

Alzheimers Disease Brain Blood Samples

Janis Corona

1/17/2020

This analysis is done using the NCBI gene expression samples taken from a study that used the middle temporal gyrus side of the brain of frozen samples taken from Alzheimer's patients that totalled 78 samples of control and repeat microarray gene expression results. The file was able to be downloaded and unzipped complete with the platform gene symbol ID already attached. A separate file of the meta information that includes the age, gender, tissue type and disease as Alzheimer's Disease (AD) or control was made from the Series information for each sample. The age range for these samples of healthy and AD patients is from 70-95 years of age.

This study can be linked to with all the sample and meta information at:

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE109887> The link to the 41 mb text file of series of samples with ID that is too large for github, or my github account type is:

ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE109nnn/GSE109887/matrix/GSE109887_series_matrix.txt.gz

Open the original Sample ID values text file and write out as a csv file.

```
alz <- read.delim('GSE109887_series_matrix.txt', sep='\t', header=TRUE,
na.strings=c('', ' '), comment.char='!')
```

```
write.csv(alz, 'alzheimersamples.csv', row.names=FALSE)
```

```
head(alz)
```

```
## ID_REF GSM2973262 GSM2973263 GSM2973264 GSM2973265 GSM2973266 GSM2973267
## 1 7A5 6.436441 6.500922 6.513507 6.480839 6.478978 6.509372
## 2 A1BG 6.736136 6.649034 6.612224 6.559642 6.574546 6.471425
## 3 A1CF 6.545395 6.479783 6.588104 6.482503 6.466453 6.442931
## 4 A26C3 6.529725 6.504956 6.531813 6.549229 6.458589 6.602654
## 5 A2BP1 6.773307 7.675723 7.159857 7.334894 7.928057 7.427863
## 6 A2M 11.253437 10.758069 10.790378 11.280206 10.550165 11.015546
## GSM2973268 GSM2973269 GSM2973270 GSM2973271 GSM2973272 GSM2973273
GSM2973274
## 1 6.558079 6.539648 6.476178 6.384209 6.474428 6.506147
6.547314
## 2 6.601043 6.680376 6.619720 6.695035 6.636312 6.549436
6.631196
## 3 6.448482 6.511453 6.537003 6.485914 6.481272 6.470623
6.556559
## 4 6.551262 6.618060 6.473897 6.475091 6.494137 6.539573
```

6.504725
 ## 5 7.753590 7.542770 6.962004 7.844319 7.045881 8.032980
 7.021767
 ## 6 10.374510 10.232084 11.219453 10.558733 11.928947 10.029828
 11.261613
 ## GSM2973275 GSM2973276 GSM2973277 GSM2973278 GSM2973279 GSM2973280
 GSM2973281
 ## 1 6.482653 6.431043 6.467538 6.496945 6.455699 6.533167
 6.502415
 ## 2 6.637651 6.610124 6.629145 6.633296 6.574900 6.686115
 6.594453
 ## 3 6.500758 6.521914 6.561131 6.483240 6.495440 6.545356
 6.540573
 ## 4 6.574710 6.434496 6.499997 6.456096 6.506443 6.601116
 6.455450
 ## 5 7.237758 7.626402 7.702040 7.108037 8.086046 7.320380
 7.262326
 ## 6 10.869890 10.028131 10.710539 10.902636 10.507890 10.889177
 10.479438
 ## GSM2973282 GSM2973283 GSM2973284 GSM2973285 GSM2973286 GSM2973287
 GSM2973288
 ## 1 6.444786 6.501353 6.477281 6.544612 6.400074 6.520184
 6.539472
 ## 2 6.571460 6.662062 6.610356 6.586472 6.590251 6.587090
 6.597562
 ## 3 6.462481 6.499808 6.487859 6.539648 6.521903 6.523406
 6.469701
 ## 4 6.449598 6.531614 6.493402 6.426615 6.542165 6.577142
 6.421726
 ## 5 7.229908 7.817468 7.986829 8.202127 7.727549 7.299809
 8.396471
 ## 6 10.736578 10.754942 9.644938 10.630864 9.750951 11.035407
 10.141873
 ## GSM2973289 GSM2973290 GSM2973291 GSM2973292 GSM2973293 GSM2973294
 GSM2973295
 ## 1 6.424923 6.474643 6.589264 6.445287 6.529792 6.537969
 6.459321
 ## 2 6.680176 6.550771 6.679956 6.569058 6.675714 6.579930
 6.559305
 ## 3 6.530320 6.473031 6.517640 6.506745 6.516516 6.464695
 6.556136
 ## 4 6.533016 6.370309 6.488512 6.445287 6.455889 6.412834
 6.477753
 ## 5 8.017758 8.169229 8.477709 7.172333 7.303684 8.116424
 8.008762
 ## 6 10.213843 10.347774 10.304462 9.771581 11.610263 10.145408
 9.672881
 ## GSM2973296 GSM2973297 GSM2973298 GSM2973299 GSM2973300 GSM2973301
 GSM2973302
 ## 1 6.418835 6.503889 6.485979 6.485081 6.530189 6.450836

6.489789						
## 2	6.616573	6.893967	6.556532	6.569898	6.896213	6.618346
6.637531						
## 3	6.481994	6.476141	6.525448	6.504118	6.471415	6.439531
6.522903						
## 4	6.443488	6.490153	6.502554	6.540274	6.610700	6.443019
6.483445						
## 5	7.650064	7.083986	7.212323	7.138866	6.818649	7.580220
6.876141						
## 6	11.445590	10.917707	10.639809	10.877692	11.223275	9.632867
10.446053						
##	GSM2973303	GSM2973304	GSM2973305	GSM2973306	GSM2973307	GSM2973308
GSM2973309						
## 1	6.450708	6.461540	6.470023	6.446682	6.580688	6.506489
6.335251						
## 2	6.593549	6.678117	6.591719	6.636234	6.648021	6.622491
6.577416						
## 3	6.503260	6.528314	6.487072	6.525948	6.540932	6.499622
6.479403						
## 4	6.592281	6.483814	6.582309	6.535979	6.521488	6.538749
6.492057						
## 5	7.380896	6.899655	8.136078	7.597110	7.000424	8.066728
6.978048						
## 6	11.354353	10.837673	9.561554	10.928431	10.706460	10.236279
10.857606						
##	GSM2973310	GSM2973311	GSM2973312	GSM2973313	GSM2973314	GSM2973315
GSM2973316						
## 1	6.435910	6.383139	6.504311	6.442846	6.495440	6.423849
6.439607						
## 2	6.715207	6.776508	6.606429	6.632080	6.614676	6.610304
6.641509						
## 3	6.589606	6.485514	6.522903	6.514882	6.493620	6.504014
6.494156						
## 4	6.473449	6.466644	6.498143	6.367646	6.512105	6.483955
6.578028						
## 5	6.770394	8.196010	8.175675	8.239208	7.745435	7.878781
8.135434						
## 6	11.231447	10.371160	9.715824	9.869439	10.131814	10.695212
10.372269						
##	GSM2973317	GSM2973318	GSM2973319	GSM2973320	GSM2973321	GSM2973322
GSM2973323						
## 1	6.476767	6.432943	6.567727	6.590029	6.518445	6.402109
6.462368						
## 2	6.647461	6.583375	6.641342	6.559408	6.603364	6.667564
6.643683						
## 3	6.522168	6.490471	6.612509	6.502178	6.495266	6.483240
6.539573						
## 4	6.577480	6.422356	6.485102	6.515997	6.489093	6.380886
6.519553						
## 5	7.967045	7.416130	7.679170	8.173208	7.996740	7.893309

```

8.566021
## 6 10.452866 10.354096 10.449631 9.839133 9.939212 10.397824
10.141047
## GSM2973324 GSM2973325 GSM2973326 GSM2973327 GSM2973328 GSM2973329
GSM2973330
## 1 6.429510 6.461409 6.404062 6.527984 6.519651 6.447827
6.462642
## 2 6.610161 6.570311 6.587222 6.672783 6.565616 6.645337
6.667337
## 3 6.473020 6.468053 6.550716 6.533538 6.514122 6.499146
6.540705
## 4 6.397061 6.451695 6.484181 6.441520 6.464654 6.473196
6.557970
## 5 8.215478 8.026517 7.850848 8.178894 7.749528 8.342166
8.141308
## 6 9.837524 10.109630 11.115016 9.570170 11.066659 9.929106
9.642897
## GSM2973331 GSM2973332 GSM2973333 GSM2973334 GSM2973335 GSM2973336
GSM2973337
## 1 6.536339 6.506450 6.429433 6.427672 6.488852 6.503583
6.508713
## 2 6.551605 6.557762 6.600702 6.648831 6.660864 6.617121
6.541810
## 3 6.491767 6.479403 6.535446 6.487697 6.515856 6.546361
6.457605
## 4 6.505190 6.585256 6.594342 6.523304 6.526753 6.492019
6.623703
## 5 8.334120 8.201316 7.891050 8.061283 7.794727 7.156839
6.903667
## 6 10.061636 11.116617 10.779104 10.415084 10.889901 11.026483
10.926880
## GSM2973338 GSM2973339
## 1 6.442999 6.465497
## 2 6.547078 6.611912
## 3 6.469584 6.521874
## 4 6.467831 6.453058
## 5 8.187274 7.777883
## 6 9.636975 10.613131

