promise that the same **genes** will have the same intensities, only the same distribution of intensities).

Quantile normalization is done by sorting the gene expression levels. let E_i^j be the expression level of gene i in chip j. After sorting, let $\hat{E}_k^j$ be the k-th largest expression level for chip j. Of course, this is the expression level of gene i for some i : $\hat{E}_k^j = E_i^j$. We now

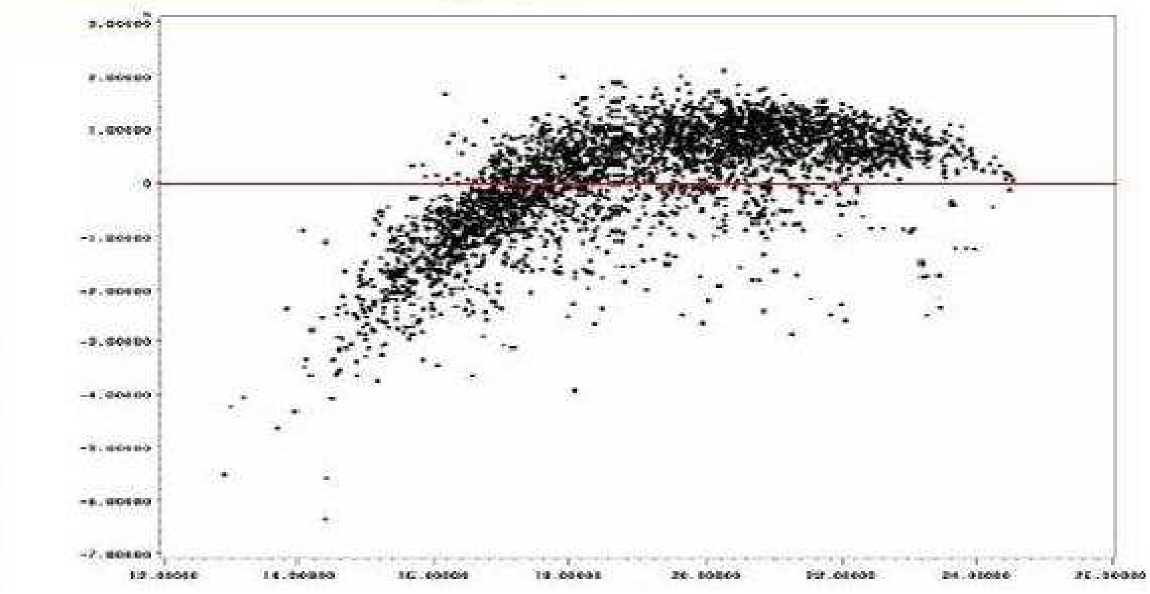


Figure 4.14: M vs A with bias.

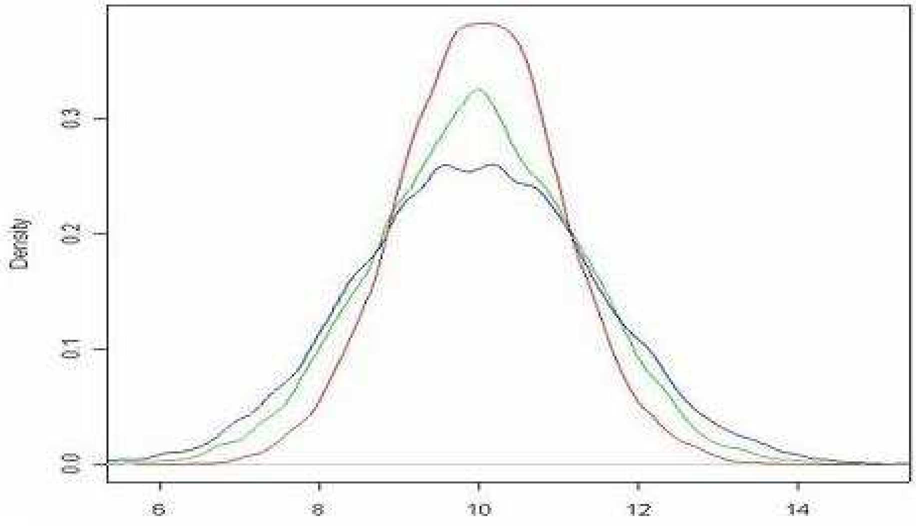


Figure 4.15: After loess normalization.

