Final Project - Differential Expression Analysis in sporadic Alzheimer's disease patients

Varsha Neelakantan

11/19/2018

Introduction

One of the pathologies of **Alzheimers disease** is dysfunctional/disrupted BBB. **Pericytes** are cells that belong to the Blood-Brain Barrier and play an important role in maintaining the **blood brain barrier** integrity as well as in the functioning of the barrier. I want to identify differentially expressed genes in primary brain pericytes from sAD (Sporadic AD) patients compared to normal/WT pericytes. RNA seq data was obtained from Normal/Healthy patients and two sAD patients. The goal is to identify if the genetic basis of this disease is similar and what could be the major players in Alzheimers Disease

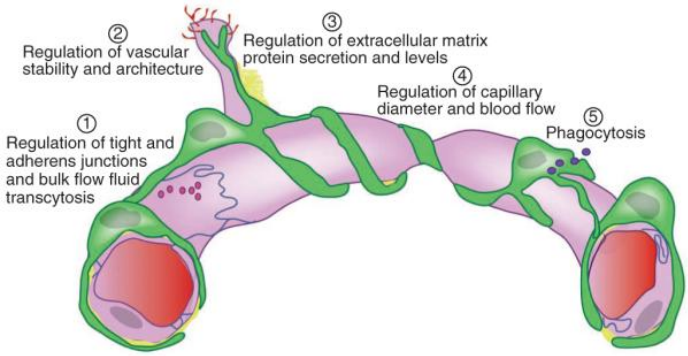


Figure1: Shows the localization of pericytes and its role in the blood brain barrier (Green represents pericytes and the pink represents endothelial cells)

Datasets used:

RNA Sequence data in FASTQ format are available from 2 sAD patients(in duplicates) and 1 WT.

Data Retrieval and Alignment

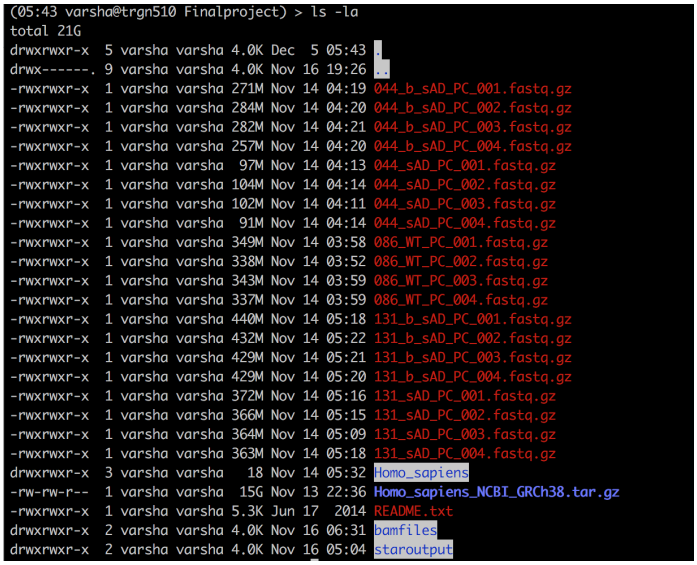
Get gene annotation files for STAR aligner (In Bash/Terminal)

```
mkdir Finalproject
cd Finalproject
wget ftp://igenome:G3nom3s4u@ussd-ftp.illumina.com/Homo_sapiens/NCBI/GRCh38/Homo_sapi
ens_NCBI_GRCh38.tar.gz
gunzip Homo_sapiens_NCBI_GRCh38.tar.gz
mkdir Homo_sapiens_NCBI_GRCh38/Homo_sapiens/NCBI/GRCh38/Sequence/STAR_Index
```

Transfer the files from local computer to remote server (In Bash/Terminal)

```
scp /Users/varshaneelakantan/Desktop/TRGN510\ FINAL\ PROJECT/044_sAD_PC_002.fastq.gz
varsha@trgn.bioinform.io:./Finalproject
```

It is easier to have all the files in one folder so running it would be simpler for example:



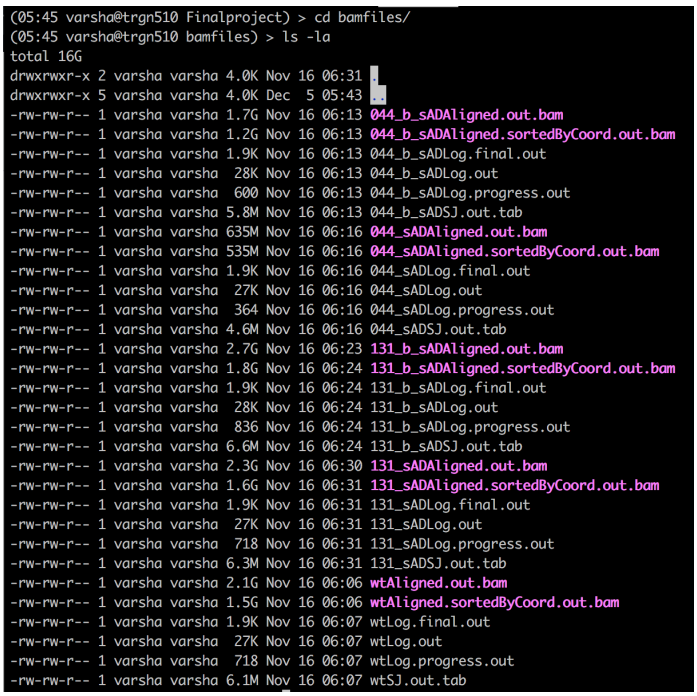
Run STAR (In Bash/Terminal)