```

Lets read in the meta information for age, gender, and disease type as AD or control.

```

meta <- read.csv('AlzheimerAgeGenderTissueSamplesMeta.csv', sep=',',
                 header=TRUE, na.strings=c('', ' '))

```

```
head(meta)
```

```

## sampleID          GSM2973262
## 1 gender              M
## 2 age                91
## 3 disease            AD
## 4 tissue brain, middle temporal gyrus blood

```

##	GSM2973263	GSM2973264
## 1	M	F
## 2	87	82
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973265	GSM2973266
## 1	F	M
## 2	73	94
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973267	GSM2973268
## 1	M	F
## 2	72	90
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973269	GSM2973270
## 1	F	F
## 2	86	87
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973271	GSM2973272
## 1	M	M
## 2	92	81
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973273	GSM2973274
## 1	M	F
## 2	87	92
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973275	GSM2973276
## 1	F	M
## 2	95	75
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973277	GSM2973278
## 1	F	M
## 2	87	95
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973279	GSM2973280
## 1	M	F
## 2	90	77
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973281	GSM2973282
## 1	F	M
## 2	84	85
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood

##	GSM2973283	GSM2973284
## 1	M	F
## 2	89	89
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973285	GSM2973286
## 1	F	F
## 2	82	78
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973287	GSM2973288
## 1	M	F
## 2	70	86
## 3	AD	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973289	GSM2973290
## 1	F	F
## 2	75	94
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973291	GSM2973292
## 1	M	M
## 2	82	82
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973293	GSM2973294
## 1	M	F
## 2	73	77
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973295	GSM2973296
## 1	M	M
## 2	85	92
## 3	AD	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973297	GSM2973298
## 1	F	F
## 2	84	87
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973299	GSM2973300
## 1	M	F
## 2	86	92
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973301	GSM2973302
## 1	M	F
## 2	92	90
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood

##	GSM2973303	GSM2973304
## 1	F	F
## 2	82	82
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973305	GSM2973306
## 1	F	M
## 2	89	90
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973307	GSM2973308
## 1	M	M
## 2	87	78
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973309	GSM2973310
## 1	F	F
## 2	88	86
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973311	GSM2973312
## 1	F	M
## 2	88	86
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973313	GSM2973314
## 1	F	F
## 2	92	81
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973315	GSM2973316
## 1	M	F
## 2	82	92
## 3	AD	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973317	GSM2973318
## 1	F	M
## 2	81	89
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973319	GSM2973320
## 1	F	M
## 2	85	94
## 3	AD	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973321	GSM2973322
## 1	F	F
## 2	85	82
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood

##	GSM2973323	GSM2973324
## 1	M	F
## 2	81	77
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973325	GSM2973326
## 1	F	M
## 2	81	79
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973327	GSM2973328
## 1	M	F
## 2	78	78
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973329	GSM2973330
## 1	M	M
## 2	79	86
## 3	control	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973331	GSM2973332
## 1	M	M
## 2	91	82
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973333	GSM2973334
## 1	M	F
## 2	84	91
## 3	AD	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973335	GSM2973336
## 1	F	M
## 2	87	86
## 3	control	AD
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973337	GSM2973338
## 1	M	M
## 2	88	81
## 3	AD	control
## 4	brain, middle temporal gyrus blood	brain, middle temporal gyrus blood
##	GSM2973339	
## 1	F	
## 2	85	
## 3	AD	
## 4	brain, middle temporal gyrus blood	

Transpose the meta to fields instead of rows

```
meta1 <- as.data.frame(t(meta))
names <- as.character(meta$sampleID)
```



```
colnames(meta1) <- names
meta2 <- meta1[,-1,]
head(meta2)
```

```
##           gender age disease                                tissue
## GSM2973262      M  91      AD brain, middle temporal gyrus blood
## GSM2973263      M  87      AD brain, middle temporal gyrus blood
## GSM2973264      F  82      AD brain, middle temporal gyrus blood
## GSM2973265      F  73      AD brain, middle temporal gyrus blood
## GSM2973266      M  94      AD brain, middle temporal gyrus blood
## GSM2973267      M  72      AD brain, middle temporal gyrus blood
```

Separate the data into gender and control or AD data sets respectively in each gender and for both genders.

Data tables of males and females for meta only:

```
row.names(alz) <- alz$ID_REF
alz1 <- alz[, -1]

row.names(meta) <- meta$sampleID
Meta <- meta[, -1]

names <- row.names(Meta)
Meta1 <- as.data.frame(t(Meta))
colnames(Meta1) <- names

fem <- grep('F', Meta1$gender)
mal <- grep('M', Meta1$gender)

Fem <- Meta1[fem,]
Mal <- Meta1[mal,]

Fem$sampleID <- as.factor(row.names(Fem))
Mal$sampleID <- as.factor(row.names(Mal))
```

Display the fields of these gender specific tables by meta information:

```
colnames(Fem)

## [1] "gender" "age" "disease" "tissue" "sampleID"

colnames(Mal)

## [1] "gender" "age" "disease" "tissue" "sampleID"
```

The same indices of each row name is the same as the alz1, alzheimer table colnames indices. So, use those same female and male indices values from the regex commands to separate the alzheimer table into male and female tables.

```
Females <- alz1[,fem]
Males <- alz1[,mal]
```

Data tables of control or AD for males and for females for meta only:

```
femControl <- grep('control', Fem$disease)
malControl <- grep('control', Mal$disease)

femAD <- grep('AD', Fem$disease)
malAD <- grep('AD', Mal$disease)

FemCtrl <- Fem[femControl,]
MalCtrl <- Mal[malControl,]

FemAD <- Fem[femAD,]
MalAD <- Mal[malAD,]
```

Display the AD or Control tables by gender for meta information:

```
colnames(FemCtrl)
## [1] "gender" "age" "disease" "tissue" "sampleID"

colnames(FemAD)
## [1] "gender" "age" "disease" "tissue" "sampleID"

colnames(MalCtrl)
## [1] "gender" "age" "disease" "tissue" "sampleID"

colnames(MalAD)
## [1] "gender" "age" "disease" "tissue" "sampleID"
```

Now use that same index information within each gender table derived from the alzheimer gene expression data:

```
Females_AD <- Fem[femAD,]
Females_control <- Fem[femControl,]

males_AD <- Mal[malAD,]
males_control <- Mal[malControl,]
```

Write these last tables to csv file:

```
write.csv(Females_AD, 'females_Alzheimers.csv', row.names=TRUE)
write.csv(Females_control, 'females_healthy_control.csv', row.names=TRUE)
write.csv(males_AD, 'males_Alzheimers.csv', row.names=TRUE)
write.csv(males_control, 'males_healthy_control.csv', row.names=TRUE)
```

What is the age range on these samples of healthy controls and AD patients?

```

age1 <- range(as.numeric(as.character(FemAD$age)))
age2 <- range(as.numeric(as.character(FemCtrl$age)))
age3 <- range(as.numeric(as.character(MalAD$age)))
age4 <- range(as.numeric(as.character(MalCtrl$age)))

char <- c('The minimum age: ', 'The maximum age: ')

AD_fem <- paste0(char,age1, sep='')
AD_mal <- paste0(char,age3, sep='')
ctrl_fem <- paste0(char,age2, sep='')
ctrl_mal <- paste0(char, age4, sep='')

adf <- c('Alzheimer females: ')
adm <- c('Alzheimer males: ')
cf <- c('healthy females: ')
cm <- c('healthy males: ')

ranges <- c(adf,AD_fem, adm, AD_mal, cf, ctrl_fem, cm, ctrl_mal)
ranges

## [1] "Alzheimer females: " "The minimum age: 73" "The maximum age: 95"
## [4] "Alzheimer males: " "The minimum age: 70" "The maximum age: 95"
## [7] "healthy females: " "The minimum age: 77" "The maximum age: 92"
## [10] "healthy males: " "The minimum age: 73" "The maximum age: 94"

```

The age range in all of this data for these healthy and AD patients is age 70 to age 95 years of age. Just useful to know when cross comparing blood microarray samples across the females from our overweight females using epigallocatechin (EGCG) in their 40s, the males from Russia who are also in their 40s with or without myocardial infarction (MI), the hemochromatosis (iron toxicity) males and females averaging their 40s in years of age with or without iron toxicity, and the flu vaccinated blood samples of males and females also in their 40s with or without antibiotic treatment combined with flu vaccination.

