DSC 2003 Working Papers
(Draft Versions)
http://www.ci.tuwien.ac.at/Conferences/DSC-2003/

A Graphical Users Interface to Normalize Microarray Data

Fatima Sanchez Cabo[‡§], Zlatko Trajanoski[§], Kwang-Hyun Cho[*] and Olaf Wolkenhauer[‡†]

[‡]Department of Biomolecular Sciences
[†]Department of Electrical Engineering and Electronics, Control Systems Centre
UMIST, P.O. Box 88, Manchester M60 1QD, U.K.

[§]Institute of Biomedical Engineering
Graz University of Technology
Krenngasse 37, 8010 Graz, Austria

[*] School of Electrical Engineering, University of Ulsan
Ulsan, 680-749, South Korea

Abstract

Microarray technology is becoming an essential tool in functional genomics. The possibility of monitoring the expression level of thousands of genes simultaneously, as the response to a particular biological condition, gives to the biologists the chance to widen the aims of their experiments and opens a door to the understanding of cellular transcription processes. In order to extract valuable information from the big amount of data that microarrays experiments generate, suitable and powerful statistical and computational methods are required. An example of the effort of statisticians and computer scientists is the release of the first Bioconductor software and the increasing number of functions for microarray data analysis implemented

in several programming languages (e.g. R, MATLAB, Java) by different research teams all around the world.

In this paper, we describe a Graphical Users Interface (GUI) written in MATLAB to deal with the normalization of microarray data. In our opinion, not enough importance has been given yet either to the assessment of the effect of the normalization on the data or to the study of the most suitable normalization methods according to the experimental design. To aim these objectives, a great variety of normalization methods were implemented in the interface here described, allowing the user to visualize the data before and after every step of the normalization process. Our interface suggests an example of what should be done using also other softwares such as R.

The features implemented in this interface were validated using data sets from microarray experiments carried out for *Mycobacterium tuberculosis* by the Bacterial Microarray Group St.George's Hospital, Medical School in London and for *Streptomyces coelicolour* by the Streptomyces group at UMIST.

Introduction

Two color microarrays measure the relative abundance of messenger RNA (mRNA) of thousands of genes in two different samples. To obtain an estimator of the mRNA abundance, the two pools of mRNA from the cell populations to be studied are reversed transcribed to complementary DNA (cDNA) and labelled using two different fluorescent dyes (usually cyanine dyes Cy3 and Cy5), as described in [4] and [11]. The two pools are then combined and applied to the microarray itself, where products of the polymerase chain reaction (PCR) generated from cDNA libraries or clone collections were printed as spots at defined locations. Labelled cDNA or genomic DNA (gDNA) in the pools hybridize to complementary sequences on the array and unhybridized DNA is washed off. The slide is then scanned using two different wavelengths and the intensity of the same spot in both channels is compared. This results in a measurement of the ratio of transcript levels for each gene represented on the array.

The statistical analysis starts with the scanning file itself. Different location parameters for the distribution of the pixels in a particular spot are given (e.g., mean, median, mode, standard deviation) and the most suitable one to explain the intensity value of a given spot in both channels should be chosen. The scan file gives the location parameters for both channels foreground intensities (Cy3 and Cy5) and for their background. The background intensity measures the intensity of the mRNA that binds to the slide even if there is no material spotted. Using all this information the next step is to filter those spots with bad quality that should not be used for further analysis. Yet, before proper analysis of the data we need to normalize the data in order to remove all the non-biological variation introduced by the experimental process and to enable the comparison of the intensity values within and across slides.

There are many different methods to normalize microarray data. Some of them such as ANOVA [7, 8] or SVD [1] remove in one step all the non-biological variation. However, they can be considered by biologists as "black-boxes". In consequence, it can be of a greater interest the use of a sequential method, allowing the user to choose different options at different stages of the normalization process, according to the particularities of the experiment. The interface here described normalizes

Table 1: General approach to the sequential process. All the options were implemented in our toolbox in order to enable the user to choose the most suitable one according to its experimental design.

