Bioconductor's marray package: Plotting component

Yee Hwa Yang[1] and Sandrine Dudoit[2]

October 18, 2010

1. Department of Medicine, University of California, San Francisco, `jean@biostat.berkeley.edu`
2. Division of Biostatistics, University of California, Berkeley,
`http://www.stat.berkeley.edu/~sandrine`

Contents

1 Overview

This document provides a detailed discussion of the plotting functions in `marray` package, which is a packages for diagnostic plots of two-color spotted microarray data. This docuement provides functions for diagnostic plots of microarray spot statistics, such as boxplots, scatter–plots, and spatial color images. Examination of diagnostic plots of intensity data is important in order to identify printing, hybridization, and scanning artifacts which can lead to biased inference concerning gene expression. We encourage users to read the shorter *overview* quick start guide on this package given in the `inst/doc` directory.

2 Getting started

To load the `marray` package in your R session, type `library(marray)`. We demonstrate the functionality of this R packages using gene expression data from the Swirl zebrafish experiment. These data are included as part of the package, hence you will also need to install this package. To load the swirl dataset, use `data(swirl)`, and to view a description of the experiments and data, type `?` `swirl`.

3 Diagnostic plots

Before proceeding to normalization or any higher–level analysis, it is instructive to look at diagnostic plots of spot statistics, such as red and green foreground and background log–intensities, intensity log–ratio, area, etc. Such plots are useful for the purpose of identifying printing, hybridization, and scanning artifacts as demonstrated below. Three main types of functions were defined to operate on pre– and post–normalization microarray objects: functions for boxplots, scatter–plots, and spatial images. The main arguments to these functions are microarray objects of classes `marrayRaw`, `marrayNorm` and arguments specifying which spot statistics to display (e.g. Cy3 and Cy5 background intensities, intensity log–ratios) and which subset of spots to include in the plots. Default graphical parameters are chosen for convenience using the function `maDefaultPar` (e.g. color palette, axis labels, plot title), but the user has the option to overwrite these parameters at any point. Note that by default the plots are done for the first array in a batch. To produce plots for other arrays, subsetting methods may be used. For example, to produce diagnostic plots for the second array in the batch of zebrafish arrays `swirl`, the argument `swirl[,2]` should be passed to the plot functions.

To read in the data for the Swirl experiment and generate the plate IDs (see `marrayClasses` and `marrayInput` for greater details)

```
> library(marray)
> data(swirl)
> maPlate(swirl) <- maCompPlate(swirl, n = 384)
```

4 Spatial plots of spot statistics – `image`

The function `image` creates *images* of shades of gray or colors that correspond to the values of a statistic for each spot on an array. Details on the arguments of the function are given in ? `maImage`. The statistic can be the intensity log–ratio M, a spot quality measure (e.g. spot size or shape), or a test statistic. This function can be used to explore whether there are any spatial effects in the data, for example, print–tip or cover–slip effects. In addition to existing color palette functions, such as `rainbow` and `heat.colors`, a new function `maPalette` was defined to generate color palettes from user supplied low, middle, and high color values. To create white–to–green, white–to–red, and green–to–red palettes for microarray images

```
> Gcol <- maPalette(low = "white", high = "green", k = 50)
> Rcol <- maPalette(low = "white", high = "red", k = 50)
> RGcol <- maPalette(low = "green", high = "red", k = 50)
```

Useful diagnostic plots are images of the Cy3 and Cy5 background intensities; these images may reveal hybridization artifacts such as scratches on the slides, drops, cover–slip effects etc. The following commands produce images of the Cy3 and Cy5 background intensities for the Swirl 93 array (third array in the batch) using white–to–green and white–to–red color palettes, respectively.

```
> tmp <- image(swirl[, 3], xvar = "maGb", subset = TRUE, col = Gcol,
+     contours = FALSE, bar = FALSE)

[1] FALSE
```

2

```
> tmp <- image(swirl[, 3], xvar = "maRb", subset = TRUE, col = Rcol,
+     contours = FALSE, bar = FALSE)
```

```
[1] FALSE
```

