```
# Running script.r

# Mosleth et al. “Transcriptome of Subcutaneous Adipose Tissue reflect Membrane Dysfunction and Impaired Oxidative Processes in T2D across Heterogenous Populations”


library(readxl);library(pls);library(gemR)

rm(list = ls())


# Run functions

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

propVar <- function(object){

 vars <- object$sdev^2

 vars/sum(vars)

}


check.error <- function(pls.mod,ncomp,y){

 # Argument: pls model

 # output:   gives the proportion of errornes classification

 error <- vector()


 n <- length(y)

 for(i in 1:ncomp){

  y.predicted <- pls.mod$fitted.values[,1,i]

  y.pred.sign <- sign(y.predicted)

  error[i]   <- sum(sign(y)!=sign(y.predicted))/n

  y.yhat <- cbind(y, y.pred.sign)

 }

 out <- list(error,y.yhat)

 return(out)

}
```

```
Gene2AssGN <- function(Genes.common,Probes.Cohort.12,Probes.Cohort.3){
 # Arguments:
 # Genes.common: the selected genes identified across the cohorts, as gene symbol
 # Probes.Cohort.12 <- association identity of the selected probes for cohort 1 and 2
 # Probes.Cohort.3  <- association identity of the selected probes for cohort 3

 # Outputs:
 # Gene symbols, and gene association identity of each cohort

 DF   <- data.frame(G = NA, N.x = NA, E.x = NA, P.x = NA, N.y = NA, E.y = NA, P.y = NA)
 GN.12 <- gconvert(colnames(Features.12))[,1:6]
 GN.3  <- gconvert(colnames(Features.3))[,1:6]

 k <- length(Genes.common);k
 for (i in 1:k){
  gene.name <-  Genes.common[i]

  # Cohort 1 and 2
  GN     <- GN.12
  N.sel   <- Probes.Cohort.12
  id       <- which(GN$name==gene.name);id
  N       <- GN$input[id];N
  G       <- GN$name[id];G
  E       <- GN$target[id];E
  x        <- GN$description[id];x
  P       <- sub('\\[.*', '', x);P
  NGP    <- data.frame(N,E,G,P)
  id       <- which(N %in% N.sel);id
  NGP.12  <- NGP[id,]
```

```
  # Cohort 3
  GN     <- GN.3
  N.sel  <- Probes.Cohort.3
  id     <- which(GN$name==gene.name);id
  N      <- GN$input[id];N
  G      <- GN$name[id];G
  E      <- GN$target[id];E
  x      <- GN$description[id];x
  P      <- sub('\\[.*', '', x);P
  NGP    <- data.frame(N,E,G,P)
  id     <- which(N %in% N.sel);id
  NGP.3  <- NGP[id,]

  x  <- NGP.12; colnames(x)
  y  <- NGP.3;  colnames(y)
  xy <- merge(x,y,by.x='G',by.y='G')
  DF <- rbind(DF,xy)              # Append new row
 }
 Out <- DF[-1,]
 return(Out)
}

# Import the data
######################
Cohort1.Features      <- t(read.table(file='./Data/Cohort1.Features.t.txt',sep='\t'))
Cohort2.Features      <- t(read.table(file='./Data/Cohort2.Features.t.txt',sep='\t'))
Cohort3.Features      <- t(read.table(file='./Data/Cohort3.Features.t.txt',sep='\t'))
Cohort1.Design.Clinics <- read.table(file='./Data/Cohort1.Design.Clinics.txt',sep='\t')
Cohort2.Design.Clinics <- read.table(file='./Data/Cohort2.Design.Clinics.txt',sep='\t')
Cohort3.Design        <- read.table(file='./Data/Cohort3.Design.txt',sep='\t')
```

```
# The design:
# A= -1; data before the surgery; A= 1; data afer the surgery
# B= -1; nonD;              B= 1; T2D
# C= -1; cohort 1           C= 1; cohort 2

# Organise the data
######################
# Cohort 1, before the surgery
id      <- which(Cohort1.Design.Clinics$A== -1);id # data before the surgery
