



Using Our Discipline to Enhance Human Welfare August 7 - 11, 2005

Lab 2 - Loading, pre-processing and normalizing two-color microarray data (part II)

James Wettenhall. August 6-7, 2005

1. Software required for this lab.

You will need R 2.1.0 or 2.1.1 (<http://www.R-Project.org>) for this lab. This section lists all of the R packages you need to have installed and also lists some additional R packages which are recommended. Note that most of the R packages can be installed from the Bioconductor site automatically using:

```
source("http://www.bioconductor.org/getBioC.R")
getBioC()
```

However, Bioconductor releases only occur once every six months, whereas the authors of these labs would typically use a more up-to-date version of some packages e.g. `limma`, which is why it is advisable to check the R package version numbers below and install them from the links provided if your installation is not up-to-date.

1.1 Required R packages

It is highly desirable to have these R packages in a directory in which you have write permission. You can use `.libPaths("C:/Custom/R/library/directory")` or `.libPaths("C:\\Custom\\R\\library\\directory")` before you run `install.packages()` (or equivalent) to install the package(s) in a customized directory location.

Package	Windows	MacOS X	Source
arrayQuality_1.2.2	arrayQuality_1.2.2.zip	arrayQuality_1.2.2.tar.gz	arrayQuality_1.2.2.tar.gz
convert_1.1.11	convert_1.1.11.zip	convert_1.1.11.tgz	convert_1.1.11.tar.gz
gridBase_0.4-1	gridBase_0.4-1.zip	gridBase_0.4-1.tar.gz	gridBase_0.4-1.tar.gz
hexbin_1.0.10	hexbin_1.0.10.zip	hexbin_1.0.10.tgz	hexbin_1.0.10.tar.gz
limma_2.0.2	limma_2.0.2.zip	limma_2.0.2.tar.gz	limma_2.0.2.tar.gz

marray_1.6.3	marray_1.6.3.zip	marray_1.6.3.tgz	marray_1.6.3.tar.gz
mclust_2.1-11	mclust_2.1-11.zip	mclust_2.1-11.tar.gz	mclust_2.1-11.tar.gz
statmod_1.1.1 or statmod_1.2.0	statmod_1.2.0.zip	statmod_1.2.0.tar.gz	statmod_1.2.0.tar.gz

1.2 Required data

The cell line dataset is required for this lab, and can be downloaded from the following URLs:

<http://bioinf.wehi.edu.au/marray/jsm2005/CellLine.zip>

<http://bioinf.wehi.edu.au/marray/jsm2005/CellLineImages.zip>

1.3 Details on the files used

After conducting a microarray experiment on one or more arrays printed with a particular library of probes, the arrays are scanned to produce TIFF images, one for each channel (Cy3 and Cy5). The TIFF images are processed using an image analysis program such as ArrayVision, ImaGene, GenePix, QuantArray or **SPOT** to acquire the red and green foreground and background intensities for each spot, along with other measurements. The spot intensities are then exported from the image analysis program into a series of text files. There should be one file for each array or, in the case of ImaGene, two files for each array.

To analyze microarray data, we require (i) a file which describes the probes, often a GenePix Array List (GAL) file, and (ii) the image analysis output files. In most cases it is also desirable to have a Targets File, describing which RNA sample was hybridized to each channel of each array. A further optional file is the Spot Types file (STF) which identifies special probes such as control spots.

The Targets File

The Targets File is normally in tab-delimited text format. It should contain a row for each microarray in your experiment. It should contain a FileName column, giving the file from image-analysis containing raw foreground and background intensities for each slide, a Cy3 column giving the RNA type reverse transcribed and labelled with Cy3 dye for that slide (e.g. Wild Type) and a Cy5 column giving the RNA type reverse transcribed and labelled with Cy5 dye for that slide. For ImaGene files, the FileName column is split into a FileNameCy3 column and a FileNameCy5. As well as the essential columns, you can have a Name column giving an alternative slide name to the default name, "Slide n", where n is the SlideNumber and you can have a Date column, listing the date of the hybridization. Additional columns are allowed, provided that the column names are unique. Targets Files can be created in excel or a text editor, and should be saved in Text (Tab Delimited) .txt format.

The Spot Types File

The Spot Types File (STF) is a tab-delimited text file which allows you to identify different types of spots from the gene list. The STF is typically used to distinguish control spots from those corresponding to genes of interest, to distinguish positive from negative controls, ratio from calibration controls and so on. In the first column of this file (named SpotType), names for each class of spot (eg gene, control) on the array should be specified. One or more other columns should have the same names as columns in the gene list file and should contain patterns or regular expressions sufficient to identify the spot-type. Asterisks are wildcards which can represent anything. Be careful to use upper or lower case as appropriate and don't insert any extra spaces. Any other columns are assumed to contain plotting parameters, such as colors (column name Color) or plotting characters (column name cex) to be associated with the different types of points. STF can be created in excel or a text editor, and should be saved in Text (Tab Delimited) .txt format.