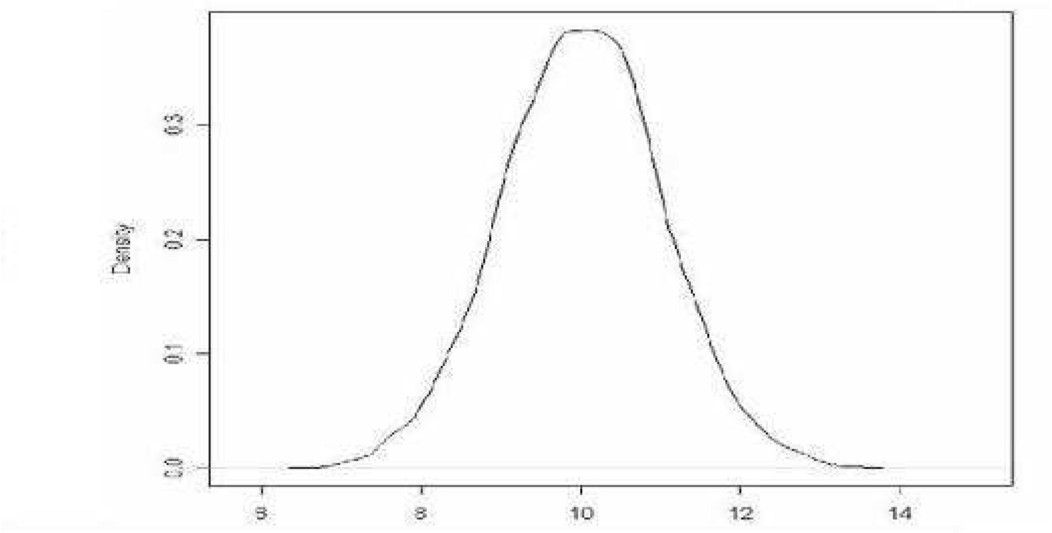


Figure 4.16: After quantile normalization.

compute the median intensity for each rank:

$$\alpha_k = \frac{\sum \hat{E}_k^j}{T}$$

Now we should normalize by replacing each gene i with expression level $\hat{E_k^j}$ with this median. In this way we make sure that for each rank k, we will have an expression level on each chip with the same value, thus the chips will have the same expression level distribution.

Summary

It appears that Quantile normalization is the best normalization method. Loess has comparable results, but Global normalization is not satisfactory([?] [?]). There are a number of normalization tools available:

- BioConductor can be used on both Affymetrix and cDNA microarrays
- dCHIP can be used only for Affymetrix and is based on Quantile normalization, using the Invariant set method to choose normalization genes
- Expander can be used on both Affymetrix and cDNA microarrays and can use both Quantile normalization and Loess.

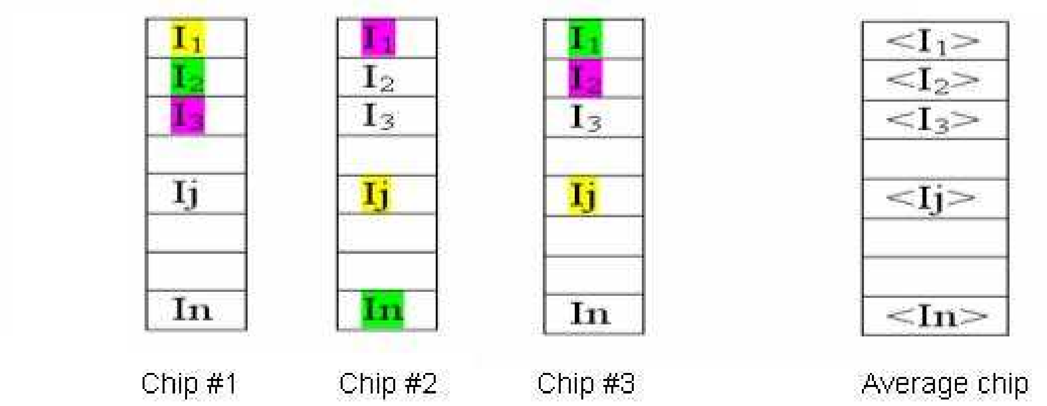


Figure 4.17: Quantile normalization. Each color is a specific gene, I_i denotes the ranked intensity of a gene.

