Project Description: Genomic Data

Jyotishka Datta

November 2, 2018

Final Project : Analysing Genomic Data

- Possible Inferential Goals:

1. Compare the baseline expression values between two groups using a nonparametric test, e.g. Wilcoxon/Mann-Whitney test.
2. We have p genes: p simultaneous tests for each of the variables. We need to correct for multiple testing.

Large Scale Testing

- Suppose for the i^{th} variable x_i the two group means are $\theta_{i,1}$ and $\theta_{i,2}$.

$$H_{0i} : \theta_{i,1} = \theta_{i,2} \text{ vs. } H_{1i} : \theta_{i,1} \neq \theta_{i,2}$$

- If H_{0i} is true, the group means $\bar{x}_{i,1}$ and $\bar{x}_{i,1}$ should be close.
- We can do an independent samples test for each of the p variables.
- It is not necessary to compare the means via a t-test. We can compare medians using Mann-Whitney / variations using a Siegel-Tukey or an omnibus test using two-sample Kolmogorov-Smirnon test.

Example from T-cell lymphoma

```
## required for gene expression data classification example
require(ALL)
data(ALL)
dim(ALL)
```

```
## Features  Samples
##    12625      128
```

Simplifying features

We are going to use the first three features.

```
resp <- gsub("B[0-9]", "B", ALL$BT)
## B-cell tumors of type B, B1,B2, T, T1, T2
resp <- factor(gsub("T[0-9]", "T", resp))
xmat <- t(exprs(ALL))
mydata <- data.frame(y = resp,x1 = xmat[,1],x2=xmat[,2],x3=xmat[,3])
head(mydata,n=3)
```

```
##       y       x1       x2       x3
## 01005 B 7.597323 5.046194 3.900466
## 01010 B 7.479445 4.932537 4.208155
## 03002 B 7.567593 4.799294 3.886169
```

Further Analysis

- Let's look at the results of molecular biology testing for the 128 samples:

```
table(ALL$mol.biol)
```

```
##
## ALL1/AF4  BCR/ABL E2A/PBX1      NEG   NUP-98  p15/p16
##       10       37        5       74        1        1
```

- Not all levels are frequent !

Filter

Ignoring the samples which came back negative on this test (NEG), most have been classified as (BCR/ABL) or (ALL1/AF4).

For the purposes of this example, we are only interested in these two subgroups, so we will create a filtered version of the dataset using this as a selection criteria:

```
eset <- ALL[, ALL$mol.biol %in% c("BCR/ABL", "ALL1/AF4")]
dim(eset)
```

```
## Features  Samples
##    12625       47
```

How do we analyze this data?

- We have the expression levels 12,625 genes for 47 samples.
- We also have a factor `ALL$mol.biol` that has two levels: BCR/ABL and ALL1/AF4.
- We can ask for which genes, the gene expression values differ between these two subgroups?
- Two sample tests !

Simple test

- One idea would be use a two sample t-test for equality of group mean for each of the 12,625 genes.
- The `mt.teststat` function from the `multtest` library does this for you.

```
require(multtest)
```

```
## Loading required package: multtest
```

```
all.mat = exprs(eset)
all.cl <- factor(as.character(eset$mol.biol))
teststat = mt.teststat(all.mat,all.cl,test="t")
```

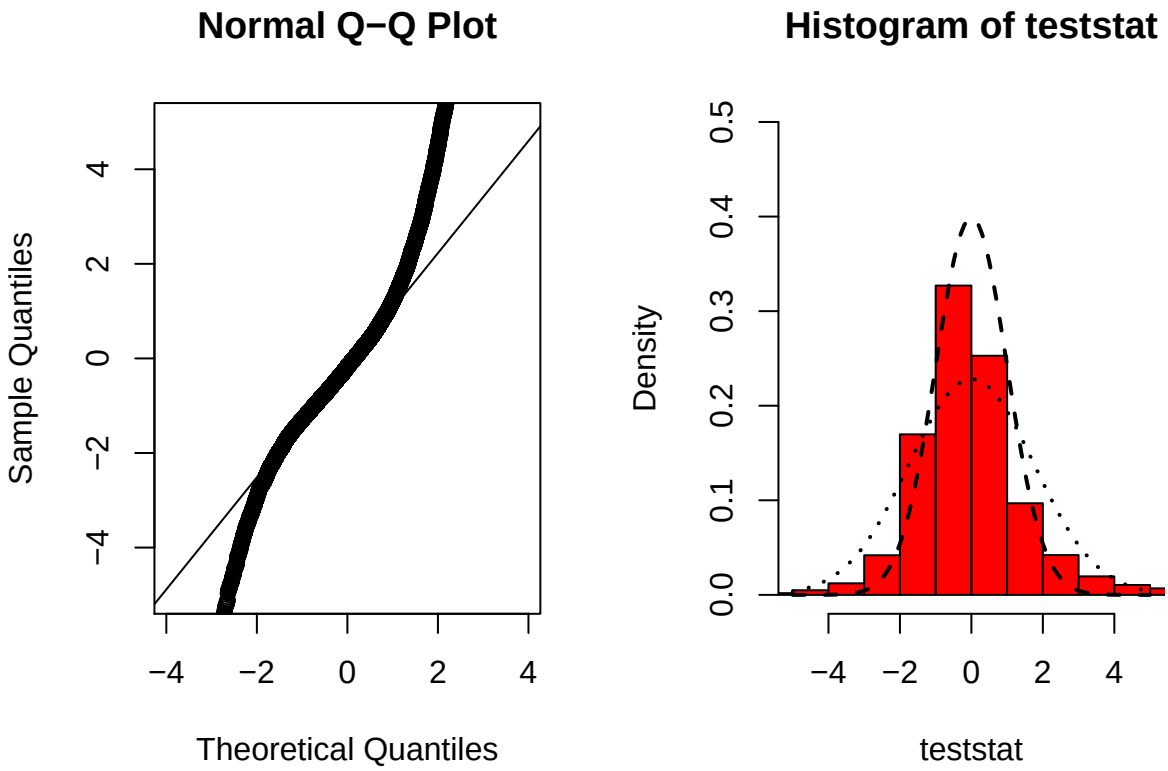
- The two datasets `all.mat` and `all.cl` are all you need.
- `all.mat` has the gene expression values and `all.cl` gives the class labbels.
- mt.teststat {multtest}: **Package for computing test statistics for each row of a data frame**.

