Problem 1 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.

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

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

A)

Rscript:

```
biocLite("ArrayExpress")
library(ArrayExpress)
yeast.raw = ArrayExpress('E-MEXP-1551')
eset <- expresso(yeast.raw,bgcorrect.method="mas",
normalize.method="quantiles",pmcorrect.method="pmonly",
summary.method="medianpolish")
```

Answer:

```
background correction: mas
normalization: quantiles
PM/MM correction : pmonly
expression values: medianpolish
background correcting...done.
normalizing...done.
10928 ids to be processed
|                    |
|####################|
```

B)

Rscript:

```
expressionvalues<- exprs(eset)
expressionvalues[1:5,]
```

Answer:

	Gre_MCA_2822	Gre_MCA_5014	Gre_MCA_3174	Gre_MCA_4108	Gre_MCA_3864
1769308_at	9.049451	8.968067	9.203207	8.862262	8.929268
1769309_at	5.582373	5.671969	5.504717	5.718791	5.833522
1769310_at	5.697383	5.536656	5.358618	5.573735	5.720136
1769311_at	11.427141	11.100304	11.550417	11.407650	11.152456
1769312_at	9.870687	9.760277	9.673015	9.888257	9.706339

	Gre_MCA_9147	Gre_MCA_8454	Gre_MCA_4493	Gre_MCA_9211	Gre_MCA_3824
1769308_at	8.858289	9.074716	8.453985	8.883578	8.806324
1769309_at	5.700246	5.479024	5.831566	5.516686	5.940651
1769310_at	5.598420	5.828015	5.537205	5.545576	5.626586
1769311_at	11.465841	11.391676	11.395314	11.234055	11.377907
1769312_at	9.822943	9.764995	9.626873	9.882233	9.854645

	Gre_MCA_8298	Gre_MCA_6510	Gre_MCA_7031	Gre_MCA_6671	Gre_MCA_2172
1769308_at	8.990849	8.915155	8.836336	9.060353	8.860005
1769309_at	5.734243	5.956699	5.707443	5.942985	5.782136
1769310_at	5.793489	5.760122	5.700734	5.688692	5.687390
1769311_at	11.269368	11.320259	11.483310	11.078343	11.452286
1769312_at	9.718603	9.716625	9.769313	9.539288	9.767874

	Gre_MCA_8016	Gre_MCA_7817	Gre_MCA_6886	Gre_MCA_6857	Gre_MCA_9353
1769308_at	9.235485	8.852164	8.853089	8.801891	9.198097
1769309_at	5.413773	5.852047	5.730331	5.740296	5.203052
1769310_at	5.531625	5.865332	5.815449	5.571610	5.559456
1769311_at	11.604265	11.452660	11.296400	11.466942	11.623890
1769312_at	9.629088	9.771854	9.793377	9.764324	9.688466

	Gre_MCA_1726	Gre_MCA_0356	Gre_MCA_8052	Gre_MCA_9301	Gre_MCA_5948
1769308_at	9.134587	8.872302	8.859122	8.807088	8.982770
1769309_at	5.741257	5.764816	5.723271	5.554010	5.573772
1769310_at	5.543069	5.927136	5.792484	5.569385	5.688268
1769311_at	11.500176	11.372680	11.414907	11.305940	11.320536
1769312_at	9.734023	9.883580	9.815190	9.779764	9.910259

	Gre_MCA_7528	Gre_MCA_1103	Gre_MCA_6052	Gre_MCA_2282	Gre_MCA_3378
1769308_at	8.886055	9.068463	8.745493	8.997327	9.038056
1769309_at	5.707336	5.379089	5.673917	5.419490	5.601698
1769310_at	5.470141	5.629583	5.853489	5.389162	5.655055
1769311_at	11.462510	11.449835	11.393147	11.463031	11.195200
1769312_at	9.807047	9.730797	9.764790	9.579366	9.560505

C)
Rscript:
dim(exprs(eset))

Answer:
10928 30

10928 genes and 30 samples

Problem 2 Searching Annotations

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

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

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

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

A)

Rscript:

```
annotation(yeast.raw)
```

Answer:

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

B)

Rscript:

```
biocLite("yeast2.db")
library("yeast2.db")
go<-get("1769308_at", env = yeast2GO)
gomf<-getOntology(go,"MF")
gomf
```

Answer:

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

I get 7 GO numbers.