Note that the same images can be obtained using the default arguments of the function by the shorter commands

```
> image(swirl[,3], xvar="maGb")
> image(swirl[,3], xvar="maRb")
```

If `bar=TRUE`, a calibration color bar is displayed to the right of the images. The `image` function returns the values and corresponding colors used to produce the color bar, as well as a six number summary of the spot statistics. The resulting images are shown in Figure 1. It can be noted that the Cy3 and Cy5 background intensities are not uniform across the slide and are higher in the top right corner, perhaps due to cover slip effects or tilt of the slide during scanning. Such patterns were not as clearly visible in the individual Cy3 and Cy5 TIFF images. Similar displays of the Cy3 and Cy5 foreground intensities do not exhibit such strong spatial patterns. For other arrays, such as the Swirl 81 array, background images revealed the existence of a scratch with very high background in print–tip–groups (3,2) and (3,3).

The `image` function may also be used to generate an image of the pre–normalization log–ratios M (or any other statistic of interest), using a green–to–red color palette. Figure 2 displays such an image for the Swirl 93 array, highlighting only those spots with the highest and lowest 10% pre–normalization log–ratios M. Other options include displaying contours and altering graphical parameters such as axis labels and plot title. Figure 2 suggests the existence of spatial dye biases in the intensity log–ratio, with higher values in grid (3,3) and lower values in grid column 1 of the array.

```
> tmp <- image(swirl[, 3], xvar = "maM", bar = FALSE, main = "Swirl array 93: image of pre--nor
```

```
[1] FALSE
```

```
> tmp <- image(swirl[, 3], xvar = "maM", subset = maTop(maM(swirl[,
+     3]), h = 0.1, l = 0.1), col = RGcol, contours = FALSE, bar = FALSE,
+     main = "Swirl array 93:\nimage of pre--normalization M for  10 %  tails")
```

```
[1] FALSE
```

Note that the `image` function (and other functions `boxplot` and `plot` to be described next) can be used to plot other statistics than fluorescence intensities. They can be used to plot layout parameters such as spot coordinates `maSpotRow`, print–tip–group coordinates `maPrintTip`, or plate IDs `maPlate` (Figure 3).

```
> tmp <- image(swirl[, 3], xvar = "maSpotCol", bar = FALSE)
```

```
[1] FALSE
```

```
> tmp <- image(swirl[, 3], xvar = "maPrintTip", bar = FALSE)

[1] FALSE

> tmp <- image(swirl[, 3], xvar = "maControls", col = heat.colors(10),
+     bar = FALSE)

[1] FALSE

> tmp <- image(swirl[, 3], xvar = "maPlate", bar = FALSE)

[1] FALSE
```

(a)   (b)
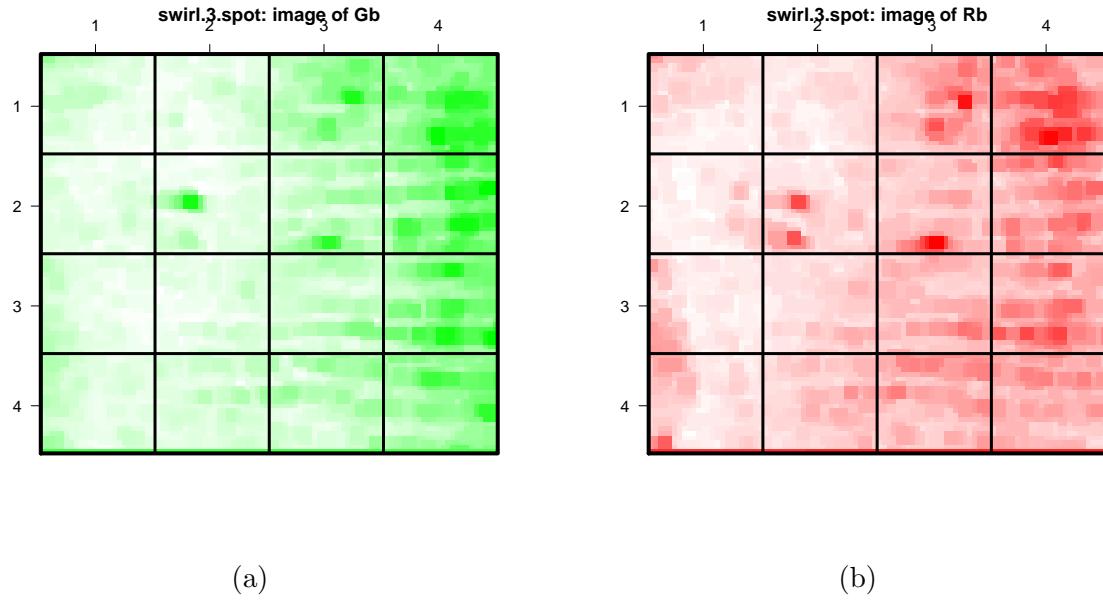
Figure 1: Images of background intensities for the Swirl 93 array. Panel (a): Cy3 background intensities using white–to–green color palette. Panel (b): Cy5 background intensities using white–to–red color palette.