Troubleshooting: STAR Requires upto 32GB of RAM. So the alignment was done in the TRGN server.

```
STAR --runThreadN 6 --genomeDir /home/varsha/Finalproject/Homo_sapiens/NCBI/GRCh38/Se
quence/STAR_Index --readFilesIn 131_sAD_PC_001.fastq.gz,131_sAD_PC_002.fastq.gz,131_s
AD_PC_003.fastq.gz,131_sAD_PC_004.fastq.gz --readFilesCommand zcat  --outFileNamePref
ix /home/varsha/Finalproject/bamfiles/131_sAD --outSAMtype BAM Unsorted SortedByCoord
inate
```

Saved all the output to another folder called bamfiles. In the output, the data we are interested in are the ones in pink labelled as XXXAligned.out.bam

These bam files are then transferred back to my local computer using "sftp" command



Install and Load Libraries required for Featurecounts

```
BiocManager::install("Rsubread")
BiocManager::install("DESeq2")
BiocManager::install("Biobase")
BiocManager::install("limma")
BiocManager::install("EnhancedVolcano")
```

```
library(BiocManager)
library(Rsubread)
library(DESeq2)
library(RColorBrewer)
library(gplots)
library(ggplot2)
library(EnhancedVolcano)
library(grid)
library(gridExtra)
library(genefilter)
```

Set working Directory

```
setwd("/Users/varshaneelakantan/Desktop/test")
```

RunFeatureCounts

Note: my gtf file has the Gene names already. In cases where the gtf file only has the gene ID you need to get the gene names from some other resource like NCBI.

The Subread package allows us to analyse next gen sequencing data. The featurecounts function is for counting reads to genomic features

```
featureCounts(files=c("wtAligned.out.bam", "044_b_sADAligned.out.bam", "044_sADAligne
d.out.bam" , "131_sADAligned.out.bam" , "131_b_sADAligned.out.bam"),annot.ext="genes.
gtf", isGTFAnnotationFile=TRUE,GTF.featureType="exon",GTF.attrType="gene_id")
fc <- featureCounts(files=c("wtAligned.out.bam", "044_b_sADAligned.out.bam", "044_sAD
Aligned.out.bam" , "131_sADAligned.out.bam" , "131_b_sADAligned.out.bam"),annot.ext=
"genes.gtf", isGTFAnnotationFile=TRUE,GTF.featureType="exon",GTF.attrType="gene_id")
```

Save data as a txt file

```
write.table(x=data.frame(fc$annotation[,c("GeneID","Length")],fc$counts,stringsAsFact
ors=FALSE),file="readcounts.txt",quote=FALSE,sep="\t",row.names=FALSE)
```

Differential Analysis using DESeq

Reading Data

Make "Coldata" as a table that contains the sample names and the type/condition (example: WT/Control, Sample1/Test etc). You may get an error because of dupplicates in the featurecounts file. In order to go past that you need to look for those duplicates and delete them.

```
dds <- read.delim("featurecounts_output.txt" , sep = "\t" , header=TRUE, row.names =
1)  #Removed Duplicates from the file or else error keeps popping up
dds <- dds[-c(1)] #Ignoring the Length column
data <- as.data.frame(dds)
coldata <- read.delim("coldata.txt", sep = "\t") #Made an excel with the column names
 of the different samples
coldata <- as.data.frame(coldata)
head(coldata)
```

```
##      X Condition
## 1   WT   Control
## 2 sAD1     test1
## 3 sAD2     test1
## 4 sAD3     test2
## 5 sAD4     test2
```

Run DESEQ2

Run the libararies first.

```
dds_output <- DESeqDataSetFromMatrix(countData = data,
                              colData = coldata,
                              design = ~ Condition)

deseq_dds <- DESeq(dds_output)

res_test2 <- results(deseq_dds, contrast = c("Condition", "Control" , "test2"))
res_test1 <- results(deseq_dds, contrast = c("Condition", "Control" , "test1"))

resa <- results(deseq_dds, contrast = c("Condition", "test1" , "Control"))
resb <- results(deseq_dds, contrast = c("Condition", "test2" , "Control"))
res_sad1_sad2 <- results(deseq_dds, contrast = c("Condition", "test2" , "test1"))
```

Size factors or normalization factors for stabilizing the variance in the samples

```
sizeFactors(deseq_dds)
```

```
##        WT      sAD1      sAD2      sAD3      sAD4
## 1.1636215 1.0305322 0.3502768 1.4225433 1.7041582
```