4.2 Identification of Differential Genes

The most common microarray experiment is a comparison between 2 samples - a treatment sample and a control sample. The goal to identify genes that are differently expressed in the two samples. The number of microarrays is usually very low (2-4). There are a number of methods to identify the differently expressed genes.

An important perquisite of these methods is the ability to asses the chance of *false positives*[2]. Without it, it is impossible to know whether the results of the experiment are reliable.

4.2.1 Fold change

In this method, all of the genes with expression level change (between treatment and control samples) of more then a given percentage (e.g. 100 percent) are treated as differential genes. This naive method has a number of major limitations:

- no estimation is given for the chance of false positives

- this method is biased toward lower expression genes, because for those genes, even a small change due to an error could be enough to mark them as differential. (see Figure 4.17) This situation could be improved by using a cutoff to filter genes with a too low expression level.

[2] The chance that a gene will be detected as differential even though it isn't one

- there is no consideration of the variability of gene expression levels over a number of microarrays. For example, it is enough for one treatment microarray to show a very high expression level for a gene, for this gene to be marked as differential. yet, in other treatment microarrays this genes might have low expression level, possibly of some other biological phenomena in the specific sample analyzed by the first microarray. (see Figure 4.18 for example)

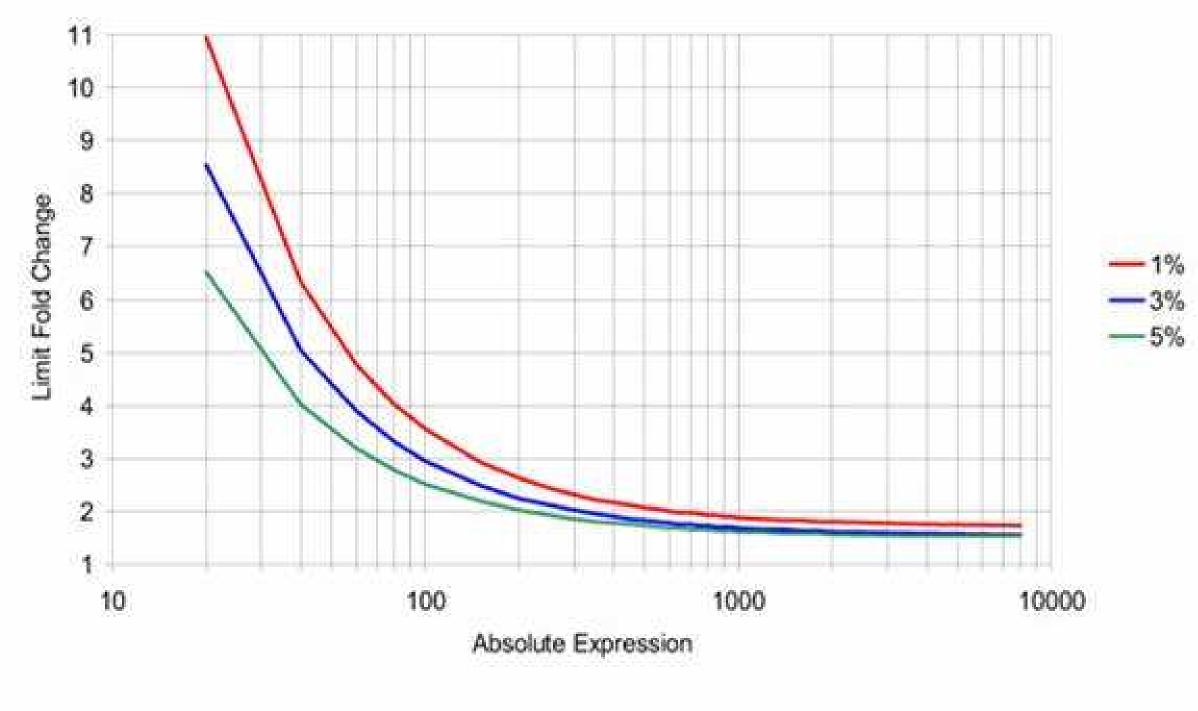
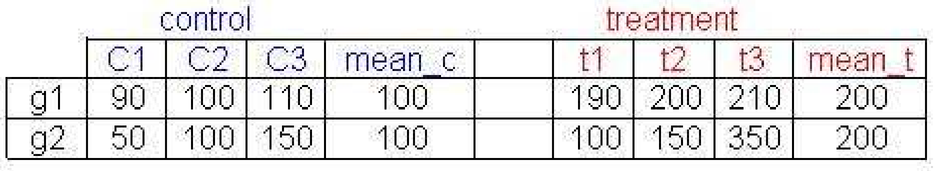


