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

design\_filter\_16s<-read.table("design\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

tax\_filter\_16s<- read.table("tax\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

otu\_rarefy\_16s<-read.table("otu\_rarefy\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

otu\_filter\_16s<-read.table("otu\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

otu\_filter\_16s <- as.matrix(otu\_filter\_16s) #rownames (otu\_filter\_16s); #colnames (otu\_filter\_16s)

dim(otu\_filter\_16s)

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

design <- design\_filter\_16s; otu.data <- otu\_filter\_16s

rep=6

design$Trt <- factor(rep(c('control', 'treat'), each = rep), levels = c('control', 'treat'))

edgeR\_enrich <- DGEList(counts= otu.data,

group= design $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) # 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)

**write.table**(tt\_enrich, file="VOC.tt\_enrich.DDS.txt", sep="\t", quote=F)

CH\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC > 1 & tt\_enrich$table$FDR < 0.01,] ; dim (CH\_enrich) #处理富集的

CK\_enrich <- tt\_enrich$table[tt\_enrich$table$logFC < -1 & tt\_enrich$table$FDR < 0.01,] ; dim (CK\_enrich) #对照富集的

**write\_rds** (rownames(CH\_enrich), "CH\_enrich.DDS.rds")

**write\_rds** (rownames(CK\_enrich), "CK\_enrich.DDS.rds")

design\_filter\_16s<-read.table("design\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

tax\_filter\_16s<- read.table("tax\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

otu\_filter\_16s<-read.table("otu\_filter\_16s.DDS.txt", row.names=1, sep="\t", header=T, blank.lines.skip=F, check.names=F)

otu\_filter\_16s <- as.matrix(otu\_filter\_16s) #rownames (otu\_filter\_16s); #colnames (otu\_filter\_16s)

dim(otu\_filter\_16s)

design\_filter\_16s$Trt <- as.factor(design\_filter\_16s$Trt)

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

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

I\_enrich <- tt\_enrich[tt\_enrich$logFC > 1 & tt\_enrich$FDR < 0.01,]; dim (I\_enrich)

M\_enrich <- tt\_enrich[tt\_enrich$logFC < -1 & tt\_enrich$FDR < 0.01,]; dim (M\_enrich)

#table(tax\_filter\_16s [rownames (I\_enrich),]$Class)

tt\_enrich <- as.data.frame(tt\_enrich); forManh <- tt\_enrich[, c("logCPM", "logFC", "FDR") ]

forManh$logCPM <- 2^forManh$logCPM; colnames(forManh) <- c("CPM", "logFC", "signif")

forManh$Label <- tax\_filter\_16s [rownames(forManh),]$labels; sort(table(forManh$Label))

forManh$Class <- tax\_filter\_16s [rownames(forManh),]$Class

forManh$Phylum <- tax\_filter\_16s [rownames(forManh),]$Phylum; sort(table(forManh$Phylum))

Proteobacteria.Class = c("Alphaproteobacteria", "Betaproteobacteria", "Gammaproteobacteria", "Deltaproteobacteria")

forManh [!(forManh$Class %in% Proteobacteria.Class)&forManh$Phylum=="Proteobacteria",]$Label = "Proteobacteria.Other"

Abundant.ClassPhylum=c("Alphaproteobacteria", "Betaproteobacteria", "Gammaproteobacteria", "Deltaproteobacteria", "Actinobacteria", "Acidobacteria", "Chloroflexi", "Gemmatimonadetes", "Bacteroidetes", "Firmicutes")

forManh [!(forManh$Label %in% Abundant.ClassPhylum),]$Label = "ZOthers"

forManh[rownames(forManh)[forManh$Label=="Alphaproteobacteria" ], ]$Label <- "01Alphaproteobacteria" #

forManh[rownames(forManh)[forManh$Label=="Betaproteobacteria" ], ]$Label <- "02Betaproteobacteria"

forManh[rownames(forManh)[forManh$Label=="Gammaproteobacteria" ], ]$Label <- "03Gammaproteobacteria"

forManh[rownames(forManh)[forManh$Label=="Deltaproteobacteria" ], ]$Label <- "04Deltaproteobacteria"

# forManh[rownames(forManh)[forManh$Label=="Proteobacteria.Other" ], ]$Label <- "05Proteobacteria.Other"

forManh[rownames(forManh)[forManh$Label=="Actinobacteria" ], ]$Label <- "05Actinobacteria"

forManh[rownames(forManh)[forManh$Label=="Acidobacteria"],]$Label <- "06Acidobacteria"

forManh[rownames(forManh)[forManh$Label=="Chloroflexi"],]$Label<-"07Chloroflexi"

forManh[rownames(forManh)[forManh$Label=="Gemmatimonadetes" ], ]$Label <- "08Gemmatimonadetes"

forManh[rownames(forManh)[forManh$Label=="Bacteroidetes" ], ]$Label <- "09Bacteroidetes"

forManh[rownames(forManh)[forManh$Label=="Firmicutes" ], ]$Label <- "10Firmicutes"

forManh$OTUnames <- rownames (forManh); forManh <- forManh [order(forManh $ OTUnames), ]

forManh <- forManh[order(forManh$Class), ]; forManh <- forManh [order(forManh$Phylum), ]

forManh <- forManh [order(forManh$Label), ]

forManh$order <- 1:nrow(forManh)

phylum\_num <- as.numeric(table(forManh $Label))

phylum\_name <- names(table(forManh $Label))

a=0; phylum\_numhalf=0

for(i in 1:length(phylum\_num))