The STF uses simplified regular expressions to match patterns. For example, 'AA*' means any string starting with 'AA', '*AA' means any code ending with 'AA', 'AA' means exactly these two letters, '*AA*' means any string containing 'AA', 'AA.' means 'AA' followed by exactly one other character and 'AA\.' means exactly 'AA' followed by a period and no other characters. For

those familiar with regular expressions, any other regular expressions are allowed but the codes ^ for beginning of string and \$ for end of string should be excluded. Note that the patterns are matched sequentially from first to last, so more general patterns should be included first. For example, it is often a good idea to include a default spot-type as the first line in the STF with pattern '*' for all the pattern-matching columns and with default plotting parameters.

2. Cell line comparison data

In this exercise we consider an experiment where two RNA sources are compared directly on 6 replicate arrays.

Background. This data is taken from a set of quality control hybridizations, which were performed to assess how well the spots were printed on the arrays, rather than a question of biological interest. The RNA compared was from two very different cell lines, which ensured many of the genes were differentially expressed at varying degrees. Data for each slide is available from two image analysis packages (GenePix and SPOT), and will be used to highlight some of the differences between the programs, particularly in terms of the background levels measured for each spot.

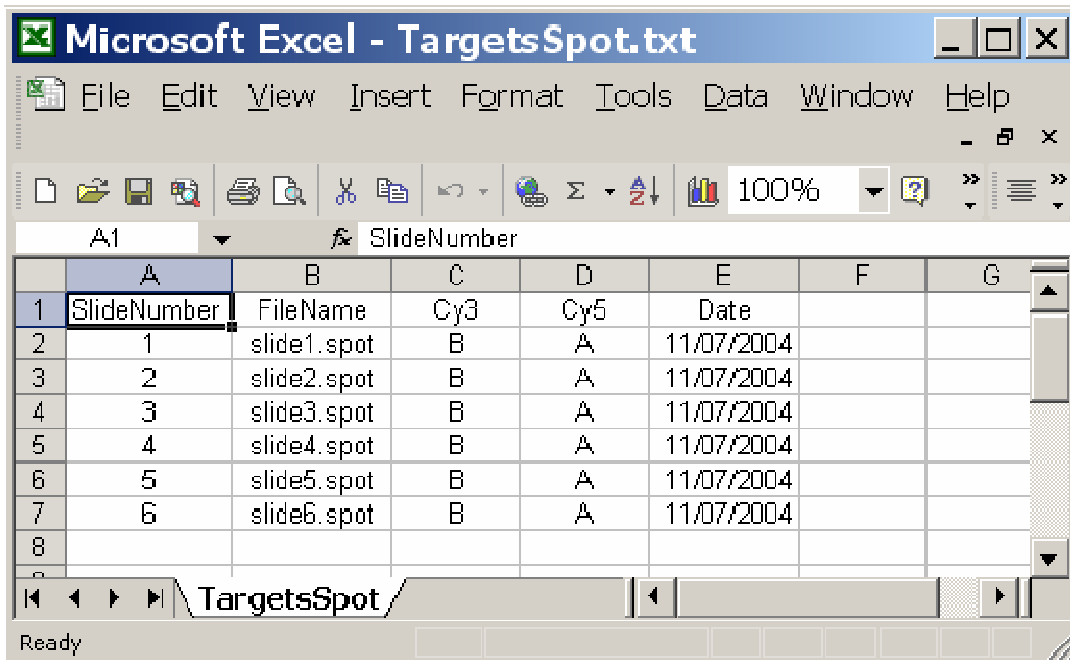
The hybridizations. Six microarrays were each hybridized with RNA from cell line A (Cy5) and cell line B (Cy3).

The arrays. A total of 10944 spots were printed on each array. They were arranged in 6x4 print-tip grids, each containing 19x24 rows and columns of spots. The Lucidea Universal ScoreCard controls (Samartzidou et al, 2001) were printed on the arrays, and spiked into the RNA mixes of each experiment. This control series is made up of artificial genes, some of which are added in equal quantities, or at 3 and 10 times as much in one sample compared to the other to give known fold-changes for particular spots.

For this example, you will need the GAL file called "genelist.gal" and the raw SPOT (.spot) and GenePix (.gpr) output files in the current working directory.