NORMALISATION IN MADE (MicroArray Data Explorer)		
Effects corrected		***Options***
Background effect		1. Background subtraction 2. No subtraction
Spatial effect		$p_i = \frac{r_i}{g_i}$
Dye effect	*Using all genes*	1. Global constant 2. Linear regression 3. LOWESS function 4. LOWESS for print-tips
	Quality control elements	1.Dye-swap normalization 2.Use of spotted controls
Array effect: Across replicates normalization		
Average experimental replicates (slides/spots)		
Array effect: Across samples normalization		1. Against all arrays 2. Against arrays in J
Transformation of the data		1.$\log_2(\bullet)$ transformation 2. $\sqrt{\bullet}$ transformation 3. lin-$\log_2$ transformation 4. arsinh($\bullet$) transformation

the data in this way. Table 1 is an example of the very general sequential method implemented in our toolbox called MADE (MicroArray Data Explorer).

The paper is organized as follows: Firstly, the motivation for normalization of microarray data is explained and the main sources of variability in microarray data are defined. These sources of variability can be introducing in the data biological variability but also random and systematic errors. The main sources of non-biological variation are the background effect, the dye effect and the array effect. The paper describes then the different features that allow the user to visualize and correct every of those effects.

Normalization of microarray data

A proper understanding of the intrinsical errors in a measurement requires a suitable mathematical approach. Errors in a measurement can be of two types: systematic or random.

- *Systematic error or inaccuracy* is a fixed positive or negative error that is the same if the measurement is repeated (systematic error).
- *Random error or imprecision* is a random positive or negative error that varies every time the measurement is made.

In the microarray production process, many systematic and random errors are introduced. These errors are masking the biological variation in which we are interested. Normalization is the process of removing all this non-biological variation. As described in the previous section, microarrays is a tool used to estimate the amount of mRNA for a gene across different conditions. For such an estimation, we rely on two intensity values per spot, one for every channel. In a very simple approach, according to [6], every intensity value can be modelled as:

$$I = N \cdot A + error$$

where A is the abundance of mRNA for the gene in the given sample. N is then the normalization factor that corrects all systematic errors and $error$ summarizes the random error. The objective of the normalization process is to make I a reliable estimator of A. For that it is essential to estimate N and $error$. According to [7] and [8], there are four main sources of variation in microarray data. Some of them introduce non-biological variability (contributing to N and $error$) and should be removed in order to understand the real biological variation. The four main sources of variation are:

- Dye effect. The different incorporation properties of the dyes and their different physical characteristics make this the most important source of systematic error in two-color microarrays.
- Array effect. The difference in the overall intensity across different arrays can be due to real biological variation from one condition to another or just to some experimental noise.
- Gene effect. The different expression level of a particular gene in a particular array can be due to the biological variability of the gene or to some noise.
- Sample effect. If the overall intensity of the hybridized samples is different, it can be due to some experimental error or to real biological activity. This factor will be important in the choice of the within array normalization method.

  Besides these four factors it must be also considered:
- Background effect. Some part of the probe will attach to the slide even when there is not spotted material, contributing to the foreground intensity. However, it has not been yet provided a reliable estimator for the background intensity.

The increasing number of methods described to correct all the systematic and random bias mostly summarized in the four effects previously described (see [13]) may lead to confusion in the analysis of microarray data. Which method should I use? Should I use all of them? These are typical questions when facing the normalization of a data set. With the aim of helping to answer these questions, we implemented an interface in MATLAB$^{©}$ (Mathworks Inc.)

A normalization interface implemented in MATLAB$^©$

We chose MATLAB to analyze the data from our microarray experiments due, among other reasons, to the variety of representation features that this software provides. However, MATLAB has many limitations in terms of memory and speed and many important statistical tests are not implemented in its library. For example, it is limited in the functions for multivariate analysis of variance, which is becoming an increasingly important tool for the normalization and analysis of microarray data. For all those purposes, programs based on R, such as Bioconductor, or functions written in Java or C++ can be more appropriated. In this direction, the Jackson Laboratory [2] has implemented a number of functions in MATLAB with C++ core functions, improving the efficiency of functions such as LOWESS [3] and implementing factorial designs that were not written as default MATLAB functions. Some ideas about the implementation of GUI interfaces in R has been presented in [12].

One of the main problems in the normalization of microarray data is the organization of all the information. The MATLAB interface described in this paper summarizes the different steps that must be performed in the normalization of microarray data, allowing the user to visualize different plots in order to decide at every stage which is the most suitable option among all those available. A compromise between particularity and generalization must be taken in the normalization process. For this reason, although the methods must be as general as possible to allow its application to all kind of data sets, it is essential to visualize the data to define particularities associated with it.

