

# limma: Assessing differential expression

Natalie P. Thorne

October 13, 2008

## Detecting differential expression

■ With only a single microarray experiment, detecting differential expression amounts to selecting those genes whose  $\log(R)$  and  $\log(G)$  intensities are distinctly different. In general, arbitrary cut-offs such as two-fold ( $M = 1$ ) differential expression are applied to single microarray experiments. However, when there are no repeats of the experiment, it is hard to judge whether observed differential expression is real or not. Indeed, it is only when there are replicates that we can begin to objectively assess and better judge the genes that might truly be differentially expressed as opposed to those appearing differentially expressed due to some spurious effect from an artifact or something similar. Statistical approaches can be used to formalise this process of judging the evidence for differential expression from replicate arrays for each gene. The simplest statistic for assessing evidence for differential expression is the average (or equivalently the median). In the long run, when there are many replicates, the average is very useful, particularly when standardised by the standard error and distributional assumptions can be employed to assess significance. However, microarray experiments, are rarely repeated many times (usually only 2-8 replicates are done). Moreover expression arrays tend to be very noisy. Couple highly variable measurements with low numbers of replicates and thousands of genes and you have a challenging statistical problem to identify the real differentially expressed genes. In fact, what we realistically aim to achieve is a sensible ranking of genes based on evidence for differential expression. The approach in `limma` incorporates average differential expression and a measure of variability similar to that used in a t-statistic (i.e. standard error). However the genewise standard errors are modified.

**Exercise 1:** What is the main reason why ordinary t-statistics are not suitable for assessing differential expression of genes in microarray experiments? Describe, in general terms, how the standard errors are modified in `limma`.

## Simple replicated design

■ In the next few sections we will look at various data sets to find the differentially expressed genes in each. You will begin to get a practical sense of how to assess differential expression. Read the chapter in the `limma` User's Guide on **Statistics for Differential Expression**. In this lab we will look mostly at the B-statistic rather than the suggested modified p-values. For our purposes, it doesn't make too much difference (to the ranking

obtained on DE for the genes), just bear in mind that we have made numerous model assumptions in order to fit the B-statistic (including the assumption of the proportion of genes we expect to be differentially expressed). We use  $B$  to rank genes, rather than for a literal interpretation of the log odds of being differentially expressed.

**Exercise 2:** Change the working directory in R to the `data` folder. Use `load` to get the swirl data object contained in this folder. The object you just loaded is called `MA` and contains the targets information, spot types, gene annotation, weights etc and has been background corrected and normalised. Look at the `targets` in this object, look at the `Cy3` and `Cy5` columns to remind yourself what samples were hybridised to each array and in what dye orientation. Which are the dye-swaps? Draw a diagram of the experimental design. There are two samples, so there can only be one parameter. Write down the design matrix for this data for the `Sw-Wt` effect. Now make this `design` matrix in R. Fit the linear model which estimates this parameter for each gene.

```
> load("swirl.RData")
> library(limma)
> design = c(-1, 1, -1, 1)
> fit = lmFit(MA, design)
> names(fit)
```

☞ In the above fit object, `coefficients` is a matrix of the parameter estimates obtained from fitting the linear model. For this model, only one parameter was estimated, so there is only one column in `coefficients`. `sigma` is the sample standard deviation for each gene. Ordinary t-statistics for comparing swirl to wt could be computed. Do you remember the main pitfall of using ordinary t-statistic? Below we obtain the ordinary t-statistics and get an index (numbers indicating the gene rows) for the top 200 most extreme t-statistics. These are the 200 genes with high evidence for differential expression according to the t-statistic.

```
> ordinary.t = fit$coef/fit$stdev.unscaled/fit$sigma
> topt = order(abs(ordinary.t), decreasing = T)[1:200]
> topt
```

**Exercise 3:** Make a volcano plot and highlight the 200 genes based on the B-statistic, then make another volcano plot (for comparison) this time highlighting the 200 highest absolute ordinary t-statistics.