2.1 Defining Targets files

For the cell line comparison arrays, a targets file would look like



	A	B	C	D	E	F	G
1	SlideNumber	FileName	Cy3	Cy5	Date		
2	1	slide1.spot	B	A	11/07/2004		
3	2	slide2.spot	B	A	11/07/2004		
4	3	slide3.spot	B	A	11/07/2004		
5	4	slide4.spot	B	A	11/07/2004		
6	5	slide5.spot	B	A	11/07/2004		
7	6	slide6.spot	B	A	11/07/2004		
8							

for the Spot data, and

Microsoft Excel - TargetsGenePix.txt

File Edit View Insert Format Tools Data Window Help

100%

A1 SlideNumber

	A	B	C	D	E	F	G
1	SlideNumber	FileName	Cy3	Cy5	Date		
2	1	slide1.gpr	B	A	11/07/2004		
3	2	slide2.gpr	B	A	11/07/2004		
4	3	slide3.gpr	B	A	11/07/2004		
5	4	slide4.gpr	B	A	11/07/2004		
6	5	slide5.gpr	B	A	11/07/2004		
7	6	slide6.gpr	B	A	11/07/2004		
8							
9							
10							

TargetsGenePix

Ready

for the GenePix data. Files named "TargetsSpot.txt" and "TargetsGenePix.txt" are included in the zip file for the cell line data.

2.2 Defining a Spot Types file

An appropriate STF for these arrays which contain Lucidea Universal ScoreCard controls is:

Microsoft Excel - SpotTypes.txt

File Edit View Insert Format Tools Data Window Help

100%

A1 SpotType

	A	B	C	D	E	F	G
1	SpotType	ID	Name	Color	cex		
2	Gene	*	*	black	0.2		
3	Blank	Blank	*	yellow	1		
4	Reserved	.reserved	*	pink	1		
5	NC	.NC	*	orange	1		
6	PC	.PC	*	brown	1		
7	HG	.(1,2)HG	*	magenta	1		
8	DR	.DR	*	blue	1		
9	U03	1RC	*	red	1		
10	D03	2RC	*	green	1		
11	U10	3RC	*	darkred	1		
12	D10	4RC	*	darkgreen	1		
13							

SpotTypes

Ready

A STF named "SpotTypes.txt" is included in the zip file for the cell line data.

2.3 Loading the data using limma

To load the `limma` library, type

```
library(limma)
```

To read in the STF and targets information, use the following commands

```
SpotTypes <- readSpotTypes()
```

and

```
TargetsSpot <- readTargets("TargetsSpot.txt")
```

The file names stored in the targets file can be used to read in the intensity information contained in each of the image analysis output files with the command

```
RGspot <- read.maimages(TargetsSpot$FileName, source="spot")
```

The object `RGspot` is an `RGList` object which contains five components: `R`, `G`, `Rb`, `Gb` and `targets`. You can verify this by typing:

```
names(RGspot)
```

The foreground intensities are stored in `RGspot$R` (Cy5) and `RGspot$G` (Cy3) and the backgrounds are stored in `RGspot$Rb` (Cy5) and `RGspot$Gb` (Cy3).

Other useful information can also be added to the `RGList`, such as the probe ID information

```
RGspot$genes <- readGAL()
```

This information is read in from a file in the current working directory with an extension `.gal`. If no such file, or multiple files with this extension exist, an error message will be generated.

The `SpotTypes` information can now be used to determine the status of each spot (ie control or gene), using the command

```
RGspot$genes$Status <- controlStatus(SpotTypes, RGspot$genes)
```

This information will be used later to highlight the control spots on an MA plot. The printer information is also needed if intensity based print-tip normalization is to be applied to the data. The easiest way to acquire this information is from the GAL file, ie

```
RGspot$printer <- getLayout(RGspot$genes)
```

Now all of the necessary information is stored in the `RGList` object, `RGspot`. View a summary of the contents of `RGspot` by typing its name and pressing enter.

```
RGspot
```

This process can be repeated for the GenePix data:

```
TargetsGenePix <- readTargets("TargetsGenePix.txt")
```

```
RGgpr <- read.maimages(TargetsGenePix$FileName, source="genepix")
```

```
RGgpr$genes <- readGAL()
```

```
RGgpr$printer <- getLayout(RGgpr$genes)
```

```
RGgpr$genes$Status <- controlStatus(SpotTypes, RGgpr$genes)
```

The default for GenePix output is that the F635 Mean (Cy5) and F532 Mean (Cy3) columns of each .gpr file are used as foreground intensities and B635 Median (Cy5) and B532 Median (Cy3) are used as background intensities.

An alternative way of reading data from the image analysis output files without using the Targets file, is

```
spotFiles <- dir(pattern="\\.spot$")
RGspot <- read.maimages(spotFiles, source="spot")
```

for SPOT data and

```
gprFiles <- dir(pattern="\\.gpr$")
RGgpr <- read.maimages(gprFiles, source="genepix")
```

for GenePix data. In this case, the order of the files in the working directory (which depends on their names) will determine the order they are read in. A benefit of using a targets file is that the order is controlled by the user. The targets file can also be used later in the linear modelling analysis to set up the design matrix. Assigning the gene ID's, control status of the spots and the printer layout will need to be repeated as before.

