

R/BioC Exercises: Linear models and differential expression

Perry Moerland

April 15, 2010

☞ Information on how to log on to a PC in the exercise room and the UNIX server can be found here: <http://bioinformaticslaboratory.nl/twiki/bin/view/BioLab/EducationBioinformaticsII>. Don't forget to enable X11 forwarding when starting up PuTTY.

☞ Background reading for these exercises is Chapter 23 (Smyth 2005) in the course reader. Further information can be found at <http://bioinformaticslaboratory.nl/twiki/bin/view/BioLab/EducationBioinformaticsIILab3>

1 Introduction

Many microarray studies are designed to detect genes associated with different phenotypes, for example, the comparison of cancer tumors to normal cells. In the more complex experiments genetic networks are perturbed with various treatments to understand the effects of these treatments and their interactions with each other in a dynamic cellular network. This tutorial presents some commonly used experimental designs used to explore differential expression of genes between phenotypic groups and their analyses using linear models via the *limma* package.

2 Simple comparisons

Assume that we have three replicate two-color arrays comparing the same two RNA sources: wild-type (wt, Cy3) and mutant (mu, Cy5) RNA. For this three-array experiment, the targets frame may be represented as

FileName	Cy3	Cy5
File1	wt	mu
File2	wt	mu
File3	wt	mu

Exercise 1: Suppose that we are mainly interested in the fold change (=ratio) information. What does a negative value of M imply? What does a positive value of M imply? Suggest a possible test of hypotheses that may be performed on the M -values.

A very general way of analyzing microarray experiments uses *linear models* (Smyth 2005). The idea is that an experiment can be concisely described using matrices and simple linear algebra. In our three-array experiment this can be seen as follows. The comparison of interest

is the mean log-ratio $\log_2(\mu/wt) = M$. Let y_{gi} represent the log-ratio of gene g on array i . Then we can write

$$\begin{pmatrix} y_{g1} \\ y_{g2} \\ y_{g3} \end{pmatrix} = \begin{pmatrix} M_g \\ M_g \\ M_g \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} M_g = X\alpha_g.$$

Indeed, $(y_{g1}, y_{g2}, y_{g3}) = X\alpha_g$ is a linear model. The matrix - or in this case vector - X is the *design matrix*. Each row of the design matrix corresponds to an array and each column corresponds to a *coefficient* which is used to describe the different RNA targets that have been hybridised to the arrays. In this example the design matrix is a single column of ones because each array compares the same two RNA targets: wild-type (Cy3) and mutant (Cy5). The coefficient α_g is the mean log-ratio to estimate. The coefficients α_g are in general estimated using ordinary least squares.

The linear model defined above can be used to perform a classical single-sample t -test. Suppose variable `MA` contains pre-processed data. The following commands may be used to fit the linear model

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

As we will see later on, the `fit` object of class *MArrayLM* then contains all elements needed to compute t -statistics and the corresponding p -values.

3 Two groups

Valk et al. (2004) determined the gene-expression profiles in samples of peripheral blood or bone marrow from 285 patients with acute myeloid leukemia (AML). Unsupervised cluster analyses identified 16 groups of patients with AML on the basis of molecular signatures. The clustering was driven by the presence of chromosomal lesions, particular genetic mutations, and abnormal oncogene expression. A unique cluster with a distinctive gene-expression signature included cases of AML with a poor treatment outcome.

We will import the normalized Affymetrix (single-channel) gene expression data for a subset of the AML data as a matrix into R. This dataset contains 2856 transformed expression values (after selection) for 20 AML samples (11-30), the first ten belonging to Cluster 12 and the second ten belonging to Cluster 3. We will identify genes that are differentially expressed between these two clusters.

You can access the expression data by logging on to SARA and then copying the data to your account (the dot at the end of the command line is part of the command)

```
stud01@amc-app1:~$ cp -r /data/home/stud00/Valk .
stud01@amc-app1:~$ cd Valk
```

Now we are ready to start R:

```
stud01@amc-app1:~/Valk$ R 2.10
```

Execute the following commands to read in the data

```

> dataset <- read.table("valk_finding.txt", row.names = 1, header = TRUE)
> ngenes <- nrow(dataset)
> dim(dataset)

[1] 2856  20

> dataset[1:5, 1:5]

```

	Cluster12_11	Cluster12_12	Cluster12_13	Cluster12_14	Cluster12_15
117_at	3.475390	2.555139	2.880877	2.498155	3.605545
1405_i_at	3.360019	4.328886	4.715796	4.224703	3.582136
1598_g_at	4.767637	4.099571	3.988363	4.013836	4.918065
200067_x_at	6.081005	6.122393	5.470254	5.762660	5.886500
200075_s_at	5.285296	5.583601	5.075593	5.312251	4.680153

Here the expression data is just a data frame with each column corresponding to the log₂-intensities of an array. A two-sample *t*-test is then a simple way of identifying genes that are differentially expressed between Cluster 12 and Cluster 3.

