**DATA PREPROCESSING STEPS**

**#WGCNA FOR multiple datasets**

setwd("D:/Machine learning/PD/Parkinsons\_disease")## you have to set your own directory

getwd()

install.packages("BiocManager")

BiocManager::install("EnrichmentBrowser")

BiocManager::install("hgu133a.db")

BiocManager::install("hgu133b.db")

BiocManager::install("hgu133plus2.db")

BiocManager::install("oligo")

BiocManager::install("EnrichmentBrowser")

BiocManager::install("limma")

BiocManager::install("sva")

**##calling library for function##**

library(BiocManager)

library(affy)

library(oligo) #if both affy and oligo packaages are used specify the function with package name #

library(Biobase)

library(preprocessCore)

library(limma)

library(sva)

library("annotate")

library(ggfortify)

library(tibble)

library(plyr)

library(ggplot2)

library(lumi)

#DATASET 1

#preprocessing dataset GSE6613#annotation#Affymetrix Human Genome U133A Array#

celpath = "D:/Machine learning/PD/Parkinsons\_disease/PD\_Datasets/GSE6613\_RAW"

list = list.files(celpath,full.names=TRUE)

GSE6613 = read.celfiles(list)

Norm1<-rma(GSE6613)

par(mar=c(5,5,5,5))

boxplot(GSE6613, las=2)

boxplot(Norm1, las=2)

PROBE1<-EnrichmentBrowser::probe2gene(Norm1, chip = "hgu133plus2", from = "PROBEID",to = "SYMBOL",multi.to = "first",multi.from = "mean")

Expr1<-assay(PROBE1, "exprs")

STR1<-strsplit(colnames(Expr1), split = ".CEL.gz")

colnames(Expr1)<-STR1

ColS1<-rownames\_to\_column(as.data.frame(Expr1), var = "SYMBOL")

REP1<-avereps(ColS1, ColS1$SYMBOL)

df1<-as.data.frame(REP1)

write.csv(df1, file="df1\_GSE6613.csv")

#DATASET 2

#preprocessing dataset GSE6613#annotation#Affymetrix Human Genome U133A Array#

celpath2 = "D:/Machine learning/PD/Parkinsons\_disease/PD\_Datasets/GSE72267\_RAW"

list2 = list.files(celpath2,full.names=TRUE)

GSE72267 = read.celfiles(list2)

Norm2<-rma(GSE72267)

par(mar=c(5,5,5,5))

boxplot(GSE72267, las=2)

boxplot(Norm2, las=2)

PROBE2<-EnrichmentBrowser::probe2gene(Norm2, chip = "hgu133plus2", from = "PROBEID",to = "SYMBOL",multi.to = "first",multi.from = "mean")

Expr2<-assay(PROBE2, "exprs")

STR2<-strsplit(colnames(Expr2), split = ".CEL.gz")

colnames(Expr2)<-STR2

ColS2<-rownames\_to\_column(as.data.frame(Expr2), var = "SYMBOL")

REP2<-avereps(ColS2, ColS2$SYMBOL)

df2<-as.data.frame(REP2)

write.csv(df2, file = "df2\_GSE72267.csv")

#DATASET 3

#preprocessing dataset GSE99039#annotation#Affymetrix Human Genome U133Aplus2 Array#

celpath3 = "D:/Machine learning/PD/Parkinsons\_disease/PD\_Datasets/GSE99039\_RAW"

list3 = list.files(celpath3,full.names=TRUE)

GSE99039 = read.celfiles(list3)

Norm3<-rma(GSE99039)

par(mar=c(10,5,5,5))

boxplot(GSE99039, las=2)

boxplot(Norm3, las=2)

PROBE3<-EnrichmentBrowser::probe2gene(Norm3, chip = "hgu133plus2", from = "PROBEID",to = "SYMBOL",multi.to = "first",multi.from = "mean")