Design.1  <- Cohort1.Design.Clinics[id,];dim(Design.1)
Features.1<- Cohort1.Features[id,];dim(Features.1)
my.DF.1   <- data.frame(
          Design  = I(Design.1),
          A     = Design.1$A, # before vs after bariatric surgery
          B     = Design.1$B, # nonD vs T2D
          C     = Design.1$C, # cohort affiliation
          Features = I(Features.1))

# Cohort 2
Design.2   <- Cohort2.Design.Clinics
Features.2  <- Cohort2.Features
my.DF.2   <- data.frame(
          Design  = I(Design.2),
          A     = Design.2$A, # batch 1 vs batch 2
          B     = Design.2$B, # nonD vs T2D
          C     = Design.2$C, # cohort affiliation
          Features    = I(Features.2))
```

```
# Cohort 3
Design.3   <- Cohort3.Design;dim(Design.3)
Features.3  <- Cohort3.Features;dim(Features.3)
my.DF.3  <- data.frame(
          Design  = I(Design.3),
          B      = Design.3$B,
          C      = Design.3$C,
          Features    = I(Features.3))

# Set colors and points
############################
cls.2       <- c('blue','deeppink1')
cls.4       <- c("blue","deeppink1","blue","deeppink1")
cls.incl.lean <- c("gray30", "blue","deeppink1","blue","deeppink1")
points.2     <- c(1,19)

# Remove bath effects in cohort 2
########################################

# Remove batch effect by gemR
gem.2          <- GEM(Features ~ A * B, data = my.DF.2)
gem            <- gem.2
Me.er.values.B.2  <- as.matrix(as.data.frame(unclass(gem$ER.values$B)))

# use the ER values for cohort 2, as batch effects are removed
my.DF.2  <- within(my.DF.2, {Me.er.values.B<-I(Me.er.values.B.2)})
```

```
# Combine cohort 1 and 2
################################
N.sel        <-
intersect(colnames(my.DF.1$Features),colnames(my.DF.2$Me.er.values.B))


# Select common probes and scale both data to means of 0 and sd of 1
dataset1      <- scale(my.DF.1$Features[,N.sel]);      dim(dataset1)
dataset2      <- scale(my.DF.2$Me.er.values.B[,N.sel]); dim(dataset2)
Features.12    <- scale(rbind(dataset1,dataset2));dim(Features.12)
common.variables<-
intersect(colnames(my.DF.1$Design),colnames(my.DF.2$Design));common.variables
Design.12     <-
rbind(my.DF.1$Design[,common.variables],my.DF.2$Design[,common.variables]);dim(D
esign.12)


my.DF.12   <- data.frame(
        Features = I(Features.12),
        Design   = I(Design.12),
        B = Design.12$B,
        C = Design.12$C)


gem.12          <- GEM(Features ~ B * C, data = my.DF.12)
er.values.B.12  <- as.matrix(as.data.frame(unclass(gem.12$ER.values$B)))
er.values.c.12  <- as.matrix(as.data.frame(unclass(gem.12$ER.values$C)))
er.values.BC.12 <- as.matrix(as.data.frame(unclass(gem.12$ER.values$`B:C`)))


# Run PLS across cohort 1 and 2
#####################################
l <- 0.05 # choose significance level in Jackknife
gem       <- gem.12
my.ncomp.B  <- 2; my.ncomp.C  <- 2; my.ncomp.BC <- 2 # components included in the
PLS analyses
```

```
Design     <- my.DF.12$Design

B          <- my.DF.12$B

C          <- my.DF.12$C

plsMod.B   <- pls(gem, 'B',   my.ncomp.B, validation = "CV", segment.type =
"interleaved", length.seg = 2,jackknife=TRUE,df.used = 2)

plsMod.C   <- pls(gem, 'C',   my.ncomp.C, validation = "CV", segment.type =
"interleaved", length.seg = 2,jackknife=TRUE,df.used = 2)