☞ The moderated t-statistics use sample standard deviations which have been shrunk towards a pooled standard deviation value. This modified t-statistic is the t-statistic you find in the linear model fit object (not the ordinary t-statistic which we had to calculate by hand). The lods score (or B-statistic) calculated the log odds of differential expression versus no differential expression. Therefore, a B value less than zero has the interpretation that there is more evidence against being differentially expressed than there is for being differentially expressed. However, this interpretation cannot be taken literally. There are many assumptions used in the calculation of this statistic that do not hold. In particular, the assumption that the proportion of differentially expressed genes is 0.01 will most likely not be true. Therefore, the location of the true zero for the B-statistics may be further up or down the volcano

plot. However, despite not being able to interpret the B-statistic literally, the actual ranking is informative.

```
> fit.eb = eBayes(fit)
> names(fit.eb)
> par(mfrow = c(1, 2))
> volcanoplot(fit.eb, highlight = 200)
> plot(fit.eb$coef, fit.eb$lods, pch = 16, cex = 0.4, xlab = "avM", ylab = "B")
> points(fit.eb$coef[topt], fit.eb$lods[topt], pch = 16, cex = 0.4, col = "blue")
```

☞ Note that the points in the volcano plot might not display correctly if you are working with X Window System (X11) on Mac or Linux. If that is the case, try the command `x11(pointsize=20)`.

☞ We have plotted (against the average M), the B-statistic which is the empirical Bayes log odds of differential expression. This kind of plot is often referred to as the “volcano” plot. On the right, we highlighted the top 200 ordinary t-statistics. Notice that in many cases the ordinary t-statistic selects genes with small average M-values. If you investigate further you will find that these genes have very small sample standard deviations. If we focus on selecting genes using the B-statistic (as in the left plot), you will notice that some genes with large absolute average M-values have low B-values. These genes are presumably quite variable across replicate experiments resulting in too much uncertainty in their average M-value.

☞ The issues surrounding multiple testing and finding DE genes in microarray experiments are quite important. However, we will not spend much time on false discovery rates and methods for adjusting p-values. Such methods are implemented in `limma` and you can read about their use in the `limma` guide. In practise it is ok to simply rank genes by the B-statistic or the modified t-statistic and select a suitable number of genes for follow up and validation. Settling on a certain list of differentially expressed genes will obviously be dependent on how many genes “appear” to be differentially expressed, and the nature of the follow up on these genes.

**Exercise 4:** Make volcano plots highlighting different numbers of genes in each. As you increase the number of genes highlighted on the plot, at what point do you think you begin to select genes with rather low average M-values? Make a conservative decision about the number of genes you would feel comfortable following up (validating) as *differentially expressed*. For this number of genes, what is the corresponding B-value?

```
> par(mfrow = c(2, 2), mar = c(2, 2, 2, 2))
> volcanoplot(fit.eb, highlight = 30)
> volcanoplot(fit.eb, highlight = 100)
> volcanoplot(fit.eb, highlight = 150)
> volcanoplot(fit.eb, highlight = 300)
```


**Exercise 5:** Lets look at the top 150 genes selected by the B-statistic. Now write this table to a file.

```
> options(digits = 3)
> topTable(fit.eb, number = 150)
> tt = topTable(fit.eb, number = 150)
> write.table(tt, "swirltopable.txt", sep = "\t")
```

**Exercise 6:** Save the current workspace for the swirl analysis, call the workspace `SwirlAnalysis.RData`. Then use the command below to remove all objects from the workspace before starting the next data set.

```
> save.image("SwirlAnalysis.RData")
> rm(list = ls(all = TRUE))
> objects()
```

## Two-sample common reference

 We now consider the APOAI experiment, another familiar data set, which formed the basis of an early paper on differential expression. Mice with the apolipoprotein AI (ApoAI) gene knocked out are compared to wild type (or control) mice. You can read about the design of the experiment in the `limma` User's Guide. Basically, eight KO and eight WT mice are compared via a common reference of a pool of the eight WT mice.

