Math7340 HW10

Chengbo Gu

Problem 1 (20 points) Preprocessing a data set

Install the "ArrayExpress" package from Bioconductor. Load the yeast microarray data using R commands:

library(ArrayExpress)

yeast.raw = ArrayExpress('E-MEXP-1551')

(a) Preprocess the raw data set into an expression data set using: the "mas" background correction method, the "quantiles" normalization method, "pmonly" pm correction method and "medianpolish" summary method. Give the R command here for doing this task.

```
library(ArrayExpress)
library(affy)
```

```
yeast.raw <-  ReadAffy(celfile.path= 'E:/yeast' )
eset <- expresso(yeast.raw,
                 bgcorrect.method="mas",
                 normalize.method="quantiles",
                 pmcorrect.method="pmonly",
                 summary.method="medianpolish")
```

(b) Print out the mean expression values for the first five genes across all samples.

```
firstFive <- exprs(eset)[1:5,]
means <- apply(firstFive, 1, mean)
means
```

```
## 1769308_at 1769309_at 1769310_at 1769311_at 1769312_at
##   8.936128   5.666040   5.650467  11.380948   9.752480
```

(c) How many genes and how many samples are in the preprocessed expression data set?

```
dim(eset)
```

```
## Features  Samples
##    10928       30
```

There are 10928 genes and 30 samples in the preprocessed expression data set.

Problem 2 (30 points) Searching Annotations

(a) What is the annotation package for the yeast data set in question 1?

Install the annotation package from Bioconductor.

```
anno <- annotation(yeast.raw)
anno
```

```
## [1] "yeast2"
```

```
# db <- paste(anno, ".db", sep="")
# source("https://bioconductor.org/biocLite.R")
# biocLite(db)
```

The annotation package for the yeast data set is "yeast2".

(b) Search the 1769308_at gene GO numbers related to Molecular Function (MF). How many GO numbers do you get?

```
library(yeast2.db)
library(annotate)
```

```
go1769308 <- get("1769308_at", env = yeast2GO)
gonr <- getOntology(go1769308, "MF")
gonr
```

```
## [1] "GO:0003824" "GO:0016616" "GO:0016853" "GO:0016491" "GO:0016829"
## [6] "GO:0004300" "GO:0003857"
```

```
length(gonr)
```

```
## [1] 7
```

There are 7 GO numbers.

(c) Find the GO parents of the GO IDs in part (b). How many GO parents are there?

```
library(GO.db)
```

```
gP <- getGOParents(gonr)
pa <- sapply(gP, function(x) x$Parents)
length(unique(pa))
```

```
## [1] 5
```

There are 5 GO parents.

(d) Find the GO children of the GO IDs in part (b). How many GO children are there?

```
gC <- getGOChildren(gonr)
ch <- sapply(gC, function(x) x$Children)
length(unique(unlist(ch)))
```

```
## [1] 434
```

There are 434 GO children.

Problem 3 (30 points) Gene filtering on B-cell ALL patients

We work with the patients in stages "B2","B3".

(a) We look for genes expressed differently in stages B2 and B3. Use genefilter to program the Wilcoxon test and the Welch t-test separately for each gene. For each test, we select the genes with p-value<0.001. To save computational time, we set exact=F in the Wilcoxon test function.

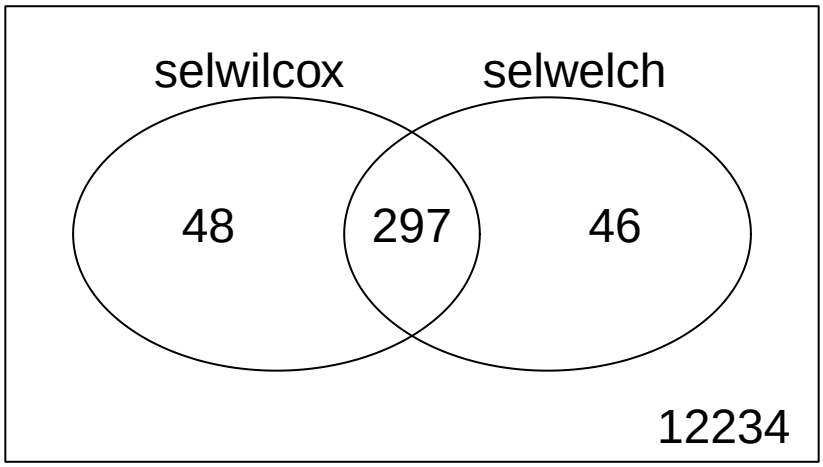
```
library(genefilter)
library(ALL)
```

```
data(ALL)

patientB2 <- factor(ALL$BT %in% c("B2"))
patientB3 <- factor(ALL$BT %in% c("B3"))
wilcox <- function(x) ( wilcox.test (x[patientB2 == TRUE], x[patientB3 == TRUE],
                                     paired = F, exact = F)$p.value < 0.001 )

selwilcox <- genefilter(exprs(ALL), filterfun(wilcox))

welch <- function(x) ( t.test(x[patientB2 == TRUE], x[patientB3 == TRUE],
                              paired = F) $p.value < 0.001 )

selwelch <- genefilter(exprs(ALL), filterfun(welch))
```