(a)   (b)
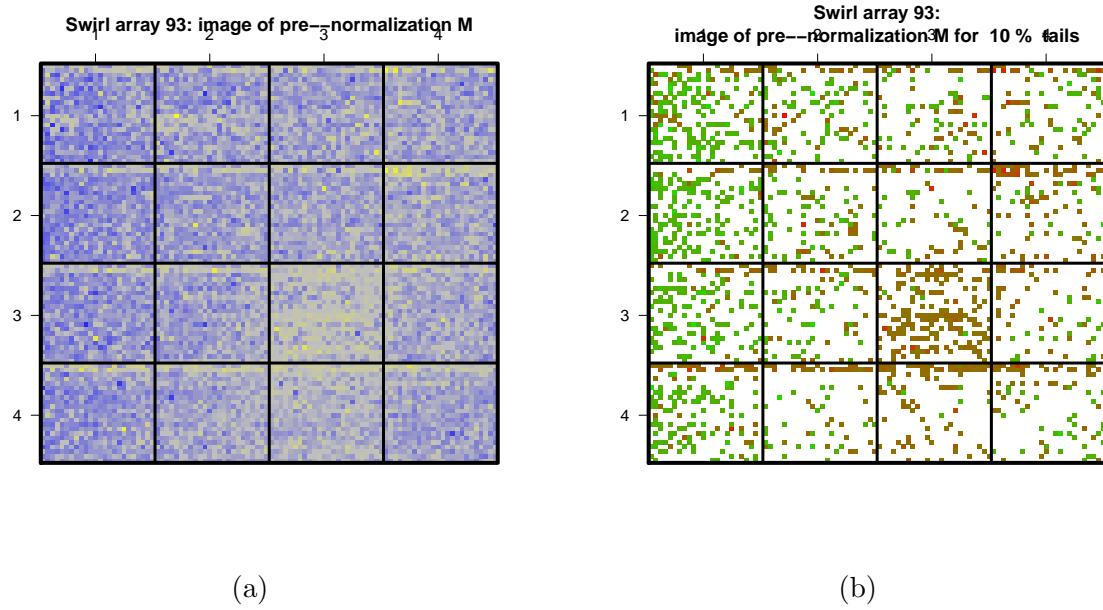
Figure 2: Images of the pre–normalization intensity log–ratios M for the Swirl 93 array, using a green–to–red color palette. Panel (a): All spots are displayed. Panel (b): only spots with the highest and lowest 10% log–ratios are highlighted.