Exercise 2: Use the function *rowttests* in the package *genefilter* to perform row-by-row (=gene-by-gene) tests for a significant difference in mean expression between the two clusters. In this case, you can use the information about cluster membership in the column names of *dataset* to define the groups.

```

> library(genefilter)

```

Take a look at the histogram of resulting *p*-values using the function *hist*. How many genes are differentially expressed, say with a *p*-value < 0.05?

From the lectures last week you probably remember that performing a large number of hypothesis tests (2856 in this case) potentially leads to a large number of falsely significant genes. Many methods have been devised to deal with this problem of *multiple testing* and some have been implemented in R.

Exercise 3: Use the function *p.adjust* to either control the family-wise error rate using the Bonferroni correction or the false discovery rate with Benjamini-Hochberg. Plot both histograms of the resulting corrected *p*-values and explain the differences between them.

Exercise 4: Instead of doing a two-sample *t*-test ourselves, we could also fit a linear model to the data. Try to specify the design matrix for this experiment. Remember that each row of the design matrix corresponds to an array in the experiment and each column corresponds to a coefficient which is used to describe the RNA sources in the experiment. How many rows does your design matrix need to have and how many columns?

We create a design matrix which includes separate coefficients for Cluster 12 (C12) and Cluster 3 (C3).

```
> library(limma)
> design <- cbind(C12 = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
+ 0, 0, 0, 0, 0, 0, 0), C3 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1))
> colnames(design) <- c("C12", "C3")
> rownames(design) <- c(11:30)
> design
```

	C12	C3
11	1	0
12	1	0
13	1	0
14	1	0
15	1	0
16	1	0
17	1	0
18	1	0
19	1	0
20	1	0
21	0	1
22	0	1
23	0	1
24	0	1
25	0	1
26	0	1
27	0	1
28	0	1
29	0	1
30	0	1

Here the first coefficient (=column of design matrix) estimates the mean log-intensity for Cluster 12 and the second coefficient estimates the mean log-intensity for Cluster 3. To find differentially expressed genes, we have to make a *contrast matrix*, which allows the coefficients defined by the design matrix to be combined into contrasts of interest. The contrasts are arithmetic combinations of the parameters estimated in the model. The contrast matrix must have a number of rows equal to the number of coefficients in the linear model. 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. A contrast, in the contrast matrix, consists of a linear combination of the effects (coefficients) in the linear model. In this dataset we are interested in the difference between Cluster 12 and Cluster 3 as a contrast.¹

¹Since we always work on a log scale and because $\log_2(C12) - \log_2(C3) = \log_2(C12/C3)$ this indeed represents the log-ratio.

```

> fit <- lmFit(dataset, design)
> cont.matrix <- makeContrasts(C12vsC3 = C12 - C3, levels = design)
> fit2 <- contrasts.fit(fit, cont.matrix)

```

The objects `fit` and `fit2` contain several items that might be of interest, including model coefficient estimates and standard errors. In order to find out what `fit` contains, use

```

> names(fit)

 [1] "coefficients"      "rank"              "assign"            "qr"
 [5] "df.residual"      "sigma"             "cov.coefficients" "stdev.unscaled"
 [9] "pivot"            "genes"             "method"            "design"

```

If you wish to access separately anyone of these items, use `fit` followed by the `$` sign and by the name of the item you wish to extract.

Exercise 5: The estimated coefficients are given in `fit$coef`. Check that indeed the first coefficient corresponds to the mean log-intensity for Cluster 12 and the second coefficient to the mean log-intensity for Cluster 3. Can you explain the values in `fit2$coef`?