Why t-test?

- These functions provide a convenient way to compute test statistics, e.g., two-sample Welch t-statistics, **Wilcoxon statistics**, F-statistics, paired t-statistics, block F-statistics, for each row of a data frame.
- Should we use a t-test or a different test?

- How do you know?

Visualize



- Histogram and N(0,1) density different on the tails - a few interesting genes?

Multiple Testing Issues

- We have a large number of tests: 12,625. If we use standard hypothesis testing at a 5% significance level, 5% of all tests will be falsely rejected (type 1 error) just by pure chance.
- We need some kind of multiplicity control.
- The most stringent is Bonferroni: Divide each α by the total number of tests $p = 12,625$.

Bonferroni

- Bonferroni's correction controls for the familywise error rate (FWER) instead of each α.

$$FWER = P(\text{at least one false rejection}) \leq \alpha$$

- Bonferroni leads to a stringent test, since α/p could be very small if we are carrying out a large number of p tests simultaneously.
- In R, we can apply the `p.adjust` function for this task.

- `p.adjust` also has other useful methods such as "Benjamini-Hochberg False Discovery Rate control procedure".

Bonferroni

```
rawp = 2 * (1 - pnorm(abs(teststat)))
selected  <- p.adjust(rawp, method = "bonferroni") <0.05
esetSel <- eset [selected, ]
sum(selected)
```

```
## [1] 343
```

- Bonferroni's correction leads to rejection of 343 tests - these genes significantly differ between two groups

Bonferroni

- M hypothesis tests: H_{0m} vs. H_{1m} for $m = 1, \ldots, M$.
- Let $p_1, \ldots, p_M$ be the p-values for these M tests.
- In our case $M = p$ (no. of genes)
- Bonferroni method:

$$\text{Reject null hypothesis } H_{0m} \text{ if } p_m \leq \frac{\alpha}{M}$$

- Outcome: The probability of falsely rejecting any null hypothesis is less than or equal to α.

Benjamini-Hochberg

- Let M_0 be the number of null hypotheses that are true, $M_0 = M - M_1$.

	H_0 acc	H_0 rej	Total
H_0 true	U	V	M_0
H_0 false	T	S	M_1
Total	M-R	R	M

Define the false discovery proportion (FDP):

$$FDP = \begin{cases} V/R \text{ if } R > 0 \\ 0 \text{ otherwise} \end{cases}$$

Benjamini-Hochberg

- M hypothesis tests We order the p-values in increasing order. $p_{(1)} \leq \ldots \leq p_{(M)}$.
- *Benjamini-Hochberg Method*
  1. For a given α find the largest k such that

$$p_{(k)} \leq k\frac{\alpha}{M}$$

  2. Then reject all H_{0m} for $m = 1, \ldots, k$.

- *Theorem* :

$$FDR = E(FDP) \leq \frac{M_0}{M}\alpha \leq \alpha$$

- *Outcome*: For a given significance level α, the Benjamini Hochberg method bounds the false discovery rate.

Benjamini-Hochberg

```
rawp = 2 * (1 - pnorm(abs(teststat)))
selected  <- p.adjust(rawp, method = "BH") <0.05
esetSel <- eset [selected, ]
sum(selected)
```

```
## [1] 947
```

- Benjamini-Hochberg method leads to rejection of 947 tests - less stringent than Bonferroni.

Nonparametric test

- We can use the `mt.teststat` function from the `multtest' package as shown above  but choose a different test other than‘` test.
- Since we know how to do any nonparametric test covered in class for two independent samples, we can use them on each gene and calculate P-values for each gene, using a for loop.

```
dim(all.mat)
```

```
## [1] 12625    47
```

```
dim(all.cl)
```

```
## NULL
```

```
str(all.cl)
```

```
##  Factor w/ 2 levels "ALL1/AF4","BCR/ABL": 2 2 1 2 2 2 2 2 2 2 ...
```

A simple two-sample nonparametric test for the first row, i.e. one single gene is shown below:

```
all.mat.af4 = all.mat[,all.cl=='ALL1/AF4']
all.mat.abl = all.mat[,all.cl=='BCR/ABL']

wilcox.test(all.mat.af4[1,],all.mat.abl[1,])
```

```
##
##  Wilcoxon rank sum test
##
## data:  all.mat.af4[1, ] and all.mat.abl[1, ]
## W = 110, p-value = 0.05179
## alternative hypothesis: true location shift is not equal to 0
```

Main task

- You need to perform this test for all the 12,625 rows, get P-values for each of them and then apply multiple testing correction as shown above.
- Perform at least two different tests and both Bonferroni and Benjamini-Hochberg corrections.
- Write your conclusions clearly.

Help

- If you get stuck with any of the steps, please let me know at jd033@uark.edu.