swirl.3.spot: image of SpotCol

swirl.3.spot: image of PrintTip

(a)　　　　　　　　　　　　(b)

swirl.3.spot: image of Plate

**swirl.3.spot: image of Controls**

(c)　　　　　　　　　　　　(d)
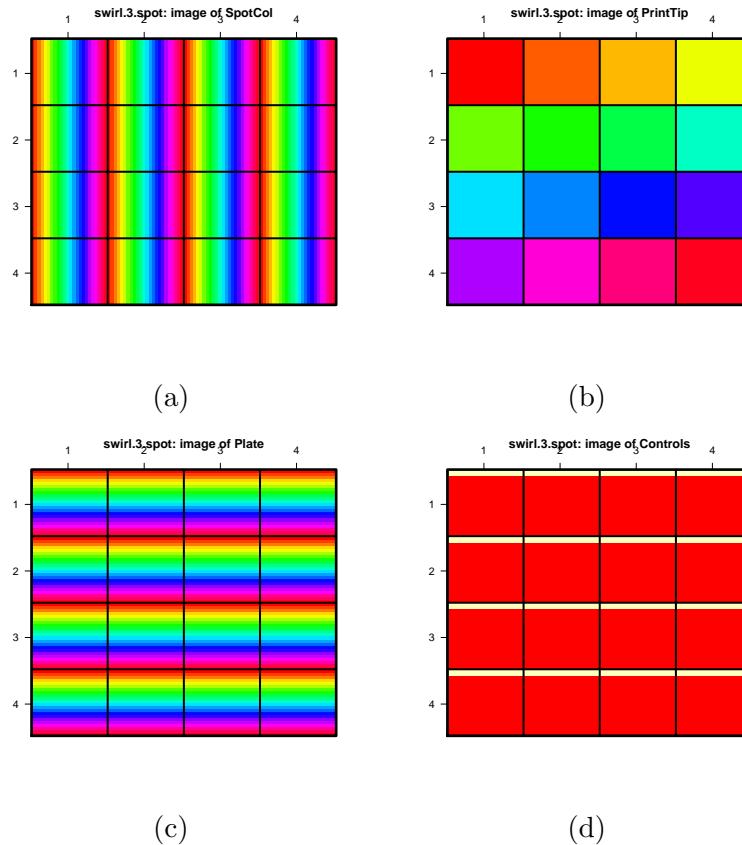
Figure 3: Images of layout parameters for the Swirl 93 array. Panel (a): Spot matrix column coordinate. Panel (b): Print–tip–group. Panel (c): Plate index. Panel (d): Control status.

5 Boxplots of spot statistics – `boxplot`

Boxplots of spot statistics by plate, print–tip–group, or slide can also be useful to identify spot or hybridization artifacts. *Boxplots*, also called *box–and–whisker plots*, were first proposed by Tukey in 1977 as simple graphical summaries of the distribution of a variable. The summary consists of the median, the upper and lower quartiles, the range, and, possibly, individual extreme values. The central box in the plot represents the *inter–quartile range (IQR)*, which is defined as the difference between the 75th percentile and 25th percentile, i.e., the upper and lower quartiles. The line in the middle of the box represents the median; a measure of central location of the data. Extreme values, greater than 1.5 IQR above the 75th percentile and less than 1.5 IQR below the 25th percentile, are typically plotted individually.

The function `boxplot` produces boxplots of microarray spot statistics for the classes `marrayRaw`, `marrayNorm`. The function `boxplot` has three main arguments:

x: Microarray object of class `marrayRaw` or `marrayNorm`.

xvar: Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object such as `maPlate` or a method such as `maPrintTip`. If `xvar` is `NULL`, the data are not stratified.

yvar: Name of accessor method for the spot statistic of interest, typically a slot name for the microarray object `m`, such as `maM`.

Figure 4 panel (a) displays boxplots of pre–normalization log–ratios M for each of the 16 print–tip–groups for the Swirl 93 array. This plot was generated by the following commands

```
> boxplot(swirl[, 3], xvar = "maPrintTip", yvar = "maM", main = "Swirl array 93: pre--normaliza
```

