DEG group comparison analysis for Alzheimer's

Project: DEG Analysis on gene expression data of patients in different stages of Alzheimer's Disease.

Purpose: To find out what are the significant differences of gene expression between groups of subjects at various stages of the disease.

Team 4: My team is tasked with comparing moderate group vs. incipient group and sever group vs. moderate group.

This file contains analysis between gene expression data of moderate group and incipient group of patients.

1a - Read in clinical data

```
#30 patients
clinData <- read.csv(file = "input/Blalock_clin_final.csv", header = T, stringsAsFactors = F, row.names


head(clinData)
```

```
##              GENDER AGE DISEASE_STATUS BIOSPECIMEN_ID
## Control 1003   MALE  80        Control      GSM697309
## Control 1008 FEMALE  92        Control      GSM697310
## Control 1012   MALE  80        Control      GSM697311
## Control 1015   MALE  75        Control      GSM697312
## Control 1018 FEMALE  97        Control      GSM697313
## Control 1030   MALE  95        Control      GSM697314
```

```
knitr::kable(head(clinData), caption = "Clinical Data with rows as patients and attributes as columns")
```

Table 1: Clinical Data with rows as patients and attributes as columns

	GENDER	AGE	DISEASE_STATUS	BIOSPECIMEN_ID
Control 1003	MALE	80	Control	GSM697309
Control 1008	FEMALE	92	Control	GSM697310
Control 1012	MALE	80	Control	GSM697311
Control 1015	MALE	75	Control	GSM697312
Control 1018	FEMALE	97	Control	GSM697313
Control 1030	MALE	95	Control	GSM697314

Checking the shape and overall information about the clinical dataset

```r
dim(clinData)
```

```
## [1] 30  4
```

1b: Read in processed normalized gene expression data in log2 scale, includes gene annotation

```r
# Read in gene expression file - features (rows), 30 patients (columns)
geneExp <- read.table(file = "input/GSE62232_Blalock_geneexp_final.tsv",
                      sep="\t",
                      row.names = 1,
                      header = T,
                      stringsAsFactors = F)
head(geneExp[1:5, 1:4])
```

```
##                   GSM697308 GSM697309 GSM697310 GSM697311
## 1007_s_at|DDR1     6.170842  5.437054  5.849658  5.474436
## 1053_at|RFC2       1.759196  1.858930  1.934719  2.256952
## 117_at|HSPA6       3.880619  3.864713  5.161341  4.112907
## 121_at|PAX8        5.000021  4.503467  4.413525  5.037864
## 1255_g_at|GUCA1A   3.985697  3.188188  2.657556  3.449957
```

```r
knitr::kable(head(geneExp), caption = "Processed gene expression data in log2 scale with gene annotation
```

Table 2: Processed gene expression data in log2 scale with gene annotation

	[illegible]	GSM697337
1007[illegible]	6.1[illegible]	[illegible]761
1053[illegible]	1.75[illegible]	[illegible]9028
117_[illegible]	3.88[illegible]	[illegible]3792
121_[illegible]	5.00[illegible]	[illegible]6781
1255[illegible]	3.98[illegible]	[illegible]0517
1294[illegible]	4.63[illegible]	[illegible]6918