plsMod.BC  <- pls(gem, 'B:C', my.ncomp.BC, validation = "CV",segment.type =
"interleaved", length.seg = 2,jackknife=TRUE,df.used = 2)

coeff.C    <- plsMod.C$coefficients[,,my.ncomp.C]

coeff.BC   <- plsMod.BC$coefficients[,,my.ncomp.BC]

error.PLS.B    <- check.error(pls.mod=plsMod.B,ncomp=my.ncomp.B,y=Design$B)[[1]]

error.PLS.C    <- check.error(pls.mod=plsMod.C,ncomp=my.ncomp.C,y=Design$C)[[1]]

error.PLS.BC   <-
check.error(pls.mod=plsMod.BC,ncomp=my.ncomp.BC,y=Design$B*Design$C)[[1]]

scores.PLS.B   <- plsMod.B$scores; loadings.PLS.B  <- plsMod.B$loadings

scores.PLS.C   <- plsMod.C$scores; loadings.PLS.C  <- plsMod.C$loadings

scores.PLS.BC  <- plsMod.BC$scores;loadings.PLS.BC <- plsMod.BC$loadings

coeff.B        <- plsMod.B$coefficients[,,my.ncomp.B]

id             <- which(coeff.B>0);pos.B <- names(coeff.B)[id]

id             <- which(coeff.B<0);neg.B <- names(coeff.B)[id]

n.s.jkn.B      <- names(which(plsMod.B$jack[,1,my.ncomp.B]<l));  length(n.s.jkn.B)

n.s.jkn.B.neg.12<- intersect(n.s.jkn.B,neg.B)

n.s.jkn.B.pos.12<- intersect(n.s.jkn.B,pos.B)

length(n.s.jkn.B);length(n.s.jkn.B.pos.12);length(n.s.jkn.B.neg.12)


# PLS plots

tiff('Cohort 1 2. GEM-PLS on the combined file of cohort 1 and 2.tiff', width=3*1500,
height=3*1500, compression="lzw", type="cairo", res=500)

par(mfrow=c(2,2))

dataset  <- ' Cohort 1 and 2'

my.cls.s <- cls.2[(my.DF.12$B+3)/2]
```

```
my.pch.s <- points.2[(my.DF.12$C+3)/2]
my.cex.main <- 1.3
my.cex.lab  <- 1.2
my.cex.axis <- 1.2

# B
comps   <- c(1,2)
scores   <- scores.PLS.B
loadings <- loadings.PLS.B
error    <- error.PLS.B
effect   <- 'T2D vs nonD'
my.ylim <- c(min(scores[,comps[2]]),2*max(scores[,comps[2]]))
plot(scores[,comps],col=my.cls.s,pch=my.pch.s,cex.axis=my.cex.axis,cex.lab=my.cex.lab,ylim=my.ylim)
legend("topright",c("nonD Cohort 1", "nonD Cohort 2", "T2D Cohort 2", "nonD Cohort 2"),
     col=rep(cls.2,each=2), pch=points.2,ncol=2)
abline(v=0,h=0,lty=2,col='gray')
title(paste0(letters[1],") PLS scores ",dataset,'\n',effect), adj = 0,line = 1,cex.main=my.cex.main)

# C
scores   <- scores.PLS.C
loadings <- loadings.PLS.C
error    <- error.PLS.C
effect   <- 'Cohort affiliation'
my.xlim <- c(min(scores[,comps[1]]),2*max(scores[,comps[1]]))
my.ylim <- c(min(scores[,comps[2]]),2*max(scores[,comps[2]]))
plot(scores[,comps],col=my.cls.s,pch=my.pch.s,
    cex.axis=my.cex.axis,cex.lab=my.cex.lab,xlim=my.xlim,ylim=my.ylim)
legend("topright",c("nonD Cohort 1", "nonD Cohort 2", "T2D Cohort 2", "nonD Cohort 2"),
     col=rep(cls.2,each=2), pch=points.2,ncol=2)
```

```
abline(v=0,h=0,lty=2,col='gray')

title(paste0(letters[2],") PLS scores ",dataset,'\n',effect), adj = 0,line =
1,cex.main=my.cex.main)


# BC

scores   <- scores.PLS.BC

loadings <- loadings.PLS.BC

error    <- error.PLS.BC

effect   <- 'Interacting effect'

my.ylim <- c(min(scores[,comps[2]]),2*max(scores[,comps[2]]))

plot(scores[,comps],col=my.cls.s,pch=my.pch.s,cex.axis=my.cex.axis,cex.lab=my.cex.l
ab,ylim=my.ylim)

legend("topright",c("nonD Cohort 1", "nonD Cohort 2", "T2D Cohort 2", "nonD Cohort 2"),

     col=rep(cls.2,each=2), pch=points.2,ncol=2)

abline(v=0,h=0,lty=2,col='gray')

title(paste0(letters[3],") PLS scores ",dataset,'\n',effect), adj = 0,line =
1,cex.main=my.cex.main)


dev.off()