The boxplots clearly reveal the need for normalization, since most log–ratios are negative in spite of the fact that only a small proportion of genes are expected to be differentially expressed in the mutant and wild–type zebrafish. As is often the case, this corresponds to higher signal in the Cy3 channel than in the Cy5 channel even in the absence of differential expression. In addition, the boxplots show the existence of spatial dye biases in the log–ratios. In particular, print–tip–group (3,3) clearly stands out from the remaining ones, as suggested also in the image of Figure 2. The function `maBoxplot` may also be used to produce boxplots of spot statistics for all arrays in a batch. Such plots are useful when assessing the need for between array normalization, for example, to deal with scale differences among different arrays. The following command produces a boxplot of the pre–normalization intensity log–ratios M for each array in the batch `swirl`. Figure 5 panel (a) suggest that different normalizations may be required for different arrays, including possibly scale normalization.

```
> boxplot(swirl, yvar = "maM", main = "Swirl arrays: pre--normalization")
```

The function `maNorm` from the `marrayNorm` package can be used for different types of within-array location normalization. The following command normalizes all four arrays in the Swirl experiment simultaneously. Please refer to the vignette on normalization for more information. The following command performs within print-tip group loesss normalization.

```
> swirl.norm <- maNorm(swirl, norm = "p")
```

The following commands can be used to produce post–normalization boxplots of the log–ratios.
The plots are shown in panel (b) of Figures 4 and 5.

```
> boxplot(swirl.norm[, 3], xvar = "maPrintTip", yvar = "maM", main = "Swirl array 93: post--nor

> boxplot(swirl.norm, yvar = "maM", col = "green", main = "Swirl arrays: post--normalization")
```
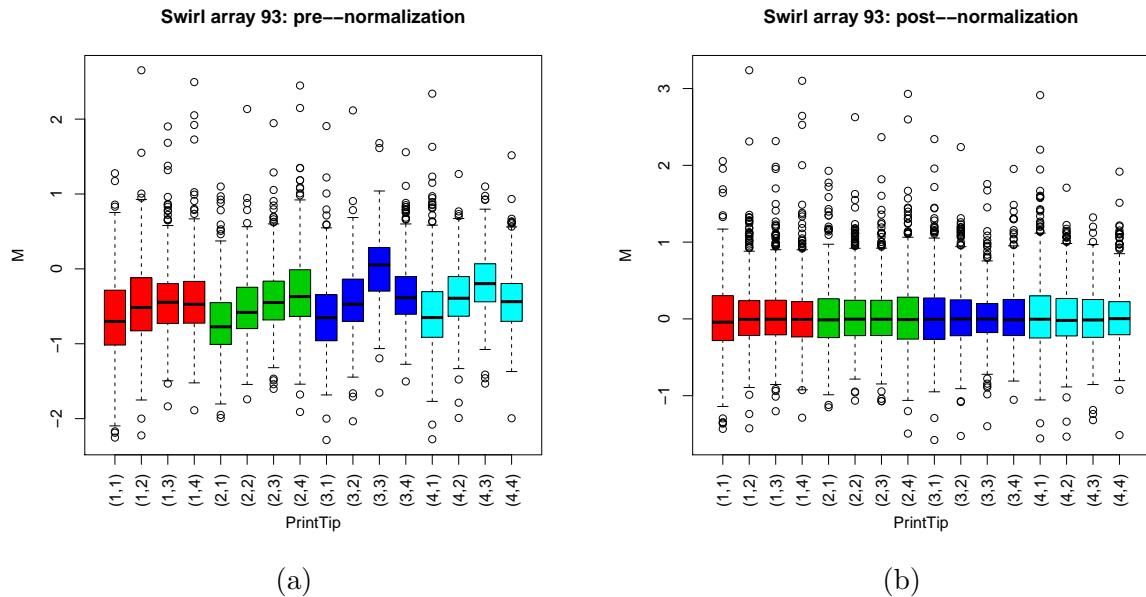
Figure 4: Boxplots by print–tip–group of the pre– and post–normalization intensity log–ratios M for the Swirl 93 array.
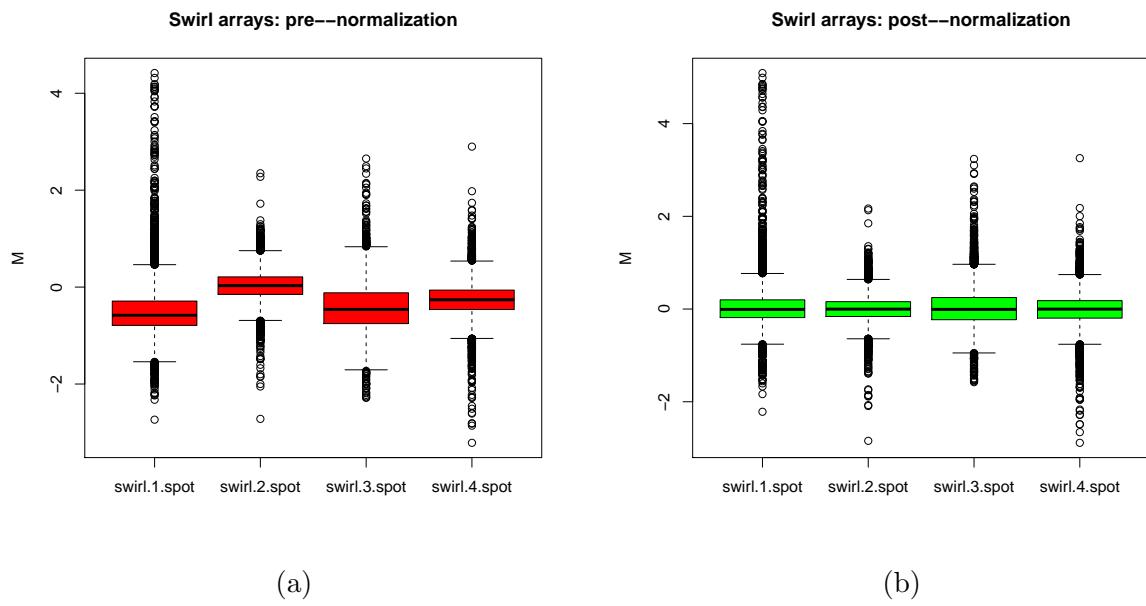
Figure 5: Boxplots of the pre–and post–normalization intensity log–ratios M for the four arrays in the Swirl experiment.

