Microarray Analysis: Self-Study Exercises

Chao-Jen Wong

Fred Hutchinson Cancer Research Center

9-10 December, 2010

For this lab activity we use a subset of `ALL` called `ALLfilt_bcrneg` for differential analysis. It was constructed to obtained samples from B-cell tumors harboring the BCR/ABL mutation and from B-cell tumors with no observed cytogenetic abnormalities (NEG). The goal is to find genes that are differentially expressed between these two distinct molecular subtypes of B-cell leukemia.

1 Loading data

`ALLfilt_bcrneg` is an instance of `ExpressionSet` and is available in the package *SeattleIntro2010*. First of all, we load the dataset to the workspace:

```
> library(SeattleIntro2010)
> data(ALLfilt_bcrneg)
```

2 *t*-tests

We can now perform gene-by-gene tests for differential expression using a simple two-class *t*-tests. The function `rowttest` in the *genefilter* package uses the *t*-test, row by row, to detect significant differences in the location of the distribution of expression values of two groups of samples defined by a factor variable.

Exercise 1
Use `rowttest` to perform t-tests for each genes.

Solution:

```
> library(genefilter)
> table(ALLfilt_bcrneg$mol.biol)

BCR/ABL     NEG
     37      42

> tt <- rowttests(ALLfilt_bcrneg, "mol.biol")
```

Exercise 2
Consult the manual for `rowttest` *for the meaning of the four elements of the return value* `tt`*.*

Exercise 3
Create a histogram of the resulting p-values.

3 Multiple testing correction

The *multtest* package (by K. Pollard, Y. Ge and S. Dudoit) provides a wide variety of p-value correction methods for controlling a broad class of Type I error rates. These methods including single-step procedures, FWER-controlling and FDR-controlling procedures. The results of p-value correction can be more informative for choosing selection cut-offs. Here, we are going to adjust the p-value using the procedure of Benjamini and Hochberg for controlling the false discovery rate (FDR). Note that FDR is the expected proportion of false positives among the genes that are called differentially expressed.

Exercise 4
Use the function `mt.rawp2adjp` *and Benjamini and Hochberg procedure to adjust the p-value. How many genes have adjusted p-values less than* 0.05*? Print out the ten highest-ranking genes with respect to the adjusted p-value.*

Solution:

```
> library(multtest)
> mt <- mt.rawp2adjp(tt$p.value, proc="BH")
> head(mt$adjp) ## print out the first 6 genes with their raw p and adjusted p

             rawp           BH
[1,] 2.445693e-10 1.075860e-06
[2,] 1.280912e-09 2.817365e-06
[3,] 5.265146e-09 7.720459e-06
[4,] 2.740257e-08 2.450276e-05
[5,] 2.785038e-08 2.450276e-05
[6,] 1.536188e-07 1.126282e-04

> sum(mt$adjp[, "BH"] < 0.05)

[1] 206

> mt$index[1:10]

 [1] 1308 3475 2511  897 3230 2001 3553
 [8]  729  585 2646

> featureNames(ALLfilt_bcrneg)[mt$index[1:10]]
```

```
 [1] "1635_at"    "1674_at"
 [3] "40504_at"   "37015_at"
 [5] "40202_at"   "32434_at"
 [7] "37027_at"   "39837_s_at"
 [9] "41274_at"   "40167_s_at"
```

4 Linear model and moderate t-tests

limma is a Bioconductor package that provides advanced statistical methods for linear modeling of microarray data and for identifying differentially expressed genes. Our goal here is to get familiar with the steps of fitting a linear model to the data and using moderate t-tests for assessing differential expression.

4.1 Example 1

First, define the design matrix.

```
> library(limma)
> design <- model.matrix( ~mol.biol, ALLfilt_bcrneg)
> head(design)
```

```
      (Intercept) mol.biolNEG
01005           1           0
01010           1           1
03002           1           0
04007           1           1
04008           1           1
04010           1           1
```

