Analysis of Designed Microarray Experiments

Statistical Methods in Microarray Analysis Tutorial
Institute for Mathematical Sciences National University of Singapore
January 3, 2004

Gordon Smyth
Walter and Eliza Hall Institute of Medical Research

Hands-On Lab

0. Prerequisites

We will assume that readers are running R 1.8.1 under Windows and that you have limma 1.3.9 installed.

We also assume that you have the data files associated with this tutorial in the directory c:\R on your computer. The data files can be obtained as zip files from http://bioinf.wehi.edu.au/Singapore2004.

First Steps

Start R on your computer and change the working directory to c:\R (use the menus File > Change dir ...). Load the limma library:

```
> library(limma)
> help.start()
```

Follow the links Packages > limma. Have a look at the User's Guide and the introductory help topics.

1. A Single Array

We will start off by looking at a single array of Genepix data. The zip file associated with this experiment is firstarray.zip. This contains three files which should be unpacked into your working directory, assumed to be c:\R.

First read the data into your R session:

```
RG <- read.maimages("firstarray.gpr",source="genepix",wt.fun=wtflags(0))
```

Have a look at the data:

```
RG
plotMA(RG)
```

The identities of the genes are contained in the Genepix Allocation List (GAL) file:

```
RG$genes <- readGAL("human10_5k.gal")
RG$genes[1:30,]
```

We can make the plot more fancy by highlighting various sorts of control spots:

```
spottypes <- readSpotTypes()
spottypes
RG$genes$Status <- controlStatus(spottypes, RG)
plotMA(RG)
```

Now let's try the same plot but without background correction:

```
RGnb <- backgroundCorrect(RG, method="none")
plotMA(RGnb)
```

We have to figure out how many spots there are on the array and how many pins (print head tips) were used to print it:

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

Have a look at spatial variation of background on the plot:

```
imageplot(RG$Rb, RG$printer, low="white", high="red")
imageplot(RG$Gb, RG$printer, low="white", high="green")
```

Print-tip loess normalization:

```
MA <- normalizeWithinArrays(RG)
plotMA(MA)
```

Not very satisfactory! Try instead

```
RGnb$printer <- RG$printer
MA <- normalizeWithinArrays(RGnb)
plotMA(MA)
```

2. One-Sample Experiments

Swirl Zebrafish Data

The zip file associated with this experiment is swirl.zip. This file should be unpacked into your working directory, assumed to be c:\R. The data consists of a GAL file fish.gal and four SPOT output files.

Background. The experiment was carried out using zebrafish as a model organism to study the early development in vertebrates. Swirl is a point mutant in the BMP2 gene that affects the dorsal/ventral body axis. The main goal of the Swirl experiment is to identify genes with altered expression in the Swirl mutant compared to wild-type zebrafish.

The hybridizations. Two sets of dye-swap experiments were performed making a total of four replicate hybridizations. Each of the arrays compares RNA from swirl fish with RNA from normal ("wild type") fish. The experimenters have prepared a tab-delimited targets file called "SwirlSamples.txt" which describes the four hybridizations:

```
targets <- readTargets("SwirlSample.txt")
targets
```

You'll see that slide numbers 81, 82, 93 and 94 were used to make the arrays. On slides 81 and 93, swirl RNA was labelled with green (Cy3) dye and wild type RNA was labelled with red (Cy5) dye. On slides 82 and 94, the labelling was the other way around.

Each of the four hybridized arrays was scanned on an Axon scanner to produce a TIFF image, which was then processed using the image analysis software SPOT. The data from the arrays are stored in the four output files listed under `FileName`. Now we read the intensity data into an `RGList` object in R. The default for SPOT output is that `Rmean` and `Gmean` are used as foreground intensities and `morphR` and `morphG` are used as background intensities:

```
RG <- read.maimages(targets$FileName, source="spot")
RG
```

The data object you have read in behaves like a complex sort of matrix. You can subset it or treat it like a matrix in lots of ways. Try

```
dim(RG)
nrow(RG)
RG[1,]
RG[,c(1,3)]
cbind(RG[,1],RG[,3])
```

The arrays. The microarrays used in this experiment were printed with 8448 probes (spots), including 768 control spots. The array printer uses a print head with a 4x4 arrangement of print-tips and so the microarrays are partitioned into a 4x4 grid of tip groups. Each grid consists of 22x24 spots that were printed with a single print-tip. The gene name associated with each spot is recorded in a GenePix array list (GAL) file:

```
RG$genes <- readGAL("fish.gal")
RG$genes[1:30,]
```

The 4x4x22x24 print layout also needs to be set. The easiest way to do this is to infer it from the GAL file:

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

Normalization. Print-tip loess normalization. Now we plot the individual MA-plots for each of the print-tip groups on this array, together with the loess curves which will be used for normalization:

```
plotPrintTipLoess(RG)
MA <- normalizeWithinArrays(RG)
plotPrintTipLoess(MA)
```

