library (readr); library(DESeq2); library(dplyr); library(reshape2); library(stringr); library (ggplot2); library(ggpubr); library (ggrepel)

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

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

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

otu\_filter\_16s<-read.table("otu\_filter\_16s.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

Pct <- 0.05; BM=10; CK.rep=4; Trt.rep=4

#################Wild type#######################################

library (dplyr)

design.FW <- **filter**(design, Trt !="Cured"); design.FW <- **filter**(design.FW, Trt !="Restored");

otu.data.FW <- otu.data[, rownames(design.FW)]

Myotu.data <- as.data.frame(otu.data.FW)

Mydesign <- data.frame(Trt = factor(c(rep(c('control'), CK.rep), rep(c('treat'), Trt.rep)), levels = c('control', 'treat')))

dds <- DESeqDataSetFromMatrix( countData = Myotu.data, colData = Mydesign , design= ~Trt )

dds1 <- DESeq(dds, fitType = 'mean', minReplicatesForReplace = 7, parallel = F)

res <- results(dds1, contrast = c('Trt', 'treat', 'control')) # 默认 pAdjustMethod="BH"

res

res1 <- data.frame(res, stringsAsFactors = FALSE, check.names = FALSE)

###################作图MA plots##########

tt\_enrich <- as.data.frame(res1)

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

forMA <- tt\_enrich

forMA <- mutate(forMA, signif = ifelse(padj <= Pct, "T", "F"))

forMA.W <- forMA

Enriched.W <- filter(forMA, log2FoldChange>0&baseMean>BM & padj< Pct); dim (Enriched.W) # Enriched &baseMean>5

Depleted.W <- filter(forMA, log2FoldChange<0&baseMean>BM & padj< Pct); dim (Depleted.W) # Depleted &baseMean>5

cs <- c(rownames(Enriched.W), rownames(Depleted.W) )

# cs.TC.En <- rownames(T\_enrich); cs.TC.De <- rownames(CK\_enrich)

# S.Taxa<-tax\_filter\_16s[rownames(T\_enrich), ]; rownames(S.Taxa)[ S.Taxa $Genus=="Bacillus" ]

**write\_rds** (rownames(Enriched.W), "Enriched.W.rds")

**write\_rds** (rownames(Depleted.W), "Depleted.W.rds")

## define colors

forMA$col <- "dimgrey"

forMA [which(rownames(forMA) %in% rownames(Depleted.W) ), ]$col <- "darkorange"

forMA [which(rownames(forMA) %in% rownames(Enriched.W) ), ]$col <- "dodgerblue"

# order for plotting colors

forMA$ord[forMA$col=="dimgrey"] <- 1; forMA$ord[forMA$col=="darkorange"] <- 2; forMA$ord[forMA$col=="dodgerblue"] <- 3

forMA <- forMA[sort(forMA$ord, ind=T, decr=F) $ ix, ]

## define pch

forMA$pch <- 1 ; forMA [which(rownames(forMA) %in% cs ), ] $pch <- 16

## define size

forMA$size <- 1; forMA [which(rownames(forMA) %in% cs ), ]$size <- 1.5

forMA$label <- NA

forMA $ OTUnames <- rownames(forMA)

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

forMA [rownames(forMA)[forMA$Genus=="unassigned" & rownames(forMA) %in% cs ], ]$label <- NA

xrng <- range(log2(forMA$baseMean)) ; yrng <- range(forMA$log2FoldChange)

**write.table**(forMA, 'forMA.WD.txt', col.names = NA, sep = '\t', quote = FALSE)

MA.plot.W <- ggplot(data=forMA, aes(x= log2(baseMean), y= log2FoldChange, colour=col))+

geom\_point(aes(shape= factor(pch), size=size) )+

scale\_color\_manual(values=c("dodgerblue", "dimgrey", "forestgreen"), guide="none" )+

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

scale\_size\_continuous(range = c(0.6, 1.2), guide="none")+ #点的大小

labs(x="Average abundance (log2(BM))", y="log2 fold change")+

**geom\_text\_repel**(data=forMA, aes(x=log2(baseMean), y=log2FoldChange, label=label), colour="black", size=3) +

annotate("text",label=paste("Depleted: ", format(length(rownames(Depleted.W))), sep=""), color= "dodgerblue",

x=xrng[2]-2, y= -4.5, size=4, vjust="inward", hjust="inward") +

annotate("text", label=paste("Enriched: ", format(length(rownames(Enriched.W))), sep=""), color= "forestgreen",

x=xrng[2]-2, y= 4.5, size=4, vjust="inward", hjust="inward") +

theme\_bw()+

# geom\_hline(yintercept = c(-1, 1), lty = 3, color = 'black') + #添加阈值线

theme(panel.grid=element\_blank(),

axis.text=element\_text(size=10, color="black"), axis.title=element\_text(size=12),