C)

Rscript:

```
library("GO.db")
go<-get("1769308_at", env = yeast2GO)
gonr <- getOntology(go, "MF")
gP <- getGOParents(gonr)
```

```
pa <- sapply(gP,function(x) x$Parents)
pa
length(unlist(pa))
```

Answer:

```
GO:0016491.is_a GO:0003824.is_a GO:0016616.is_a GO:0016829.is_a
GO:0016853.is_a
  "GO:0003824"    "GO:0003674"    "GO:0016614"    "GO:0003824"
"GO:0003824"
GO:0004300.is_a GO:0003857.is_a
  "GO:0016836"    "GO:0016616"
```

There are 7 GO parents

D)

Rscript:

```
library("GO.db")
go<-get("1769308_at", env = yeast2GO)
gonr <- getOntology(go, "MF")
gC <- getGOChildren(gonr)
ch <- sapply(gC,function(x) x$Children)
ch
length(unlist(ch))
```

Answer:

423 GO children.

Problem 3 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.

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

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

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

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

A)

Rscript:

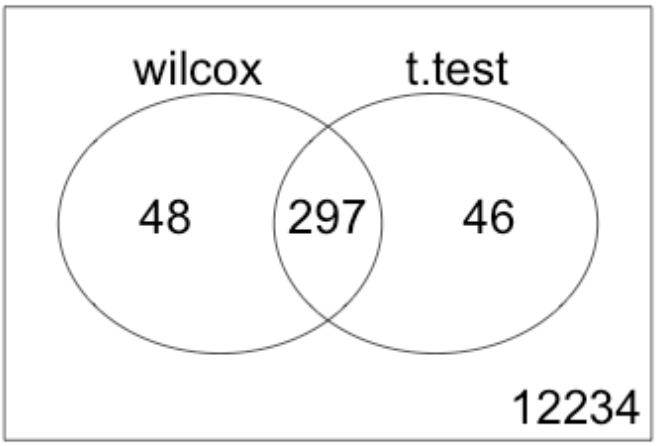
```
library("genefilter")
library("ALL"); data(ALL, package = "ALL");
patientB <- exprs(ALL)[,(ALL$BT %in% c("B2","B3"))]
factor <- droplevels(ALL$BT[ALL$BT %in% c("B2","B3")])
f1 <- function(x) (wilcox.test(x ~ factor, exact = F)$p.value < 0.001)
f2 <- function(x) (t.test(x ~ factor)$p.value < 0.001)
wilcox <- genefilter(patientB, filterfun(f1))
t.test <- genefilter(patientB, filterfun(f2))
```

B)

Rscript:

```
library(limma)
x <- apply(cbind(wilcox,t.test), 2, as.integer)
vc <- vennCounts(x, include="both")
vennDiagram(vc)
```

Answer:



C)
Answer:
From the vein diagram:

How many pass the Wilcoxon filter – 48 + 297 =345

How many passes both filters – 297

D)
Rscript:

```
annotation(ALL)
library("GO.db"); library("annotate"); 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")
```

Answer:
annotation package for the ALL data set - hgu95av2

the GO numbers for “oncogene” - "GO:0090402"

E)
Rscript:

```
selected<- wilcox & t.test
ALLs<-ALL[selected,]
tran <- hgu95av2GO2ALLPROBES$"GO:0090402"
inboth <- tran %in% row.names(exprs(ALLs))

print(sum(inboth))
```

Answer:
0 genes passing the filters in (a) are oncogenes

Problem 4

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.

(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**.

(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.

A)

Rscript:

```
library("limma"); library("ALL"); data(ALL, package = "ALL");
allB <- ALL[,which(ALL$BT %in% c("B1","B2","B3"))] #Patients in 3 stages
exprs(allB)
```

B)

Rscript:

```
library("limma"); library("ALL"); data(ALL, package = "ALL");
allB <- ALL[,which(ALL$BT %in% c("B1","B2","B3"))]
design.ma <- model.matrix(~ 0 + factor(allB$BT))
colnames(design.ma) <- c("B1","B2","B3")
fit <- lmFit(allB,design.ma)
fit <- eBayes(fit)

sum(topTable(fit, number=Inf,adjust.method="fdr")$adj.P.Val<0.05)

print( topTable(fit, coef=3, number=5, adjust.method="fdr"), digits=4)
```

Answer:

```
              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
```

Above we see all 12625 genes have p value less than 0.05 hence we can reject the null hypothesis off all zero group means and conclude that all the genes express differently.

Top five genes with nonzero means in B3 group are:
AFFX-hum_alu_at
32466_at
31962_at
32748_at
35278_at

C)
Rscript:

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

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

Answer:

	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

With false discovery rate of 0.01, 314 genes are found to be differentially expressed

The top five genes that express differently among the three groups are:
1389_at
1914_at
33358_at
38555_at
40763_at