Ordinary t -statistics and p -values for the comparison between Cluster 12 and Cluster 3 can now be computed from `fit2` as follows

```

> ordP <- function(fit, coeff = 1, method = "none") {
+   ordinary.t <- fit$coef/fit$stdev.unscaled/fit$sigma
+   ordinary.p <- p.adjust(2 * pt(abs(ordinary.t[, coeff]), df = fit$df.residual,
+     lower.tail = FALSE), method = method)
+   return(ordinary.p)
+ }
> ordinary.p <- ordP(fit2, 1)

```

Exercise 6: Take a look at the histogram of resulting p -values `ordinary.p` using the function `hist`. Are the p -values indeed equal to the p -values calculated in Exercise 2?

Until now we have used the classical t -test for assessing statistical significance. However, since a typical microarray experiment is noisy and the number of arrays per group tends to be small, the standard error (=denominator of t -statistic) is hard to estimate reliably. The consequence is that some genes have small p -values only because, by chance, the denominator of the t -statistic was very small. Several researchers have proposed alternative statistics to obtain a more stable estimate of the gene-specific variance. In `limma` this has been implemented via a *moderated* t -statistic, details are described in (Smyth 2005). The moderated t -statistic is simply calculated from `fit2` using the function `eBayes`

```

> fit2.eb <- eBayes(fit2)
> names(fit2.eb)

 [1] "coefficients"      "rank"              "assign"            "qr"
 [5] "df.residual"      "sigma"             "cov.coefficients" "stdev.unscaled"

```

```

[9] "genes"          "method"          "design"           "contrasts"
[13] "df.prior"       "s2.prior"        "var.prior"       "proportion"
[17] "s2.post"        "t"               "p.value"         "lods"
[21] "F"              "F.p.value"

```

The object `fit2.eb` is mainly an extension of the old `fit2` with, for example, moderated t -statistics and p -values. The object `fit2.eb$genes` only contains the Affymetrix probe ids:

```

> fit2.eb$genes[1:3, ]

[1] "117_at"      "1405_i_at" "1598_g_at"

```

To add more annotation, in this case gene names, we now import the list of gene names from a tab-delimited text file:

```

> gene.names <- read.delim("gene_names.txt", header = TRUE)

```

To insert this information in the `fit2.eb` object use

```

> fit2.eb$genes <- gene.names

```

A list of the ten most significant differentially expressed genes can be obtained by

```

> topTable(fit2.eb, adjust = "fdr")

```

	Gene_ID	Gene_Name
1492	209905_at	HOXA9; homeo box A9
2130	214651_s_at	HOXA9; homeo box A9
586	204069_at	MEIS1; Meis1, myeloid ecotropic viral integration site 1 homolog (mouse)
1654	210998_s_at	HGF; hepatocyte growth factor (hepapoietin A; scatter factor)
1653	210997_at	HGF; hepatocyte growth factor (hepapoietin A; scatter factor)
2039	213844_at	HOXA5; homeo box A5
1963	213150_at	HOXA10; homeo box A10
1617	210755_at	HGF; hepatocyte growth factor (hepapoietin A; scatter factor)
1962	213147_at	HOXA10; homeo box A10
790	205110_s_at	FGF13; fibroblast growth factor 13

	logFC	t	P.Value	adj.P.Val	B
--	-------	---	---------	-----------	---

```

1492 -5.243925 -36.31979 4.678267e-21 1.336113e-17 36.86834
2130 -4.788110 -22.69391 1.061529e-16 1.128571e-13 28.11324
586 -3.333739 -22.57505 1.185473e-16 1.128571e-13 28.01040
1654 4.872853 21.19411 4.454252e-16 3.180336e-13 26.76944
1653 4.917051 20.94238 5.717967e-16 3.266103e-13 26.53366
2039 -3.440078 -19.35168 2.959946e-15 1.408934e-12 24.96960
1963 -3.287779 -18.33262 9.055099e-15 3.694480e-12 23.89499
1617 3.958380 17.17765 3.443029e-14 1.229162e-11 22.60108
1962 -3.116935 -16.96749 4.427186e-14 1.404894e-11 22.35635
790 3.349237 14.11919 1.787578e-12 5.105324e-10 18.72016

```

The *topTable* function yields in this case seven columns:

- Gene_ID
- Gene_Name
- logFC: the log fold change represents the effect of the contrast, that is, how much difference there is between expression values in Cluster 12 compared with Cluster 3
- t: the moderated *t*-statistic from the linear model with an improved variance estimate
- P.Value: the *p*-value corresponding to this *t*-statistic
- adj.P.Val: the *p*-value after correction for multiple testing, in this case FDR
- B: the log-odds ratio that the gene is differentially expressed, see section 23.12 (Smyth 2005)

Note that the default *topTable* command returns the top 10 significant genes. We can get more or all genes by adding the term *number*.