Lets do some data analysis of the means and fold change between the groups to create some additional statistical information on these Alzheimers blood samples and healthy controls of elderly patients. We will use dplyr for this and plot with ggplot2 later.

```
library(dplyr)
```

First lets look at the control and the AD tables within the gender specific tables: Females_AD, Females_control, males_AD, males_control. Get the means of each gene and compare into one table. This and these tables of data were already cleaned to only have one gene per row, due to not having to merge the platform fields to the series table and duplicate genes for missing data in the additional rows like previous studies mentioned earlier required.

Get the row means of each table and add to each table.

```

names <- row.names(Females_AD)
Females_AD1 <- rowMeans(Females_AD)

```

```

#colnames(Females_AD1) <- 'fem_AD_Mean'

Females_control1 <- rowMeans(Females_control)
#colnames(Females_control1) <- 'fem_ctrl_Mean'

males_AD1 <- rowMeans(males_AD)
#colnames(males_AD1) <- 'mal_AD_Mean'

males_control1 <- rowMeans(males_control)
#colnames(males_control1) <- 'mal_ctrl_Mean'

means <- cbind(Females_AD1, Females_control1, males_AD1, males_control1)
Means <- as.data.frame(means)
colnames(Means) <- paste(colnames(Means), '_Mean', sep='')
str(Means)

## 'data.frame':    31700 obs. of  4 variables:
## $ Females_AD1_Mean      : num  6.48 6.63 6.52 6.51 7.42 ...
## $ Females_control1_Mean: num  6.47 6.65 6.5 6.48 7.97 ...
## $ males_AD1_Mean       : num  6.48 6.6 6.5 6.51 7.51 ...
## $ males_control1_Mean  : num  6.48 6.62 6.51 6.5 8.01 ...

```

Now for the fold change values between the Means of the female control and AD patients, then for the male control and AD patients.

```

names <- row.names(Means)
FC_females <- Means %>% mutate(FC_fem_ctrl_AD =
                                Females_AD1_Mean/Females_control1_Mean)
str(FC_females)

## 'data.frame':    31700 obs. of  5 variables:
## $ Females_AD1_Mean      : num  6.48 6.63 6.52 6.51 7.42 ...
## $ Females_control1_Mean: num  6.47 6.65 6.5 6.48 7.97 ...
## $ males_AD1_Mean       : num  6.48 6.6 6.5 6.51 7.51 ...
## $ males_control1_Mean  : num  6.48 6.62 6.51 6.5 8.01 ...
## $ FC_fem_ctrl_AD       : num  1.001 0.998 1.004 1.005 0.931 ...

row.names(FC_females) <- names

names <- row.names(FC_females)
FC_both <- FC_females %>% mutate(FC_male_ctrl_AD =
                                males_AD1_Mean/males_control1_Mean)
row.names(FC_both) <- names
str(FC_both)

## 'data.frame':    31700 obs. of  6 variables:
## $ Females_AD1_Mean      : num  6.48 6.63 6.52 6.51 7.42 ...
## $ Females_control1_Mean: num  6.47 6.65 6.5 6.48 7.97 ...
## $ males_AD1_Mean       : num  6.48 6.6 6.5 6.51 7.51 ...
## $ males_control1_Mean  : num  6.48 6.62 6.51 6.5 8.01 ...

```

```
## $ FC_fem_ctrl_AD      : num  1.001 0.998 1.004 1.005 0.931 ...
## $ FC_male_ctrl_AD     : num  0.999 0.997 0.999 1.002 0.938 ...

write.csv(FC_both, 'FC_both.csv', row.names=TRUE)
```

Combine the data of the samples with IDs in the alz1 table with the stats.

```
both <- cbind(FC_both, alz1)

Both <- round(both, 3)

write.csv(Both, 'Both.csv', row.names=TRUE)
```

Top 100 genes Most expressed in fold change for females with AD:

```
top100_fem <- Both[order(Both$FC_fem_ctrl_AD, decreasing=TRUE)[0:100],]
a <- row.names(top100_fem)
a
```

## [1]	"SLC5A3"	"LOC339879"	"DDIT4L"	"SLC6A12"	"SLC7A2"
## [6]	"SFMBT2"	"LOC727908"	"CHST6"	"TEAD2"	"GPER"
## [11]	"FLJ38717"	"FCGBP"	"HEY2"	"ITPRIPL2"	"SIPA1"
## [16]	"DDX27"	"RAPGEF3"	"C15ORF52"	"CALD1"	"ITSN1"
## [21]	"AEBP1"	"PLXNB1"	"HS.37648"	"LOC100129828"	"NBPF8"
## [26]	"GSDMD"	"ITPKB"	"PTH1R"	"SYTL4"	"MYOT"
## [31]	"TMEM137"	"FAM65C"	"FZD9"	"VCAN"	
	"ZCCHC24"				
## [36]	"HS.505676"	"HS.534061"	"LOC100133019"	"LRRC32"	"AHNAK"
## [41]	"LYL1"	"NOTCH1"	"APLNR"	"HS.379253"	"PTRF"
## [46]	"RPPH1"	"C10RF110"	"INPPL1"	"LOC649841"	"DLC1"
## [51]	"ITGB5"	"RASL12"	"AKR1C3"	"FOXC1"	
	"LOC100132532"				
## [56]	"RAB13"	"WSCD1"	"LOC643287"	"ACACB"	"CUEDC1"
## [61]	"INPP5D"	"LOC100129681"	"LOC642031"	"TNFRSF1B"	"CFLAR"
## [66]	"FXYS5"	"LOC100131541"	"PDGFRB"	"RAXL1"	"ANGPT2"
## [71]	"LOC100131277"	"SLC15A3"	"FLNC"	"LOC648921"	"NACC2"
## [76]	"RUNDC2C"	"UBXN2A"	"VIL2"	"ZNF786"	
	"DCLRE1C"				
## [81]	"FLJ46906"	"HS.143018"	"LOC727948"	"HS.193767"	"LRCH4"
## [86]	"MT1H"	"SCRIB"	"KIF1C"	"LASS1"	
	"LOC100130598"				
## [91]	"NDE1"	"RN7SK"	"ZNF621"	"C9ORF130"	"CAPS"
## [96]	"EML3"	"HS.576072"	"KCNJ10"	"LOC100131096"	
	"LOC653158"				

Top 100 genes most expressed in fold change for males with AD:

```
top100_male <- Both[order(Both$FC_male_ctrl_AD, decreasing=TRUE)[0:100],]
b <- row.names(top100_male)
b
```

##	[1]	"RGS1"	"SLC5A3"	"LOC649362"	"TTR"	"CD44"
##	[6]	"FCGBP"	"SLC7A2"	"C100RF10"	"EMP1"	"ITPRIP"
##	[11]	"DDIT4"	"PLXNB1"	"ITPKB"	"S100A8"	
		"SERPINA3"				
##	[16]	"TEAD2"	"ADAMTS1"	"CCL2"	"ZFP36"	"HEY2"
##	[21]	"PTH1R"	"CHST6"	"GFAP"	"HLA-DRA"	
		"SLC6A12"				
##	[26]	"ACACB"	"GPR4"	"SRGN"	"CD14"	"CDKN1A"
##	[31]	"LRRC32"	"FOS"	"HLA-DMB"	"KIAA1881"	
		"TNFRSF1B"				
##	[36]	"HLA-DMA"	"MS4A6A"	"RAB13"	"CALD1"	"RASL12"
##	[41]	"MT2A"	"SYTL4"	"AHNAK"	"CEBPD"	"FAM65C"
##	[46]	"TGFB1"	"TMC6"	"CAPS"	"DDIT4L"	"ID3"
##	[51]	"ITGB2"	"LCP1"	"RNU1G2"	"S1PR3"	"SIPA1"
##	[56]	"GADD45G"	"GSDMD"	"NUPR1"	"ANLN"	"GEM"
##	[61]	"PDGFRB"	"VCAN"	"AEBP1"	"CD163"	"KIF1C"
##	[66]	"LOC387763"	"RUNDC2C"	"ZFP36L1"	"CFLAR"	"CTGF"
##	[71]	"IL8"	"PDK4"	"PLIN2"	"SLC16A9"	"CSF1R"
##	[76]	"IL17RB"	"LOC100133692"	"LOC643287"	"SCIN"	"TAGLN"
##	[81]	"TMEM106A"	"ADM"	"EDN1"	"FOXC1"	
		"GADD45B"				
##	[86]	"HSPA1A"	"MT1M"	"APLNR"	"APOLD1"	"C7"
##	[91]	"HS.37648"	"IGFBP5"	"MYOT"	"RN7SK"	
		"C15ORF52"				
##	[96]	"GJA4"	"GPER"	"ITGB5"	"LEP"	"PLXDC2"