# legend.title=element\_blank(), legend.position=c(0.15, 0.75),

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

legend.text=element\_text(size=12) )

MA.plot.W

#################Cured FOL#######################################

library (dplyr)

design.FC <- **filter**(design, Trt !="WD"); design.FC <- **filter**(design.FC, Trt !="Restored");

otu.data.FC <- otu.data[, rownames(design.FC)]

Myotu.data <- as.data.frame(otu.data.FC)

Mydesign <- data.frame(Trt = factor(c(rep(c('control'), CK.rep), rep(c('treat'), Trt.rep)), levels = c('control', 'treat')))

dds <- DESeqDataSetFromMatrix( countData = Myotu.data, colData = Mydesign , design= ~Trt )

dds1 <- DESeq(dds, fitType = 'mean', minReplicatesForReplace = 7, parallel = F)

res <- results(dds1, contrast = c('Trt', 'treat', 'control')) # 默认 pAdjustMethod="BH"

res

res1 <- data.frame(res, stringsAsFactors = FALSE, check.names = FALSE)

###################作图MA plots##########

tt\_enrich <- as.data.frame(res1)

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

forMA <- tt\_enrich

forMA <- mutate(forMA, signif = ifelse(padj <= Pct, "T", "F"))

forMA.C <- forMA

Enriched.C <- filter(forMA, log2FoldChange>0&baseMean>BM & padj< Pct); dim (Enriched.C) # Enriched &baseMean>5

Depleted.C <- filter(forMA, log2FoldChange<0&baseMean>BM & padj< Pct); dim (Depleted.C) # Depleted &baseMean>5

cs <- c(rownames(Enriched.C), rownames(Depleted.C) )

**write\_rds** (rownames(Enriched.C), "Enriched.C.rds")

**write\_rds** (rownames(Depleted.C), "Depleted.C.rds")

## define colors

forMA$col <- "dimgrey"

forMA [which(rownames(forMA) %in% rownames(Depleted.C) ), ]$col <- "darkorange"

forMA [which(rownames(forMA) %in% rownames(Enriched.C) ), ]$col <- "dodgerblue"

# order for plotting colors

forMA$ord[forMA$col=="dimgrey"] <- 1; forMA$ord[forMA$col=="darkorange"] <- 2; forMA$ord[forMA$col=="dodgerblue"] <- 3

forMA <- forMA[sort(forMA$ord, ind=T, decr=F) $ ix, ]

## define pch

forMA$pch <- 1 ; forMA [which(rownames(forMA) %in% cs ), ] $pch <- 16

## define size

forMA$size <- 1; forMA [which(rownames(forMA) %in% cs ), ]$size <- 1.5

forMA$label <- NA

forMA $ OTUnames <- rownames(forMA)

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

forMA [rownames(forMA)[forMA$Genus=="unassigned" & rownames(forMA) %in% cs ], ]$label <- NA

xrng <- range(log2(forMA$baseMean)) ; yrng <- range(forMA$log2FoldChange)

**write.table**(forMA, 'forMA.Cured.txt', col.names = NA, sep = '\t', quote = FALSE)

MA.plot.Cured <- ggplot(data=forMA, aes(x= log2(baseMean), y= log2FoldChange, colour=col))+

geom\_point(aes(shape= factor(pch), size=size) )+

scale\_color\_manual(values=c("dodgerblue", "dimgrey", "darkorange2"), guide="none" )+

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

scale\_size\_continuous(range = c(0.6, 1.2), guide="none")+ #点的大小

labs(x="Average abundance (log2(BM))", y="log2 fold change")+

**geom\_text\_repel**(data=forMA, aes(x=log2(baseMean), y=log2FoldChange, label=label), colour="black", size=3) +

annotate("text",label=paste("Depleted: ", format(length(rownames(Depleted.C))), sep=""), color= "dodgerblue",

x=xrng[2]-2, y= -3, size=4, vjust="inward", hjust="inward") +

annotate("text", label=paste("Enriched: ", format(length(rownames(Enriched.C))), sep=""), color= "darkorange2",

x=xrng[2]-2, y= 3, size=4, vjust="inward", hjust="inward") +

theme\_bw()+

# geom\_hline(yintercept = c(-1, 1), lty = 3, color = 'black') + #添加阈值线

theme(panel.grid=element\_blank(),

axis.text=element\_text(size=10, color="black"), axis.title=element\_text(size=12),

# legend.title=element\_blank(), legend.position=c(0.15, 0.75),

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

legend.text=element\_text(size=12) )

MA.plot.Cured

#################Restored FOL#######################################

library (dplyr)

design.FR <- **filter**(design, Trt !="WD"); design.FR <- **filter**(design.FR, Trt !="Cured");

otu.data.FR <- otu.data[, rownames(design.FR)]