Reordering based on Lowest p value

Showing and example of how the data looks like as a data frame

```
resOrdered_1 <- res_test1[order(res_test1$pvalue),]
resOrdered_2 <- res_test2[order(res_test2$pvalue),]
resOrdered_sAD1vsctrl <- resa[order(resa$pvalue),]
resOrdered_sAD2vsctrl <- resb[order(resb$pvalue),]
resOrdered_sAD2vssAD1 <- res_sad1_sad2[order(res_sad1_sad2 $pvalue),]
head(resOrdered_sAD2vsctrl)
```

```
## log2 fold change (MLE): Condition test2 vs Control
## Wald test p-value: Condition test2 vs Control
## DataFrame with 6 rows and 6 columns
##                 baseMean    log2FoldChange             lfcSE
##                <numeric>         <numeric>         <numeric>
## APOE    2768.01794915885 -6.80076247248609 0.186422171547044
## COL11A1 3557.97961520115 -6.28152590065817  0.18516383465743
## IGFBP1  2917.73668218714  5.85824992910463 0.211900048724944
## FBLN2    4148.1372313505 -4.15399864838396 0.168099986897092
## FRZB    739.167133623431 -6.11591288236176 0.248191304163045
## CX3CL1  575.737859500636 -6.68475510162572 0.285019654411804
##                     stat                pvalue                  padj
##                <numeric>             <numeric>             <numeric>
## APOE    -36.4804380082543 2.26543295529474e-291 3.47426798024001e-287
## COL11A1 -33.9241510756113 2.93481598524462e-252 2.25041689748557e-248
## IGFBP1   27.6462887307256 3.09277386184194e-168  1.5810259981736e-164
## FBLN2   -24.7114751467944 8.05099422312859e-135  3.0867511851475e-131
## FRZB    -24.6419305583085 4.49133817174888e-134 1.37758324403882e-130
## CX3CL1  -23.4536636268859  1.2128824016795e-121  3.1001274186928e-118
```

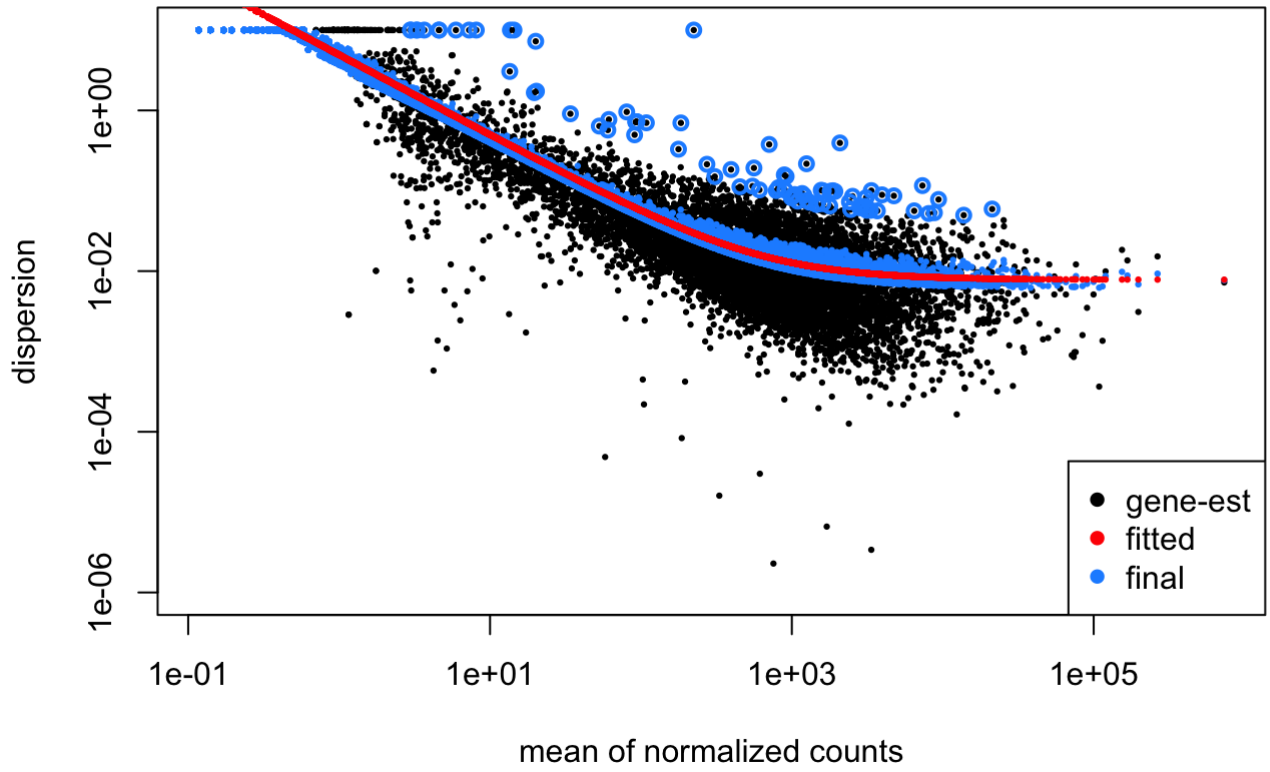
Exporting data into csv files

```
write.csv(as.data.frame(resOrdered_1), file="dds_ctrlvssAD1.csv")
write.csv(as.data.frame(resOrdered_2), file="dds_ctrlvssAD2.csv")
write.csv(as.data.frame(resOrdered_sAD1vsctrl), file="dds_sAD1vsctrl.csv")
write.csv(as.data.frame(resOrdered_sAD2vsctrl), file="dds_sAD2vsctrl.csv")
write.csv(as.data.frame(resOrdered_sAD2vssAD1), file="dds_sAD2vssAD1.csv")
```

Plot for how the data is dispersed

This shows how deseq runs the program to give you a final data point

```
plotDispEsts(deseq_dds, ylim = c(1e-6, 1e1) )
```