Expr3<-assay(PROBE3, "exprs")

STR3<-strsplit(colnames(Expr3), split = "\_HG-U133\_Plus\_2\_.CEL.gz")

#T.Expr3<-t(Expr3)

#rownames(T.Expr3)<- gsub("\_.\*", "", rownames(T.Expr3))

colnames(Expr3)<-STR3

COLS3<-rownames\_to\_column(as.data.frame(Expr3), var ="SYMBOL")

REP3<-avereps(COLS3, COLS3$SYMBOL)

df3<-as.data.frame(REP3)

write.csv(df3, file = "df3\_GSE99039.csv")

#illumina dataset preproccessing

#DATASET 4

#Seperately processed for GSE57475 as its illumina and file saved as GSE57475\_final, script name GSE57475\_illumn\_script.R

getwd()

mywd57475=setwd("D:/Machine learning/PD/Parkinsons\_disease/PD\_Datasets/GSE57475\_RAW")

getwd()

GSE57475<-lumiR("GSE57475\_non-normalized.txt", sep = NULL, na.rm = TRUE, convertNuID = TRUE, lib.mapping = 'lumiHumanIDMapping', dec = '.', parseColumnName = TRUE, checkDupId = TRUE, QC = TRUE,

columnNameGrepPattern = list(exprs='SAMPLE', se.exprs='NA', detection="Detection", beadNum='NA'))

# filter probes based on detection p value, such that the applied condition is satisfied in atleast

# >= 3

fil\_probes\_57475=rowSums(GSE57475@assayData$detection<0.05)>=3 #changing probe to gene

View(fil\_probes\_57475)#The number of 'TRUE' character in the object represents the actual number

# of probes satisfied the condition.

GSE57475=GSE57475[fil\_probes\_57475,]# Add these filtered probes to lumiBatchobject

dim(GSE57475) #16236 142

# Pheno data\_View

# the series matrix from the GEO had more sample information compared to the GSEphenodata from the non normalised file

# The information from the series matrix file was merged with the phenodata from GSE38481

GSE57475@phenoData

pData(GSE57475)

# Read series matrix file

Pheno\_57475=read.delim("GSE57475\_series\_matrix\_phenodata.txt", colClasses="factor", header = T)

Pheno\_57475[]=lapply(Pheno\_57475, as.character)

phenoData(GSE57475) <- AnnotatedDataFrame(Pheno\_57475) #Add phenotypic info (source and Gender) to the pDATA

phenoData(GSE57475)

pData(GSE57475)

colnames(GSE57475)=GSE57475@phenoData@data$GSM\_ID # Change the sample ID in GSE38481 S4 class object to GSM IDs from phenodata

View(exprs(GSE57475))

dim(GSE57475)

# Quantile normalization

colnames(phenoData(GSE57475))

colnames(GSE57475)

norm57475=lumiN(GSE57475, method = "quantile") # watch for warning message for NANs

enorm57475<-exprs(norm57475)

# Check for negative values and assign 0

has.neg57475=exprs(norm57475)

has.neg57475 <- apply(has.neg57475, 1, function(row) any(row < 0))

length(which(has.neg57475))#1407 rows with negative values

df57475\_exp=exprs(norm57475)

df57475\_exp[df57475\_exp<0]=0 # Assign 0 to negative values

df57475\_exp[df57475\_exp<0] # Confirm if there are any negative values

# Gene symbols/EntrezIDs (nuID2EntrezID=for Entrez IDs and nuID2targetID for Gene Symbols)

library(magrittr) #works well w/o this package

library(tibble)

df57475\_exp=as.data.frame(df57475\_exp)

df57475\_exp=rownames\_to\_column(df57475\_exp)

entrezid\_df57475=nuID2targetID(df57475\_exp$rowname, lib.mapping='lumiHumanIDMapping')

entrezid\_df57475=rownames\_to\_column(data.frame(entrezid\_df57475), var = "rowname")

df57475\_join=plyr::join(entrezid\_df57475,df57475\_exp , by= "rowname", type="inner")

