Colloquium Biometricum 39
2009, 9–19

THE METHODS OF NORMALIZATION USED IN THE ANALYSIS OF TWO-COLOR MICROARRAYS

Idzi Siatkowski[1], Joanna Zyprych[1], Luiza Handschuh[2,3], Marek Figlerowicz[2]

[1]Department of Mathematical and Statistical Methods
Poznan University of Life Sciences
Wojska Polskiego 28, 60-637 Poznań, Poland
idzi@au.poznan.pl
zjoanna@au.poznan.pl
[2] Institute of Bioorganic Chemistry PAS
Noskowskiego 12/14, 61-704 Poznań, Poland
luizahan@ibch.poznan.pl
marekf@ibch.poznan.pl
[3] Poznań University of Medical Sciences Departament of Hematology,
Szamarzewskiego 84, 60-569 Poznań, Poland

Summary

In this work the basic aspects of microarray data normalization are presented. Due to high level of complexity of microarray experiments their results are usually distorted. The normalization process allows to eliminate bias and to make comparison between distinct microarrays reliable. The main types of normalization of two-color microarray data are reviewed and presented using R and Bioconductor tools.

Keywords and phrases: microarray cDNA, normalization, R software, bioconductor

Classification AMS 2000: 62-07, 62-09

1. The aim of normalization

The analysis of microarray experiment provides a lot of information regarding genome, its structure and functioning. Spotted cDNA or oligonucleotide microarrays are widely used to learn which genes are expressed in the cells and tissues and what is the level of their expression. Unfortunately, the data obtained in such experiments are usually loaded with many biological and technical errors which disguise the results to some extend. Normalization is a process that enables to remove these errors in order to make the comparison between the different microarrays reasonable. Systematic biases accompanying technical replications can be sourced from the dye effect (efficiency of various dye incorporation into a sample and different sensitivity of dyes, for example, red dye is more sensitive than green), the scanner effect (different scanner settings during individual experiments do not provide the same results), and the printer effect (different pins used simultaneously for spotting a microarray).

2. The types of normalization

The simplest possible microarray experiment is one with a series of replicate two-color arrays, all comparing two RNA samples obtained from the same organism, e.g. coming from homogenous cancer tissue and control one. Simple way to modify the above experiment would be to swap the dyes for at least one set of the arrays. Such an operation is called dye-swap. In the first experiment, the cancer sample is labeled with a red dye and the control sample with a green dye, and in the second experiment vice-versa (the cancer sample is labeled with a green dye and the control sample with a red dye). The aim of this approach is to eliminate the dye effect.

Another important question one needs to ask is which genes should be used for normalization. Yang et al. (2001) suggests three approaches. First one uses all genes on the microarray. This global method assumes that the majority of the genes represented on the microarray have a constant level of expression or that there is symmetry in the number of up- and down-regulated genes. However, such an assumption is not true for small dedicated microarrays. Here, instead of using all genes we can use a smaller subset of the so-called housekeeping genes, which are characterized by a stable expression regardless of the conditions of

the experiment. However, these genes tend to be highly expressed, and they cannot be considered a representative population in relation to all other genes. The best choice is to use so called spiking controls -control probes and exogenous RNA complementary to these probes, added to the both samples (control and investigated one in the same amount) before labelling. Therefore, they are expected to give equal intensities in both channels. Usually the probes specific for spike genes are spotted on a microarray in a number of replications. Consequently, the observed differences in the intensity of spike genes within and between arrays come from bias introduced by hybrydization and printing processes.

Another problem to solve is to ascertain whether the received data should be normalized within or between arrays (if diagnostic plots suggest a difference in scale between the arrays), or both, what is the most frequent case. In the attempt to answer this question, an important step is to analyse the variation of raw data points for each subgrid and for each microarray separately. Many useful tools to assess the quality of array data are available in Bioconductor, e.g. scatter plots, MA plots, boxplots, spatial plots, plotDensities. Interpretation of these graphs helps researchers to make decision which methods of normalization should be chosen to obtain optimal results. Within-array normalization is carried out for each array separately and it is applied when in the MAplot graph ($\boldsymbol{M}$ is the difference in red and green fluorophores intensity and $\boldsymbol{A}$ is the arithmetic mean of the logarithms of red and green fluorophores intensity) a large dispersion of the results for each individual subgrid is observed. Changes in genes expression are interpreted as follows:

- $\boldsymbol{M} = 0$ in the absence of any changes in the expression level of genes labeled in red and green dye,
- $\boldsymbol{M} = 1$ means that the genes highlighted in red are overexpressed twice comparing to genes marked in green,
- $\boldsymbol{M} = -1$ means that the genes highlighted in green are twice as overexpressed as genes marked in red,
- $\boldsymbol{M} = 2$ is a 4 - fold change, etc.