Now lets combine the dataset using the Patient's GSMID as the unique identifier

```r
clinData$BIOSPECIMEN_ID # these are the identifiers in the clinical data
```

```
##  [1] "GSM697309" "GSM697310" "GSM697311" "GSM697312" "GSM697313" "GSM697314"
##  [7] "GSM697315" "GSM697308" "GSM697319" "GSM697320" "GSM697321" "GSM697322"
## [13] "GSM697316" "GSM697317" "GSM697318" "GSM697327" "GSM697328" "GSM697329"
## [19] "GSM697330" "GSM697323" "GSM697324" "GSM697325" "GSM697326" "GSM697337"
## [25] "GSM697331" "GSM697332" "GSM697333" "GSM697334" "GSM697335" "GSM697336"
```

```r
colnames(geneExp) # same identifiers in the gene exp data
```

```
##  [1] "GSM697308" "GSM697309" "GSM697310" "GSM697311" "GSM697312" "GSM697313"
##  [7] "GSM697314" "GSM697315" "GSM697316" "GSM697317" "GSM697318" "GSM697319"
## [13] "GSM697320" "GSM697321" "GSM697322" "GSM697323" "GSM697324" "GSM697325"
## [19] "GSM697326" "GSM697327" "GSM697328" "GSM697329" "GSM697330" "GSM697331"
## [25] "GSM697332" "GSM697333" "GSM697334" "GSM697335" "GSM697336" "GSM697337"
```

```r
matchingSamples = which(colnames(geneExp) %in% (clinData$BIOSPECIMEN_ID)) # 30 patients
matchingSamples
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30
```

```r
subsetGeneExp <- geneExp[, matchingSamples]
head(subsetGeneExp)
```

```
##                 GSM697308 GSM697309 GSM697310 GSM697311 GSM697312 GSM697313
## 1007_s_at|DDR1   6.170842  5.437054  5.849658  5.474436  6.285563  5.436715
## 1053_at|RFC2     1.759196  1.858930  1.934719  2.256952  2.251992  1.813054
## 117_at|HSPA6     3.880619  3.864713  5.161341  4.112907  4.719614  3.260608
## 121_at|PAX8      5.000021  4.503467  4.413525  5.037864  4.078362  4.464851
## 1255_g_at|GUCA1A 3.985697  3.188188  2.657556  3.449957  3.076214  3.441813
## 1294_at|UBA7     4.638791  4.088865  4.369077  4.264800  3.329961  4.168488
##                 GSM697314 GSM697315 GSM697316 GSM697317 GSM697318 GSM697319
## 1007_s_at|DDR1   5.867604  6.097429  6.036111  5.969330  6.362341  6.418384
## 1053_at|RFC2     1.831469  1.669274  1.774645  1.895006  2.004433  2.183885
## 117_at|HSPA6     3.578447  3.712416  3.341475  4.784170  3.903785  3.981612
## 121_at|PAX8      4.102676  4.317918  4.203470  4.936325  4.195753  4.865569
## 1255_g_at|GUCA1A 3.335569  2.670690  2.776477  2.386652  3.907316  2.715844
## 1294_at|UBA7     4.068348  3.647039  4.258798  5.439319  4.017352  4.117107
##                 GSM697320 GSM697321 GSM697322 GSM697323 GSM697324 GSM697325
## 1007_s_at|DDR1   5.836035  6.216795  5.623314  6.332254  6.265997  6.268595
## 1053_at|RFC2     1.897643  2.089310  2.559907  1.745056  1.611377  1.853608
## 117_at|HSPA6     3.297919  3.611989  3.677024  3.853263  4.106096  3.781244
## 121_at|PAX8      4.334031  4.309009  4.223607  4.786270  4.474938  4.434910
## 1255_g_at|GUCA1A 2.462689  2.723987  2.377768  3.265388  2.917575  2.510591
## 1294_at|UBA7     4.262979  4.227645  3.846864  4.481490  4.315105  4.177122
##                 GSM697326 GSM697327 GSM697328 GSM697329 GSM697330 GSM697331
## 1007_s_at|DDR1   6.081696  5.087591  6.698347  5.677410  6.509104  6.283506
## 1053_at|RFC2     1.638858  1.727018  1.748224  2.098544  1.771952  2.246568
## 117_at|HSPA6     3.251731  3.494550  3.886255  4.190944  3.798462  3.726193
## 121_at|PAX8      4.350030  4.738870  4.562518  4.887229  4.377327  4.431056
## 1255_g_at|GUCA1A 2.922083  2.378818  3.559576  2.596038  2.890413  2.692837
## 1294_at|UBA7     3.952723  4.449213  4.376851  4.380287  4.457181  4.128005
##                 GSM697332 GSM697333 GSM697334 GSM697335 GSM697336 GSM697337
## 1007_s_at|DDR1   5.983837  6.573478  6.007344  6.183228  6.633673  6.834761
## 1053_at|RFC2     2.005551  1.870484  1.609222  1.968057  1.973800  1.939028
## 117_at|HSPA6     8.476790  4.105462  4.355081  4.195202  3.949483  3.721792
## 121_at|PAX8      4.741944  4.549978  4.765414  4.904877  4.586500  4.193781
## 1255_g_at|GUCA1A 2.443887  2.443819  3.171695  3.237434  3.167422  4.010517
## 1294_at|UBA7     4.104519  4.408773  4.655337  4.799650  4.482796  4.076918
```

Step 3 - Identifying the groups to be compared

Identifying the groups to be compared (Baseline and Comparison Grps)

In this case Baseline = Incipient and Comparison = Moderate

```
# Labels (row numbers) that can identify the baseline group patients
baselineGrpLabels <-  which(clinData$DISEASE_STATUS == "Incipient") #7 samples
head(baselineGrpLabels)
```

```
## [1]  9 10 11 12 13 14
```

```
length(baselineGrpLabels)
```