View(df57475\_join)

df57475\_GID=df57475\_join[,c(-1)]

View((df57475\_GID))

dim(df57475\_GID) #16023 143

n\_occur57475 <- data.frame(table(df57475\_GID$entrezid\_df57475))## Check for duplicates (optional)

# gives you a data frame with a list of ids and the number of times they occurred.

n\_occur57475[n\_occur57475$Freq > 1,] #tells you which ids occurred more than once.

df57475\_GID[df57475\_GID$entrezid\_df57475 %in% n\_occur57475$Var1[n\_occur57475$Freq > 1],]# returns the records with more than one occurrence.

## Averaging out the intensities for duplicates using avereps function

df57475\_final=limma::avereps(df57475\_GID, ID= df57475\_GID$entrezid\_df57475) %>% data.frame

dim(df57475\_GID) #16236 143

dim(df57475\_final) # reduced to 14442

# check and remove unannotated genes

table(is.na(df57475\_final))# Assuming that unannotated genes would have NA is.na was used

# No NAs were detected.

# GeneIDs/entrezids as Rownames

df57475\_final=column\_to\_rownames(df57475\_final, "entrezid\_df57475")

dim(df57475\_final) #14442 142

df57475\_final[df57475\_final<0]

df4<-rownames\_to\_column(df57475\_final)

colnames(df4)[1] <- "SYMBOL"

# df57475\_final is a df with log2 transformed values and can be used for meta-analysis

write.csv(df4, "df4\_GSE57475\_ill.csv")

##################################################################################

## before we use the combined dataset we need to do batch correction to remove batch effect

#following codes can be used for batch correction

#Combining illumina and affymetrix datasets using combat##

#merging 4 datasets in form of dataframe#

library(dplyr)

library(COMBAT)

df\_datasets2<-merge(df1, df2, by="SYMBOL") #GSE6613 & GSE72267

df\_datasets3<-merge(df\_datasets2, df3, by="SYMBOL") #df\_datasets2 & GSE99039

df\_datasets4<-merge(df\_datasets3, df4, by="SYMBOL") #df-datasets3 & df\_57475

#final merged datasets will be df\_datasets4

write.csv(df\_datasets4, file = "PD\_metadata\_sn712.csv")

#after merge check for similar dimentions#

dim(df\_datasets4) #sample 712, features=3886

#Specifying the type of batch samples belong to#

pdata<- read.csv("D:/Machine learning/Nikita\_machine\_learning/PD\_Meta-phenodata.csv")

dim(pdata)

#very imp when you have to convert matrix to numeric#

data\_PD<-column\_to\_rownames(df\_datasets4, var = "SYMBOL")

is.numeric(data\_PD) #gives false

mutdata<-data\_PD %>% mutate\_if(is.character,as.numeric)#this also converts matrix to numeric but also gives gene symbol#

#T1D <- apply(as.matrix(T1Ddata), 2, as.numeric) this converts matrix to numeric but deletes gene symbols#

#Batch correction using GSEID

Batch<- pdata$GSE\_ID

modcombat = model.matrix(~1, data=pdata)

#Batch correction with matrix with telling groups depending on GSEID#

batch\_correct<- ComBat (mutdata,

batch = Batch,

mod = modcombat,

par.prior = TRUE,

prior.plots = FALSE)

color=pdata$GSE\_ID

color

autoplot(prcomp(t(mutdata)), scale=T, data = pdata, colour="GSE\_ID")+ ggtitle("PCA plot for raw data")

autoplot(prcomp(t(batch\_correct)), scale=T, data = pdata, colour="GSE\_ID")+ ggtitle("PCA plot for Batch Corrected data")

write.csv (batch\_correct, file = "PD\_BatchCorrected\_metadata.csv")