library(ropls); library (dplyr); library(caret); library(mixOmics)

Design.0 <- read.table("Design.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

Metab.0 <- read.table("Metab.RES.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F, quote="")

anno <- Metab.0

Metab <- Metab.0

Metab <- Metab[, -c(1, 2, 3, 4, 5, 6)]

Metab <- Metab[, -c(1, 2, 3, 4, 5)]

head (Metab); dim(Metab)

design <- Design.0 [-c(1, 2, 3, 4, 5), ]

Pcut = 0.001

Myplsda <- plsda(t(Metab), design$Trt, ncomp = 10)

Mycol <- c("cyan", "chartreuse", "orange")

Mypch <- c(16, 16, 16)

plotIndiv(Myplsda, col = Mycol, pch= Mypch , ellipse = TRUE, ellipse.level = 0.95, legend= T,

legend.position= "right", cex=3, abline = T )

############# WD vs. Cured ############################################################

Design.WC <- filter(design, Trt !="Restored")

Design.WC <- filter(Design.WC, Trt !="EB")

Design.WC$Trt <- as.factor(Design.WC$Trt)

Metab.WC<- Metab [, rownames(Design.WC)]

Metab.data <- list()

Metab.data$dataMatrix <- t(Metab.WC)

Metab.data$sampleMetadata <- Design.WC

sacurine.plsda <- opls(x = Metab.data$dataMatrix, y = Metab.data$sampleMetadata [, 'Trt'] ,

orthoI = NA, crossvalI = 4 , permI=200) #, permI=200

vipVn <- getVipVn(sacurine.plsda)

vipVn\_select <- vipVn[vipVn > 1]

head(vipVn\_select)

vipVn <- as.data.frame(vipVn)

library (readr); library(edgeR); library(phyloseq); library (ggplot2); library(ggpubr); library (ggrepel)

edgeR\_enrich <- DGEList(counts= Metab.WC,

group= Design.WC$Trt)

edgeR\_enrich <- calcNormFactors(edgeR\_enrich)

otu\_norm\_enrich <- **cpm**(edgeR\_enrich, normalized.lib.sizes=T, log=F)

model\_mat\_enrich <- model.matrix(~Trt, data= Design.WC)

dge\_enrich <- estimateGLMRobustDisp(edgeR\_enrich, design=model\_mat\_enrich)

#plotBCV(dge\_enrich)

#plotSmear(dge\_enrich, pair=c("M","I"))

fit\_enrich <- glmFit(dge\_enrich, design=model\_mat\_enrich)

lrt\_enrich <- glmLRT(fit\_enrich, coef=2)

tt\_enrich <- **topTags**(lrt\_enrich, n=Inf, p.value=1)

head(tt\_enrich$table)

WD\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC > 0 & tt\_enrich$table$ PValue < Pcut,] ; dim (WD\_enrich)

Cu\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC < 0 & tt\_enrich$table$ PValue < Pcut,] ; dim (Cu\_enrich)

tt\_enrich <- as.data.frame(tt\_enrich)

tt\_enrich$VIP <- vipVn[rownames(tt\_enrich), ]

tt\_enrich <- tt\_enrich [order(tt\_enrich $VIP, decreasing = TRUE), ]

tt\_enrich.WC.all <- tt\_enrich

tt\_enrich <- tt\_enrich[tt\_enrich$PValue< Pcut, ]

# tt\_enrich <- tt\_enrich[tt\_enrich$logCPM > 50, ]

tt\_enrich <- tt\_enrich[abs(tt\_enrich$logFC) > 0.5, ]

tt\_enrich <- tt\_enrich[tt\_enrich$VIP > 1, ]

tt\_enrich$anno <- anno[rownames(tt\_enrich),]$Compound

dim (tt\_enrich)

tt\_enrich.WC <- tt\_enrich

#tt\_enrich$logFC <- -tt\_enrich$logFC

tt\_enrich <- tt\_enrich[1:30, ]

range (tt\_enrich$logFC)