Bottom 100 genes that are least expressed in fold change in females with AD:

```
bottom100_fem <- Both[order(Both$FC_fem_ctrl_AD, decreasing=FALSE)[0:100],]
c <- row.names(bottom100_fem)
c
```

##	[1]	"SYT1"	"CHGB"	"PCSK1"	"VGF"	"MAL2"
##	[6]	"RGS4"	"GABRA1"	"NPTX2"	"SST"	"STAT4"
##	[11]	"C10RF173"	"CBLN4"	"DYNC1I1"	"STXBP1"	"GNG2"
##	[16]	"RGS7"	"TSPAN13"	"EFCBP1"	"NELL2"	"HPRT1"
##	[21]	"TAGLN3"	"DCLK1"	"ST6GALNAC5"	"GLRB"	"BEX5"
##	[26]	"DIRAS2"	"SYP"	"KIAA1107"	"PAK1"	"PTPRN"
##	[31]	"SERPINI1"	"ZCCHC12"	"NELL1"	"DACH2"	"ELMOD1"
##	[36]	"NECAB1"	"NEFM"	"PARM1"	"RTN1"	"VAMP2"
##	[41]	"ENC1"	"TMEM16C"	"CREG2"	"STX1A"	"SVOP"
##	[46]	"CAP2"	"SCG2"	"UCHL1"	"CALY"	
		"KIAA0748"				
##	[51]	"NAP1L3"	"RIMBP2"	"YWHAG"	"GABRG2"	
		"LOC100128403"				
##	[56]	"NAP1L2"	"NAP1L5"	"STMN2"	"ADCYAP1"	"SCN3B"
##	[61]	"GAD1"	"HS.390250"	"INA"	"SCN2B"	
		"GPRASP2"				
##	[66]	"ITFG1"	"SNX10"	"SV2B"	"ATP6V1G2"	"GABBR2"
##	[71]	"GAD2"	"HPCA"	"NUDT11"	"LPPR4"	

```

"TMEM155"
## [76] "CAMK1G"      "EPHA4"      "GABRA5"      "MYT1L"
"HS.553187"
## [81] "CDH13"       "VSNL1"      "XK"           "CKMT1A"
"HS.31961"
## [86] "PCSK1N"      "PRKCG"      "SYT13"       "NPY"         "PRKCB"
## [91] "RASL11B"     "SYN2"       "CADPS"       "TSPYL1"
"C12ORF53"
## [96] "HSPB3"       "MKL2"       "PPP3R1"      "SNCA"
"C2ORF80"

```

Bottom 100 genes that are least expressed in fold change in males with AD:

```

bottom100_male <- Both[order(Both$FC_male_ctrl_AD, decreasing=FALSE)[0:100],]
d <- row.names(bottom100_male)
d

```

```

## [1] "VGF"      "PCSK1"    "RGS4"     "CHGB"     "STAT4"
## [6] "GABRA1"   "ST6GALNAC5" "ADCYAP1"  "MAL2"     "SYT1"
## [11] "CREG2"    "NPTX2"    "TMEM16C"  "CBLN4"    "SVOP"
## [16] "KIAA1107" "RGS7"     "BEX5"     "C10RF173" "NELL1"
## [21] "SST"      "PARM1"    "DYNC1I1"  "NAP1L5"   "NEFM"
## [26] "VAMP2"    "HS.390250" "PTPRN"    "HSPB3"
"LOC100128403"
## [31] "STXBP1"   "ENC1"     "GAD2"     "PAK1"     "ANO3"
## [36] "INA"      "LOC387856" "SLC30A3"  "STX1A"
"ZCCHC12"
## [41] "SCN2B"    "C2ORF80"  "CRYM"     "EPHA4"    "GLRB"
## [46] "HPRT1"    "KIAA0748" "STMN2"    "C11ORF87" "CAP2"
## [51] "GAD1"     "SNX10"    "EFCBP1"   "RPH3A"
"SERPINI1"
## [56] "DACH2"    "DCLK1"    "GPRASP2"  "SYP"
"TSpan13"
## [61] "VIP"      "C12ORF53" "CALY"     "OLFM3"    "SV2B"
## [66] "COPG2IT1" "DIRAS2"   "ELMOD1"   "NELL2"    "SCG2"
## [71] "SLC39A10" "GABRG2"   "SLITRK4"  "TAGLN3"
"TMEM155"
## [76] "CKMT1B"   "CYP26B1"  "GABBR2"   "CPNE4"
"HS.553187"
## [81] "MFF"      "PRKCG"    "TUBB2A"   "ELAVL4"   "LRFN5"
## [86] "NCALD"    "RIMBP2"   "RTN1"     "SLC6A17"  "SYT13"
## [91] "YWHAE"    "CCKBR"    "FGF12"    "G3BP2"    "HOPX"
## [96] "NAP1L3"   "NMNAT2"   "ZNF365"   "C20ORF103" "CADPS"

```

```

ab <- c(a,b)
topBoth <- unique(ab)

cd <- c(c,d)
bottomBoth <- unique(cd)

```

The unique top genes are:

topBoth

##	[1]	"SLC5A3"	"LOC339879"	"DDIT4L"	"SLC6A12"	"SLC7A2"
##	[6]	"SFMBT2"	"LOC727908"	"CHST6"	"TEAD2"	"GPER"
##	[11]	"FLJ38717"	"FCGBP"	"HEY2"	"ITPRIPL2"	"SIPA1"
##	[16]	"DDX27"	"RAPGEF3"	"C15ORF52"	"CALD1"	"ITSN1"
##	[21]	"AEBP1"	"PLXNB1"	"HS.37648"	"LOC100129828"	"NBPF8"
##	[26]	"GSDMD"	"ITPKB"	"PTH1R"	"SYTL4"	"MYOT"
##	[31]	"TMEM137"	"FAM65C"	"FZD9"	"VCAN"	
"ZCCHC24"						
##	[36]	"HS.505676"	"HS.534061"	"LOC100133019"	"LRRC32"	"AHNAK"
##	[41]	"LYL1"	"NOTCH1"	"APLNR"	"HS.379253"	"PTRF"
##	[46]	"RPPH1"	"C10RF110"	"INPL1"	"LOC649841"	"DLC1"
##	[51]	"ITGB5"	"RASL12"	"AKR1C3"	"FOXCI"	
"LOC100132532"						
##	[56]	"RAB13"	"WSCD1"	"LOC643287"	"ACACB"	"CUEDC1"
##	[61]	"INPP5D"	"LOC100129681"	"LOC642031"	"TNFRSF1B"	"CFLAR"
##	[66]	"FXVD5"	"LOC100131541"	"PDGFRB"	"RAXL1"	"ANGPT2"
##	[71]	"LOC100131277"	"SLC15A3"	"FLNC"	"LOC648921"	"NACC2"
##	[76]	"RUNDC2C"	"UBXN2A"	"VIL2"	"ZNF786"	
"DCLRE1C"						
##	[81]	"FLJ46906"	"HS.143018"	"LOC727948"	"HS.193767"	"LRCH4"
##	[86]	"MT1H"	"SCRIB"	"KIF1C"	"LASS1"	
"LOC100130598"						
##	[91]	"NDE1"	"RN7SK"	"ZNF621"	"C9ORF130"	"CAPS"
##	[96]	"EML3"	"HS.576072"	"KCNJ10"	"LOC100131096"	
"LOC653158"						
##	[101]	"RGS1"	"LOC649362"	"TTR"	"CD44"	
"C10ORF10"						
##	[106]	"EMP1"	"ITPRIP"	"DDIT4"	"S100A8"	
"SERPINA3"						
##	[111]	"ADAMTS1"	"CCL2"	"ZFP36"	"GFAP"	"HLA -
DRA"						
##	[116]	"GPR4"	"SRGN"	"CD14"	"CDKN1A"	"FOS"
##	[121]	"HLA-DMB"	"KIAA1881"	"HLA-DMA"	"MS4A6A"	"MT2A"
##	[126]	"CEBPD"	"TGFB1"	"TMC6"	"ID3"	"ITGB2"
##	[131]	"LCP1"	"RNU1G2"	"S1PR3"	"GADD45G"	"NUPR1"
##	[136]	"ANLN"	"GEM"	"CD163"	"LOC387763"	
"ZFP36L1"						
##	[141]	"CTGF"	"IL8"	"PDK4"	"PLIN2"	
"SLC16A9"						
##	[146]	"CSF1R"	"IL17RB"	"LOC100133692"	"SCIN"	"TAGLN"
##	[151]	"TMEM106A"	"ADM"	"EDN1"	"GADD45B"	"HSPA1A"
##	[156]	"MT1M"	"APOLD1"	"C7"	"IGFBP5"	"GJA4"
##	[161]	"LEP"	"PLXDC2"			