6 Scatter–plots of spot statistics – `maPlot` or `plot`

The function `plot` produces *scatter–plots* of microarray spot statistics for the classes `marrayRaw` and `marrayNorm`. It also allows the user to highlight and annotate subsets of points on the plot, and display fitted curves from robust local regression or other smoothing procedures (see details in `?  maPlot`). The function `maPlot` has seven main arguments:

x: Microarray object of class `marrayRaw` or `marrayNorm`.

xvar: Name of accessor function for the abscissa spot statistic, typically a slot name for the microarray object m, such as `maA`.

yvar: Name of accessor function for the ordinate spot statistic, typically a slot name for the microarray object m, such as `maM`.

zvar: Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object such as `maPlate` or a method such as `maPrintTip`. If `zvar` is NULL, the data are not stratified.

lines.func: Function for computing and plotting smoothed fits of `yvar` as a function of `xvar`, separately within values of `zvar`, e.g. `maLoessLines`. If `lines.func` is NULL, no fitting is performed.

text.func: Function for highlighting a subset of points, e.g., `maText`. If `text.func` is NULL, no points are highlighted.

legend.func: Function for adding a legend to the plot, e.g. `maLegendLines`. If `legend.func` is NULL, there is no legend.

As usual, optional graphical parameters may be supplied and these will overwrite the default parameters set in the plot functions. A number of functions for computing and plotting the fits are provided, such as `maLowessLines` and `maLoessLines` for robust local regression using the R functions `lowess` and `loess`, respectively (type `?  loess` or `?  lowess` for a brief description of R functions for robust local regression). Functions are also provided for highlighting points (e.g. `text`) and adding a legend to the plot (e.g. `maLegendLines`).