With our interface, we tried to offer a wide sample of methods to correct the effects previously defined. Among them, all the gene specific errors (e.g. short PCR products) and the spatial noise affect both channels in the same amount. Consequently, these sources of noise are removed just by taking the ratio of both channels. Hence, the interface has three main blocks to correct the background, dye and array effects. For every of them the interface enables:

1. Visualization of different plots for the pre-corrected data in order to choose the most suitable normalization method.

2. A variety of effect-correction options to be chosen.

3. Visualization of the corrected data in order to asses the effect of the method that was chosen.

The data will follow the flow shown in Table 1. Those steps are sequential, although not all are compulsory. In the interface, every block appears when the data has been corrected for the previous effect.

Background correction

Most of the published literature recommends the subtraction of the background intensity from the foreground intensity of every spot. The background intensity is defined as the intensity of the probe that attaches to the array, even when there is no cDNA available, contributing this intensity to the foreground intensity. However, the chemical properties of the array surface are still not completely known. This makes difficult to determine which is the contribution of the background intensity to the measured foreground intensity. The later will be the estimator of the abundance of mRNA for a particular gene in a particular sample.

In order to choose a suitable background correction, the interface allows the visualization of different features:

- Scatter plots. There are three different types of scatter plots that can be used for the analysis.

  - Background against foreground intensities. This plot can help to decide whether the background intensity is additive to the foreground intensity. The background intensities are usually much lower than the foreground intensities. In consequence, there is an unreal idea of linearity, and this would suggest an additive effect of the background to the foreground intensities.
  - Scatter plot of the background of both channels
  - Scatter plot of the foreground of both channels.

  The last two will give some clues to clarify if the relationship in both, foreground and background, is similar. We could then extend conclusions from the background to the foreground and vice-versa.

- 3D plots and contour plots. These plots are useful to study the distribution of the background and foreground intensities across the array and to determine areas where the background intensity is extremely high. As shown in Figure 1 these plots gave us the chance to detect some of the gene effects. Areas around the controls shown a lower overall intensity than the rest of the array. This was due to shorter PCR products than in the rest of the array. Thanks to the possibility of plotting the contour plot of the ratio, we realized how this effect was cancelled.

All these visualization tools help to take a decision about the most suitable background correction to perform in the data set. The interface allows both possibilities: subtraction of the background or not. After choosing one of both possibilities, the interface allows to visualize scatter plots and contour plots of the corrected data set. This can be useful to check the different effect of the background correction in the data. For our data set, background subtraction appeared to pass the noise from the background to the foreground, increasing the noise in the experiment instead of reducing it. It is shown also in [5] how the background subtraction would increase the variability of the data using the $\log_2$ transformation while this variability is not so great if the background is not subtracted. Although background subtraction is currently the most popular approach, we are investigating new methods to use the information about the background intensities to correct the foreground intensities.

Dye correction

After background correction, systematic errors must be corrected. The most important of all of them is the one introduced by the different properties of both fluorescent dyes labelling the two RNA pools. We have detected four properties that are different for both dyes. The most important of them is the lower incorporation rate of $Cy5$, but as well the quantum yield, the photobleatching and the

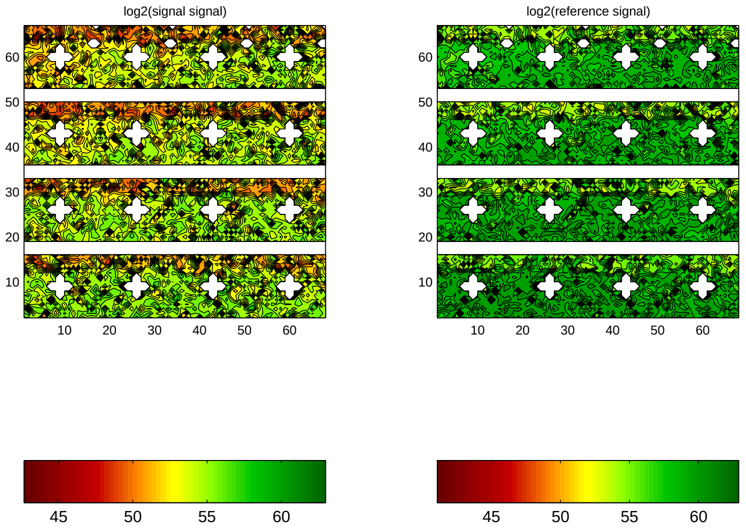


Figure 1: Contour plots for the foreground intensity of green and red channels in a particular microarray. The level curves give an idea of the distribution of the intensities according to its location in the array.