**Exercise 7:** Load the ApoAI data set, look at the targets information in the `RG` object. Draw the experimental design, and make the design matrix for the parameters `KO-Ref` and `WT-Ref`. Normalise the data and fit the linear model. Then perform empirical Bayes smoothing of the standard errors and produce the volcano plot for each parameter. Usually, much more extensive exploratory data analysis would be performed before doing analyses such as linear model fitting. However for the purpose of this lab, we focus on finding differentially expressed genes.

```
> load("ApoAI.RData")
> RG$design = cbind("KO-Ref" = c(rep(0, 8), rep(1, 8)), "WT-Ref" = c(rep(1,
+      8), rep(0, 8)))
> RG$design
> MA = normalizeWithinArrays(RG)
> names(MA)
> fit = lmFit(MA, design = RG$design)
> colnames(fit$coef)
> fit.eb = eBayes(fit)
> volcanoplot(fit.eb)
```

**Exercise 8:** Make a matrix of the contrasts you are interested in. The contrasts are arithmetic combinations of the parameters estimated in the model. The contrast matrix must have `number(rows)=number(coef)` in the linear model. Here we have fitted two effects in the linear model, `KO-Ref` and `WT-Ref`. The contrast of interest is `KO-WT=KO-Ref-(WT-Ref)`, but we can also include the original two effects in the contrast matrix (as shown below). Each column in the contrast matrix corresponds to a different contrast of interest where the rows

correspond to the parameters estimated by the linear model fit and design matrix. A contrast, in the contrast matrix, consists of coefficients representing a linear combination of the effects (parameters) in the linear model fit.

```
> cont.matrix = cbind(KO = c(1, 0), WT = c(0, 1), "KO-WT" = c(1, -1))
> cont.matrix
> cont.matrix = as.matrix(cbind("KO-WT" = c(1, -1)))
> cont.matrix
> fit2 = contrasts.fit(fit, cont.matrix)
> fit2.eb = eBayes(fit2)
> topTable(fit2.eb, coef = 1, number = 10)
> topTable(fit2.eb, coef = 1, number = 10, resort.by = "M")
```

☞ Note that the last command might not work correctly if you are working with an R console on a Mac.

**Exercise 9:** Save the current workspace for the ApoAI analysis, call the workspace `ApoAnalysis.RData`. Then use the command below to remove all objects from the workspace before starting the next data set.

```
> save.image("ApoAnalysis.RData")
> rm(list = ls(all = TRUE))
```

## Dye-swap mistakes

☞ In the following experiment, the biological scientist provided a `targets.txt` file describing the hybridisations that had been performed. In each array a cell line was compared before (`bf`) and after (`af`) a treatment inducing the cells to transform to a morphologically more differentiated state. Three biological replicates of the before/after comparison were done in dye-swap. Again these data (and any accompanying files) were read into R using `limma` and the session saved prior to this lab. The data are provided as an `RGList` object, called `RG` in this lab.

**Exercise 10:** Look at the targets information and confirm for yourself that the design matrix for the `af-bf` parameter is correct. Make the `volcanoplot` for the parameter estimated in the linear model fit. Inspect the plot carefully; why might you suspect a problem with this array data?

```
> load("RG1.RData")
> MA.nb = MA.RG(RG, bc.method = "none")
> MA = normalizeWithinArrays(MA.nb, method = "loess")
> MA$targets
> MA$design = c(1, 1, 1, -1, -1, -1)
> MA$design
> fit = lmFit(MA)
> fit.eb = eBayes(fit)
```

```
> par(mfrow = c(1, 1))
> volcanoplot(fit.eb, highlight = 30, names = RG$genes$GeneName)
```

**Exercise 11:** Make MA-plots to check the quality of the data from each slide. The slides used for this experiment are commercial **Agilent** arrays. Notice that the MA-plots for these data seem to be considerably less noisy than observed in the cDNA array data from the **swirl** experiment.

```
> par(mfcol = c(3, 2), mar = c(1, 1, 1, 1))
> for (i in 1:6) {
+   plotMA(MA.nb, array = i, cex = 0.5, main = colnames(MA)[i])
+ }
```

☞ Based on MA-plots of the data, it is clear that the data appear to be of high quality. The first array has more intensity dependent dye-bias than all the other arrays, however this is not considered too bad, especially when there are 6 replicates available. In addition the background levels are low (and very consistent) compared to the foreground intensities, so any spatial trends in the background will be of little consequence.