#tt\_enrich$VIP <- tt\_enrich$VIP - 1

tt\_enrich$logFC <- - (tt\_enrich$logFC)

ggdotchart(tt\_enrich, x ="anno", y = "VIP", color = "logFC", # Color by groups

sorting = "descending", # Sort value in descending order

# ylim=c(1.0, 2.5),

add = "segments", # Add segments from y = 0 to dots

rotate = TRUE, dot.size = 4, ggtheme = theme\_pubr() ) + # ggplot2 theme

scale\_color\_gradient2(name="logFC", low = "dodgerblue2", mid = "white", high = "red3" ) # , limit = c(-3.5, 3.5)

# + scale\_y\_continuous(limits=c(0, 0.75), breaks=seq(0, 2, 0.25) )

############# Restored vs. Cured ####################################################

Design.RC <- filter(design, Trt !="WD")

Design.RC <- filter(Design.RC, Trt !="EB")

Design.RC$Trt <- as.factor(Design.RC$Trt)

Metab.RC<- Metab [, rownames(Design.RC)]

Metab.data <- list()

Metab.data$dataMatrix <- t(Metab.RC)

Metab.data$sampleMetadata <- Design.RC

sacurine.plsda <- opls(x = Metab.data$dataMatrix, y = Metab.data$sampleMetadata [, 'Trt'] ,

orthoI = NA, crossvalI = 4 , permI=200) #, permI=200

vipVn <- getVipVn(sacurine.plsda)

vipVn\_select <- vipVn[vipVn > 1]

head(vipVn\_select)

vipVn <- as.data.frame(vipVn)

edgeR\_enrich <- DGEList(counts= Metab.RC,

group= Design.RC$Trt)

edgeR\_enrich <- calcNormFactors(edgeR\_enrich)

otu\_norm\_enrich <- **cpm**(edgeR\_enrich, normalized.lib.sizes=T, log=F)

model\_mat\_enrich <- model.matrix(~Trt, data= Design.RC) # Site+Type Type

dge\_enrich <- estimateGLMRobustDisp(edgeR\_enrich, design=model\_mat\_enrich)

fit\_enrich <- glmFit(dge\_enrich, design=model\_mat\_enrich)

lrt\_enrich <- glmLRT(fit\_enrich, coef=2)

tt\_enrich <- **topTags**(lrt\_enrich, n=Inf, p.value=1)

head(tt\_enrich$table)

Res\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC > 0 & tt\_enrich$table$ PValue < Pcut,] ; dim (Res\_enrich)

Cu.R\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC < 0 & tt\_enrich$table$ PValue < Pcut,] ; dim (Cu.R\_enrich)

tt\_enrich <- as.data.frame(tt\_enrich)

tt\_enrich$VIP <- vipVn[rownames(tt\_enrich), ]

tt\_enrich <- tt\_enrich [order(tt\_enrich $VIP, decreasing = TRUE), ]

tt\_enrich.RC.all <- tt\_enrich

tt\_enrich <- tt\_enrich[tt\_enrich$PValue< Pcut, ]

tt\_enrich <- tt\_enrich[abs(tt\_enrich$logFC) > 0.5, ]

tt\_enrich <- tt\_enrich[tt\_enrich$VIP > 1, ]

tt\_enrich$anno <- anno[rownames(tt\_enrich),]$Compound

dim (tt\_enrich)

tt\_enrich.RC <- tt\_enrich

tt\_enrich <- tt\_enrich[1:30, ]

range (tt\_enrich$logFC)

tt\_enrich$logFC <- - (tt\_enrich$logFC)

ggdotchart(tt\_enrich, x ="anno", y = "VIP", color = "logFC", # Color by groups

sorting = "descending", # Sort value in descending order

# ylim=c(1.0, 2.5),

add = "segments", # Add segments from y = 0 to dots

rotate = TRUE, dot.size = 4, ggtheme = theme\_pubr() ) + # ggplot2 theme

scale\_color\_gradient2(name="logFC", low = "dodgerblue2", mid = "white", high = "red3" ) # , limit = c(-3.5, 3.5)

