Multiple testing

In this lab, we will be working with an Affymetrix data set that was run on the human HGU95A array. This experiment was designed to assess the gene expression events in the frontal cortex due to aging. A total of 18 male and 12 female postmortem brain samples were obtained to assess this. (link 1-4: can pick 1 or 2, and 3 or 4. TCGA at bottom)

The analysis that we are interested in conducting is a direct follow up to the previous lab of differential expression. We first want to identify those genes/probes that are differentially expressed in the frontal cortex between old and young subjects, then between males and females. Next, we would like to evaluate the differences between a couple of multiple testing adjustment methods. As explained in the lecture and the course website, multiple testing is a necessary step to reduce false positives when conducting more than a single statistical test. You will generate some p-value plots to get an idea of the how conservative some methods are compared to others.

2 gene vectors have been identified to use below, so do not calculate the t-test or adjustments on the entire array of genes/probes.

For the second part of this lab, you will be working with RNA-sequencing data from The Cancer Genome Atlas (TCGA), specifically a breast invasive carcinoma dataset of 119 patient tumors. The data matrix and annotation files are on the course website. We will be trying to confirm an observation from a meta-analysis performed by Mehra et al, 2005 in Cancer Research. The authors identified the gene (using arrays) and protein (using immunohistochemistry) GATA3 as a prognostic factor in breast cancer, where patients with low expression of GATA3 experienced overall worse survival. The PubMed abstract is here: http://www.ncbi.nlm.nih.gov/pubmed/16357129.

1.) Download the GEO Brain Aging study from the class website. Also obtain the annotation file for this data frame.

#Done

2.) Load into R, using read.table() function and the header=T/row.names=1 arguments for each data file.

```
>cortexFile<- "C:/Users/fermi/Documents/Fall 2020/Lab
6/agingStudy11FCortexAffy.txt"
> cortexAnn<- "C:/Users/fermi/Documents/Fall 2020/Lab
6/agingStudy1FCortexAffyAnn.txt"

>cortexDat<- read.table(cortexFile, header=T, row.names=1)
>anno<- read.table(cortexAnn, header=T, row.names=1)
```

3.) Prepare 2 separate vectors for comparison. The first is a comparison between male and female patients. The current data frame can be left alone for this, since the males and females are all grouped together. The second vector is comparison between patients >= 50 years of age and those < 50 years of age.

To do this, you must use the annotation file and logical operators to isolate the correct arrays/samples.

```
>males<-cortexDat[,1:18]
>females<-cortexDat[,19:30]

> malesM<-unname(as.matrix(males))
> femalesM<-unname(as.matrix(females))

>cortexL50<-cortexDat
>cortexL50<-cortexL50[,-8:-18]
>cortexL50<-cortexL50[,-13:-19]

> cortexO50<-cortexDat
> cortexO50<-cortexO50[,-1:-7]
> cortexO50<-cortexO50[,-12:-16]

>cortexL50<-unname(as.matrix(cortexL50))
>cortexO50<-unname(as.matrix(cortexO50))

>dat<-unname(as.matrix(cortexDat))
```

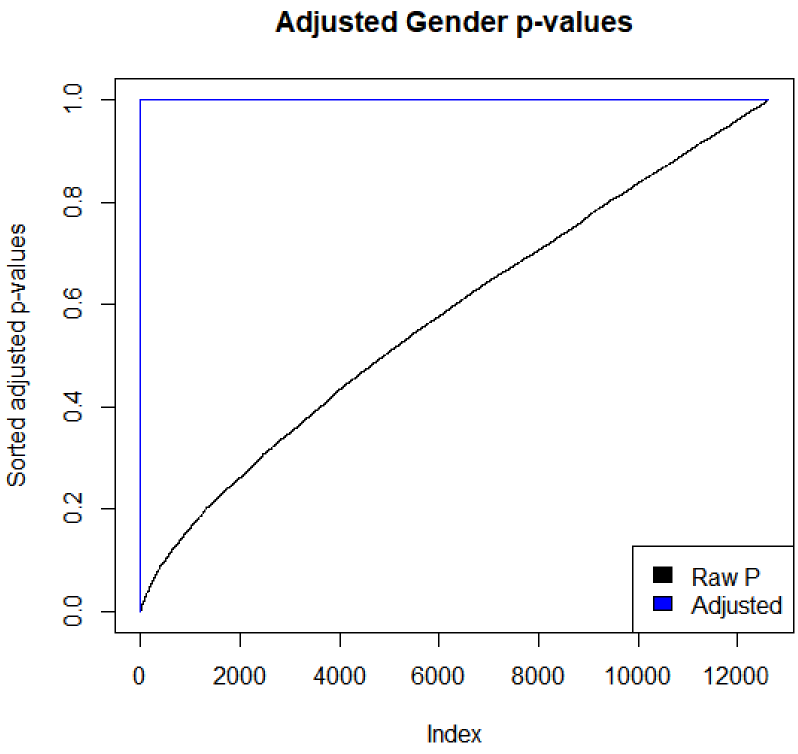
4.) Run the t.test function from the notes using the first gene vector below for the gender comparison. Then use the second gene vector below for the age comparison. Using these p-values, use either p.adjust in the base library or mt.rawp2adjp in the multtest library to adjust the values for multiple corrections with the Holm's method.

```
> 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)
+ }

#Gender Comparison
> ganno<-c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1)
rawpGender<-apply(dat,1,t.test.all.genes,s1=ganno==0,s2=ganno==1)
```