To make it rlog transformed data

Showing and example of how the data looks like as a data frame

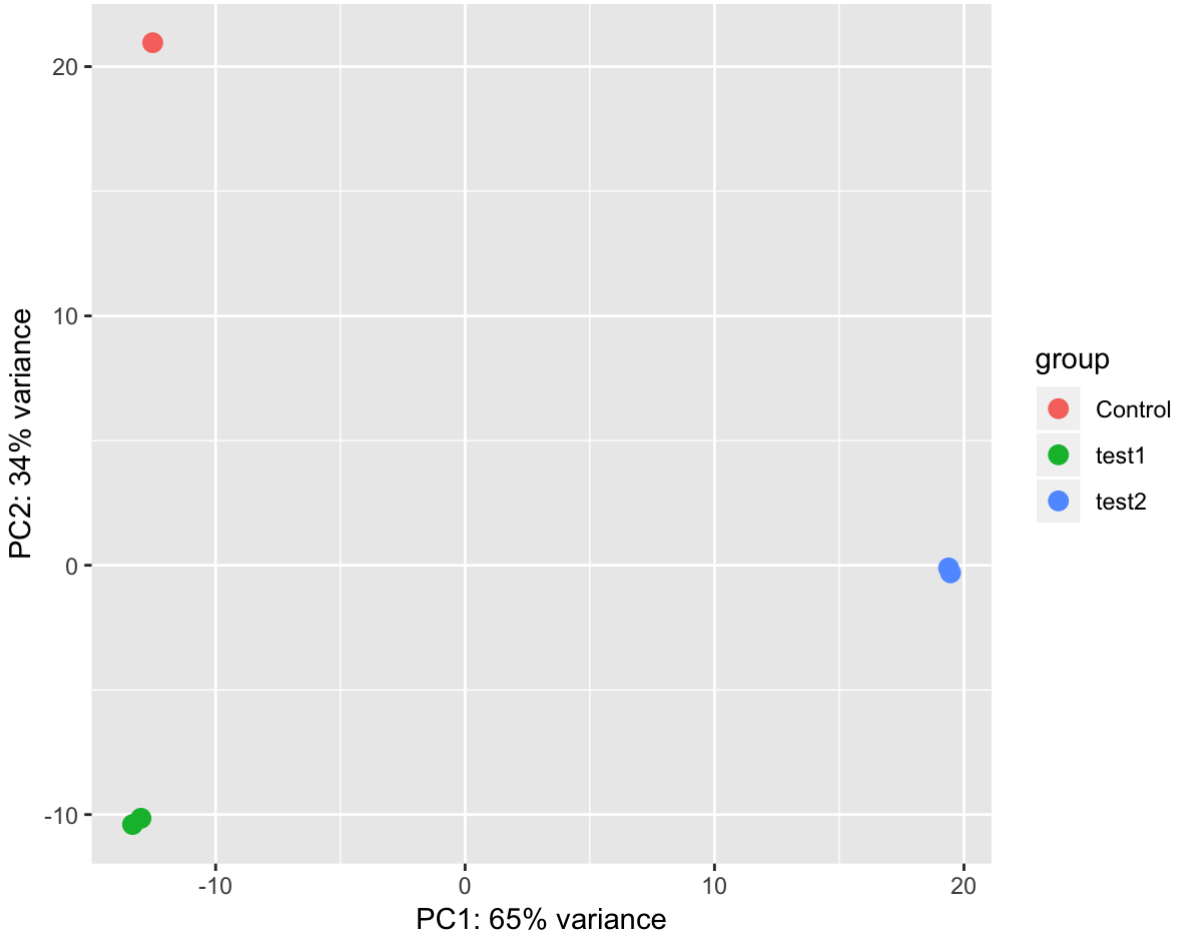
```
rld <- rlog(deseq_dds)
head( assay(rld) )
```

```
##                  WT      sAD1      sAD2      sAD3      sAD4
## DDX11L1    4.238392  3.901767  4.090172  4.085234  4.124920
## WASH7P     9.795133  9.270708  9.526410  9.649355  9.447682
## MIR6859-1  3.551767  3.353717  3.440006  3.368766  3.422490
## MIR1302-2 -1.874578 -1.860017 -1.866338 -1.875957 -1.877196
## FAM138A   -1.861909 -1.873430 -1.866812 -1.875407 -1.876515
## OR4F5      0.000000  0.000000  0.000000  0.000000  0.000000
```

PCA - To know how different are the patient pericytes in comparison to wild type or control

The PCA shows that there is very little variance in the replicates but there is not too much variance between the replicates of the patient RNA but they are different from each other and the wild type.

```
plotPCA(rld, intgroup = "Condition")
```