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

```
# Use the labels (row numbers) to subset baseline patients in clinical data file
clinBase <- clinData[baselineGrpLabels, ]
clinBase
```

```
##                GENDER AGE DISEASE_STATUS BIOSPECIMEN_ID
## Incipient 1019   MALE  88      Incipient      GSM697319
## Incipient 1029 FEMALE  91      Incipient      GSM697320
## Incipient 1034   MALE  88      Incipient      GSM697321
## Incipient 1043 FEMALE  97      Incipient      GSM697322
## Incipient 715  FEMALE 101      Incipient      GSM697316
## Incipient 720  FEMALE  95      Incipient      GSM697317
## Incipient 994  FEMALE  83      Incipient      GSM697318
```

```
# Labels (row numbers) that can identify the comp group patients
compGrpLabels <- which(clinData$DISEASE_STATUS == "Moderate") #8 samples
head(compGrpLabels)
```

```
## [1] 16 17 18 19 20 21
```

```
length(compGrpLabels)
```

```
## [1] 8
```

```
# Use the labels (row numbers) to subset comp patients in clinical data file
clinComp <- clinData[compGrpLabels, ]
clinComp
```

```
##               GENDER AGE DISEASE_STATUS BIOSPECIMEN_ID
## Moderate 1020 FEMALE  79       Moderate      GSM697327
## Moderate 1025   MALE  81       Moderate      GSM697328
## Moderate 1031 FEMALE  86       Moderate      GSM697329
## Moderate 1037   MALE  82       Moderate      GSM697330
## Moderate 826  FEMALE  85       Moderate      GSM697323
## Moderate 832  FEMALE  89       Moderate      GSM697324
## Moderate 856  FEMALE  83       Moderate      GSM697325
## Moderate 965  FEMALE  82       Moderate      GSM697326
```

```
#### Use the clinBase and clinComp objects to subset gene expression data
geneExpBase <- subsetGeneExp[, clinBase$BIOSPECIMEN_ID] # 43135 feature (rows), 7 samples columns
geneExpComp <- subsetGeneExp[, clinComp$BIOSPECIMEN_ID] # 43135 feature (rows), 8 samples columns

head(geneExpBase)
```

```
##                  GSM697319 GSM697320 GSM697321 GSM697322 GSM697316 GSM697317
## 1007_s_at|DDR1    6.418384  5.836035  6.216795  5.623314  6.036111  5.969330
## 1053_at|RFC2      2.183885  1.897643  2.089310  2.559907  1.774645  1.895006
## 117_at|HSPA6      3.981612  3.297919  3.611989  3.677024  3.341475  4.784170
## 121_at|PAX8       4.865569  4.334031  4.309009  4.223607  4.203470  4.936325
## 1255_g_at|GUCA1A  2.715844  2.462689  2.723987  2.377768  2.776477  2.386652
## 1294_at|UBA7      4.117107  4.262979  4.227645  3.846864  4.258798  5.439319
##                  GSM697318
## 1007_s_at|DDR1    6.362341
## 1053_at|RFC2      2.004433
## 117_at|HSPA6      3.903785
## 121_at|PAX8       4.195753
## 1255_g_at|GUCA1A  3.907316
## 1294_at|UBA7      4.017352
```

```
head(geneExpComp)
```

```
##                  GSM697327 GSM697328 GSM697329 GSM697330 GSM697323 GSM697324
## 1007_s_at|DDR1    5.087591  6.698347  5.677410  6.509104  6.332254  6.265997
## 1053_at|RFC2      1.727018  1.748224  2.098544  1.771952  1.745056  1.611377
## 117_at|HSPA6      3.494550  3.886255  4.190944  3.798462  3.853263  4.106096
## 121_at|PAX8       4.738870  4.562518  4.887229  4.377327  4.786270  4.474938
## 1255_g_at|GUCA1A  2.378818  3.559576  2.596038  2.890413  3.265388  2.917575
## 1294_at|UBA7      4.449213  4.376851  4.380287  4.457181  4.481490  4.315105
##                  GSM697325 GSM697326
## 1007_s_at|DDR1    6.268595  6.081696
## 1053_at|RFC2      1.853608  1.638858
## 117_at|HSPA6      3.781244  3.251731
## 121_at|PAX8       4.434910  4.350030
## 1255_g_at|GUCA1A  2.510591  2.922083
## 1294_at|UBA7      4.177122  3.952723
```