## Cohort 3

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


my.DF.3   <- data.frame(

 Design =I(Design.3),

 B=Design.3$B,

 Features=I(Features.3))


# PLS

######

x       <- scale(as.matrix(Features.3));dim(x)

y       <- as.numeric(Design.3$B);length(y)
```

```
my.data  <- data.frame(y = y, x = x)

my.ncomp    <- 3

pls.mod     <- plsr(y~x, data = my.data, ncomp = my.ncomp,jackknife=TRUE, validation = "CV",segment.type = "interleaved", length.seg = 2)

scores.PLS  <- pls.mod$scores;  scores.PLS.3  <- scores.PLS

loadings.PLS  <- pls.mod$loadings; loadings.PLS.3 <- loadings.PLS

PLS.error   <- check.error(pls.mod=pls.mod,ncomp=my.ncomp,y=y)[[1]]

pls.mod     <- plsr(y~x, data = my.data, ncomp = my.ncomp,jackknife=TRUE, validation = "CV",segment.type = "interleaved", length.seg = 2)

y.predicted  <- pls.mod$fitted.values[, , my.ncomp]

y.pred.sign  <- sign(y.predicted)

coeff       <- pls.mod$coefficients[,,my.ncomp]

jkn         <- jack.test(pls.mod, ncomp = my.ncomp, use.mean = TRUE)

p.jkn       <- jkn$pvalues[,1,1]

n.s.jkn     <- names(which(p.jkn<l))

id          <- which(coeff>0);pos <- names(coeff)[id]

id          <- which(coeff<0);neg <- names(coeff)[id]

n.s.jkn     <- n.s.jkn

n.s.jkn.B.neg.3  <- intersect(n.s.jkn,neg)

n.s.jkn.B.pos.3  <- intersect(n.s.jkn,pos)



## Get gene symbols and names of significant probes

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


# Cohort 1 and 2

Draft  <- gconvert(n.s.jkn.B.neg.12)[,1:6]

x    <- Draft$description

desc  <- sub('\\[.*', '', x)

GN.s.jkn.B.neg.12 <- data.frame(Draft[,1:5],desc)
```

```
Draft  <- gconvert(n.s.jkn.B.pos.12)[,1:6]
x     <- Draft$description
desc   <- sub('\\[.*', '', x)
GN.s.jkn.B.pos.12 <- data.frame(Draft[,1:5],desc)

# Cohort 3
Draft  <- gconvert(n.s.jkn.B.neg.3)[,1:6]
x     <- Draft$description
desc   <- sub('\\[.*', '', x)
GN.s.jkn.B.neg.3 <- data.frame(Draft[,1:5],desc)

Draft  <- gconvert(n.s.jkn.B.pos.3)[,1:6]
x     <- Draft$description
desc   <- sub('\\[.*', '', x)
GN.s.jkn.B.pos.3 <- data.frame(Draft[,1:5],desc)

G.s.jkn.B.pos.12 <- unique(GN.s.jkn.B.pos.12$name)
G.s.jkn.B.neg.12 <- unique(GN.s.jkn.B.neg.12$name)
G.s.jkn.B.pos.3  <- unique(GN.s.jkn.B.pos.3$name)
G.s.jkn.B.neg.3  <- unique(GN.s.jkn.B.neg.3$name)

# Common genes significant negatively or positively associated with T2D across the cohorts
temp     <- intersect(G.s.jkn.B.neg.12,G.s.jkn.B.neg.3);
G.neg.123 <- setdiff(temp,'None') # gene symbols, negatively associated with T2D by Jakknife in PLS in cohort 1 and 2 and in cohort 3

temp     <- intersect(G.s.jkn.B.pos.12,G.s.jkn.B.pos.3);
G.pos.123 <- setdiff(temp,'None') # gene symbols, positively associated with T2D by Jakknife in PLS in cohort 1 and 2 and in cohort 3
```

```
# Get the gene probes for each cohort based on the common genes significant across the cohorts

# Note here the gconvert online is run in the code, which gives the updated version, which may change upon time


GN.common.neg <- Gene2AssGN(Genes.common=G.neg.123,Probes.Cohort.12=n.s.jkn.B.neg.12,Probes.Cohort.3=n.s.jkn.B.neg.3)

sort(unique(GN.common.neg$G))


GN.common.pos <- Gene2AssGN(Genes.common=G.pos.123,Probes.Cohort.12=n.s.jkn.B.pos.12,Probes.Cohort.3=n.s.jkn.B.pos.3)

sort(unique(GN.common.pos$G))
```