quenching properties are different. All these differences distort the real intensity values of both channels and both channels intensities must be balanced. However, we must be careful at this stage. The most popular dye correction methods are based on the idea that the majority of the genes are equally expressed in both channels. But this is not going to be the case of all experiments. For this reason, two different approaches were implemented in our interface. Using the terms defined in [6], the correction of the dye effect -as well known as within-array normalization- can be performed by:

- using the whole data set to normalize the data (see Figure 2), as well known as self-consistency.
- using the quality control elements provided in the experiment. This includes the dye-swap normalization [9], the use of spotted controls or the use of a reference channel (see Figure 3).

As seen in Figure 2 and Figure 3, both approaches can be selected in our interface.

Dye-effect correction by self-consistency

Assuming that most of the genes are going to be equally expressed in both channels, an expression ξ is estimated to force the overall intensity of both channels to be the same. Both channels intensities would be then related according to the expression:

$$R = \xi \cdot G,$$

where $R \equiv red$ and $G \equiv green$. The estimation of this expression ξ is going to result in many different methods to correct the different properties of the dyes. Four of them can be selected in our interface:

Global normalization In this case, we assume that the systematical bias due to the different dyes properties is affecting all spotted genes in the array in the same amount. A constant k relating both channels is estimated. If most of the genes are expected to be equally expressed, then a good representative value of the distribution of the ratios is:

$$k = med_i \frac{R_i}{G_i}$$

and $\xi = k$. For experiments for which a high percentage of genes is differentially expressed comparing both channels, the use of the first or third quartiles are more suitable options. The three choices are implemented in our interface.

Linear regression normalization In [10] a regression line is fitted to the scatter plot (G,R). Under the assumption that most of the genes should be equally expressed for both channels, the regression line should have a slope one. Hence,

$$R = m \cdot G + n \to \frac{R}{m} - \frac{n}{m} = G \ .$$

From that follows $\xi \simeq m$, where m is the slope of the regression line fitted to the scatter plot and n is the intercept with the ordinate.

LOWESS normalization As suggested in [9, 13], by looking at the (A,M) plot implemented in our interface we can detect if the distribution of the log ratios depends on the intensity. In this case, it is not appropriated to correct every spot in the same amount as the global method does. At the same time, the linear regression method is very sensitive to outliers, so a more robust alternative is required. For these reasons the use of a LOWESS function to correct the dye bias is becoming more important in the normalization of microarray data. (A,M) scatter plot will show:

$$M = \log_2 \frac{S_i}{R_i} \ ,$$

$$A = \frac{1}{2} \cdot (\log_2 S_i + \log_2 R_i) \ .$$

The *LOWESS* function $c(A_i) : \mathrm{I} \mapsto \mathrm{R}$ can be calculated from this plot, where the set of indexes I denotes all genes spotted on the array. The fitting of the *LOWESS* function $c(A)$ from the (A,M) scatterplot leads to:

$$M = \log_2 \left(\frac{R}{G} \right) \cong c(A) \ \Rightarrow \xi = k(A) = 2^{c(A)}.$$

To estimate this function in MATLAB takes extremely long, but we can improve its efficiency using a C function implemented by the Jackson's laboratory (see [2]). LOWESS is computationally efficient also in R.

LOWESS for different print tips During the spotting process, the spots located in the same "grid" are printed by the same print tip. As suggested in [13] different LOWESS functions should be fitted for the different print tip subgroups. In our interface we have implemented the scatter plots that show the genes ordered as in the array to detect print tip effects and correct them if necessary. However, we would expect this effect to cancel with the ratios in two color-microarrays.

Regardless to the method used to estimate ξ, any of those methods corrects the data to get,

$$\frac{R_i}{G_i} \cong 1 \Rightarrow M = \log_2 \frac{R_i}{G_i} \cong 0$$

For this reason, to look at the scatter plots, boxplots and kernel fitted functions before and after the correction is essential.

We must not forget that the interface allows as well the visualization of different probability plots before and after the dye correction. This feature is important in the study of the distribution of both channels intensities and this must be considered if we want to use ANOVA or any other probabilistic framework for further analysis of the data.

