Gene Expression Data Analysis and Visualization
410.671
HW #3

1.) Load the golub data training set in the multtest library. Also load Biobase and annotate libraries, if they are not loaded with the multtest library. Remember that the golub data training set is in the multtest library, so see the help file for information on this data set (2.5 pts)

```
BiocManager::install('multtest')
library(multtest)
library(Biobase)
library(annotate)
data(golub)
```

2.) Cast the matrix to a data frame and label the gene names as numbers (e.g. "g1","g2",etc). (2.5 pts)

```
golub.datfr <- as.data.frame(golub)
growname <- paste('g',1:nrow(golub.datfr),sep = '')
rownames(golub.datfr) <- growname
colnames(golub.datfr) <- paste(colnames(golub.datfr),golub.cl,sep = ' ')
```

3.) Get the sample labels (see lecture notes) and set the sample labels to the data frame. (2.5 pts)

```
golub.datfr <- as.data.frame(golub)
growname <- paste('g',1:nrow(golub.datfr),sep = '')
rownames(golub.datfr) <- growname
colnames(golub.datfr) <- paste(colnames(golub.datfr),golub.cl,sep = ' ')
```

4.) Use the t-test function in the lecture #7 notes and modify it to "wilcox.test" instead of "t.test". Change the "$p.value" argument to "$statistic". Assign the following arguments to the function: (2.5 pts)
exact=F
alternative="two.sided"
correct=T
Run the function on all of the genes in the dataset and save it as "original.wmw.run"

```
wilcox.test.all.genes <- function(x,s1,s2) {
  x1 <- x[s1]
  x2 <- x[s2]
  x1 <- as.numeric(x1)
  x2 <- as.numeric(x2)
  t.out <- wilcox.test(x1,x2,exact=F,alternative="two.sided",correct=T)
  out <- as.numeric(t.out$statistic)
  return(out)
}


original.wmw.run <- apply(golub.datfr,1,wilcox.test.all.genes,s1=golub.cl==0,s2=golub.cl==1)
```

5.) Now write a for loop to iterate 500 times, where in each iteration, the columns of the data frame are shuffled (class labels mixed up), the WMW test is calculated on all of the genes, and the maximum test statistic (W) is saved in a list. (5 pts)

Hints:
Use sample() to sample the number of columns
Get the maximum test statistic across all genes with max()

```
wmw.max.list <- c()
for (i in 1:500) {
  golub.datfr <- golub.datfr[,sample(ncol(golub.datfr))]
  wmw.process <- apply(golub.datfr, 1, wilcox.test.all.genes, s1=c(1:27), s2=c(28:38))
  wmw.max.list<-c(wmw.max.list, max(wmw.process))
}
```

6.) Once you have the list of maximum test statistics, get the 95% value test statistic. Subset the original.wmw.run list of values with only those that have a higher test statistic than the 95% value that you calculated. Print the gene names and test statistics out. (5 pts)

```
original.wmw.run.95.max <- original.wmw.run[original.wmw.run >quantile(wmw.max.list, probs=0.95)]
summary(original.wmw.run.95.max)
attributes(original.wmw.run.95.max)
summary(original.wmw.run.95.max)
```

```
> summary(original.wmw.run.95.max)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    267     272     274     276     281     291
> attributes(original.wmw.run.95.max)
$names
 [1] "g96"   "g283"  "g329"  "g345"  "g394"  "g422"  "g523"  "g546"  "g561"  "g621"  "g648"  "g703"  "g704"  "g717"  "g738"
[16] "g746"  "g835"  "g838"  "g839"  "g849"  "g866"  "g922"  "g984"  "g1006" "g1037" "g1042" "g1045" "g1086" "g1271" "g1327"
[31] "g1368" "g1524" "g1585" "g1598" "g1811" "g1817" "g1834" "g1869" "g1883" "g1909" "g1916" "g1920" "g1939" "g1959" "g1978"
[46] "g1995" "g2002" "g2122" "g2179" "g2266" "g2289" "g2386" "g2418" "g2489" "g2616" "g2645" "g2702" "g2801" "g2829" "g2851"
[61] "g2860" "g2879" "g2939" "g2955" "g3046"
```

7.) Now we want to compare these results to those using the empirical Bayes method in the limma package. Load this library and calculate p-values for the same dataset using the eBayes() function. (5 pts)

```
design <-cbind(Grp1=1,Grp2vs1=c(rep(0,length(golub.cl[golub.cl==0])),rep(1,length(golub.cl[golub.cl==1]))))
fit <- lmFit(golub.datfr,design)
eb.fit <- eBayes(fit)
attributes(eb.fit)
eb.p.value <- eb.fit$p.value[,2]
eb.p.value
```

8.) Sort the empirical Bayes p-values and acquire the lowest *n* p-values, where *n* is defined as the number of significant test statistics that you found in problem 6. Intersect the gene names for your two methods and report how many are in common between the two differential expression methods, when choosing the top *n* genes from each set. (2.5 pts)

```
s.eb.p.value <- sort(eb.p.value)[1:length(original.wmw.run.95.max)]
intersect(names(s.eb.p.value),names(original.wmw.run.95.max))
```