The unique bottom genes are:

bottomBoth

##	[1]	"SYT1"	"CHGB"	"PCSK1"	"VGF"	"MAL2"
##	[6]	"RGS4"	"GABRA1"	"NPTX2"	"SST"	"STAT4"
##	[11]	"C10RF173"	"CBLN4"	"DYNC1I1"	"STXBP1"	"GNG2"
##	[16]	"RGS7"	"TSPAN13"	"EFCBP1"	"NELL2"	"HPRT1"
##	[21]	"TAGLN3"	"DCLK1"	"ST6GALNAC5"	"GLRB"	"BEX5"
##	[26]	"DIRAS2"	"SYP"	"KIAA1107"	"PAK1"	"PTPRN"
##	[31]	"SERPINI1"	"ZCCHC12"	"NELL1"	"DACH2"	"ELMOD1"
##	[36]	"NECAB1"	"NEFM"	"PARM1"	"RTN1"	"VAMP2"
##	[41]	"ENC1"	"TMEM16C"	"CREG2"	"STX1A"	"SVOP"
##	[46]	"CAP2"	"SCG2"	"UCHL1"	"CALY"	
		"KIAA0748"				
##	[51]	"NAP1L3"	"RIMBP2"	"YWHAG"	"GABRG2"	
		"LOC100128403"				
##	[56]	"NAP1L2"	"NAP1L5"	"STMN2"	"ADCYAP1"	"SCN3B"
##	[61]	"GAD1"	"HS.390250"	"INA"	"SCN2B"	
		"GPRASP2"				
##	[66]	"ITFG1"	"SNX10"	"SV2B"	"ATP6V1G2"	"GABBR2"
##	[71]	"GAD2"	"HPCA"	"NUDT11"	"LPPR4"	
		"TMEM155"				
##	[76]	"CAMK1G"	"EPHA4"	"GABRA5"	"MYT1L"	
		"HS.553187"				
##	[81]	"CDH13"	"VSNL1"	"XK"	"CKMT1A"	
		"HS.31961"				
##	[86]	"PCSK1N"	"PRKCG"	"SYT13"	"NPY"	"PRKCB"
##	[91]	"RASL11B"	"SYN2"	"CADPS"	"TSPYL1"	
		"C12ORF53"				
##	[96]	"HSPB3"	"MKL2"	"PPP3R1"	"SNCA"	
		"C2ORF80"				
##	[101]	"ANO3"	"LOC387856"	"SLC30A3"	"CRYM"	
		"C11ORF87"				
##	[106]	"RPH3A"	"VIP"	"OLFM3"	"COPG2IT1"	
		"SLC39A10"				
##	[111]	"SLITRK4"	"CKMT1B"	"CYP26B1"	"CPNE4"	"MFF"
##	[116]	"TUBB2A"	"ELAVL4"	"LRFN5"	"NCALD"	
		"SLC6A17"				
##	[121]	"YWHAE"	"CCKBR"	"FGF12"	"G3BP2"	"HOPX"
##	[126]	"NMNAT2"	"ZNF365"	"C20ORF103"		

The top genes in common are those genes most expressed in fold change in females AND males who have Alzheimer:

```
A <- as.data.frame(a)
B <- as.data.frame(b)

topBoth <- merge(A,B,by.x='a', by.y='b')
topBoth$a
```

##	[1]	ACACB	AEBP1	AHNAK	APLNR	C15ORF52	CALD1	CAPS
##	[8]	CFLAR	CHST6	DDIT4L	FAM65C	FCGBP	FOXC1	GPB
##	[15]	GSDMD	HEY2	HS.37648	ITGB5	ITPKB	KIF1C	LOC643287

```
## [22] LRRC32      MYOT      PDGFRB    PLXNB1    PTH1R     RAB13     RASL12
## [29] RN7SK       RUNDC2C   SIPA1     SLC5A3    SLC6A12   SLC7A2    SYTL4
## [36] TEAD2       TNFRSF1B  VCAN
## 100 Levels: ACACB AEBP1 AHNAK AKR1C3 ANGPT2 APLNR C15ORF52 C10RF110 ...
ZNF786
```

The bottom genes in common are the least expressed genes in females AND males who have Alzheimer:

```
C <- as.data.frame(c)
D <- as.data.frame(d)

bottomBoth <- merge(C,D, by.x='c', by.y='d')
bottomBoth$c

## [1] ADCYAP1      BEX5          C12ORF53      C10RF173      C20RF80
## [6] CADPS        CALY          CAP2          CBLN4         CHGB
## [11] CREG2        DACH2         DCLK1         DIRAS2        DYNC1I1
## [16] EFCBP1       ELMOD1        ENC1          EPHA4         GABBR2
## [21] GABRA1       GABRG2        GAD1          GAD2          GLRB
## [26] GPRASP2      HPRT1         HS.390250     HS.553187     HSPB3
## [31] INA          KIAA0748      KIAA1107      LOC100128403  MAL2
## [36] NAP1L3       NAP1L5        NEFM          NELL1         NELL2
## [41] NPTX2        PAK1          PARM1         PCSK1         PRKCG
## [46] PTPRN        RGS4          RGS7          RIMBP2        RTN1
## [51] SCG2         SCN2B         SERPINI1      SNX10         SST
## [56] ST6GALNAC5   STAT4         STMN2         STX1A         STXBP1
## [61] SV2B         SVOP          SYP           SYT1          SYT13
## [66] TAGLN3       TMEM155       TMEM16C       TSPAN13       VAMP2
## [71] VGF          ZCCHC12
## 100 Levels: ADCYAP1 ATP6V1G2 BEX5 C12ORF53 C10RF173 C20RF80 CADPS ...
ZCCHC12
```