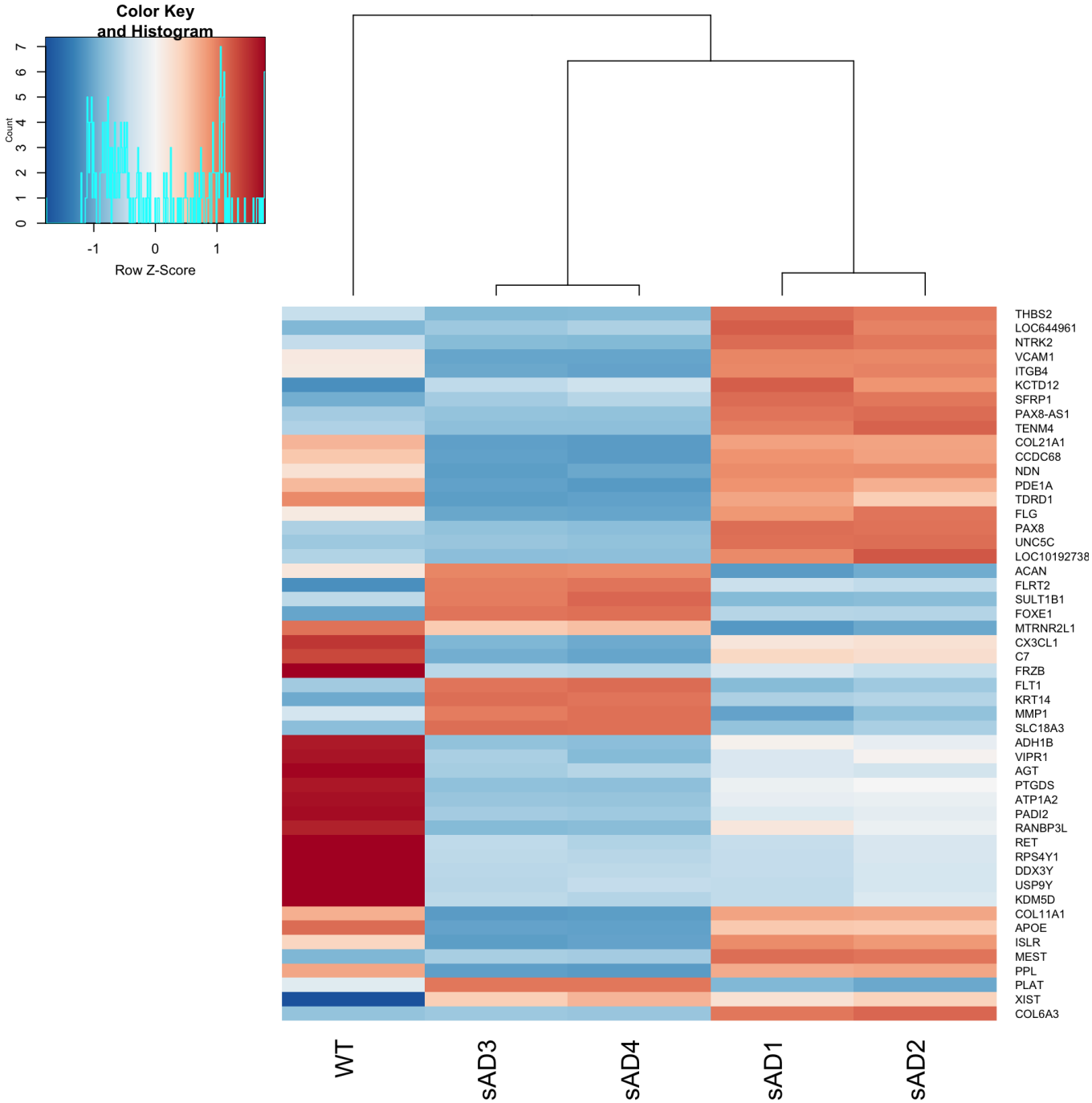


Generating Plots

Heatmap for top 50 variable genes

Data used is rlog transformed

```
topVarGenes <- head( order( rowVars( assay(rld) ), decreasing=TRUE ), 50 )
heatmap.2( assay(rld)[ topVarGenes, ], scale="row", sepwidth=c(0.5,0.5),
           trace="none", dendrogram="column", cexRow=0.75,
           col = colorRampPalette( rev(brewer.pal(9,"RdBu")) )(255))
```