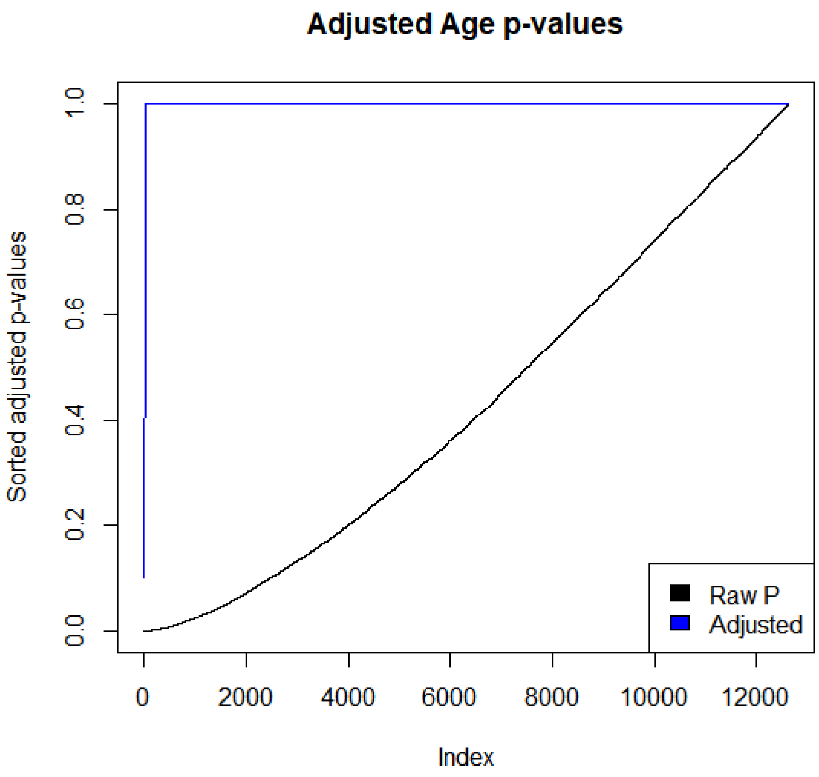
```
#Age Comparison
> aanno<-c(0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,1)
>rawpAge<-apply(dat,1,t.test.all.genes,s1=aanno==0,s2=aanno==1)

#Holm’s correction
>pAge.cor<-p.adjust(rawpAge,method="holm")
> pGender.cor<-p.adjust(rawpGender,method="holm")
```

5.) Sort the adjusted p-values and non-adjusted p-values and plot them vs. the x-axis of numbers for each comparison data set. Make sure that the two lines are different colors. Also make sure that the p-values are sorted before plotting (low to high… two different plots?).

```
> rawpAge<-sort(rawpAge)
> pAge.cor<-sort(pAge.cor)

> rawpGender<-sort(rawpGender)
> pGender.cor<-sort(pGender.cor)

> plot(rawpGender, type="l", main= "Adjusted Gender p-values",ylab= "Sorted
adjusted p-values")
> lines(pGender.cor, col="blue")
> legend("bottomright", legend= c("Raw P", "Adjusted"), fill=c("black", "blue")
```