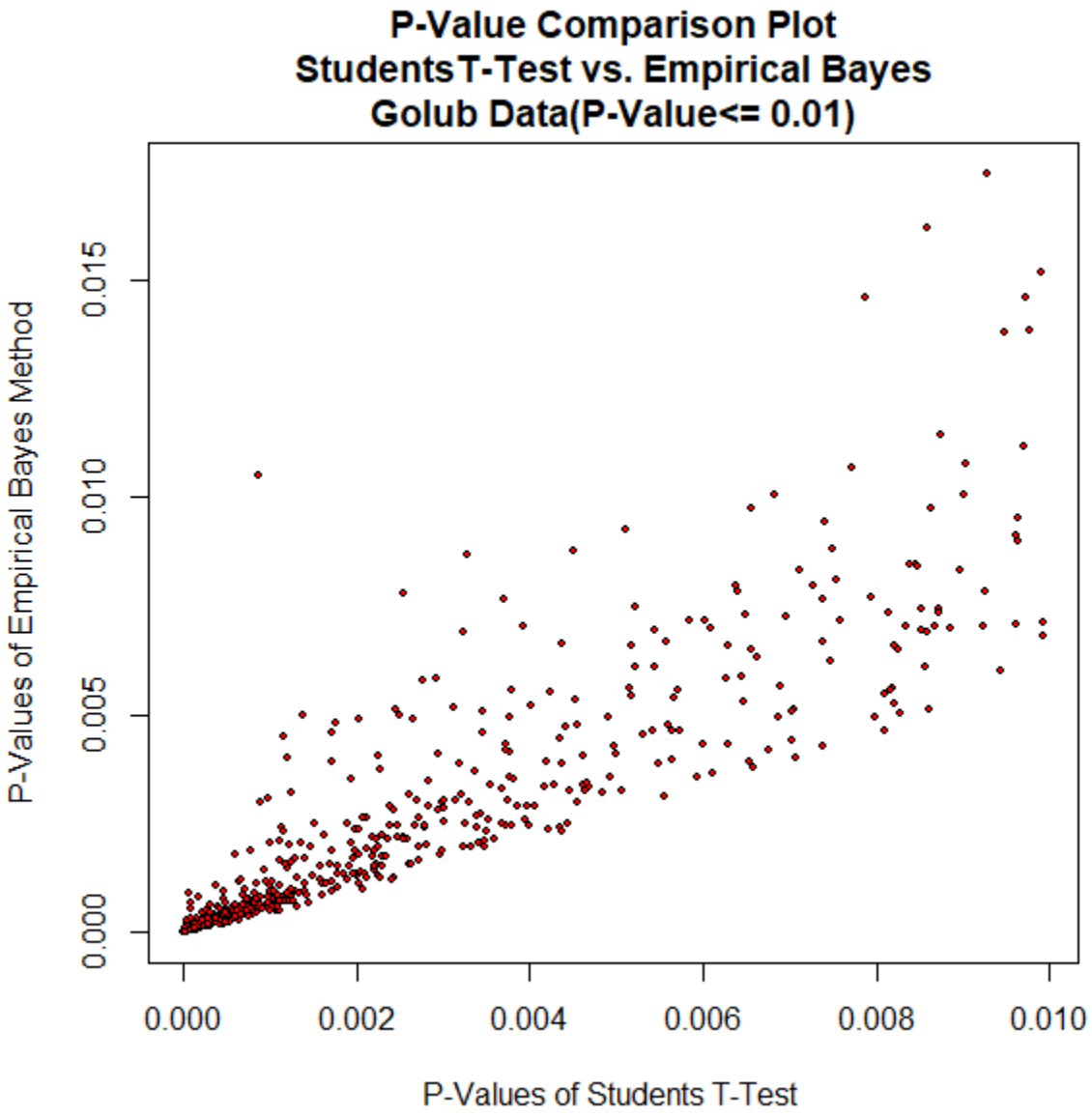
```
> intersect(names(s.eb.p.value),names(original.wmw.run.95.max))
 [1] "g2489" "g1995" "g394"  "g2939" "g717"  "g1042" "g523"  "g2702" "g1037" "g181
1" "g1883" "g2386" "g849"  "g746"  "g1834" "g2266" "g561"  "g1524" "g2289" "g2851"
 "g738"
```

9.) Finally, compare the results from a Student's t-test with the empirical Bayes method. To do this, first calculate a two sample (two-tailed) Student's t-test on all genes. Make sure that you are running a Student's t-test and not a Welch's t-test. Then extract only those genes with a p-value less than 0.01 from this test. Plot the gene p-values<0.01 for the Student's t-test vs. the same genes in the empirical Bayes method. Make sure to label the axes and title appropriately. (7.5 pts)

```r
t.test.all.genes <- function(x,s1,s2) {
  x1 <- x[s1]
  x2 <- x[s2]
  x1 <- as.numeric(x1)
  x2 <- as.numeric(x2)
  t.out <- t.test(x1,x2,alternative="two.sided",var.equal=T)
  out <- as.numeric(t.out$p.value)
  return(out)
}

t.result <- apply(golub.datfr, 1, t.test.all.genes, s1=golub.cl==0, s2=golub.cl==1)
t.result.95.max <- t.result[t.result < 0.01]

plot(t.result.95.max, eb.p.value[names(t.result.95.max)], xlab='P-Values of Students T-Test ',ylab='P-Values of Empirical Bayes Method',
     main='P-Value Comparison Plot\n StudentsT-Test vs. Empirical Bayes \nGolub Data(P-Value<= 0.01)', col='black', bg='red', pch=21, cex=0.5)
```



Paste all information into a PDF.