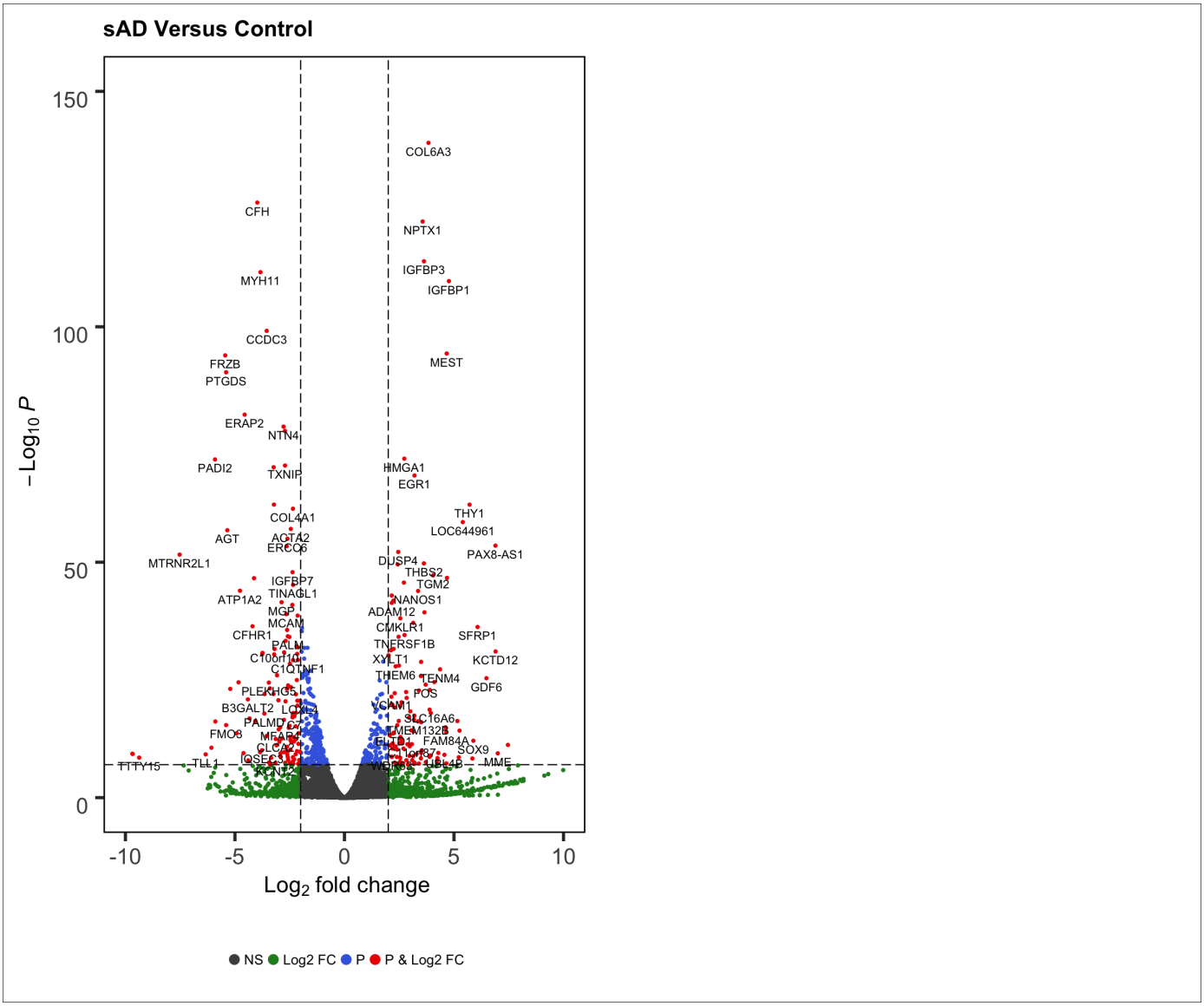
Volcano plot

(Using Enhanced volcano plots package by Kevin Blighe)

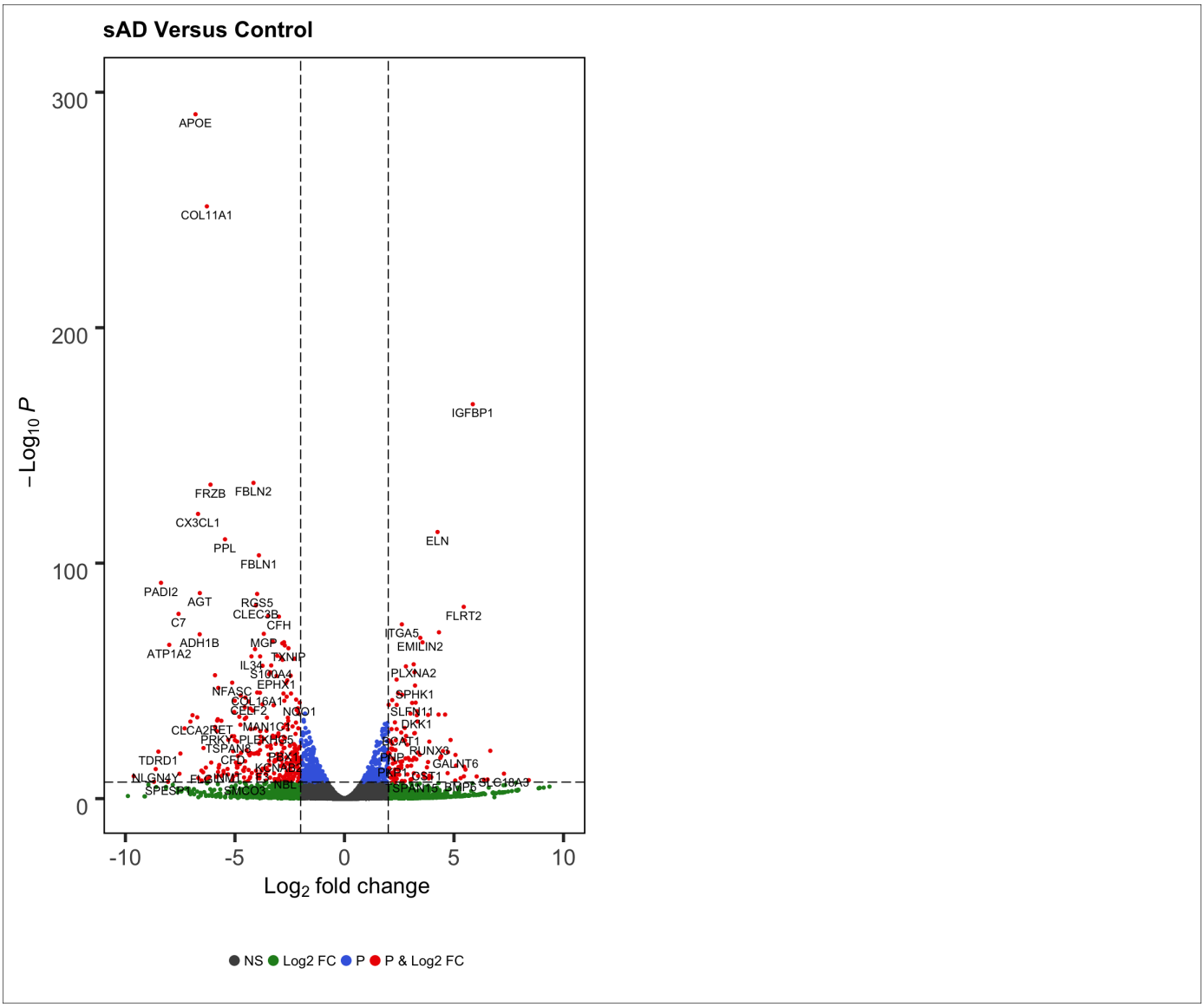
NOTE: sADa refers to sAD 1 and 2 in the heatmap data. sADb refers to sAD 3 and 4