Lets look at three top expressed and three least expressed genes in common for females and males:

```
b3 <- as.character(bottomBoth$c[1:3])
t3 <- as.character(topBoth$a[1:3])

mix <- c(b3,t3)
mix

## [1] "ADCYAP1" "BEX5"      "C12ORF53" "ACACB"     "AEBP1"     "AHNAK"

Mix <- as.data.frame(mix)
mBoth <- Both
mBoth$gene <- row.names(mBoth)
MixBoth <- merge(Mix, mBoth, by.x='mix', by.y='gene')

toPlot <- MixBoth[,c(1,6,7)]
```

```

colnames(toPlot)[1] <- 'gene'
toPlot

##      gene FC_fem_ctrl_AD FC_male_ctrl_AD
## 1  ACACB      1.092      1.085
## 2 ADCYAP1      0.889      0.895
## 3  AEBP1      1.107      1.075
## 4  AHNAK      1.099      1.079
## 5   BEX5      0.877      0.907
## 6 C12ORF53     0.902      0.922

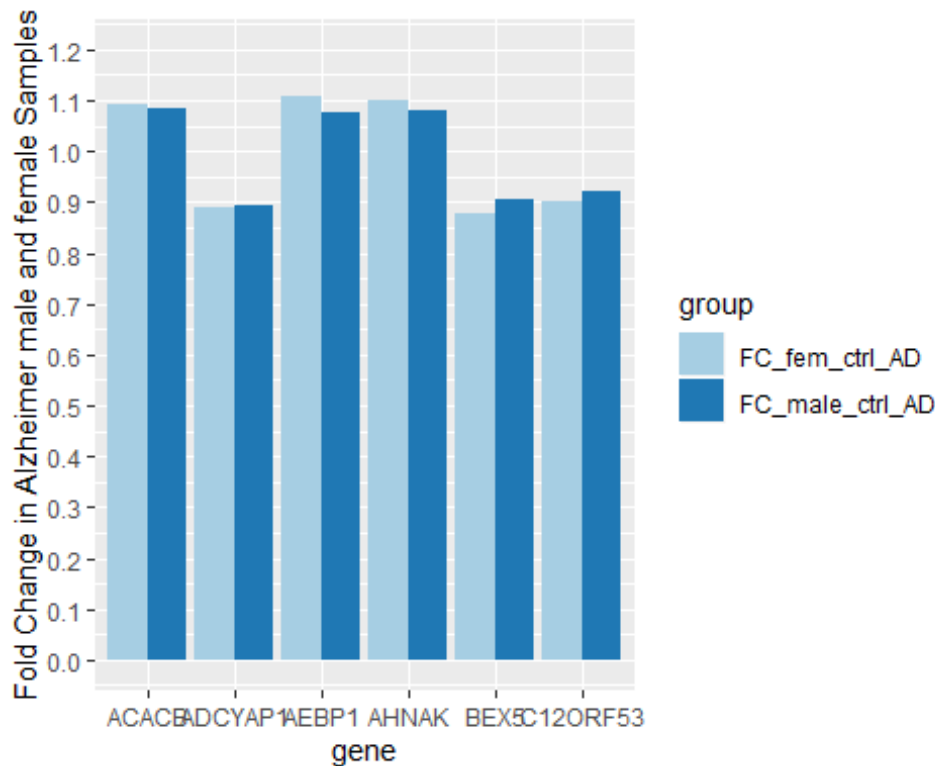
library(ggplot2)
library(tidyr)

to_plot <- gather(toPlot, 'group', 'foldChange', 2:3)
to_plot

##      gene      group foldChange
## 1  ACACB FC_fem_ctrl_AD      1.092
## 2 ADCYAP1 FC_fem_ctrl_AD      0.889
## 3  AEBP1 FC_fem_ctrl_AD      1.107
## 4  AHNAK FC_fem_ctrl_AD      1.099
## 5   BEX5 FC_fem_ctrl_AD      0.877
## 6 C12ORF53 FC_fem_ctrl_AD      0.902
## 7  ACACB FC_male_ctrl_AD      1.085
## 8 ADCYAP1 FC_male_ctrl_AD      0.895
## 9  AEBP1 FC_male_ctrl_AD      1.075
## 10 AHNAK FC_male_ctrl_AD      1.079
## 11  BEX5 FC_male_ctrl_AD      0.907
## 12 C12ORF53 FC_male_ctrl_AD      0.922

ggplot(data = to_plot, aes(x=gene, y=foldChange, fill=group)) +
  geom_bar(stat='identity', position=position_dodge()) +
  scale_y_continuous(breaks = seq(0, 1.2, by=.1), limits=c(0,1.2)) +
  scale_fill_brewer(palette='Paired') +
  ylab('Fold Change in Alzheimer male and female Samples')

```



Some other data sets on diseases would be interesting to compare to this data on Alzheimer patients. The other data sets to compare this blood tissue type of microarray gene expression profiles are the: hemochromatosis (iron toxicity) from <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE121620>, epigallocatechin EGCG (green tea extract) use by overweight females in 40s found at <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74560>, myocardial infarction MI (heart disease) in Russian males in 40s found at <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE141512>, antibiotic treatment in flu vaccinated patients of males and females between 18-45 years of age found at <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120717> and <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120719>, and an immunization blood sample data set that only has healthy donors blood six days after receiving the tetanus-diphtheria toxoids and acellular pertussis found at <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE59697>.

Those data sets are needed for their stats with Fold Change values for each study respectively, except the tetanus shot because it is only healthy donor samples analyzing the tetanus-diphtheria toxoids as an antigen to test the B lymphocytes route in the plasma cells.

The females in the overweight study, the females in the flu study, and the females in the iron toxicity study could be cross compared to the females in this study on Alzheimer female patients. The males in the heart disease study, iron toxicity study, and flu study could be compared to the males in this Alzheimer study. And both genders from those studies could also be compared to both genders in this study.

The plan is to get the genes these studies all have in common and compare between the genders on fold change stats of those genes. That being iron toxicity fold change, and flu vaccinated fold change compared to added antibiotics (not compared to healthy non vaccinated patients) for comparison with Alzheimer patients. Overweight females using EGCG for weight loss is exclusive to the female comparisons, and the heart disease study was exclusive to male patients.

This study's data on fold change, means, and sample values is the **both.csv** file. The flu study had days 0,3,7, and 21 comparisons and both genders in the **FoldChange_All.csv** file for that study. The iron toxicity fold change stats is in the **all-fc-grops.csv** file for the hemochromatosis study on both genders. The overweight females using EGCG to diet with the included fold change values for those using EGCG or EGCG+vitamin C+fish oil is in the **foldChange_EGCG.csv** file. And the males who have heart disease of type MI is in the file **foldChange_MI_males.csv**. The tetanis immunization samples don't have the fold change because they are all healthy donors with no age data or gender data and taken 6 days after the tetanis shot, but it is useful to compare genes with healthy females and males in the healthy and control groups **tetanisImmunizationBlood.csv** is that file.

Lets read in those files with their respective and appropriately described names.

```
Alzheimer <- read.csv('both.csv', sep=',', header=T, na.strings=c('', ' '))
Heart_Disease_Males <- read.csv('foldChange_MI_males.csv', sep=',',
                                header=T, na.strings=c('', ' '))
TetanisImmunity <- read.csv('tetanisImmunizationBlood.csv', sep=',',
                             header=T, na.strings=c('', ' '))
Overweight_Females_EGCG <- read.csv('foldChange_EGCG.csv', sep=',',
                                     header=T, na.strings=c('', ' '))
flu_Vaccinated <- read.csv('FoldChange_All.csv', sep=',', header=T,
                           na.strings=c('', ' '))
iron_toxicity <- read.csv('all-fc-grops.csv', sep=',', header=T,
                          na.strings=c('', ' '))
```

Look at the column names and select the ones needed or that have fold change values to cross compare, or to compare means across samples as with the tetanis data set.

The Alzheimer column names are:

```
colnames(Alzheimer)
```

## [1] "X"	"Females_AD1_Mean"	
"Females_control1_Mean"		
## [4] "males_AD1_Mean"	"males_control1_Mean"	"FC_fem_ctrl_AD"
## [7] "FC_male_ctrl_AD"	"GSM2973262"	"GSM2973263"
## [10] "GSM2973264"	"GSM2973265"	"GSM2973266"
## [13] "GSM2973267"	"GSM2973268"	"GSM2973269"
## [16] "GSM2973270"	"GSM2973271"	"GSM2973272"
## [19] "GSM2973273"	"GSM2973274"	"GSM2973275"
## [22] "GSM2973276"	"GSM2973277"	"GSM2973278"
## [25] "GSM2973279"	"GSM2973280"	"GSM2973281"
## [28] "GSM2973282"	"GSM2973283"	"GSM2973284"

## [31]	"GSM2973285"	"GSM2973286"	"GSM2973287"
## [34]	"GSM2973288"	"GSM2973289"	"GSM2973290"
## [37]	"GSM2973291"	"GSM2973292"	"GSM2973293"
## [40]	"GSM2973294"	"GSM2973295"	"GSM2973296"
## [43]	"GSM2973297"	"GSM2973298"	"GSM2973299"
## [46]	"GSM2973300"	"GSM2973301"	"GSM2973302"
## [49]	"GSM2973303"	"GSM2973304"	"GSM2973305"
## [52]	"GSM2973306"	"GSM2973307"	"GSM2973308"
## [55]	"GSM2973309"	"GSM2973310"	"GSM2973311"
## [58]	"GSM2973312"	"GSM2973313"	"GSM2973314"
## [61]	"GSM2973315"	"GSM2973316"	"GSM2973317"
## [64]	"GSM2973318"	"GSM2973319"	"GSM2973320"
## [67]	"GSM2973321"	"GSM2973322"	"GSM2973323"
## [70]	"GSM2973324"	"GSM2973325"	"GSM2973326"
## [73]	"GSM2973327"	"GSM2973328"	"GSM2973329"
## [76]	"GSM2973330"	"GSM2973331"	"GSM2973332"
## [79]	"GSM2973333"	"GSM2973334"	"GSM2973335"
## [82]	"GSM2973336"	"GSM2973337"	"GSM2973338"
## [85]	"GSM2973339"		

The heart disease column names are:

`colnames(Heart_Disease_Males)`

## [1]	"X"	"Symbol"
## [3]	"FC_MI_males"	"HealthyMale_Means"
## [5]	"MI_Male_Means"	"healthy_Males_GSM4205364"
## [7]	"healthy_Males_GSM4205363"	"healthy_Males_GSM4205362"
## [9]	"healthy_Males_GSM4205361"	"healthy_Males_GSM4205360"
## [11]	"healthy_Males_GSM4205359"	"MI_Males_GSM4205358"
## [13]	"MI_Males_GSM4205357"	"MI_Males_GSM4205356"
## [15]	"MI_Males_GSM4205355"	"MI_Males_GSM4205354"
## [17]	"MI_Males_GSM4205353"	