**Exercise 12:** Make the boxplot summaries of the foreground and background for the red and green channels to confirm that the background levels are very low for these Agilent arrays.

```
> par(mfrow = c(2, 2), mar = c(5, 4, 4, 2))
> boxplot(log2(RG$Rb) ~ col(RG$Rb), col = "red", ylim = c(4, 16))
> boxplot(log2(RG$R) ~ col(RG$R), col = "red", ylim = c(4, 16))
> boxplot(log2(RG$Gb) ~ col(RG$Gb), col = "green", ylim = c(4, 16))
> boxplot(log2(RG$G) ~ col(RG$G), col = "green", ylim = c(4, 16))
```

**Exercise 13:** A error in the dye-swaps was suspected. Perform a simple hierarchical clustering of the log-ratios of the slides (normalised, background corrected data). First calculate the pairwise distance (or similarity) measure between the log-ratios of the different arrays. The function `dist` finds distances between rows of a matrix, so we supply the transpose (rows become columns and columns become rows) of the matrix of M-values (to calculate the distances between the arrays). Study the cluster plot carefully and deduce which arrays might be dye-swap mistakes (dye labelled in the un-intended orientation).

☞ Based on the targets file, it seems clear that the experimenter planned to do the first set of three biological replicates in a fixed labelling orientation. The remaining three slides were intended to be dye-swap technical replicates of the first three slides.

```
> dist.matrix = dist(t(MA$M))
> hc = hclust(dist.matrix)
> par(mfrow = c(1, 1))
> plot(hc)
```

**Exercise 14:** Refit the linear model with the corrected design matrix that accounts for the dye-swap errors suspected in the data.

```

> design2 = c(1, 1, -1, 1, -1, -1)
> design2
> fit2 = lmFit(MA, design2)
> fit2.eb = eBayes(fit2)
> par(mfrow = c(1, 2))
> volcanoplot(fit.eb, highlight = 30, ylim = c(-10, 20), names = RG$genes$GeneName)
> volcanoplot(fit2.eb, highlight = 30, ylim = c(-10, 20), names = RG$genes$GeneName)

```

Based on the volcano plots for the original and new dye label orientations, we can be fairly confident that we have deduced the corrected assignment of dye-swaps for these arrays. But there is still another feature of this data that doesn't seem satisfactory. Some asymmetry is apparent in the volcano plot and we need to check whether this has resulted from a dye effect that has remained even after normalisation.

**Exercise 15:** Re-fit the corrected design, this time add an effect for dye. In this model you will estimate the log-ratio parameter for the difference between the red and green channel, regardless of what samples are hybridised to each channel.

The dye parameter will estimate the effect of dye for each gene. Genes with a large value for this parameter are those which always have higher intensity in one channel compared to the other, regardless of the sample hybridised in each channel. Such genes have a dye effect that cannot be removed through normalisation - they have a gene specific dye effect that is additional to any general array specific dye effect (removed by normalisation).

There are certainly some genes showing evidence of systematic intensity bias due to dye, however the overall effect of dye is not too strong in most genes.

```

> design3 = cbind(dye = c(1, 1, 1, 1, 1, 1), "bf-af" = c(1, 1,
+   -1, 1, -1, -1))
> design3
> fit3 = lmFit(MA, design3)
> fit3.eb = eBayes(fit3)
> par(mfrow = c(1, 3))
> volcanoplot(fit2.eb, highlight = 30, ylim = c(-10, 20), main = "no dye effect fitted",
+   names = RG$genes$GeneName)
> volcanoplot(fit3.eb, coef = 1, highlight = 30, ylim = c(-10,
+   20), main = "dye effect", names = RG$genes$GeneName)
> volcanoplot(fit3.eb, coef = 2, highlight = 30, ylim = c(-10,
+   20), main = "bf-af effect after fitting dye effect", names = RG$genes$GeneName)