We have normalized the M-values with each array. A further question is whether normalization is required between the arrays. The following plot shows overall boxplots of the M-values for the four arrays.

```
boxplot(MA$M~col(MA$M),names=colnames(MA$M))
```

There is some evidence that the different arrays have different spreads of M-values, so we will scale normalize between the arrays.

```
MA <- normalizeBetweenArrays(MA)
boxplot(MA$M~col(MA$M),names=colnames(MA$M))
```

Linear model. Now estimate the average M-value for each gene. We do this by fitting a simple linear model for each gene. The negative numbers in the design matrix indicate the dye-swaps.

```
design <- c(-1,1,-1,1)
fit <- lmFit(MA, design)
fit
```

In the above fit object, `coefficients` is the average M-value for each gene and `sigma` is the sample standard deviations for each gene. Ordinary t-statistics for comparing mutant to wt could be computed by

```
ordinary.t <- fit$coef / fit$stdev.unscaled / fit$sigma
```

Empirical Bayes analysis. We prefer though to use empirical Bayes moderated t-statistics. The moderated t-statistics use sample standard deviations which have been shrunk towards a pooled standard deviation value.

```
fit <- eBayes(fit)
fit
```

Notice the estimated hyperparameters which have been added to the fit.

```
options(digits=3)
topTable(fit)
topTable(fit, number=30, adjust="fdr")
     Block Row Column      ID   Name     M    A     t  P.Value    B
3721     8   2      1 control   BMP2 -2.21 12.1 -21.1 0.000357 7.96
1609     4   2      1 control   BMP2 -2.30 13.1 -20.3 0.000357 7.78
3723     8   2      3 control   Dlx3 -2.18 13.3 -20.0 0.000357 7.71
1611     4   2      3 control   Dlx3 -2.18 13.5 -19.6 0.000357 7.62
8295    16  16     15 fb94h06 20-L12  1.27 12.0  14.1 0.002067 5.78
7036    14   8      4 fb40h07  7-D14  1.35 13.8  13.5 0.002067 5.54
515      1  22     11 fc22a09 27-E17  1.27 13.2  13.4 0.002067 5.48
5075    10  14     11 fb85f09 18-G18  1.28 14.4  13.4 0.002067 5.48
7307    14  19     11 fc10h09 24-H18  1.20 13.4  13.2 0.002067 5.40
319      1  14      7 fb85a01  18-E1 -1.29 12.5 -13.1 0.002067 5.32
2961     6  14      9 fb85d05 18-F10 -2.69 10.3 -13.0 0.002067 5.29
4032     8  14     24 fb87d12 18-N24  1.27 14.2  12.8 0.002067 5.22
6903    14   2     15 control    Vox -1.26 13.4 -12.8 0.002067 5.20
4546     9  14     10 fb85e07 18-G13  1.23 14.2  12.8 0.002067 5.18
683      2   7     11 fb37b09  6-E18  1.31 13.3  12.4 0.002182 5.02
1697     4   5     17 fb26b10  3-I20  1.09 13.3  12.4 0.002182 4.97
7491    15   5      3 fb24g06  3-D11  1.33 13.6  12.3 0.002182 4.96
4188     8  21     12 fc18d12 26-F24 -1.25 12.1 -12.2 0.002209 4.89
```

```
4380     9   7     12 fb37e11  6-G21  1.23 14.0  12.0 0.002216 4.80
3726     8   2      6 control  fli-1 -1.32 10.3 -11.9 0.002216 4.76
2679     6   2     15 control    Vox -1.25 13.4 -11.9 0.002216 4.71
5931    12   6      3 fb32f06  5-C12 -1.10 13.0 -11.7 0.002216 4.63
7602    15   9     18 fb50g12  9-L23  1.16 14.0  11.7 0.002216 4.63
2151     5   2     15 control   vent -1.40 12.7 -11.7 0.002216 4.62
3790     8   4     22 fb23d08  2-N16  1.16 12.5  11.6 0.002221 4.58
7542    15   7      6 fb36g12  6-D23  1.12 13.5  11.0 0.003000 4.27
4263     9   2     15 control   vent -1.41 12.7 -10.8 0.003326 4.13
6375    13   2     15 control   vent -1.37 12.5 -10.5 0.004026 3.91
1146     3   4     18 fb22a12  2-I23  1.05 13.7  10.2 0.004242 3.76
157      1   7     13 fb38a01   6-I1 -1.82 10.8 -10.2 0.004242 3.75
```

The top gene is BMP2 which is significantly down-regulated in the Swirl zebrafish, as it should be because the Swirl fish are mutant in this gene. Other positive controls also appear in the top 50 genes in terms.