The tetanus immunization after six days taken from their blood has the following column names:

`colnames(TetanusImmunity)`

## [1]	"X"	"GENE_SYMBOL"	"GSM1443061"	"GSM1443062"
## [5]	"GSM1443063"	"GSM1443064"	"GSM1443065"	"GSM1443066"
## [9]	"Tetanus_Means"			

The EGCG overweight females column names are:

`colnames(Overweight_Females_EGCG)`

## [1]	"X"	"FC_egcg_quer"
## [3]	"FC_egcg"	"DE_EGCG"
## [5]	"DE_Quercentin"	"Pre_Means"
## [7]	"Post_EGCG_Means"	"Post_EGCG_Quercentin_Means"
## [9]	"pre_GSM1923000"	"pre_GSM1923004"

```
## [11] "pre_GSM1923010"      "pre_GSM1923012"
## [13] "pre_GSM1923007"      "pre_GSM1923020"
## [15] "pre_GSM192998"        "pre_GSM1922995"
## [17] "pre_GSM1923002"      "pre_GSM1923008"
## [19] "pre_GSM1923015"      "pre_GSM1923018"
## [21] "pre_GSM1923022"      "pre_GSM1923017"
## [23] "post_EG_GSM1923001"   "post_EG_GSM1923005"
## [25] "post_EG_GSM1923011"   "post_EG_GSM1923013"
## [27] "post_EG_GSM1923021"   "post_EG_GSM1923006"
## [29] "post_EG_GSM1923014"   "post_EQ_GSM192996"
## [31] "post_EQ_GSM1923003"   "post_EQ_GSM1923009"
## [33] "post_EQ_GSM1923016"   "post_EQ_GSM1923019"
## [35] "post_EQ_GSM1923023"   "post_EQ_GSM192997"
```

The flu vaccinated (treated with antibiotics treatment or not as 't' or 'nt' respectively) for each of the initial day 0, day 3, day 7, and day 21 profiles from three treated and three not treated samples have the following column names:

```
colnames(flu_Vaccinated)
```

```
## [1] "Gene"      "FC_t1"
## [3] "FC_t3"      "FC_t7"
## [5] "FC_t21"     "FC_nt1"
## [7] "FC_nt3"     "FC_nt7"
## [9] "FC_nt21"    "FCB_1"
## [11] "FCB_3"      "FCB_7"
## [13] "FCB_21"     "T0_Mean"
## [15] "T1_Mean"    "T3_Mean"
## [17] "T7_Mean"    "T21_Mean"
## [19] "NT0_Mean"   "NT1_Mean"
## [21] "NT3_Mean"   "NT7_Mean"
## [23] "NT21_Mean"  "GSM3409106_29_day_0"
## [25] "GSM3409107_29_day_1"  "GSM3409108_29_day_3"
## [27] "GSM3409004_29_day_7"  "GSM3409105_29_day_21_screening"
## [29] "GSM3409006_30_day_0"  "GSM3409007_30_day_1"
## [31] "GSM3409008_30_day_3"  "GSM3409009_30_day_7"
## [33] "GSM3409005_30_day_21_screening" "GSM3409013_05_day_0"
## [35] "GSM3409014_05_day_1"  "GSM3409015_05_day_3"
## [37] "GSM3409016_05_day_7"  "GSM3409012_05_day_21_screening"
## [39] "GSM3409161_33_day_0_no" "GSM3409162_33_day_1_no"
## [41] "GSM3409163_33_day_3_no" "GSM3409111_33_day_7_no"
## [43] "GSM3409160_33_day_21_screening_no" "GSM3409124_36_day_0_no"
## [45] "GSM3409125_36_day_1_no" "GSM3409126_36_day_3_no"
## [47] "GSM3409127_36_day_7_no"
"GSM3409123_36_day_21_screening_no"
## [49] "GSM3409135_38_day_0_no" "GSM3409136_38_day_1_no"
## [51] "GSM3409137_38_day_3_no" "GSM3409138_38_day_7_no"
## [53] "GSM3409134_38_day_21_screening_no"
```

The iron toxicity study put the males and females into three groups of the genotypes associated with hemochromatosis or iron toxicity. It also included healthy males and females to compare to. The fold change values are for the groups each compared to the healthy groups by gender by all three groups 'overall' and by each of the 3 groups separately. The iron toxicity column names are:

```
colnames(iron_toxicity)
```

```
## [1] "X"                "G1M_Mean"         "G2M_Mean"
## [4] "G3M_Mean"         "G1F_Mean"         "G2F_Mean"
## [7] "G3F_Mean"         "healthyFemale_Mean" "healthyMale_Mean"
## [10] "hemoFemale_Mean"  "hemoMale_Mean"    "FC_1m"
## [13] "FC_2m"            "FC_3m"            "FC_1F"
## [16] "FC_2F"            "FC_3F"            "FC_malesOverall"
## [19] "FC_femalesOverall" "GSM3440208"        "GSM3440209"
## [22] "GSM3440210"        "GSM3440211"        "GSM3440212"
## [25] "GSM3440213"        "GSM3440214"        "GSM3440215"
## [28] "GSM3440216"        "GSM3440217"        "GSM3440218"
## [31] "GSM3440219"        "GSM3440220"        "GSM3440221"
## [34] "GSM3440222"        "GSM3440223"        "GSM3440224"
## [37] "GSM3440225"        "GSM3440226"        "GSM3440227"
## [40] "GSM3440228"        "GSM3440229"        "GSM3440230"
## [43] "GSM3440231"
```

In the interest of being interesting, these tables have all the original samples in each respective table for that disease group. There are some healthy samples in some such as healthy overweight females before taking EGCG, non-Alzheimer's males and females, healthy Russian males without heart disease, and healthy females and males who do not have iron toxicity relative to the iron toxicity study. There are some studies that do not have healthy controls nor separation by gender such as the tetanis immunity and flu vaccinated samples.

Since the tetanis immunized samples don't have a control or healthy group to compare to which haven't been immunized, let us compare it to the samples of healthy overweight females, healthy Russian males, healthy females and males without iron toxicity, and flu vaccinated not treated with antibiotics and flu vaccinated treated with antibiotics. The values in the tetanis immunized samples have a very high range for some genes and doesn't get lower than 0.08 for any gene. We will compare the fields of each table that has the **mean** values across sample for each **gene** in that group by creating a table with those means.

```
tet_mean <- TetanisImmunity[,c(2,9)] #gene and tetanis immunized mean
iron_mean <- iron_toxicity[,c(1,8:11)]#gene, healthy female, healthy male
# iron toxic females, iron toxic males
flu_7day <- flu_Vaccinated[,c(1,22,17)] #gene, not treated, treated
EGCG_females <- Overweight_Females_EGCG[,c(1,6:8)]#gene, pre,post EGCG,
# post EGCG+vitamin C+Fish Oil
MI_healthy_mean <- Heart_Disease_Males[,c(1,4,5)]#gene, healthy males,
# MI males
```



```
alz_means <- Alzheimer[,c(1:5)]#gene, female AD, female control, male AD,  
# male control
```

Let us first compare the tetanis immunized to the healthy females no iron toxicity, healthy males no iron toxicity, overweight females EGCG, healthy males no MI, healthy AD females, and healthy AD males.

Gather only the healthy females and males of those studies.

```
healthy_iron <- iron_mean[,c(1:3)]  
Healthy_EGCG_females <- EGCG_females #all are healthy  
healthy_MImales <- MI_healthy_mean[,1:2]  
healthy_AD <- alz_means[,c(1,3,5)]  
  
dim(healthy_iron)  
## [1] 28433      3  
  
colnames(healthy_iron)[2:3] <-  
gsub('healthy','iron_healthy_',colnames(healthy_iron)[2:3])  
colnames(healthy_iron)  
  
## [1] "X"                                "iron_healthy_Female_Mean"  
## [3] "iron_healthy_Male_Mean"  
  
dim(Healthy_EGCG_females)  
## [1] 22831      4  
  
colnames(Healthy_EGCG_females)[2:4] <-  
paste('overweight_females_',colnames(Healthy_EGCG_females)[2:4],sep='')  
colnames(Healthy_EGCG_females)  
  
## [1] "X"  
## [2] "overweight_females_Pre_Means"  
## [3] "overweight_females_Post_EGCG_Means"  
## [4] "overweight_females_Post_EGCG_Quercentin_Means"  
  
dim(healthy_MImales)  
## [1] 30905      2  
  
colnames(healthy_MImales)[2] <-  
paste('MI_study_',colnames(healthy_MImales)[2], sep='')  
colnames(healthy_MImales)  
  
## [1] "X"                                "MI_study_HealthyMale_Means"  
  
dim(healthy_AD)  
## [1] 31700      3
```

```

colnames(healthy_AD)[2:3] <-
gsub('_control1_', '_no_AD_', colnames(healthy_AD)[2:3])
colnames(healthy_AD)

## [1] "X"                "Females_no_AD_Mean" "males_no_AD_Mean"

dim(tet_mean)

## [1] 19749      2

colnames(tet_mean)

## [1] "GENE_SYMBOL"      "Tetanis_Means"