# + scale\_y\_continuous(limits=c(0, 0.75), breaks=seq(0, 2, 0.25) )

intersect(rownames(WD\_enrich), rownames(Res\_enrich))

Metab.0 [intersect(rownames(WD\_enrich), rownames(Res\_enrich)), ] $ Compounds

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

pData <- tt\_enrich.WC.all

pData <- pData[Decreased.Metab, ]

Data.RC <- tt\_enrich.RC.all [Decreased.Metab, ]

pData <- pData [, c("logFC", "VIP")]

pData$FC.RC <- Data.RC$logFC

pData$VIP.RC <- Data.RC$VIP

pData$Compound <- rownames(pData)

P.WD = ggplot(pData, aes(x =Compound, y = VIP, color= logFC)) +

geom\_point(size=3.5) +

scale\_color\_gradient2(name="logFC", low = "dodgerblue2", mid = "white", high = "red3", limit = c(0, 5.5) ) +

coord\_flip() +

ylim(1.5, 2.5)

P.RC = ggplot(pData, aes(x =Compound, y = VIP.RC, color= FC.RC)) +

geom\_point(size=3.5) +

scale\_color\_gradient2(name="logFC", low = "dodgerblue2", mid = "white", high = "red3", limit = c(0, 5.5) ) +

coord\_flip() +

ylim(1.5, 2.5)

**ggarrange** (P.WD, P.RC, ncol=2, nrow=1, align="hv" )

####################作图##############

RC = intersect(rownames(tt\_enrich.RC), rownames(Res\_enrich))

WC= intersect(rownames(tt\_enrich.WC), rownames(WD\_enrich))

Decreased.Metab = intersect(RC, WC)

pData <- tt\_enrich.WC.all

pData <- pData[Decreased.Metab, ]

pData <- pData [, c("logFC", "VIP")]

pData$Compound <- rownames(pData)

pData$Trt <- c("WD")

Data.RC <- tt\_enrich.RC.all [Decreased.Metab, ]

Data.RC <- Data.RC [, c("logFC", "VIP")]

Data.RC $Compound <- rownames(Data.RC)

Data.RC $Trt <- c("Restored")

pData <- rbind(pData, Data.RC)

range(pData$ logFC)

range(pData$ VIP)

#pData$ VIP = pData$ VIP -1.8

range(pData$ VIP)

p.R <- ggplot(pData, aes(x=factor(Trt, levels=c('WD', 'Restored')), y= Compound)) +

#geom\_tile(aes(fill = VIP), colour = "white") +

#scale\_fill\_gradient2(name="VIP", low = "dodgerblue2", mid = "white", high = "red3", limit = c(0, 2.5) ) +

geom\_tile(aes(fill = logFC), colour = "white") +

scale\_fill\_gradient2(name="logFC", low = "dodgerblue2", mid = "white", high = "red3", limit = c(0, 5.5) ) +

geom\_point(aes(size= VIP ), colour = "grey15" ) +

scale\_size\_continuous (name="VIP", range = c(0, 5) , limit = c(0, 0.5) , breaks=c(0, 0.1, 0.2, 0.3, 0.4, 0.5) ) +

theme\_bw()+

theme(axis.text=element\_text(size=6, color="black"), axis.title= element\_blank(),

legend.text=element\_text(size=8), legend.background=element\_rect(fill='NA'),

# axis.text.x = element\_text(vjust = 0.5, hjust = 0.5, angle = 90), #axis.text.y = element\_text(size=8),

# axis.text.y=element\_blank(), axis.ticks.y= element\_blank (),

panel.background = element\_blank(), panel.grid.major=element\_line(colour=NA),

panel.grid.minor=element\_line(colour=NA) )

p.R

#########################################3