Figure 4.18: Fold Change limit, Biased to low expression levels. Notice that for small values, a large fold change occurs even for a small change

4.2.2 T-test

The T-test is based on normalizing the expression level change, with the variance of the mean expression levels (of the treatment and control samples). In case the expression level change is large in comparison with the variance of the mean expression values, one can assume there is a real difference in gene concentration, i.e. the gene is differential. On the other hand, even if the difference is large, but the gene has high variance, we will not treat it as differential.



	control					treatment			
	C1	C2	C3	mean_c		t1	t2	t3	mean_t
g1	90	100	110	100		190	200	210	200
g2	50	100	150	100		100	150	350	200

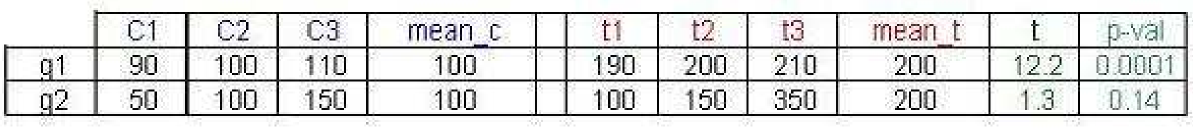
Figure 4.19: Fold Change limit, variability over replicates is ignored. Notice that cases with low and high variance get the same score

The *t-score* value is computed the following way:

$$t = \frac{M_t - M_c}{\sqrt{\frac{S_c^2}{n_c} + \frac{S_t^2}{n_t}}}$$

while S_c^2, S_t^2 are the variance estimates in control and treatment samples respectively. M_c, M_t are the mean levels in control and treatment samples respectively. n_c, n_t are the number of control and treatment samples respectively.

It is possible to calculate a P-value[3] for each T-score in order to asses the chance for a false positive (the chance is the P-value itself).



	C1	C2	C3	mean_c		t1	t2	t3	mean_t	t	p-val
g1	90	100	110	100		190	200	210	200	12.2	0.0001
g2	50	100	150	100		100	150	350	200	1.3	0.14

Figure 4.20: Example of computation of T-score and P-value when comparing control and treatment.

4.2.3 Bonferroni correction

T-score based methods are problematic when used for microarray analysis. The reason is the known statistical problem of *multiple testing*. When testing for a very large number of cases (genes), one should take into account the fact that the number of false positives is the P-value for the T-value of the cutoff threshold, multiplied by the number of tests(genes). If the P-value is 0.01,for example, and $10,000$ genes are tested, about 100 false positive genes will be detected! This poses a question on the validity of the microarray results.

[3] the chance to have a given T-score in case of a random sample

The Bonferroni correction([?]) states that in order to have a given chance of false positives q, while doing N experiments, one should aim for a P-value that is $\frac{q}{N}$. This follows immediately if one assumes that each test result is independent. For example, given the numbers described above, one should choose P-value of 0.000001 in order to have a chance of 0.01 for **one** false positive.

The problem with the Bonferroni correction is that the T-value required for such a low P-value will most probably limit the number of true positives found. In summary, using the Bonferroni correction promises a low chance for false positives but also may cause a large number of false negatives (differential genes that would be filtered because of the high T-value threshold).

4.2.4 False Discovery Rate

The idea behind *false discovery rate (FDR)*([?], [?]) is to choose an acceptable proportion of false positives among the genes declared as differential, for example 10 percent (this percentage will be marked q). The FDR method is to rank the tested genes according to their P-values and choose as differential genes, only the first k genes, those with the lowest P-value so that

$$p_i \leq i * \frac{q}{N}$$

so we will guarantee that the false positives amount is not exceeded.