```

**Exercise 16:** Save the current workspace for this analysis, call the workspace `RG1Analysis.RData`. Then use the command below to remove all objects from the workspace before starting the next data set.

```

> save.image("RG1Analysis.RData")
> rm(list = ls(all = TRUE))

```

## Biological variability

**I** This next data set is an array experiment looking at the transcriptional changes due to knocking out a gene in a certain mouse strain. The procedure of knocking out the gene was performed twice by the biologist to assess the extent to which the knock-out process was successful. Thus we fit an effect separately for each biological replicate to check the overall similarity of the two KO's before combining them.

**Exercise 17:** Fit a linear model to the data with parameters KO1-WT and KO2-WT. Look at the targets file and draw a picture of the experimental design. Write down the design matrix for yourself and check this against the one created in the exercise below.

```
> load("RG2.RData")
> RG$design <- cbind("KO1-WT" = c(1, 1, 1, -1, -1, -1, 0, 0, 0,
+   0, 0, 0), "KO2-WT" = c(0, 0, 0, 0, 0, 0, 1, 1, 1, -1, -1,
+   -1))
> RG$design
> MA.b = MA.RG(RG, bc.method = "minimum")
> MA = normalizeWithinArrays(RG, bc.method = "minimum")
> fit = lmFit(MA, design = RG$design)
> contr.matrix = cbind(KO1 = c(1, 0), KO2 = c(0, 1), "KO1-KO2" = c(1,
+   -1), "(KO1+KO2)/2" = c(0.5, 0.5))
> contr.matrix
> fit2 = contrasts.fit(fit, contr.matrix)
> fit2.eb = eBayes(fit2)
> par(mfrow = c(2, 2))
> for (i in 1:4) {
+   volcano(fit2.eb, coef = i, main = colnames(contr.matrix)[i],
+     ylim = c(-10, 20), xlim = c(-2, 2))
+ }
```

**I** There appears to be considerable expression differences between KO1 and KO2. The question is as follows: is this biological variance? The MA-plots of the array data from this experiment seem to suggest otherwise.

**Exercise 18:** Make MA-plots for arrays 1 to 6; these are the plots for the KO1 replicates. Then make MA-plots for arrays 7-12 for the KO2 replicates. Is there any noticeable difference in the appearance of the MA-plots for the KO1 and KO2 experiments?

```
> par(mfrow = c(3, 2), mar = c(1, 1, 1, 1))
> for (i in 1:6) {
+   plotMA(MA.b, array = i, cex = 0.5, xlim = c(6, 16), ylim = c(-2, 2))
+ }
> par(mfrow = c(3, 2), mar = c(1, 1, 1, 1))
> for (i in 7:12) {
+   plotMA(MA.b, array = i, cex = 0.5, xlim = c(6, 16), ylim = c(-2, 2))
+ }
```



**Exercise 19:** Make an `imageplot` and a `heatDiagram` for the KO1 and KO2 coefficients. Describe the main features of any differences you notice.

```
> par(mfrow = c(1, 2))
> imageplot(fit2.eb$coef[, 1], layout = RG$printer)
> imageplot(fit2.eb$coef[, 2], layout = RG$printer)
> results = decideTests(fit2.eb)
> par(mfrow = c(1, 1))
> heatDiagram(results[, 1:2], fit2.eb[, 1:2])
```

■✎ The MA-plots for the KO1 vs WT arrays have a systematically different intensity dependent dye-bias than the KO2 vs WT arrays. The data quality for the KO2 vs WT arrays are questionable and should perhaps be repeated. It is always important to look for effects that may confound biological effects of interest. Here the investigator was interested in detecting any differences between KO1 and KO2 mice strains. However many of the observed differences in this experiment may be due to technical difficulties which yielded poor quality data for the KO2 strain.

**Exercise 20:** Save the current workspace for this analysis, call the workspace `RG2Analysis.RData`. Then use the command below to remove all objects from the workspace before quitting R.

```
> save.image("RG2Analysis.RData")
> rm(list = ls(all = TRUE))
```