Data used is the original results from deseq analysis

```
p1 <- EnhancedVolcano(resa,
                      lab = rownames(resa),
                      x = "log2FoldChange",
                      y = "pvalue",
                      pCutoff = 10e-8,
                      FCcutoff = 2.0,
                      xlim = c(-10,10),
                      ylim = c(0, 150),
                      transcriptLabSize = 3.0,
                      title = "sAD Versus Control",
                      colAlpha = 1,
                      legendPosition = "bottom",
                      legendLabSize = 10,
                      legendIconSize = 3.0,
                      border = "full",
                      borderWidth = 1,
                      borderColour = "black",
                      gridlines.major = FALSE,
                      gridlines.minor = FALSE)
grid.arrange(p1, ncol=2)
grid.rect(gp=gpar(fill=NA))
```

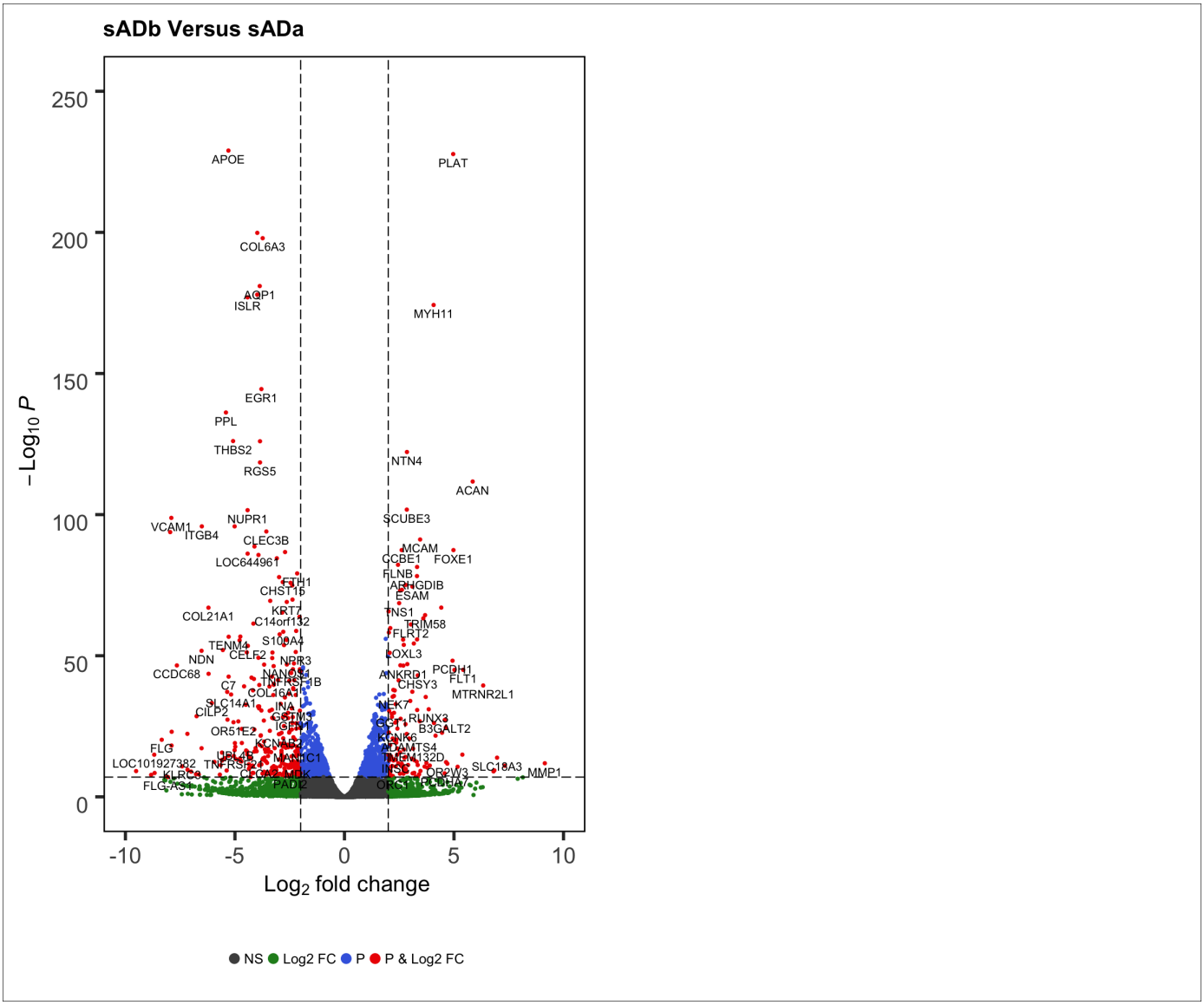


```
p2 <- EnhancedVolcano(resb,
                      lab = rownames(resb),
                      x = "log2FoldChange",
                      y = "pvalue",
                      xlab = bquote(~Log[2]~ "fold change"),
                      ylab = bquote(~-Log[10]~italic(P)),
                      pCutoff = 10e-8,
                      FCcutoff = 2.0,
                      xlim = c(-10,10),
                      ylim = c(0, 300),
                      transcriptLabSize = 3.0,
                      title = "sAD Versus Control",
                      colAlpha = 1,
                      legend=c("NS","Log2 FC","P","P & Log2 FC"),
                      legendPosition = "bottom",
                      legendLabSize = 10,
                      legendIconSize = 3.0,
                      border = "full",
                      borderWidth = 1,
                      borderColour = "black",
                      gridlines.major = FALSE,
                      gridlines.minor = FALSE)
grid.arrange(p2, ncol=2)
grid.rect(gp=gpar(fill=NA))
```



```
p3 <- EnhancedVolcano(res_sad1_sad2,
                      lab = rownames(res_sad1_sad2),
                      x = "log2FoldChange",
                      y = "pvalue",
                      xlab = bquote(~Log[2]~ "fold change"),
                      ylab = bquote(~-Log[10]~italic(P)),
                      pCutoff = 10e-8,
                      FCcutoff = 2.0,
                      xlim = c(-10,10),
                      ylim = c(0, 250),
                      transcriptLabSize = 3.0,
                      title = "sADb Versus sADa",
                      colAlpha = 1,
                      legend=c("NS","Log2 FC","P","P & Log2 FC"),
                      legendPosition = "bottom",
                      legendLabSize = 10,
                      legendIconSize = 3.0,
                      border = "full",
                      borderWidth = 1,
                      borderColour = "black",
                      gridlines.major = FALSE,
                      gridlines.minor = FALSE)

grid.arrange( p3, ncol=2)
grid.rect(gp=gpar(fill=NA))
```