Step 4: Sanity check

- See if filtering of clinical data in R matches filtering of clinical data in excel
- See if sample ids in clinical data match sample ids in gene exp data (if they don't match it means your step 1 and/or 2 is wrong)
- Verify you see correct number of samples in baseline and comp groups
- Export the column names from gene expression data to see if it contains only probe/gene names and no other garbage

```
#See if sample ids in clinical data match sample ids in gene exp data
clinBase$BIOSPECIMEN_ID == colnames(geneExpBase)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
clinComp$BIOSPECIMEN_ID == colnames(geneExpComp)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
### Checking if the samples IDs baseline and comp groups are the same ---- can put these in an excel to
write.table(clinBase$BIOSPECIMEN_ID , file = "sanity/Tazeen_ClinBaseIDs.tsv", sep="\t", quote = F  )
write.table(clinComp$BIOSPECIMEN_ID , file = "sanity/Tazeen_ClinCompIDs.tsv", sep="\t", quote = F  )
write.table(colnames(geneExpBase) ,
            file = "sanity/Tazeen_GeneExpBaseIDs.tsv",
            sep="\t", quote = F  )
write.table(colnames(geneExpComp) ,
            file = "sanity/Tazeen_GeneExpCompIDs.tsv",
            sep="\t", quote = F  )

#Export the features from gene expression data
#Open this file and check that it contains only probe/gene names and no other garbage
write.table(rownames(geneExp),file = "Tazeen_FeatureIDs.tsv", sep="\t", quote = F  )
```

Step 5: Preparing data for T-test

- Molecular data must have features (genes in this case) as rows, and samples as columns.
- Transpose data (if needed) to obtain this
- Objects must be data frame
- Numeric data only

```
### Checking to make sure data is a numeric data frame
knitr::kable(head(geneExpBase[1:5,1:4]))
```

	GSM697319	GSM697320	GSM697321	GSM697322
1007_s_at|DDR1	6.418384	5.836035	6.216794	5.623314
1053_at|RFC2	2.183885	1.897643	2.089310	2.559907
117_at|HSPA6	3.981612	3.297919	3.611988	3.677024
121_at|PAX8	4.865569	4.334030	4.309009	4.223607
1255_g_at|GUCA1A	2.715844	2.462689	2.723987	2.377768

```
knitr::kable(head(geneExpComp[1:5,1:4]))
```

	GSM697327	GSM697328	GSM697329	GSM697330
1007_s_at|DDR1	5.087591	6.698347	5.677410	6.509105
1053_at|RFC2	1.727018	1.748224	2.098544	1.771952
117_at|HSPA6	3.494550	3.886255	4.190944	3.798462
121_at|PAX8	4.738870	4.562518	4.887229	4.377327
1255_g_at|GUCA1A	2.378818	3.559576	2.596038	2.890413

```
source("fnTTest.R")

#### Call T-test function
results1 = fnTTest(baseGroup = geneExpBase,
                   compGroup = geneExpComp,
                   testName = "Tazeen_TTest_",
                   baseGroupName = "Incipient",
                   compGroupName =  "Moderate",
                   folderName = "output")
```

Function for T-test

Final Step - Sub-set top differentially expressed genes

```
#Read in the T-Test results file

ttestResults <- read.csv(file = "output/Tazeen_TTest__Moderate_(Comp).vs._Incipient_(Base).TTest.csv")

#check to make sure p-value column is imported as numeric
#sort by p-value (just in case the results are not sorted by p-value)

ttestResultsSorted <- dplyr::arrange(ttestResults, Pvalue)

#find rows with p-value < 0.01
whichSig <- which(ttestResultsSorted$Pvalue <= 0.01)

#Short list sig results
ttestResultsSig <- ttestResultsSorted[whichSig, ] #1789 rows

### Export short listed results
write.table(x = ttestResultsSig,
            file = "output/Tazeen_Moderate_Incipient_Ttest_Shortlisted.csv",
            quote = F, sep = ",")

##### First column is a list of features in thsi format : ProbeID|GeneName.
#### Use string split strsplit() function to extract gene names
funcSplit <- function(featureX) {
  f1 <- unlist(strsplit(x = featureX, split = "|", fixed = TRUE))
  f2 <- f1[2]
  return(f2)
}

# Use apply() function to run the split on every row, its faster version of a loop
geneNames1 <- apply(X = as.matrix(ttestResultsSig$Feature),
                    MARGIN = 1, FUN = funcSplit)

head(geneNames1)
```

```
## [1] "TNKS"    "EXOC7"   "NDUFA10" "FAM91A1" "KYNU"    "SAMD15"
```

```r
#print length of short listed gene names
length(geneNames1)
```

```
## [1] 157
```

```r
### Export list of gene names
write.table(x = geneNames1,
            file = "output/Tazeen_Moderate_Incipient_SigDiffExpressedGenes.csv",
            quote = F, sep = ",")
```