Transcriptomic data analysis using R/Bioconductor

Ståle Nygård
Bioinformatics core facility, OUS/UiO
staaln@ifi.uio.no

Contents

1 Before you start

...you need to install a few libraries from Bioconductor. Type the following commands

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("ALL","multtest","hgu95av2.db","edgeR"))
```

This will take a few minutes. Let it run in the background while you follow the lecture. NB! This is needed only the first time you use the specific packages.

2 Miroarray data

Material partly adapted from Anja von Heydelbreck.

2.1 Reading in data and pre-processing

The data used for this exercise come from a study of Chiaretti et al. (Blood 103:2771-8, 2004) on acute lymphoblastic leukemia (ALL), which was conducted with HG-U95Av2 Affymetrix arrays. The data package *ALL* contains an *ExpressionSet* object called `ALL`, which contains the expression data that were normalized with `rma` (intensities are on the log2scale), and annotations of the samples.

Load the ALL package. What is the dimension of the expression data matrix?

```
library(ALL)
data(ALL)
dim(exprs(ALL))
```

For this exercise you can now go directly to the next section. If you need to do the pre-processing yourself, a quick way of doing this is the following (typed in red):

1. Create a directory, move all the relevant CEL files to that directory.
2. Start R in that directory.
3. If using the Rgui for Microsoft Windows make sure your working directory contains the CEL files (use "File $->$ Change Dir" menu item).
4. Pre-process the data using the method gcrma.

   ```
   library (gcrma) #load the gcrma package
   ```

5. Make a tab-delimitted text file `target.txt` with the first column having the names of the CEL files, and the next columns describing the samples on each file (e.g. treatment, strain, sex etc.).
6. Read in the data and create an expression set.

   ```
   pd <- read.AnnotatedDataFrame("target.txt",header=TRUE)
   Data <- ReadAffy(filenames=rownames(pData(pd)),phenoData=pd)
   eset <- gcrma(Data)
   ```

2.2 Looking at a subset of samples

We want to look at the B-cell ALL samples (they can be identified by the column BT of the *ExpressionSet* `ALL`). Of particular interest is the comparison of samples with the BCR/ABL fusion gene resulting from a translocation of the chromosomes 9 and 22 (labelled BCR/ABL in the column mol.biol of the

ExpressionSet `ALL`), with samples that are cytogenetically normal (labelled NEG).

Define an *ExpressionSet* object containing only the data from the B-cell ALL samples. How many samples belong to the cytogenetically defined groups?

```
table(ALL$BT)
table(ALL$mol.biol)
subset <- intersect(grep("B", as.character(ALL$BT)),
which(as.character(ALL$mol.biol) %in% c("BCR/ABL", "NEG")))
eset <- ALL[, subset]
eset$mol.biol <- factor(eset$mol.biol)
table(eset$mol.biol)
```

2.3 Differential expression

Now we are ready to examine the selected genes for differential expression between the BCR/ABL samples and the cytogenetically normal ones.

We use the two-sample t-test to identify genes that are differentially expressed between the two groups. The function `mt.teststat` from the *multtest* package allows to compute several commonly used test statistics for all rows of a data matrix (study its help page). First, we calculate the nominal p-values. The function `pt` gives the distribution function of the t-distribution.

```
library(multtest)
cl <- as.numeric(eset$mol.biol == "BCR/ABL")
t <- mt.teststat(exprs(eset), classlabel = cl, test = "t.equalvar")
pt <- 2 * pt(-abs(t), df = ncol(exprs(eset)) - 2)
```

The function `p.adjust` contains different multiple testing adjustment procedures. For p-value adjustment in terms of the False Discovery Rate (FDR), we use the method of Benjamini and Hochberg. How many genes do you get when imposing an FDR of 0.1?

```
pAdjusted <- p.adjust(pt,method="BH")
sum(pAdjusted< 0.1)
```

2.4 Annotation

Now we want to see which genes are the most significant ones, and look at their raw and adjusted p-values from the different methods. Gene symbols are provided in the annotation package hgu95av2.

```
library(hgu95av2.db)
diff <- order(pAdjusted)[1:10]
genesymbolsDiff <- unlist(mget(featureNames(eset)[diff], hgu95av2SYMBOL))
genesymbolsDiff
```

The top 3 probe sets represent the ABL1 gene, which is affected by the translocation characterizing the BCR/ABL samples. Now we want to see whether there are further probe sets representing this gene, and whether they also indicate differential expression of the ABL1 gene.

```
geneSymbols = unlist(mget(featureNames(ALL), hgu95av2SYMBOL))
ABL1probes <- which(geneSymbols == "ABL1")
ABL1probes
tABL1 <- mt.teststat(exprs(eset)[ABL1probes, ], classlabel = cl,
test = "t.equalvar")
ptABL1 <- 2 * pt(-abs(tABL1), df = ncol(exprs(eset)) - 2)
sort(ptABL1)
```

We see that only three out of the six ABL1 probe sets show evidence (in fact, very strong evidence) for differential expression! It might be interesting to further investigate the ABL1 probe sets regarding e.g. their location in the ABL1 transcript sequence – indeed the BCR/ABL fusion gene resulting from the translocation differs from the normal ABL1 gene.

3 RNAseq data

Material adapted from Merete Molton Worren (Bioinformatics Core Facility).

3.1 Reading the data

Download the file "*bab_table.txt*" from the course web page. Read in the data and look at the first rows:

```
bab_table<-read.table("bab_table.txt",sep="\t")
head(bab_table)
```

3.2 Differential expression analysis with edgeR

Load the edgeR library.

```
library(edgeR)
```

Make a vector denoting group membership (HCHF/Chow).

```
groups <- c(rep("HCHF",3),rep("Chow", 3))
```

Make a DGE (Digital Gene Expression) object to be used when looking for differentially expressed miRNAs. Set the library sizes to the counts of each library (sample).

```
d <- DGEList(counts=bab_table, group=groups,lib.size=colSums(bab_table))
```

Find the normalization factors (based on TMM: trimmed mean of M-values):

```
d <- calcNormFactors(d)
```

Look at the numbers:

```
d$samples
```

Estimate the extra dispersion (variation) parameter:

```
d <- estimateCommonDisp(d, verbose=TRUE)
```

Test for differential expression

```
de <- exactTest(d)
```

Find the ten most differentially expressed genes (HCHF vs Chow):

```
topres <- topTags(de)
topres
```

Write the results to a tab-separated text file:

```
outfile <-  "bab_expression_analysis_top_results.txt"
write.table(topres,file=outfile, sep="\t",quote=FALSE)
```