Dye-effect correction using the quality elements provided in the experiment

In general, there are many experiments for which the assumption of most genes equally regulated cannot be known "a priori" or for which a very different number of genes is expected to be differentially expressed in both channels. In those case we would rely on the quality control elements. We have implemented two methods:

Dye-swap normalization It was first described in [9]. Given two arrays for which the same material was labelled with a different dye each time, for every spotted gene i the following expressions are considered

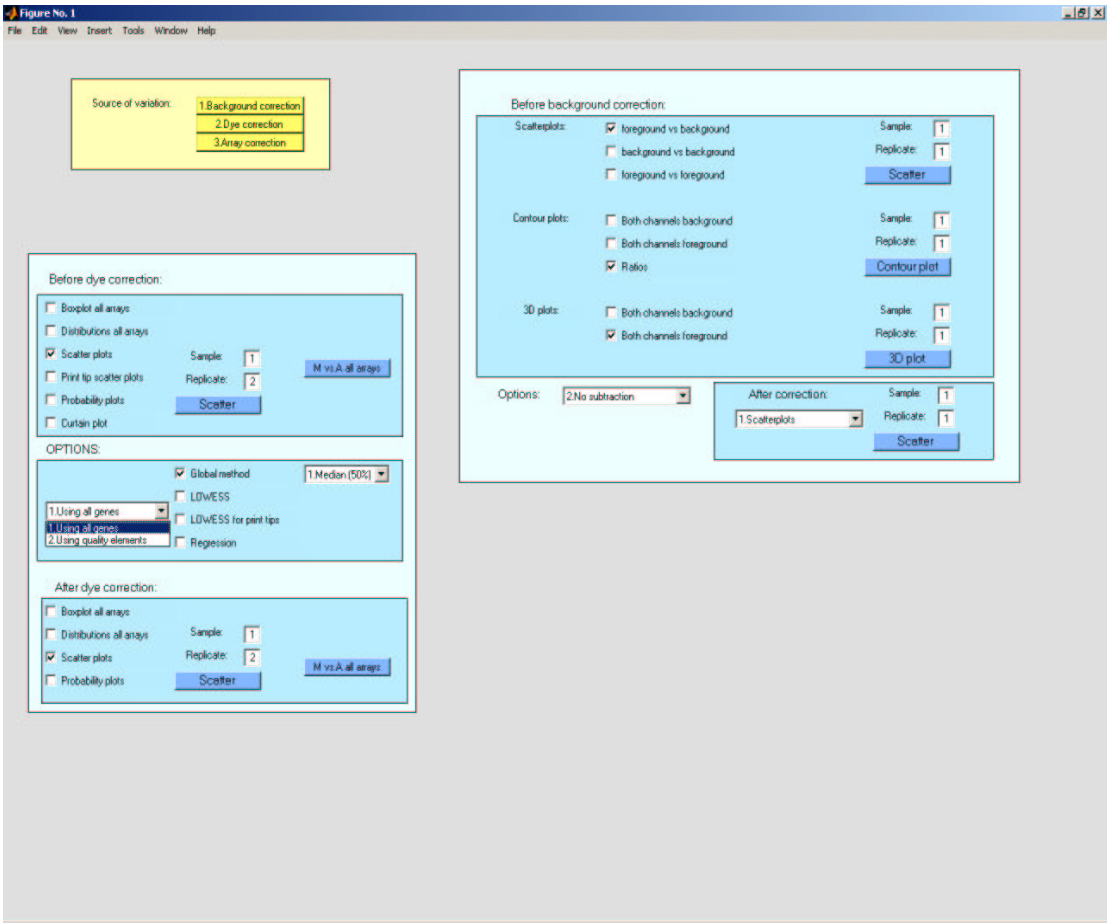
$$M_i = \log_2\left(\frac{R_i}{G_i}\right),$$
$$M'_i = \log_2\left(\frac{R'_i}{G'_i}\right).$$

From these two equations, we obtain

$$M_i = \log_2\left(\frac{R_i}{G_i}\right) = \log_2\left(\frac{s_i}{r_i}\cdot k_i\right) = \log_2\left(\frac{s_i}{r_i}\right) + \log_2 k_i = \log_2\left(\frac{s_i}{r_i}\right) + c_i,$$
$$M'_i = \log_2\left(\frac{R'_i}{G'_i}\right) = \log_2\left(\frac{r_i}{s_i}\cdot k'_i\right) = -\log_2\left(\frac{s_i}{r_i}\right) + \log_2 k'_i = -\log_2\left(\frac{s_i}{r_i}\right) + c'_i,$$

where r_i stands for the intensity of the gene i in sample r and s_i for the same value in sample s. The target is to estimate $\log_2(\frac{s_i}{r_i})$ from M_i, M'_i. Hence, it follows that

$$M_i - c_i = \log_2\left(\frac{s_i}{r_i}\right).$$
$$-M'_i + c'_i = \log_2\left(\frac{s_i}{r_i}\right).$$