The problem with FDR is that it, like the rest of the presented methods, assumes that the gene expression of different genes on the chip is independant. This is biologicaly incorrect - many genes' expressions are correlated.

4.2.5 Significance Analysis of Microarray

Significance Analysis of Microarray (SAM)([?]) is intended to deal with the fact that gene expressions are correlated in an unknown manner. The idea is to use permutations to get an 'empirical' estimate for the FDR of the reported differential genes. Instead of using the above FDR calculation, one tries to *rename* the different genes and recalculate, in order to find out the real correlation.

The SAM algorithm is :

- Compute for each gene a statistic that measures its relative expression difference in control vs 'treatment' (T-score or a variant)
- Rank the genes according to their 'difference score'

- Set a cut off d_0 and consider all genes above it as differential. the number of differential genes is N_d.
- Permute the condition labels, and count how many genes got score above d_0. the number of genes is N_p
- Repeat on many (all possible) permutations and count N_{pj}
- estimate FDR as the proportion: $\frac{<N_{pj}>}{N_d}$

4.3 Appendix

4.3.1 Boxplots

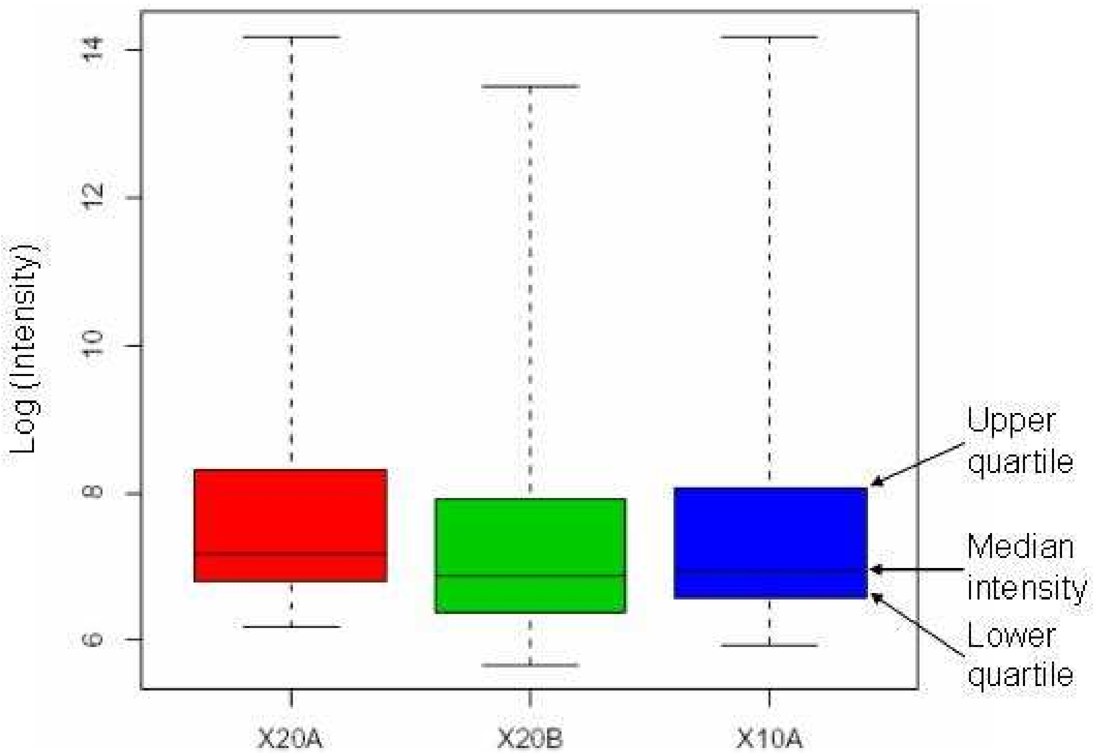


Figure 4.21: Explanation of boxplots diagrams

Boxplots are method for graphical representation of a distribution, based on representing the different quartiles. The range is divided by five values (as shown in Figure 4.21):

- The upper line indicates the maximal value.
- The upper line in the colored box indicates the upper quartile of the values.
- The middle line in the colored box indicates the median.
- the lower line in the colored box indicates the lower quartile of the values.
- The lower line indicates the minimal value.

The five number summary leads to a graphical representation of a distribution called the boxplot.