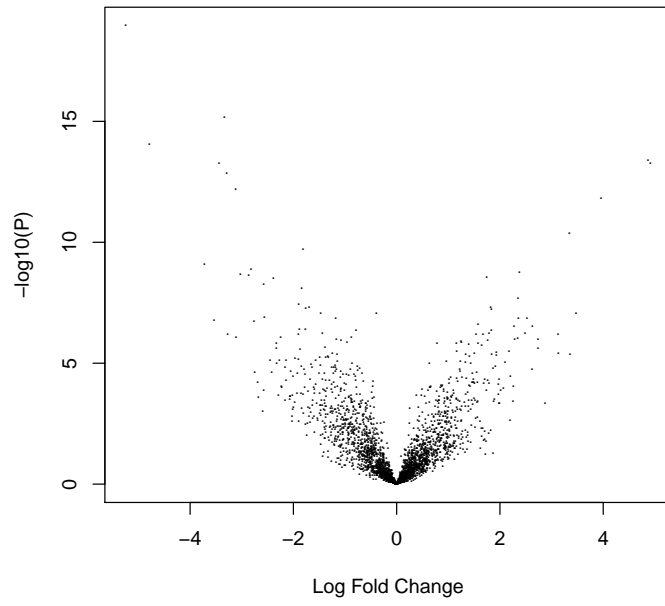
```
> C12vsC3 <- topTable(fit2.eb, number = ngenes, adjust = "fdr")
```

Exercise 7: How many genes are found with a false discovery rate of 5%?

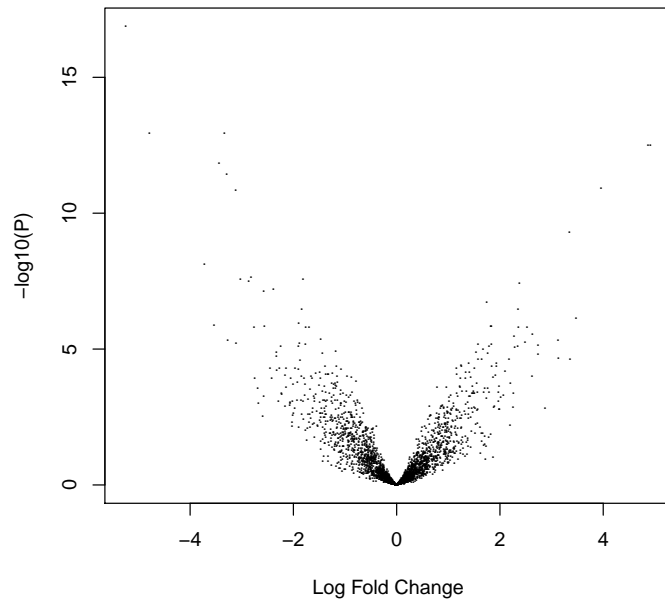
Exercise 8: Compare the FDR corrected *p*-values obtained here with those from Exercise 3, for example with a scatterplot. Can you explain the difference?

Exercise 9: As said before, the moderated *t*-statistic should give a more stable estimate of the gene-specific variance. To investigate this in some more detail have a close look at the two so-called volcano plots. Such a figure plots the *p*-values (actually $-\log_{10}$ of the *p*-values) versus the log fold change (=log-ratio). Can you explain the difference between the two plots?

volcano plot: ordinary t



volcano plot: moderated t



4 Dye swaps

In yesterday's exercise you analyzed the integrin $\alpha 4/\beta 7$ dataset. This experiment aims to study the cell adhesion molecule integrin $\alpha 4/\beta 7$ which assists in directing the migration of blood lymphocytes to the intestine and associated lymphoid tissues. The goal of the study is to identify differentially expressed genes between the $\alpha 4/\beta 7+$ and $\alpha 4/\beta 7-$ memory T helper cells. Each hybridization involved $\beta 7+$ cell RNA from a single subject (labeled with one dye) and $\beta 7-$ cell RNA from same subject (labeled with the other dye). In total 6 microarrays are used in our analysis.