Source of variation:

Before dye correction:

□ Boxplot all arrays
□ Distributions all arrays
☑ Scatter plots
□ Print tip scatter plots
□ Probability plots
□ Curtain plot

Sample: 1
Replicate: 2

OPTIONS:

1.Using all genes
1.Using all genes
2.Using quality elements

☑ Global method 1.Median (50%)
□ LOWESS
□ LOWESS for print tips
□ Regression

After dye correction:

□ Boxplot all arrays
□ Distributions all arrays
☑ Scatter plots
□ Probability plots

Sample: 1
Replicate: 2

Before background correction:

Scatterplots:
☑ foreground vs background
□ background vs background
□ foreground vs foreground
Sample: 1
Replicate: 1

Contour plots:
□ Both channels background
□ Both channels foreground
☑ Ratios
Sample: 1
Replicate: 1

3D plots:
□ Both channels background
☑ Both channels foreground
Sample: 1
Replicate: 1

Options: 2.No subtraction

After correction: 1.Scatterplots
Sample: 1
Replicate: 1

Figure 2: Interface view after background correction and chosen the option to correct the dye effect using all the genes. We can see how all the methods outlined in Table 1 to correct the dye effect in that way are implemented in the interface.

For this expression, c_i and c'_i depend on the properties of the dyes. Because they are not suppose to change significatively from one array to another it can be considered $c_i \simeq c'_i$. Adding both equations,

$$M_i - M'_i \simeq 2 \cdot \log_2 \left(\frac{s_i}{r_i} \right) \Longrightarrow \frac{1}{2} \cdot (M_i - M'_i) \simeq \log_2 \left(\frac{s_i}{r_i} \right).$$

The main advantage of the dye-swap normalization is that transforms the data preserving the characteristics of every singular gene. Note that the computational cost for the implementation of this method is very low.

Using the controls If controls covering the whole intensity range are available, we can normalize our data according to them. For controls for which the expression level in both channels is expected to be the same, a non-linear function can be fitted to the (A,M) plot of the controls and used to correct the entire data set. However, because the number of controls available per slide is usually not very large, we do not recommend to fit a LOWESS function but a more general method such as Levenberg-Marquardt. The model used will be in most of the cases a quadratic function.