MA–**plots.** Single–slide expression data are typically displayed by plotting the log–intensity $\log_2 R$ in the red channel vs. the log–intensity $\log_2 G$ in the green channel. Such plots tend to give an unrealistic sense of concordance between the red and green intensities and can mask interesting features of the data. We thus recommend plotting the intensity log–ratio $M = \log_2 R/G$ vs. the mean log–intensity $A = \log_2 \sqrt{RG}$. An MA–plot amounts to a 45° counterclockwise rotation of the $(\log_2 G, \log_2 R)$– coordinate system, followed by scaling of the coordinates. It is thus another representation of the (R, G) data in terms of the log–ratios M which directly measure differences between the red and green channels and are the quantities of interest to most investigators. We have found MA–plots to be more revealing than their $\log_2 R$ vs. $\log_2 G$ counterparts in terms of identifying spot artifacts and for normalization purposes (Dudoit et al., 2002; Yang et al., 2001, 2002).

Figure **??** panel (a) displays the pre–normalization MA–plots for the Swirl 93 array, with the sixteen lowess fits for each of the print–tip–groups (using a smoother span $f = 0.3$ for the `lowess` function). The figure was generated with the following commands

```
> defs <- maDefaultPar(swirl[, 3], x = "maA", y = "maM", z = "maPrintTip")
> legend.func <- do.call("maLegendLines", defs$def.legend)
> lines.func <- do.call("maLowessLines", c(list(TRUE, f = 0.3),
+     defs$def.lines))
> plot(swirl[, 3], xvar = "maA", yvar = "maM", zvar = "maPrintTip",
+     lines.func, text.func = maText(), legend.func, main = "Swirl array 93: pre--normalization

> plot(swirl.norm[, 3], xvar = "maA", yvar = "maM", zvar = "maPrintTip",
+     lines.func, text.func = maText(), legend.func, main = "Swirl array 93: post--normalizatio
```

The same plots can be obtain using the default arguments of the function by the commands

```
> plot(swirl[,3])
> plot(swirl.norm[,3], legend.func=NULL)
\begin{verbatim}

To highlight, say, the spots with the highest and lowest 5\%
log--ratios using purple points, or using red symbol {\tt a} use the
following commands

\begin{verbatim}
> points(swirl.norm[,3], subset=maTop(maM(swirl.norm[,3]),h=0.05,l=0.05),
pch=19, col="purple")
> text(swirl.norm[,3], subset=maTop(maM(swirl.norm[,3]),h=0.05,l=0.05),
labels="a", col="red")
\begin{verbatim}

\begin{figure}
\begin{center}
\begin{tabular}{cc}
\includegraphics[width=3in,height=3in,angle=0]{maPlot1pre} &
\includegraphics[width=3in,height=3in,angle=0]{maPlot1post} \\
(a) & (b)
\end{tabular}
\end{center}
\caption{Pre-- and post--normalization $MA$--plot for the Swirl 93
  array, with the lowess fits for individual
  print--tip--groups. Different colors are used to represent lowess
  curves for print--tips from different rows, and different line types
  are used to represent lowess curves for print--tips from different
  columns. }
\protect\label{fig:maPlot1}
```

```
\end{figure}
```

```
Figure \ref{fig:maPlot1} illustrates the non--linear dependence of the
log--ratio $M$ on the overall spot intensity $A$ and thus suggests that
an intensity or $A$--dependent normalization method is preferable to a
global one (e.g. median normalization). Also, the lowess fits vary among
print--tip--groups, again revealing the existence of spatial dye
biases. Figure \ref{fig:maPlot1} panel (b) displays the $MA$--plot after
within--print--tip--group loess location normalization.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section{Wrapper functions for basic sets of diagnostic plots -- {\tt maQualityPlots}}
The following command in another package {\tt arrayQuality} will
generate qualitative diagnostic plots for each arrays in the {\tt
marrayRaw} object and by default, saved it as different png files in the
working directory.  More details of this can be found in the package
{\tt arrayQuality}.
```

```
\begin{verbatim}
library(arrayQuality)
maQualityPlots(swirl)
```

Note: Sweave. This document was generated using the `Sweave` function from the R *tools* package. The source file is in the `/inst/doc` directory of the package *marray*.

References

S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12(1): 111–139, 2002.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty, editors, *Microarrays: Optical Technologies and Informatics*, volume 4266 of *Proceedings of SPIE*, May 2001.

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