This is the R code for performing Differential expression analysis with Limma.Code based on GEO2r analysis.

‘’’R

#Setting working directory

# Version info: R 4.2.2, Biobase 2.58.0, GEOquery 2.66.0, limma 3.54.0

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

# Loading libraries

library(GEOquery)

library(limma)

library(umap)

#Data aquisition

# Download data and subset

gset <- getGEO("GSE5058", GSEMatrix =TRUE, AnnotGPL=TRUE)

if (length(gset) > 1) idx <- grep("GPL570", attr(gset, "names")) else idx <- 1

gset <- gset[[idx]]

# make proper column names to match toptable

fvarLabels(gset) <- make.names(fvarLabels(gset))

# group membership for all samples

gsms <- "000000000000XXXXXXXXXXXXXXXXXX111111111"

sml <- strsplit(gsms, split="")[[1]

#Preprocessing

‘’’R

# filter out excluded samples (marked as "X")

sel <- which(sml != "X")

sml <- sml[sel]

gset <- gset[ ,sel]

# log2 transformation

ex <- exprs(gset)

qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))

LogC <- (qx[5] > 100) ||

(qx[6]-qx[1] > 50 && qx[2] > 0)

if (LogC) { ex[which(ex <= 0)] <- NaN

exprs(gset) <- log2(ex) }

#Differential expression analysis

# assign samples to groups and setting up design matrix for differential expression analysis using ‘limma’ package

gs <- factor(sml)

groups <- make.names(c("Normal non-smokers","smokers with E-COPD"))

levels(gs) <- groups

gset$group <- gs

design <- model.matrix(~group + 0, gset)

colnames(design) <- levels(gs)

#Removing missing values

gset <- gset[complete.cases(exprs(gset)), ]

#Fitting linear model

fit <- lmFit(gset, design)

# set up contrasts of interest and recalculate model coefficients

cts <- paste(groups[1], groups[2], sep="-")

cont.matrix <- makeContrasts(contrasts=cts, levels=design)

fit2 <- contrasts.fit(fit, cont.matrix)

# compute statistics and table of top significant genes

fit2 <- eBayes(fit2, 0.01)

,,,

This line performs empirical Bayes moderation on the fitted linear model object ‘fit2’ to estimate the coefficients and calculate moderated t-statistics.

#Extracting significant genes

tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)

,,,

This line extracts the top significant genes based on moderated t-statistics from the fitted linear model object fit2.

#subset top significant genes

tT<subset(tT,select=c("ID","adj.P.Val","P.Value","t","B","logFC","GB\_ACC","SPOT\_ID","Gene.Symbol","Gene.symbol","Gene.title"))

write.table(tT, file=stdout(), row.names=F, sep="\t")

This code segment is used to subset the table of top significant genes (tT) to include only specific columns of interest and then write this subset table to a tab-delimited text file.

# Visualize and quality control test results.

# Build histogram of P-values for all genes. Normal test

tT2 <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)

hist(tT2$adj.P.Val, col = "grey", border = "white", xlab = "P-adj",

ylab = "Number of genes", main = "P-adj value distribution")

# Venn diagram of results

vennDiagram(dT, circle.col=palette())

# volcano plot (log P-value vs log fold change)

colnames(fit2) # list contrast names

ct <- 1 # choose contrast of interest

# The following will produce basic volcano plot using limma function:

volcanoplot(fit2, coef=ct, main=colnames(fit2)[ct], pch=20,

highlight=length(which(dT[,ct]!=0)), names=rep('+', nrow(fit2)))

# MD plot (log fold change vs mean log expression)

# highlight statistically significant (p-adj < 0.05) probes

plotMD(fit2, column=ct, status=dT[,ct], legend=F, pch=20, cex=1)

abline(h=0)

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

# General expression data analysis

ex <- exprs(gset)

# box-and-whisker plot

ord <- order(gs) # order samples by group

palette(c("#1B9E77", "#7570B3"))

par(mar=c(7,4,2,1))

title <- paste ("GSE5058", "/", annotation(gset), sep ="")

boxplot(ex[,ord], boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs[ord])

legend("topleft", groups, fill=palette(), bty="n")

# expression value distribution

par(mar=c(4,4,2,1))

title <- paste ("GSE5058", "/", annotation(gset), " value distribution", sep ="")

plotDensities(ex, group=gs, main=title, legend ="topright")

# UMAP plot (dimensionality reduction)

ex <- na.omit(ex) # eliminate rows with NAs

ex <- ex[!duplicated(ex), ] # remove duplicates

ump <- umap(t(ex), n\_neighbors = 9, random\_state = 123)

par(mar=c(3,3,2,6), xpd=TRUE)

plot(ump$layout, main="UMAP plot, nbrs=9", xlab="", ylab="", col=gs, pch=20, cex=1.5)

legend("topright", inset=c(-0.15,0), legend=levels(gs), pch=20,

col=1:nlevels(gs), title="Group", pt.cex=1.5)

library("maptools") # point labels without overlaps

pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)

# mean-variance trend( helps to see if precision weights are needed)

plotSA(fit2, main="Mean variance trend, GSE5058")

# Create the heatmap

install.packages("gplots")

library(gplots)

# Calculate the correlation matrix

correlation\_matrix <- cor(exprs(gset))

# Filter out the top 250 significant DEGs

significant\_genes <- subset(tT, adj.P.Val < 0.05 & abs(logFC) > 1)

top\_250\_genes <- head(significant\_genes, 250)

# Extract the expression data for the top 250 genes

top\_250\_exprs <- exprs(gset)[rownames(top\_250\_genes), ]

# Create heatmap for the expression data of top 250 DEGs

heatmap(top\_250\_exprs, Colv = NA, Rowv = NA, scale = "row", main = "Heatmap of Top 250 DEGs")

# Calculate correlation matrix for the top 250 DEGs

correlation\_matrix <- cor(top\_250\_exprs)

# Create correlation heatmap for the top 250 DEGs

heatmap(correlation\_matrix, Colv = NA, Rowv = NA, scale = "none", main = "Correlation Heatmap of Top 250 DEGs")