Regardless of the print-tip effect, the relationship between the intensity ratios and spot positions on a microarray is often noted. These disparities may be caused by hybridization effect. In this case the special kind of normalization (location normalization) is recommended. It is also a type of global normalization, based on assumption that the intensity of green and red color is linked with a permanent factor $\boldsymbol{k}$ which fulfills the relationship $\boldsymbol{R} = \boldsymbol{kG}$ and $\log_2(\boldsymbol{R}/\boldsymbol{G}) - \boldsymbol{c} = \log_2 \boldsymbol{R}/(\boldsymbol{kG})$,

where $\boldsymbol{R}$ and $\boldsymbol{G}$ are the intensities of the red and green channels respectively. Parameter $\boldsymbol{c}$ ($\boldsymbol{c} = \log_2 \boldsymbol{k}$, the local) is often the median or the average value for a particular set of data.

Frequently observed dye effect depends on the intensity of a single probe. Before choosing the method of normalization the linearity of data should be checked. If the data are linear, median centering method can be used – the median of the log ratios for one microarray is calculated and subsequently this median is subtracted from the log ratio of every gene. If the data are non-linear we can use lowess or another local method. Two methods can be distinguished: loess and lowess, represented by separate functions. If a linear function is used for the local regression then we call this method lowess. If a quadratic function is used it means we use loess. The lowess fit is calculated at each data point in the data set. At each point, a local polynomial is fit to a local region of the data using a linear least squares regression. It is worth of attention that using loess function one can specify the model and using lowess, one needs to provide only vectors with the coordinates of the points in the scatter plot. In the print - tip loess method $\boldsymbol{M}$ is normalized by subtracting from it the corresponding value determined by the loess curve for the grid. This method is described by Yang (2001). Scaling methods depend on the choice of a base array, which determines the average intensity of all arrays.

3. Normalization with R software

Software described in this publication is based on the free statistical programming environment R available from the site http://www.bioconductor.org. Bioconductor is an open source project developed and still developing for genomic data analysis. The Bioconductor packages such as limma, marray, vsn, arrayQuality, arrayQualityMetrics were designed for quality assessment and normalization of two-color microarray data. Below, we present how to use limma (Linear Models for Microarray Data) package to normalize cDNA array data. Limma package offers two kinds of normalization: within-array normalization and normalization between arrays. To call the first one can write:

```
> normalizeWithinArrays(object, layout, method="printtiploess",
  weights=object$weights, span=0.3, iterations=4, control
  spots=NULL, df=5, robust="M", bc.method="subtract", offset=0)
```

Function `normalizeWithinArrays` normalizes $\boldsymbol{M}$-values for dye-bias within each array. There are different methods here to use: median, printtiploess, composite, control and robustspline.

Median method computes the differences between $\boldsymbol{M}$-values for each array and the weighted median. The loess methods such as: loess, printtiploess and composite were described by Yang et al. (2001, 2002). The last two methods are control and robustspline methods. The first one refers to control spots which are the basis for matching the global loess. Next this curve is applied to the all spots on the array. Robustspline methods normalize the $\boldsymbol{M}$-values for a single microarray using robustly fitted regression splines and empirical Bayes shrinkage.

Second type of normalization is between arrays normalization. These methods normalize microarray data in such a way that log-ratios or intensities across a series of arrays are comparable. Here the following function can be used:

```
> normalizeBetweenArrays(object, method="Aquantile",
  targets=NULL,...)
```

There are different methods available for this function: scale, quantile, Aquantile, Gquantile, Rquantile, Tquantile or vsn.

Quantile method ensures that the corresponding intensities across arrays and across channels have the same distribution. Other methods (Aquantile, Gquantile, Rquantile) ensure that the green channel and the red channel have the same empirical distribution of $\boldsymbol{A}$-values (average intensities) across arrays and $\boldsymbol{M}$-values are unchanged. Otherwise, it is the case for the last method, which uses a vsn function and row data as an input. This normalization method includes background correction, then log-transformation and finally normalization. One can find this function in vsn package. An input data should have the following format: for the two-color microarray, each row corresponding to one spot, and the columns to the different arrays and wave-lengths (usually red and green). This kind of normalization is particularly useful for single-channel (one-color) arrays. For example, when one has 3 two-color arrays, the data file would have 6 columns (1-3 contain intensities for green channel, and 4-6 for the red one). For one-color arrays each row corresponds to a probe, and each column to an array.