(b) Compute a Venn diagram for the Wilcoxon test and the t-test, and plot it.

```
library(limma)
```

```
x <- apply(cbind(selwilcox, selwelch), 2, as.integer)
vc <- vennCounts(x, include = "both")
vennDiagram(vc)
```



(c) How many pass the Wilcoxon filter? How many passes both filters?

```
ALLwilcoxon <- ALL[selwilcox,]
nrow(exprs(ALLwilcoxon))
```

```
## [1] 345
```

```
ALLboth <- ALL[selwilcox & selwelch,]
nrow(exprs(ALLboth))
```

```
## [1] 297
```

We can infer from both the venn diagram and the commands above.

345 genes pass the Wilcoxon filter.

297 genes pass both filters.

(d) What is the annotation package for the ALL data set? Find the GO numbers for "oncogene".

```
library(annotate)
```

```
annotation(ALL)
```

```
## [1] "hgu95av2"
```

The annotation package for the ALL data set is "hgu95av2".

```
library(GO.db)
library(hgu95av2.db)
```

```
GOTerm2Tag <- function(term) {
                GTL <- eapply(GOTERM, function(x) {grep(term, x@Term, value=TRUE)})
                Gl <- sapply(GTL, length)
                names(GTL[Gl>0])}
GOTerm2Tag("oncogene")
```

```
## [1] "GO:0090402"
```

The GO number for "oncogene" is "GO:0090402".

(e) How many genes passing the filters in (a) are oncogenes?

```
tran <- hgu95av2GO2ALLPROBES$"GO:0090402"
inboth <- tran %in% row.names(exprs(ALLboth))
ALLtran <- ALLboth[tran[inboth],]
dim(ALLtran)
```

```
## Features  Samples
##        0      128
```

There is no such gene.

Problem 4 (20 points)

Stages of B-cell ALL in the ALL data. Use the limma package to answer the questions below.

(a) Select the persons with B-cell leukemia which are in stage B1, B2, and B3.

```
library(limma)
library(ALL)
```

```
data(ALL)
allB <- ALL[, which(ALL$BT %in% c("B1", "B2", "B3"))]
```

(b) Use the linear model to test the hypothesis of all zero group means. Use "topTable()" to report the top five genes with nonzero means in B3 group.

```
design.ma <- model.matrix(~0 + factor(allB$BT))
colnames(design.ma) <- c("B1", "B2", "B3")
fit <- lmFit(allB, design.ma)
fit <- eBayes(fit)
print( topTable(fit, coef=3, number=5, adjust.method="fdr"), digits=4)
```

```
##                 logFC AveExpr     t    P.Value  adj.P.Val     B
## AFFX-hum_alu_at 13.61   13.53 355.6 5.059e-127 6.387e-123 270.8
## 32466_at        12.71   12.71 316.7 4.247e-123 2.681e-119 263.9
## 31962_at        13.05   13.09 307.1 4.695e-122 1.976e-118 262.0
## 32748_at        12.15   12.12 302.8 1.407e-121 4.406e-118 261.2
## 35278_at        12.52   12.48 302.0 1.745e-121 4.406e-118 261.0
```

(c) Use two contrasts to perform analysis of variance to test the null hypothesis of equal group means. Do this with a false discovery rate of 0.01. How many differentially expressed genes are found? Use "topTable()" to report the top five genes that express differently among the three groups.

```
cont.ma <- makeContrasts(B1-B2,B2-B3, levels=factor(allB$BT))
fit1 <- contrasts.fit(fit, cont.ma)
fit1 <- eBayes(fit1)
dim(topTable(fit1, number=Inf, p.value=0.01, adjust.method="fdr"))
```

```
## [1] 314   6
```

There are 314 genes that are expressed differentially.

```
print( topTable(fit1, number=5, adjust.method="fdr"), digits=4)
```

```
##           B1...B2  B2...B3 AveExpr     F   P.Value adj.P.Val
## 1389_at  -1.7852 -0.74038   9.678 49.15 1.532e-14 1.934e-10
## 1914_at   2.0976  0.35648   4.693 42.20 3.785e-13 2.389e-09
## 33358_at  1.4890 -0.20733   5.214 29.52 2.837e-10 1.194e-06
## 38555_at  0.8058  0.62321   6.124 25.93 2.322e-09 7.329e-06
## 40763_at  1.5921 -0.01192   3.220 23.08 1.337e-08 2.758e-05
```