In the table, t is the empirical Bayes moderated t-statistic, the corresponding P-values have been adjusted to control the false discovery rate and B is the empirical Bayes log odds of differential expression. Beware that the Benjamini and Hochberg method used to control the false discovery rate assumes independent statistics which we do not have here (see `help(p.adjust)`).

3. Two-Sample Experiments

ApoAI Knockout Data:

The zip file associated with this experiment is apoai.zip. This contains the binary data file ApoAI.RData which should be unpacked into your working directory. You can then load the data in your R session as described below or start a new R session by clicking on the file ApoAI.RData in Windows Explorer.

Background. The data is from a study of lipid metabolism by Callow et al (2000). The apolipoprotein AI (ApoAI) gene is known to play a pivotal role in high density lipoprotein (HDL) metabolism. Mice which have the ApoAI gene knocked out have very low HDL cholesterol levels. The purpose of this experiment is to determine how ApoAI deficiency affects the action of other genes in the liver, with the idea that this will help determine the molecular pathways through which ApoAI operates.

Hybridizations. The experiment compared 8 ApoAI knockout mice with 8 normal C57BL/6 ("black six") mice, the control mice. For each of these 16 mice, target mRNA was obtained from liver tissue and labelled using a Cy5 dye. The RNA from each mouse was hybridized to a separate microarray. Common reference RNA was labelled with Cy3 dye and used for all the arrays. The reference RNA was obtained by pooling RNA extracted from the 8 control mice.

Number of arrays	Red	Green
8	Normal "black six" mice	Pooled reference
8	ApoAI knockout	Pooled reference

This is an example of a single comparison experiment using a common reference. The fact that the comparison is made by way of a common reference rather than directly as for the swirl experiment makes this, for each gene, a two-sample rather than a single-sample setup.

Load the data into you session by typing:

```
load("ApoAI.RData")
```

This contains just one R object called `RG`. Try to following commands:

```
names(RG)
RG
show(RG)
RG$targets
designMatrix(RG$targets, ref="Pool")
```

Firstly we'll print-tip loess normalize the data for each array:

```
MA <- normalizeWithinArrays(RG)
MA
```

There are lots of ways to construct a design matrix for this experiment but we will use a method that generalizes to more complex experiments:

```
design <- designMatrix(RG$targets, ref="Pool")
design
fit <- lmFit(MA, design)
fit
```

This estimates the average difference between C56BL/6 and the Pooled Reference and between ApoAI-/- and the Pooled Reference for each gene. We want to test for a difference between these two coefficients for each gene, i.e., we want to test the `c(-1,1)` contrast equal to zero.

```
cont.matrix <- cbind("KO-WT"=c(-1,1))
rownames(cont.matrix) <- colnames(design)
cont.matrix
fit2 <- contrasts.fit(fit, cont.matrix)
fit2
fit2 <- eBayes(fit2)
fit2
fit2 <- ebayes(fit2)
options(digits=2)
topTable(fit2)
topTable(fit2,adjust="fdr")
                     NAME TYPE    CLID          ACC     M  A     t P.Value     B
2149       ApoAI,lipid-Img cDNA 1077520              -3.17 12 -24.0 3.0e-11 14.93
540   EST,HighlysimilartoA cDNA  439353              -3.05 12 -13.0 5.0e-07 10.81
5356  CATECHOLO-METHYLTRAN cDNA 1350232              -1.85 13 -12.4 6.5e-07 10.45
4139  EST,WeaklysimilartoC cDNA  374370              -1.03 13 -11.8 1.2e-06  9.93
1739     ApoCIII,lipid-Img cDNA  483614              -0.93 14  -9.8 1.6e-05  8.19
2537  ESTs,Highlysimilarto cDNA  483614              -1.01 14  -9.0 4.2e-05  7.30
1496                  est cDNA  484183 genome.wustl -0.98 12  -9.0 4.2e-05  7.29
4941  similartoyeaststerol cDNA  737183              -0.95 13  -7.4 5.6e-04  5.31
947   EST,WeaklysimilartoF cDNA  353292              -0.57 11  -4.6 1.8e-01  0.56
5604                       cDNA  317638              -0.37 13  -4.0 5.3e-01 -0.55
```

Notice that the top gene is ApoAI itself which is heavily down-regulated. Theoretically the M-value should be minus infinity for ApoAI because it is the knockout gene. Several of the other genes are closely related. The top eight genes here were confirmed by independent assay subsequent to the microarray experiment to be differentially expressed in the knockout versus the control line.

Acknowledgements

Thanks to Yee Hwa Yang and Sandrine Dudoit for the Swirl and ApoAI data sets. The Swirl zebrafish data were provided by Katrin Wuennenburg-Stapleton from the Ngai Lab at UC Berkeley.

References

1. Callow, M. J., Dudoit, S., Gong, E. L., Speed, T. P., and Rubin, E. M. (2000). Microarray expression profiling identifies genes with altered expression in HDL deficient mice. *Genome Research* **10**, 2022-2029. (Full Text)