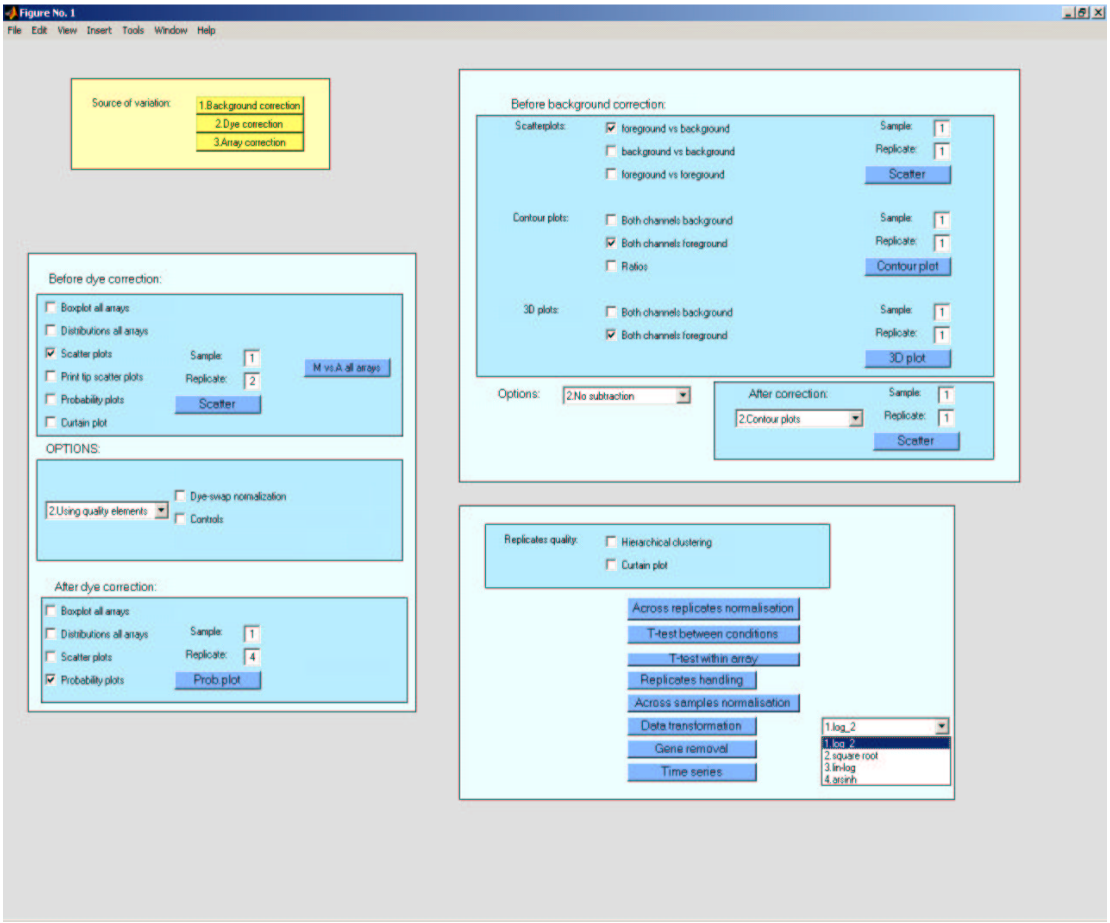


Figure 3: Interface showing the options to correct the dye-effect using the quality control elements. The different options to transform the data are as well shown.

Replicate handling

Besides the systematical error introduced in every measurement, there is an *error* factor corresponding to the random error and that cannot be estimated. The only way to reduce the intrinsical variability of a given measurement is replicating measurements. The repetitions must be carried out in completely independent conditions to warrant the applicability of statistical principles. In our interface, we first consider a feature that tests the quality of the replicates. It is called curtain plot because it appears as a curtain (see Figure 4). Different correlation measurements are allowed (Standard, Pearson and Spearman). The last of them calculates the correlation in terms of the shape of gene profiles. A percentage of the genes which profiles are correlated for the different replicates is calculated. After the dye correction, the effect can be checked as well in the replicates. Besides the curtain plot described before, hierarchical clustering on the replicates can be visualized.

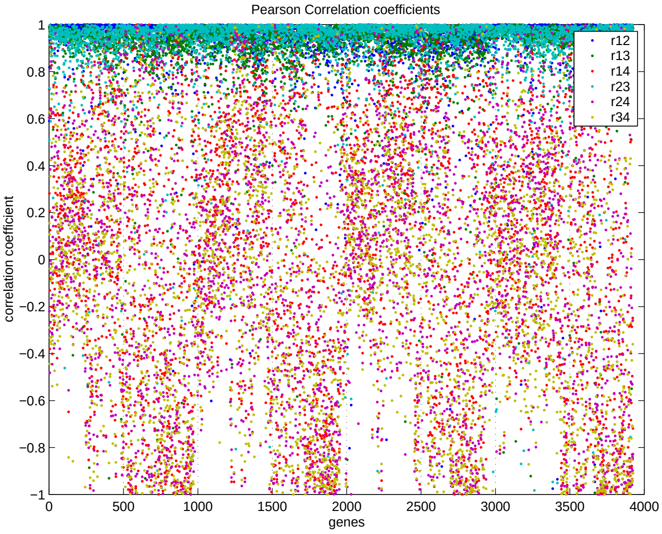


Figure 4: "Curtain" plot: The correlation coefficient for the different replicates of all the genes in the microarray are represented. It can be seen how good the quality of the replicates is. The uncorrelated replicate is the fourth, for which the dyes were swapped.

The replicates must be used to obtain a most reliable measurement of every particular ratio. A representative value of the intensity ratio of both channels for a given spot must be taken. We choose the mean. Before taking the average, the interface allows the possibility to normalize across replicates. If the overall ratio expression level is expected to be one, the different replicates can be brought to a common reference scale dividing the different replicates for a representative value of the replicates data set, which is going to be the median:

$$t_i = \frac{q_i}{med_j(q_j)}$$

where med_j is the median of the normalized ratios q_j across the different replicates. In our interface this is called *across replicates normalization* and contributes to reduce the experimental error that can come from inconsistent experimental conditions from array to array (see Figures 2, 3).