Results:

Amongst the highly varying genes, we see many genes that have an assiciation with AD. APOE- ApolipoproteinE which is known to be associated with Alzheimer's Disease

FBLN2- Is a protein that is involved in calcium ion binding and ECM binding. Pericytes are contractile cells that contract in response to calcium and potassium levels and defect in this gene could affect the pericyte function.

THBS2 - Thrombospondin 2, is a glycoprotein that is involved in cell-cell adhesion. Pericytes are usually wrapped around endothelial cells in capillaries and need to establish a strong cell contact for maintaining the blood-brain barrier.

PDE1A- is a phosphodiesterase enzyme, and is known to be involved in learning and is found in brain regions that subserve memory and learning, including frontal cortex, hippocampus.

NTRK2 - Neurotrophic tyrosine kinase receptor type 2: The NTRK family encodes the receptors TRKA, TRKB, and TRKC, to which the neurotrophins, nerve growth factor (NGF), BDNF and neurotrophin-3 (NT-3) (regulates neuronal development and plasticity, long-term potentiation, and apoptosis) bind with high affinity.

PADI2-Peptidylarginine deiminases, has been shown to be associated with amyloid beta processing.

Conclusion:

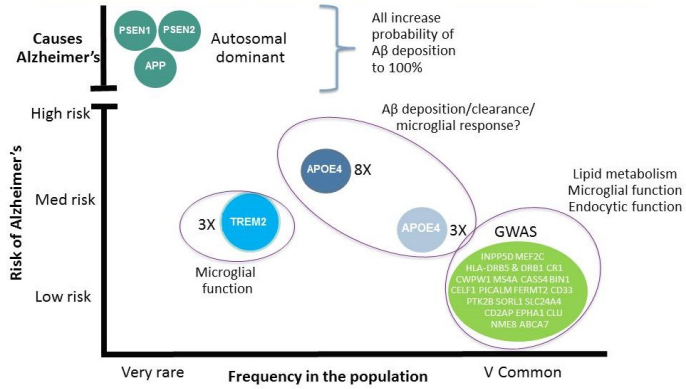


Figure2: GWAS analysis of AD patients for potential high risk genes.

From our data, we see that the genetic profiles of the two AD patients are quite different from each other and APOE (which is one of the prominent risk factors of AD) is clearly downregulated in one of the patients but the physiological effects are similar - Blood-brain barrier leakiness and amyloid beta accumulation. So this could mean that more than one gene is responsible for the disease.

References:

1. "Exome Sequencing of Extended Families with Alzheimer's Disease Identifies Novel Genes Implicated in Cell Immunity and Neuronal Function". Cukier HN (2017)
2. "Preclinical profile of ITI-214, an inhibitor of phosphodiesterase 1, for enhancement of memory performance in rats" Gretchen L. Snyder (2016)
3. "Genetic association of neurotrophic tyrosine kinase receptor type 2 (NTRK2) With Alzheimer's disease" Chen Z(2008)
4. "Increased expression of PAD2 after repeated intracerebroventricular infusions of soluble Abeta(25-35) in the Alzheimer's disease model rat brain: effect of memantine" Arif.M (2009)