{ tmp=0; if(phylum\_num[i]%%2!=0)

{ tmp=phylum\_num[i]/2+0.5}else{tmp=phylum\_num[i]/2 }

phylum\_numhalf[i]=a+phylum\_num[i]/2;a=a+phylum\_num[i] }

phylum\_range <- c(0, phylum\_num[1])

for (i in 2:length(phylum\_num)) { phylum\_range[i+1] <- phylum\_range[i] + phylum\_num[i] }

forManh$pch <- "Non-Sig"

forManh [(rownames(forManh) %in% rownames (I\_enrich)),]$pch = "I\_enrich"

forManh [(rownames(forManh) %in% rownames (M\_enrich)),]$pch = "M\_enrich"

forManh$Genus<- tax\_filter\_16s [rownames(forManh),]$Genus

forManh$label<-NA

Sig <- rownames(forManh)[forManh$pch !="Non-Sig" ]

for(i in Sig){ forManh [rownames(forManh)[ forManh $ OTUnames ==paste(i)], ]$label <- forManh [rownames(forManh)==paste(i),]$Genus}

unassi <- rownames(forManh)[forManh$Genus=="unassigned"&forManh$pch !="Non-Sig" ]

for(i in unassi){ forManh [rownames(forManh)[forManh$OTUnames ==paste(i)], ]$label <- NA }

OTU.RA <- as.data.frame(otu\_filter\_16s)

OTU.RA$RA <- rowSums(otu\_filter\_16s)/sum(otu\_filter\_16s) \*100

forManh$RA <- 0

for (i in rownames(forManh)) {forManh[rownames(forManh)==paste(i),]$RA <- OTU.RA [rownames(OTU.RA)==paste(i),]$RA }

range(forManh$RA)

r <- rownames(tt\_enrich[tt\_enrich$logFC < 0, ])

forManh <- forManh [-which(rownames(forManh) %in% r),]; dim(forManh)

Manhattan.MI <- ggplot(forManh, aes(x=order, y= -log10(signif)) ) +

annotate('rect', xmin = phylum\_range[1], xmax = phylum\_range[5], ymin = -Inf, ymax = Inf, fill = 'gray95' )+

annotate('rect', xmin = phylum\_range[6], xmax = phylum\_range[7], ymin = -Inf, ymax = Inf, fill = 'gray95' )+

annotate('rect', xmin = phylum\_range[8], xmax = phylum\_range[9], ymin = -Inf, ymax = Inf, fill = 'gray95' )+

annotate('rect', xmin = phylum\_range[10], xmax = phylum\_range[11], ymin = -Inf, ymax = Inf, fill = 'gray95' )+

annotate('rect', xmin = phylum\_range[12], xmax = phylum\_range[13], ymin = -Inf, ymax = Inf, fill = 'gray95' )+

geom\_point( aes(color=Label, size = RA, shape= factor(pch) ), alpha=1) + #log(CPM)

**geom\_text\_repel**(data=forManh, aes(x=order, y= -log10(signif), label=label), colour="black", size=3)+

scale\_size\_continuous(range = c(0.5, 4.0), breaks=c(0.01, 0.05, 0.1, 0.5, 1.0))+ #点的大小, guide =F

scale\_shape\_manual(values = c(16, 21 , 1), guide ="none")+

scale\_color\_manual(values=c("turquoise4", "springgreen3", "yellowgreen", "chartreuse", "brown2", "magenta", "darkorange", "dodgerblue", "tan4", "cyan1", "grey30") , guide ="none")+

geom\_hline(yintercept = -log10(0.01), color = 'gray36', linetype = 2, linewidth = 1) +

scale\_x\_continuous(breaks = phylum\_numhalf, labels = phylum\_name)+

labs(x=NULL, y= "-log(P)" )+

theme\_classic()+

theme(axis.text=element\_text(color="black"), axis.text.x=element\_text(angle = 45, hjust = 1, vjust = 1, size=8),

title=element\_text(size=10, face="bold"),

legend.title=element\_blank(), legend.position=c(0.45, 0.75), legend.text=element\_text(size=8),

legend.background=element\_rect(fill=NA, linewidth=0.6, linetype="dashed", colour ="grey20") )

Manhattan.MI