library("psych")

library("reshape2")

library("igraph")

library("VennDiagram")

library("pheatmap")

otu\_abundance\_file = "Bgenus.txt"

vip\_species = "NULL"

output\_prefix = "BBgenus"

correlation\_threshold = 0.7

pvalue\_threshold = 0.001

cor\_file = paste0(output\_prefix, "\_correlation.txt")

pvalue\_file = paste0(output\_prefix, "\_pvalue.txt")

edge\_file\_txt = paste0(output\_prefix, "\_edge.txt")

node\_file\_txt = paste0(output\_prefix, "\_node.txt")

edge\_file\_csv = paste0(output\_prefix, "\_edge.csv")

node\_file\_csv = paste0(output\_prefix, "\_node.csv")

degree\_file = paste0(output\_prefix, "\_network\_degree.txt")

VColor <- c("#636363","#78c679")

EColor <- c("#d7301f", "#0570b0")

setwd('C:/Users/Administrator/Desktop')

##读入丰度数据

myData <- read.table(otu\_abundance\_file, header = T, row.names=1)

#计算皮尔森相关系数

cor <- cor(t(myData), method = "pearson")

p <- corr.test(t(myData)) #计算p值

pValue <- p$p #提取p值

#输出数据

write.table(cor, cor\_file, quote = F, row.names = T, col.names = T, sep = '\t')

write.table(pValue, pvalue\_file, quote = F, row.names = T, col.names = T, sep = '\t')

##读取边相关系数文件

cor <- read.table(cor\_file, header = T, row.names=1)

##读取边p值文件

pv <- read.table(pvalue\_file, header = T, row.names=1)

cor$genus <- rownames(cor)

pv$genus <- rownames(pv)

##提取上三角矩阵并转换为边文件格式

cor[lower.tri(cor)] <- 0

pv[lower.tri(pv)] <- 0

pear.cor <- melt(cor, id.vars = "genus")

pear.pv <- melt(pv, id.vars = "genus")

pear <- data.frame(source = pear.cor[,1], target = pear.cor[,2],

pearsonCor = pear.cor[,3], pearsonPV = pear.pv[,3])

pear.trim <- pear[-which(pear[,3] == "0"),]

# pear.trim <- pear.trim[-which(pear.trim[,3] == "1"),]

pear.trim <- pear.trim[-which(pear.trim$source == pear.trim$target),]

pear.trim$weight <- abs(as.vector(pear.trim[,3])) #设置边的权重

pear.trim$class <- 0 #设置边的类别

pear.trim[which(pear.trim[,3] < 0),"class"] <- 2 #2 -> 负相关

pear.trim[which(pear.trim[,3] > 0),"class"] <- 1 #1 -> 正相关

pear.result <- pear.trim[which(pear.trim$weight > correlation\_threshold),] #剔除权重小于0.8的边

pear.result <- pear.result[which(pear.trim$pearsonPV < pvalue\_threshold),] ##剔除p值大于0.01的边

pear.result <- pear.result[,c(1,2,5,6)]

##输出边文件

##txt格式

write.table(pear.result, edge\_file\_txt, col.names = T, row.names = F, sep = "\t", quote = F)

##csv格式

write.csv(pear.result, edge\_file\_csv, row.names = F, quote = F)

##读取节点文件

node <- read.table(otu\_abundance\_file, header = T, sep = '\t', row.names = 1)

##计算节点的权重

node.result <- data.frame(id = as.character(rownames(node)), weight = rowMeans(node),

class = rep(1,length(node[,1])))

##指定VIP菌属

# node.result[1:7,3] <- 2 ##指定前7个菌为VIP菌

#也可根据菌属名指定VIP菌

# node.result[which(node.result[,1] == "Streptococcus"),3] <- 2 #指定Streptococcus为VIP菌

#根据节点的权重设定VIP菌

#node.order = node.result[rev(order(node.result[,2])),]

#node.result[rownames(node.order)[1:7],3] <- 2 #指定权重最大的前7个菌为VIP菌

if(! is.null(vip\_species)){

vip\_species\_list <- read.table(vip\_species, row.names=NULL, header=F)$V1

node.result[which(node.result[,1] %in% vip\_species\_list), 3] <- 2

}

#node.result <- node.result[-which(node.result[,2] == "0"),] ##剔除丰度为0的菌属

##输出节点文件

##txt格式

write.table(node.result, node\_file\_txt, col.names = T, row.names = F, sep = "\t", quote = F)

##csv格式

write.csv(node.result, node\_file\_csv, row.names = F, quote = F)