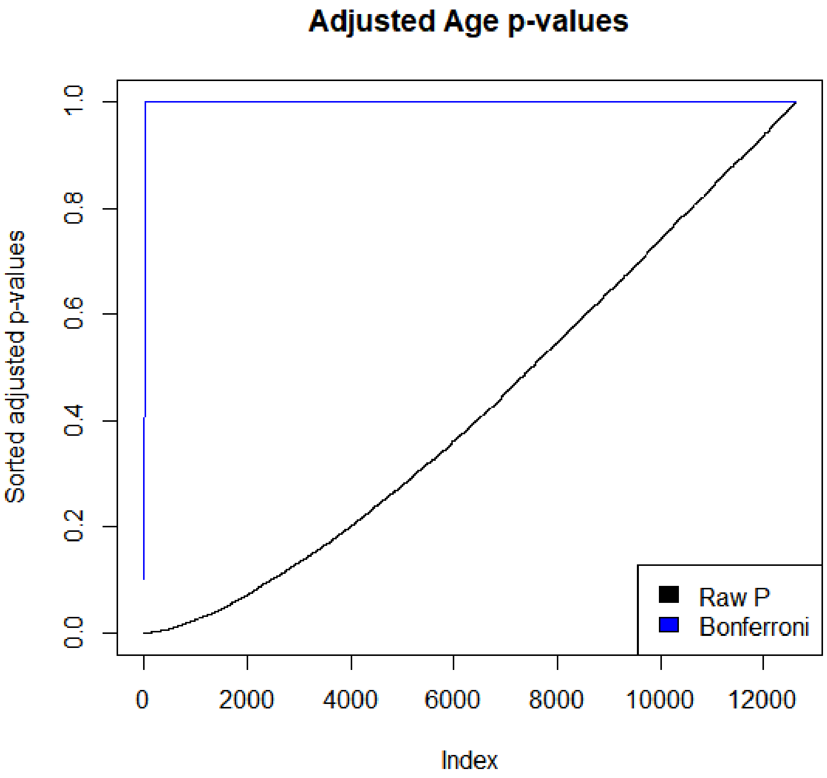


```
> plot(rawpAge, type="l", main= "Adjusted Age p-values",ylab= "Sorted adjusted p-values")
> lines(pAge.cor, col="blue")
> legend("bottomright", legend= c("Raw P", "Adjusted"), fill=c("black", "blue"))
```

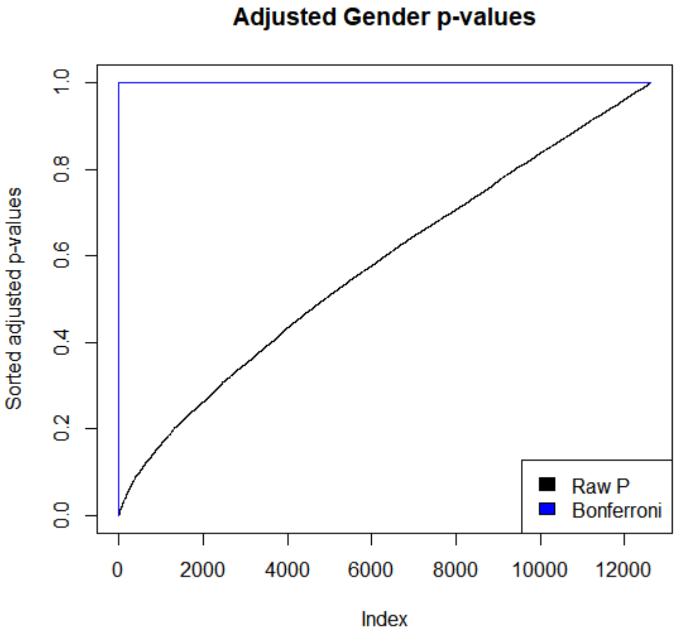


6.) Repeat #4 and #5 with the Bonferroni method.

```
#Bonferroni adjust P
>pAge.bon<-p.adjust(rawpAge,method="bonferroni")
> pGender.bon<-p.adjust(rawpGender,method="bonferroni")

> pAge.bon<-sort(pAge.bon)
> pGender.bon<-sort(pGender.bon)

> plot(rawpAge, type="l", main= "Adjusted Age p-values",ylab= "Sorted adjusted p-values")
> lines(pAge.bon, col="blue")
> legend("bottomright", legend= c("Raw P", "Bonferroni"), fill=c("black", "blue"))
```



```
> plot(rawpGender, type="l", main= "Adjusted Gender p-values",ylab= "Sorted
adjusted p-values")
> lines(pGender.bon, col="blue")
> legend("bottomright", legend= c("Raw P", "Bonferroni"), fill=c("black",
"blue"))
```



7.) Read in the $\log_2$ normalized fragments per kb per million mapped reads (FPKM) data matrix and annotation files. This is RNA-sequencing data that has normalized read counts on a similar scale to microarray intensities.

```
> tcgaFile<- "C:/Users/fermi/Documents/Fall 2020/Lab 6/tcga_brca_fpkm.txt"
> tcgaAnFile<- "C:/Users/fermi/Documents/Fall 2020/Lab 6/tcga_brca_fpkm_sam.txt"

tcgaDat<- read.table(tcgaFile, header=T, row.names=1)
tcgaAnno<- read.table(tcgaAnFile, header=T, row.names=1)
```