If you still have the Integrin data in a separate directory, you can go there from within R and copy another file:

```
> setwd("~/Integrin")
> file.copy(from = "/data/home/stud00/Integrin/TargetBeta7.txt",
+          to = "TargetBeta7.txt")
```

Otherwise you can get the entire dataset as described at the beginning of yesterday's exercises. Then read in the data once more and perform within and between array normalization

```
> library(limma)
> targets <- readTargets("TargetBeta7.txt")
> rawdata <- read.maimages(targets$FileNames, source = "genepix")
> normdata <- normalizeWithinArrays(rawdata, method = "printtiploess",
+   bc.method = "normexp")
> normdata <- normalizeBetweenArrays(normdata, method = "scale")
> save(normdata, file = "Integrin.Rdata")
```

Let us again have a look at the corresponding targets frame

FileNames	Cy3	Cy5
6Hs.195.1.gpr	b7-	b7+
6Hs.168.gpr	b7+	b7-
6Hs.166.gpr	b7+	b7-
6Hs.187.1.gpr	b7-	b7+
6Hs.194.gpr	b7-	b7+
6Hs.243.1.gpr	b7+	b7-

Exercise 10: What type of experiment is the $\alpha 4/\beta 7$ experiment? What could be the advantage of such a design?

Exercise 11: What is the contrast of interest that you would like to estimate? Can you set up the appropriate design matrix?

Exercise 12: Which genes are differentially expressed with a false discovery rate of 5%?

Exercise 13: In section 23.4 (Smyth 2005) it is explained how you can extend the design to correct for possible probe-specific dye effects. Repeat the above analysis with a dye-intercept term and compare and contrast results with the previous ones.

5 Common reference

In the above experimental design, we have assumed two RNA sources are compared directly. In many cases two RNA sources may be compared indirectly through a common reference design. The following 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.

The experiment compared 8 ApoAI knockout mice with 8 wild type (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. 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 makes this, for each gene, a two-sample rather than a single-sample setup.

You can copy the data to your account as follows

```
stud01@amc-app1:~$ cp /data/home/stud00/ApoAI.RData .
```

Load the data using the following commands

```
> library(limma)
> load("ApoAI.RData")
> names(RG)
```

```
[1] "R"      "G"      "Rb"     "Gb"     "printer" "genes"  "targets"
```

Normalize the data. The following command does print-tip loess normalization of the log-ratios by default

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

Exercise 14: Set up a design matrix to estimate the contrast of interest, viz. knock-out versus wild type. Determine the list of differentially expressed genes. Is the top-ranking gene a surprising find?

☞ Chapter 23 (Smyth 2005) and the *limma* user's guide on the course website contain many more examples of how to set up design matrices.

Acknowledgements

I would like to thank Stephen Nyangoma, Judith Boer, René de Menezes, and Gordon Smyth for ideas and making data available.

References

- Callow, M., S. Dudoit, E. Gong, T. Speed, and E. Rubin (2000). Microarray expression profiling identifies genes with altered expression in hdl-deficient mice. *Genome Res.* 10(12), 2022–2029.
- Gentleman, R., V. Carey, W. Huber, R. Irizarry, and S. Dudoit (Eds.) (2005). *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer.
- Smyth, G. (2005). limma: Linear Models for Microarray Data (Chapter 23). See Gentleman, Carey, Huber, Irizarry, and Dudoit (2005).
- Valk, P., R. Verhaak, M. Beijen, C. Erpelinck, S. B. van Waalwijk van Doorn-Khosrovani, J. Boer, H. Beverloo, M. Moorhouse, P. van der Spek, B. Lowenberg, and R. Delwel (2004). Prognostically useful gene-expression profiles in acute myeloid leukemia. *N Engl J Med.* 350(16), 1617–1628.

```
> sessionInfo()
```

```
R version 2.10.1 (2009-12-14)
i386-pc-mingw32
```

```
locale:
```

```
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] limma_3.2.3      genefilter_1.28.2 Biobase_2.6.1
```

```
loaded via a namespace (and not attached):
```

```
[1] annotate_1.24.1    AnnotationDbi_1.8.2 DBI_0.2-5
[4] RSQLite_0.8-4     splines_2.10.1      survival_2.35-7
[7] xtable_1.5-6
```