As important as getting a unique value that will be considered as the ratio of the expression level of a particular gene in both channels, is giving an estimator of how reliable this value is. For this reason we calculate as well the standard error and we use it to show in the time series plot the reliability of this value as an estimator of the log ratio. Two different t-test are available: one for reference designs for which the interest is in expression level between biological conditions and the most simple one for loop designs for which the expression level is calculated for every ratio itself.

After taking the average and calculating the standard error and t-test using the available replicates per biological condition, we still need a common reference to compare the log ratio across different biological condition. For this reason *normalization across samples* is still required. We correct the data according to:

$$e_i = \frac{t_i}{med_j(t_j)}$$

where med_j is the median of the gene expression level across the different arrays. An alternative is:

$$e_i = \frac{t_i}{med_J(t_j)}$$

where med_J is the median of the gene expression level across a fixed set of arrays. It is important to remark that across replicates normalization and across samples normalization are not compulsory and are just appropriated if the overall expression from one array to another is expected to be similar. Otherwise, we would be falsely correcting the data.

The last thing previous to proper analysis of the data (e.g. clustering, time series analysis or Principal Components)[1] is the transformation that should be chosen. The $\log_2$ transformation is the most popular but may not be the most suitable one, due to the extreme differentiation of small values. For this reason a log-linear transformation or the arcsinh transformation recommended in [5] can be more appropriated. All of them are implemented in our toolbox.

Conclusions

This paper describes a GUI for the normalization of microarray data. This interface is a good example of efficient organization of one of the approaches to normalize microarray data: the sequential method. In this paper it was demonstrated the necessity of visualizing different features of the data in order to choose among different options at every stage of the sequential normalization process. Every of those steps are described in this paper as well. Furthermore, it is needed to test how the data evolves after every different correction. The GUI for normalization allows this comparison to the user. In summary, this GUI is an example to encourage the implementation of "user friendly" environments in powerful software packages such as R. Just a few feature were included in the paper. For further information and files search http://www.systemsbiology.umist.ac.uk

Acknowledgements

The authors would like to thank St.George's Hospital Medical School microarray group (London) and to the Streptomyces group at UMIST for providing the data sets. They would also like to thank Carla Moller-Levet for implementing Figure 1 and to the Bioinformatics group at the Biomedical Engineering group of TUGraz, specially to Andi, Gerhard, Hubert, Marcel and Roland for useful discussions.

[1] Implemented in MADE

References

[1] O. Alter, P.O. Brown, and D. Botstein. Singular value decmposition for genome-wide expression data processing and modeling. *PNAS*, 97(18):10101–10106, 2000.

[2] G.A. Churchill. The ANOVA project from the Jackson's laboratory. In *http://www.jax.org/research/churchill/software/anova/Rmaanova/*, 2002.

[3] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[4] M.B. Eisen and P.O. Brown. DNA arrays for analysis of gene expression. *Methods Enzymol.*, 303:179–205, 1999.

[5] W. Huber, A. von Heydebreck, H. Sltmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 1(18):96–104, 2002.

[6] T.B. Kepler, L. Crosby, and K.T. Morgan. Normalization and analysis of DNA microarray data by self-consistency and local regression. *Genome Biology*, 3(7):research0037.1–0037.12, 2002.

[7] K. Kerr and G.A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2:183–201, 2001.

[8] K. Kerr, M. Martin, and G.A. Churchill. Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, 7:819–837, 2000.

[9] P. Luu, Y. H. Yang, S. Dudoit, and T. P. Speed. Normalization for cDNA microarray data. In *SPIE BIOS 2001*, 2001.

[10] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews Genetics*, 2(6):418–427, 2001.

[11] A. Schulze and J. Downward. Navigating gene expression using microarrays - A technology review. *Nature Cell Biology*, 3:190–195, 2001.

[12] A. Unwin. R objects, two interfaces! (r objects to interfaces?). In *DSC-2001-Proceedings. http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/index.html*, 2001.

[13] Y.H. Yang, S. Dudoit, D.M Lin, V. Peng, J. Ngai, and T.P. Speed. Normalization for cDNA microarray data: A robust composite method adressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15.1–e15.10, 2002.