For more details of remaining arguments call the function: > help(limma).

4. An example of normalization of data using R

In this work we used the ApoAI data (Callow et al., 2000) to illustrate the results of normalization process in Bioconductor. The experiment compared 8 knock-out mice (with excluded ApoAI (apolipoprotein) gene with 8 control mice. Target mRNA was isolated from mice liver. RNA from each knock-out mouse was labeled with Cy5 dye and hybridized separately. The reference RNA, obtained by pooling of RNA extracted from 8 control mice, was labeled with Cy3 dye and co-hybridized with each array.

Data input is the first step of analysis. Let's assume that our data are in the current working directory. The following commands can be used to read the data, target and spot files.

```
> TargetsSpot <-readTargets("ApoAITargets.txt")
> RGspot <-read.maimages(TargetsSpot$FileName,
  source="spot")
> MAspot <-MA.RG(RGspot)
> RGspot$genes <-readGAL()
> MAspot <-normalizeWithinArrays(RGspot)
```

The last function changes RG data format into MA data format, necessary to more far analyses. To compare the distributions of data for each array and each array subgrids, the following functions can be used to draw a graph for arrays and subgrids and store it in a "png" file.

```
> plotMA3by2(MAspot, prefix="MA", path=NULL,
  main=colnames(MA),zero.weights=FALSE,
  comon.lim=TRUE, device="png")
> plotPrintTipLoess(MA0, array=1,span=0.4,
  main="c1")
```

MA plots presented below (Fig. 1 and 2) show the distribution of the raw data for the first, fourth and eighth array. Every array is divided into 16 subarrays. The non-linear data evidently need the normalization. There is a substantial discrepancy between microarrays, (Fig. 2), what should be taken into account later, after within-array normalization.

Within-array normalization is ordered by the following function:

```
> MAprintTip <- normalizeWithinArrays
  (RG,method="printtiploess")
> MA0 <- normalizeWithinArrays(RGspot,
  method="none") # Method "none" computes M-values and A-values
  but does no normalization
```

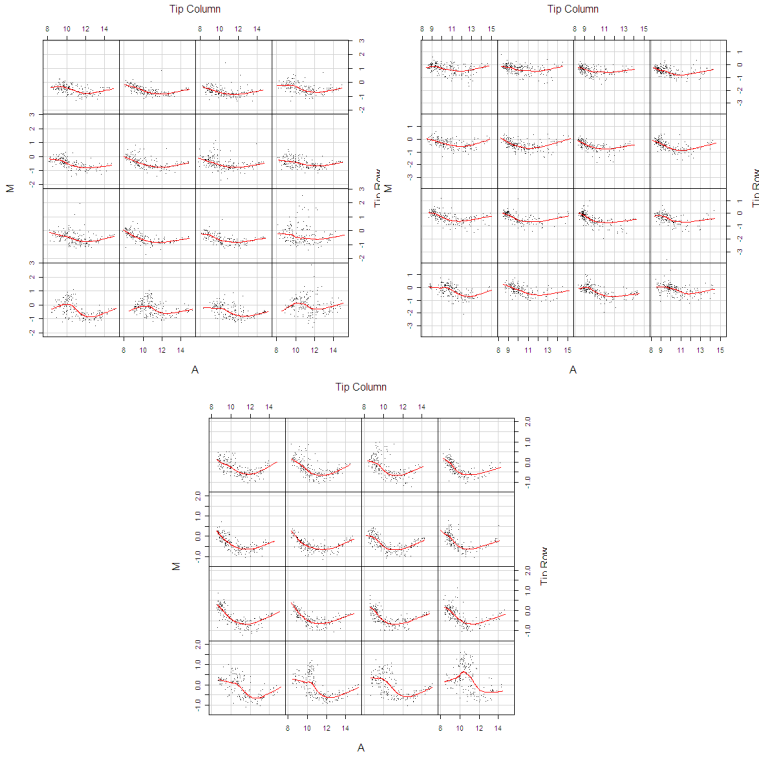

Fig. 1. MAplots for each subgrid of microarray 1, 4, 8 – data before normalization