```

The above tables all have a differing amount of genes in each table, so we need to merge the tables by genes in common to compare. The gene field is the 'X' field in these tables listed immediately above. We need to merge them all to the tetanis table, tet_mean, to compare with tetanis immunity gene expression values over 6 days.

```

healthy0 <- merge(tet_mean, healthy_iron, by.x='GENE_SYMBOL', by.y='X')
dim(healthy0)

## [1] 15059      4

healthy1 <- merge(healthy0, Healthy_EGCG_females, by.x='GENE_SYMBOL',
by.y='X')
dim(healthy1)

## [1] 14639      7

healthy2 <- merge(healthy1, healthy_MImales, by.x='GENE_SYMBOL', by.y='X')
dim(healthy2)

## [1] 14475      8

healthy3 <- merge(healthy2, healthy_AD, by.x='GENE_SYMBOL', by.y='X')
dim(healthy3)

## [1] 13193     10

colnames(healthy3)

## [1] "GENE_SYMBOL"
## [2] "Tetanis_Means"
## [3] "iron_healthy_Female_Mean"
## [4] "iron_healthy_Male_Mean"
## [5] "overweight_females_Pre_Means"
## [6] "overweight_females_Post_EGCG_Means"
## [7] "overweight_females_Post_EGCG_Quercentin_Means"
## [8] "MI_study_HealthyMale_Means"
## [9] "Females_no_AD_Mean"
## [10] "males_no_AD_Mean"

```

```
head(healthy3)
```

```
##  GENE_SYMBOL Tetanis_Means iron_healthy_Female_Mean
iron_healthy_Male_Mean
## 1      A1BG  5.991310e+06                4.816690
4.706227
## 2      A1CF  7.843977e+06                2.546950
2.229808
## 3      A2M   2.133333e-01                3.998705
4.323837
## 4     A2ML1  2.290235e+06                3.357760
3.048290
## 5      AAAS  2.676830e+07                5.287245
5.522265
## 6      AACS  2.511941e+06                5.471870
5.673802
##  overweight_females_Pre_Means overweight_females_Post_EGCG_Means
## 1                        5.789                        5.703
## 2                        4.959                        5.043
## 3                        5.221                        5.215
## 4                        4.443                        4.516
## 5                        8.117                        8.109
## 6                        6.799                        6.835
##  overweight_females_Post_EGCG_Quercentin_Means MI_study_HealthyMale_Means
## 1                        5.697                        5.653239
## 2                        4.914                        2.616675
## 3                        5.236                        3.568283
## 4                        4.483                        3.014349
## 5                        8.128                        7.043090
## 6                        6.787                        5.423620
##  Females_no_AD_Mean males_no_AD_Mean
## 1                6.647                6.619
## 2                6.495                6.506
## 3               10.231               10.377
## 4                6.571                6.568
## 5                6.875                6.828
## 6                7.946                7.907
```

We need to use dplyr to analyze this data as is for the healthy samples compared to the tetanis samples.

```
library(dplyr)
```

Get the fold change 'FC' values for the between samples of genes using the healthy3 table.

```
FC0 <- healthy3 %>% mutate(FC_iron_fem =
Tetanis_Means/iron_healthy_Female_Mean)

FC1 <- FC0 %>% mutate(FC_iron_mal = Tetanis_Means/iron_healthy_Male_Mean)

FC2 <- FC1 %>% mutate(FC_heavy_fems_no_EGCG =
```

```

Tetanis_Means/overweight_females_Pre_Means)

FC3 <- FC2 %>% mutate(FC_heavy_fems_EGCG =
Tetanis_Means/overweight_females_Post_EGCG_Means)

FC4 <- FC3 %>% mutate(FC_heavy_fems_EGCG_plus =
Tetanis_Means/overweight_females_Post_EGCG_Quercentin_Means)

FC5 <- FC4 %>% mutate(FC_healthyMI_males =
Tetanis_Means/MI_study_HealthyMale_Means)

FC6 <- FC5 %>% mutate(FC_fems_noAD = Tetanis_Means/Females_no_AD_Mean)
FC7 <- FC6 %>% mutate(FC_mals_noAD = Tetanis_Means/males_no_AD_Mean)

healthy_tetanis_FCs <- FC7
dim(healthy_tetanis_FCs)

## [1] 13193      18

colnames(healthy_tetanis_FCs)

## [1] "GENE_SYMBOL"
## [2] "Tetanis_Means"
## [3] "iron_healthy_Female_Mean"
## [4] "iron_healthy_Male_Mean"
## [5] "overweight_females_Pre_Means"
## [6] "overweight_females_Post_EGCG_Means"
## [7] "overweight_females_Post_EGCG_Quercentin_Means"
## [8] "MI_study_HealthyMale_Means"
## [9] "Females_no_AD_Mean"
## [10] "males_no_AD_Mean"
## [11] "FC_iron_fem"
## [12] "FC_iron_mal"
## [13] "FC_heavy_fems_no_EGCG"
## [14] "FC_heavy_fems_EGCG"
## [15] "FC_heavy_fems_EGCG_plus"
## [16] "FC_healthyMI_males"
## [17] "FC_fems_noAD"
## [18] "FC_mals_noAD"

```

There are 13,193 genes to compare fold change values of tetanis immunized samples to the healthy control samples from the combined studies.

```
write.csv(healthy_tetanis_FCs, 'healthy_tetanis_FCs.csv', row.names=FALSE)
```

Now, compare the tetanis immunized to the flu vaccinated samples treated and not treated with antibiotics. The flu gene samples seemed to have the most changes in values after 7 days of receiving the flu immunization and the antibiotic or no antibiotic addition. This is why the flu immunizations for day 7 means were selected. Only the tet_mean and flu_7day tables will be used for genes in common as neither has the gender information attached.

```
dim(tet_mean);colnames(tet_mean)

## [1] 19749      2

## [1] "GENE_SYMBOL"  "Tetanis_Means"

dim(flu_7day);colnames(flu_7day)

## [1] 20633      3

## [1] "Gene"      "NT7_Mean" "T7_Mean"
```

Merge these two tables of the tetanis and flu not treated (NT7 prefix) or treated (T7 prefix) means.

```
tet_flu_immune <- merge(tet_mean, flu_7day, by.x='GENE_SYMBOL', by.y='Gene')
dim(tet_flu_immune)

## [1] 17447      4

colnames(tet_flu_immune)

## [1] "GENE_SYMBOL"  "Tetanis_Means" "NT7_Mean"      "T7_Mean"
```

Now for the fold change values between the Tetanis_Means and the flu immunization means with/without added antibiotics treatment, dplyr will be used to get that information.

```
tetanis_flu_FCs <- tet_flu_immune %>% mutate(FC_tet_flu_nt7 =
Tetanis_Means/NT7_Mean)

tetanis_flu_FCs1 <- tetanis_flu_FCs %>% mutate(FC_tet_flu_t7 =
Tetanis_Means/T7_Mean)

dim(tetanis_flu_FCs1)

## [1] 17447      6

colnames(tetanis_flu_FCs1)

## [1] "GENE_SYMBOL"  "Tetanis_Means" "NT7_Mean"      "T7_Mean"
## [5] "FC_tet_flu_nt7" "FC_tet_flu_t7"

head(tetanis_flu_FCs1)

##   GENE_SYMBOL Tetanis_Means NT7_Mean  T7_Mean FC_tet_flu_nt7 FC_tet_flu_t7
## 1      A1BG  5.991310e+06 4.237321 4.283943  1.413938e+06  1.398550e+06
## 2      A1CF  7.843977e+06 3.081198 3.188847  2.545755e+06  2.459816e+06
## 3     A2BP1  3.166667e-01 2.905718 3.187804  1.089805e-01  9.933694e-02
## 4     A2LD1  8.166667e-02 3.233051 4.178542  2.525994e-02  1.954430e-02
## 5       A2M  2.133333e-01 2.411421 3.146397  8.846789e-02  6.780241e-02
## 6     A2ML1  2.290235e+06 2.327324 2.872788  9.840637e+05  7.972170e+05
```

```
write.csv(tetanis_flu_FCs1, 'tetanis_flu_FCs.csv', row.names=FALSE)
```