The above code chunk creates an $n \times 2$ binary matrix, where n is the size of the samples. This matrix is passed to the `lmFit` function to fit linear model $y_i = \mu + \beta a_{ij}$ to the expression data. Note that $a_{ij} = 1$ if $i \in \{BCR/ABL\}$, μ is the mean expression of BCR/ABL gene and β represents the effect of NEG on expression level of gene i. Next we pass this linear model to the function `eBayes` to calculate the moderated t-statistics corresponding to the coefficients of the linear model, i.e., μ and β.

```
> fit1 <- lmFit(exprs(ALLfilt_bcrneg), design)
> fit2 <- eBayes(fit1)
> topTable(fit2, coef=2, adjust.method="BH",
+          number=5)
```

```
           ID     logFC  AveExpr
1308  1635_at -1.202675 7.897095
3475  1674_at -1.427212 5.001771
```

```
2511 40504_at -1.181029 4.244478
3230 40202_at -1.779378 8.621443
897  37015_at -1.032702 4.330511
             t      P.Value
1308 -7.409899 1.015653e-10
3475 -7.059383 4.910719e-10
2511 -6.705790 2.368080e-09
3230 -6.353172 1.113659e-08
897  -6.299971 1.403744e-08
        adj.P.Val         B
1308 4.467858e-07 13.972638
3475 1.080113e-06 12.505904
2511 3.472395e-06 11.040795
3230 1.224746e-05  9.598792
897  1.235014e-05  9.383189
```

Question 1
Why is the input argument `coef` *of* `topTable` *set to* 2*?*

Exercise 5
Let the cut-off of the adjusted p-value be 0.05*. How many probe sets are corresponding to differentially expressed genes?*

Solution:

```
> result <- topTable(fit2, coef=2, adjust.method="BH",
+                    p.value=0.05, number=nrow(fit2))
> nrow(result)

[1] 214

>
```

4.2 Example 2

Create another design matrix in the following way:

```
> design <- model.matrix( ~0+mol.biol, ALLfilt_bcrneg)
> colnames(design) <- c("BCR_ABL", "NEG")
> head(design)

      BCR_ABL NEG
01005       1   0
01010       0   1
03002       1   0
04007       0   1
04008       0   1
04010       0   1
```

The linear model is then constructed in the following expression:

$$y_i = \beta_1 a_{ij} + \beta_2 b_{ij} + \varepsilon_i,$$

where β_1 is the mean expression of BCR/ABL and β_2 of NEG. In this case, we need to let `eBayes` know which contrast estimator to use. If we are interested in the effect of BCR/ABL on the expression level of genes, relative to NEG, we can construct a contrast matrix by the code below:

```
> ## note that BCR_ABL and NEG are the column names of design matrix
> contr <- makeContrasts(BCR_ABL-NEG, levels=design)

> ## or simply
> contr <- c(1, -1)
```

Now we are ready to perform further differential analysis.

```
> fit <- lmFit(exprs(ALLfilt_bcrneg), design)
> fit1 <- contrasts.fit(fit, contr)
> fit2 <- eBayes(fit1)
```

Exercise 6
Use the function `topTable` *to print out the ten highest ranking genes, which are differentially expressed in the BCR/ABL samples relative to the NEG samples.*

Solution:

```
> topTable(fit2, adjust.method="BH", number=10)

             ID     logFC  AveExpr
1308    1635_at 1.2026753 7.897095
3475    1674_at 1.4272115 5.001771
2511   40504_at 1.1810295 4.244478
3230   40202_at 1.7793784 8.621443
897    37015_at 1.0327017 4.330511
2001   32434_at 1.6785501 4.466311
3553   37027_at 1.3487023 8.444161
1639   37403_at 1.1177209 5.086540
2646 40167_s_at 0.7448175 5.573901
729  39837_s_at 0.4757069 7.144313
            t      P.Value    adj.P.Val
1308 7.409899 1.015653e-10 4.467858e-07
3475 7.059383 4.910719e-10 1.080113e-06
2511 6.705790 2.368080e-09 3.472395e-06
3230 6.353172 1.113659e-08 1.224746e-05
897  6.299971 1.403744e-08 1.235014e-05
2001 5.934867 6.759370e-08 4.955745e-05
3553 5.791117 1.243978e-07 7.817510e-05
```

```
1639 5.489363 4.393976e-07 2.242009e-04
2646 5.469906 4.762078e-07 2.242009e-04
729  5.406384 6.187354e-07 2.242009e-04
             B
1308 13.972638
3475 12.505904
2511 11.040795
3230  9.598792
897   9.383189
2001  7.919726
3553  7.352120
1639  6.178780
2646  6.104031
729   5.860814
```