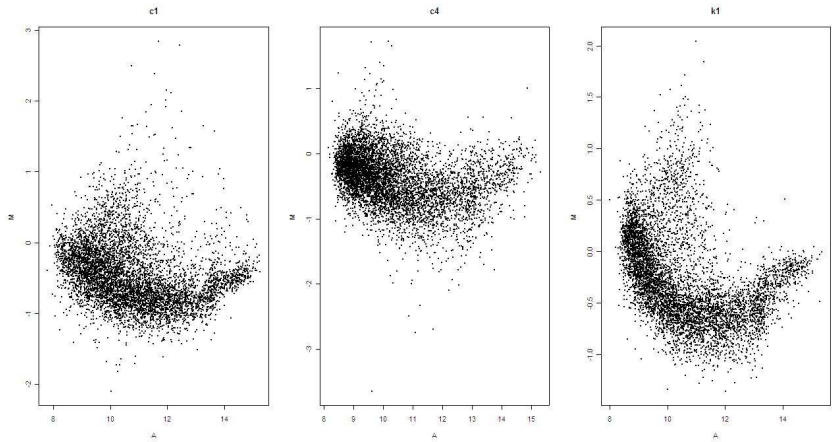

Fig. 2. MAplots for 1, 4, 8 microarray – data before normalization

After the printtiploess normalization of microarrays we obtain satisfying results. Arrays 1, 4 and 8 (Fig. 3) show significant differences when comparing the distribution of data before and after normalization.

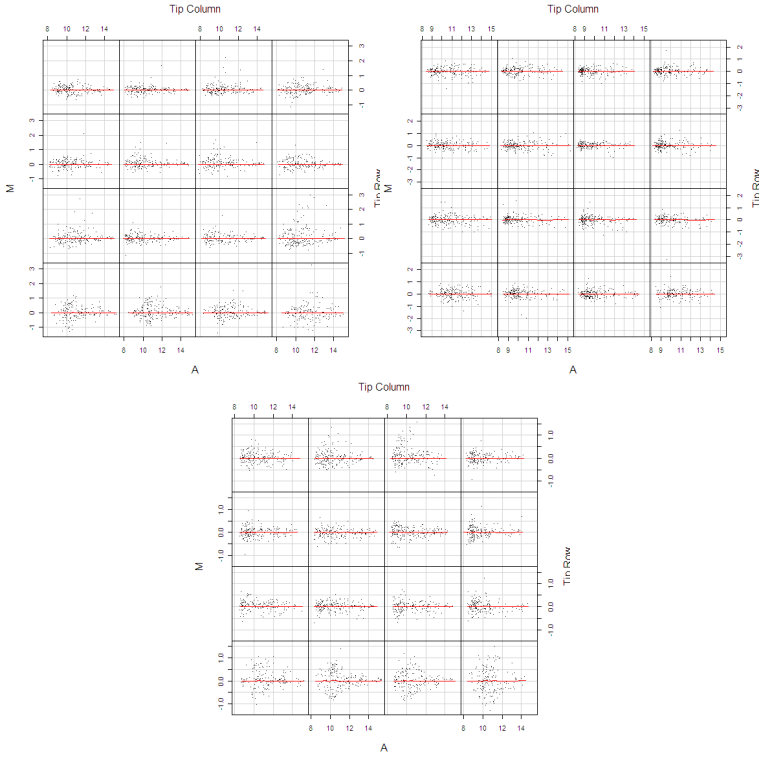

Fig. 3. MAplots for 1, 4, 8 microarray after normalization

Different types of normalization can be performed using the following functions:

```
> MA2 <- normalizeBetweenArrays(MAprintTip, method
  = "scale") # data will be normalized between arrays after within
  arrays normalization
> MA3 <- normalizeBetweenArrays(MA0, method =
  "scale") # normalization of the raw data between arrays
```

Normalization results obtained with different methods can be easily compared using boxplots. Within-array normalization can be omitted when the boxplots

are roughly at the same height. Fig. 4 and code below present the comparison between varying normalization pathways.

```
> boxplot(data.frame(MA0$M),col="bisque",main =
  "raw data", ylab = "M value", las=2)
> boxplot(data.frame(MA3$M),col="gold", main =
  "normalizeBetweenArrays_raw_data", ylab =
  "M value",las=2 )
> boxplot(data.frame(MAprintTip$M),col="red",main =
  "normalizeWithinArrays_printTip", ylab =
  "M value",las=2 )
> boxplot(data.frame(MA2$M),col="blue", main =
  "normalizeBetweenArrays_scale data", ylab =
  "M value",las=2 )
```

The first graph shows the raw data needed to be normalized between the arrays. The different slides vary with scales. The next one represents the data after normalization between arrays – scale normalization. The following one shows the effects of within-array normalization. The last is a result of normalization within-arrays and subsequent between array normalization with the scaling method. Comparing these charts one can see that the normalization between microarrays preceded by within -arrays normalization gives better results than applying only normalization between microarrays.