The `pattern` argument of the `dir` function accepts a regular expression. A detailed understanding of regular expressions is certainly not required to perform basic microarray analysis. The double backslash (`\\`) tells R that the dot immediately after it should be treated literally as a dot rather than as a special character (as it normally would be in regular expressions). The dollar sign (`$`) tells R that not only must the file names contain the pattern ".spot" (or ".gpr"), but they must in fact end with ".spot" (or ".gpr").

2.4 Normalization using limma

Un-normalized M and A values for each spot can be constructed using the function `MA.RG`.

```
MAspot <- MA.RG(RGspot)
MAgpr <- MA.RG(RGgpr)
```

MA-plots. The MA-plot of the un-normalized values for the first array is obtained using:

```
plotMA(MAspot, array=1)

# Now open a new plot window (system-dependent)
windows() # For Windows users
quartz()  # For Mac users
x11()     # For Linux users

plotMA(MAgpr, array=1)
```

Notice the automatic highlighting of the control spots specified in the STF. To do the same plot for other arrays, the `array` argument can be changed.

Normalization. For print-tip group normalization, try

```
MAspot <- normalizeWithinArrays(RGspot)
MAgpr <- normalizeWithinArrays(RGgpr)
```

To do MA plots, 6 to a page and save the output in .png format, the commands

```
plotMA3by2(MAspot, prefix="MAspot")
plotMA3by2(MAgpr, prefix="MAgpr")
```

can be used. Comparing the MA plots for a given slide, you should notice that the dynamic range of the A values is less for SPOT data compared to GenePix. There is also some fanning of the M values at lower intensities for the GenePix plots. This is a result of the different background estimates used in the two packages. The morphological background in SPOT tends to give a low, stable estimate of slide background, whereas the median of the background pixels used in GenePix is an over-estimate (see Yang et al (2002)), resulting in numerical instability for many of the low intensity log-ratios. Not background correcting GenePix data may be a good option in some instances, ie

```
RGgprnobj <- backgroundCorrect(RGgpr, method="none")
```

then proceed as before to normalize the data, using `RGgprnobj` instead of `RGgpr`.

2.5 Diagnostic plots using arrayQuality

For some other diagnostic plots, try the following functions from the `arrayQuality` library.

```
library(arrayQuality)
controlCode <- as.matrix(read.table("controlCode.txt",
header=TRUE, sep="\t"))
rawdata <- gpQuality(dir(pattern="\\.gpr$"), compBoxplot=FALSE,
controlId="Name")
```

For each slide, a summary diagnostic plot "diagPlot.<slidename>.png" will be saved in the working directory. Each slide summary shows MA plots, image plots of M (before and after normalization) and A values, dot plots of the M and A values for each class of control spot (provided `controlCode` is set correctly), as well as single channel intensity plots. These plots allow the overall quality of a slide to be assessed visually. The `arrayQuality` package can also provide a more quantitative measurement of slide quality by comparing each slide to a set of reference slides, where available (not applicable for this example).

2.6 Matching slides with their images based on results from gpQuality

For this exercise you will need the array images for the CellLine data set, which are available from <http://bioinf.wehi.edu.au/marray/jsm2005/CellLineImages.zip>. Your task is to guess which slide (1,2,3,4,5,6) corresponds to which array image (A,B,C,D,E,F) based on the quality assessment provided by `gpQuality`.

Array	Image
Slide 1	<input type="text"/>
Slide 2	<input type="text"/>
Slide 3	<input type="text"/>
Slide 4	<input type="text"/>
Slide 5	<input type="text"/>
Slide 6	<input type="text"/>

Free JavaScripts provided
by [The JavaScript Source](#)

Exiting. Once you are finished, type `q()` at the R prompt, or from the R console (Windows users) choose File > Exit to quit from your R session.

3. Acknowledgements

Thanks to Matt Ritchie and Gordon Smyth for allowing the use of material from previous microarray workshops, thanks to Gordon Smyth for allowing the use of his `limma` documentation and worked example analyses and thanks to Jean Yang and Agnes Paquet who assisted greatly with the `arrayQuality` example.

4. References

1. Samartzidou, H., Turner, L., Houts, T., Frome, M., Worley, J., and Albertsen, H. (2001) Lucidea Microarray ScoreCard: An integrated analysis tool for microarray experiments, *Life Science News*.
2. Smyth, G. K., Thorne, N. P. and Wettenhall J. (2004) *limma: Linear Models for Microarray Data User's Guide*. The Walter and Eliza Hall Institute of Medical Research.
3. Yang, Y. H., Buckley, M. J., Dudoit, S., and Speed, T. P. (2002). Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistics*, 11 (1), 108-136.