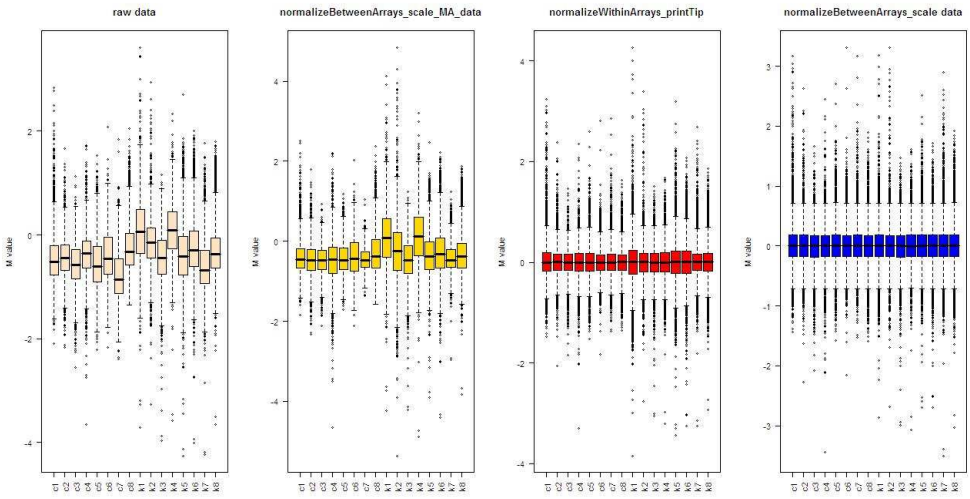

Fig. 4. Boxplots for all microarrays after applying of various normalization methods

5. Conclusion

Normalization is a very important step in the pre–processing of two-color microarray data. It has a large impact on the identification of differentially expressed genes. Normalization is required to ensure that the observed differences in fluorescence intensities indeed reflect differential gene expression, not the printing, hybridization and scanning artifacts. Microarray normalization methods will be probably further developed, however, the existing ones, e.g. print-tip loess normalization, give quite good results using a wide variety of arrays. It is important to visualize the raw data with diagnostic plots before choosing the method of normalization. When the bias in the distribution of data for separate microarrays is observed the normalization within-array should be applied. When the disparities still remain, further normalization steps such as scale-normalization between the arrays must be undertaken.

Acknowledgments

The authors wish to thank the Secretary and the reviewers' comments and suggestions that improved our manuscript. The work was partially supported by the grant from the Polish Ministry of Science and Higher Education No. PBZ-MniI-2/1/2005 to M.F.

References

Cleveland W.S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *J. Am. Stat. Assoc.* 74, 829–836.

Edwards D. (2003). Non–linear normalization and background correction on one channel cDNA microarray studies. *Bioinformatics* 19, 825–833.

Smyth G.K., Speed T.P. (2003). Normalization of cDNA microarray data. *Methods* 31, 265–273.

Yang Y.H., Dudoit S., Luu P., Speed T.P. (2001). Normalization for cDNA microarray data. In Microarrays: Optical Technologies and Informatics, M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Proceedings of SPIE*, Vol. 4266, pp. 141–152.

Yang Y.H., Dudoit S., Luu P., Lin D.M., Peng V., Ngai J., Speed T.P. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research* 30(4), e15.

METODY NORMALIZACJI W ANALIZIE DWUKOLOROWYCH MIKROMACIERZY

Streszczenie

W pracy zaprezentowano metody normalizacji danych pochodzących z dwukolorowych mikromacierzy. Omówiono typy normalizacji oraz możliwości obliczeniowe w ramach Bioconductora. Przedstawiono także funkcje wykorzystywane w analizowanych przykładach.

Słowa kluczowe: mikromacierze cDNA, analiza statystyczna, normalizacja, R, bioconductor

Klasyfikacja AMS 2000: 62-07, 62-09