Myotu.data <- as.data.frame(otu.data.FR)

Mydesign <- data.frame(Trt = factor(c(rep(c('control'), CK.rep), rep(c('treat'), Trt.rep)), levels = c('control', 'treat')))

dds <- DESeqDataSetFromMatrix( countData = Myotu.data, colData = Mydesign , design= ~Trt )

dds1 <- DESeq(dds, fitType = 'mean', minReplicatesForReplace = 7, parallel = F)

res <- results(dds1, contrast = c('Trt', 'treat', 'control'))

res

res1 <- data.frame(res, stringsAsFactors = FALSE, check.names = FALSE)

###################作图MA plots##########

tt\_enrich <- as.data.frame(res1)

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

forMA <- tt\_enrich

forMA <- mutate(forMA, signif = ifelse(padj <= Pct, "T", "F"))

forMA.R <- forMA

Enriched.R <- filter(forMA, log2FoldChange>0&baseMean>BM & padj< Pct); dim (Enriched.R) # Enriched &baseMean>5

Depleted.R <- filter(forMA, log2FoldChange<0&baseMean>BM & padj< Pct); dim (Depleted.R) # Depleted &baseMean>5

cs <- c(rownames(Enriched.R), rownames(Depleted.R) )

**write\_rds** (rownames(Enriched.R), "Enriched.R.rds")

**write\_rds** (rownames(Depleted.R), "Depleted.R.rds")

## define colors

forMA$col <- "dimgrey"

forMA [which(rownames(forMA) %in% rownames(Depleted.R) ), ]$col <- "darkorange"

forMA [which(rownames(forMA) %in% rownames(Enriched.R) ), ]$col <- "dodgerblue"

# order for plotting colors

forMA$ord[forMA$col=="dimgrey"] <- 1; forMA$ord[forMA$col=="darkorange"] <- 2; forMA$ord[forMA$col=="dodgerblue"] <- 3

forMA <- forMA[sort(forMA$ord, ind=T, decr=F) $ ix, ]

## define pch

forMA$pch <- 1 ; forMA [which(rownames(forMA) %in% cs ), ] $pch <- 16

## define size

forMA$size <- 1; forMA [which(rownames(forMA) %in% cs ), ]$size <- 1.5

forMA$label <- NA

forMA $ OTUnames <- rownames(forMA)

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

forMA [rownames(forMA)[forMA$Genus=="unassigned" & rownames(forMA) %in% cs ], ]$label <- NA

xrng <- range(log2(forMA$baseMean)) ; yrng <- range(forMA$log2FoldChange)

**write.table**(forMA, 'forMA.Restored.txt', col.names = NA, sep = '\t', quote = FALSE)

MA.plot.Restored <- ggplot(data=forMA, aes(x= log2(baseMean), y= log2FoldChange, colour=col))+

geom\_point(aes(shape= factor(pch), size=size) )+

scale\_color\_manual(values=c("dodgerblue", "dimgrey", "darkorange2"), guide="none" )+

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

scale\_size\_continuous(range = c(0.6, 1.2), guide="none")+ #点的大小

labs(x="Average abundance (log2(BM))", y="log2 fold change")+

**geom\_text\_repel**(data=forMA, aes(x=log2(baseMean), y=log2FoldChange, label=label), colour="black", size=3) +

annotate("text",label=paste("Depleted: ", format(length(rownames(Depleted.R))), sep=""), color= "dodgerblue",

x=xrng[2]-2, y= -3, size=4, vjust="inward", hjust="inward") +

annotate("text", label=paste("Enriched: ", format(length(rownames(Enriched.R))), sep=""), color= "darkorange2",

x=xrng[2]-2, y= 3, size=4, vjust="inward", hjust="inward") +

theme\_bw()+

# geom\_hline(yintercept = c(-1, 1), lty = 3, color = 'black') + #添加阈值线

theme(panel.grid=element\_blank(),

axis.text=element\_text(size=10, color="black"), axis.title=element\_text(size=12),

# legend.title=element\_blank(), legend.position=c(0.15, 0.75),

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

legend.text=element\_text(size=12) )

MA.plot.Restored

**ggarrange**(MA.plot.W, MA.plot.Cured, MA.plot.Restored, ncol=3, nrow=1, widths = c(1, 1), align="hv", legend="none" )

############# Plotting Venn Diagram ##########################

**library** (VennDiagram); require("vegan")

venn.data**<-list**()

venn.data$ Enriched.W <-rownames(Enriched.W)

venn.data$ Enriched.R <-rownames(Enriched.R)

venn.data$ Enriched.C <- rownames(Enriched.C)

grob.list<-venn.diagram(x=venn.data, filename=NULL, lwd=2, cex=1.2, cat.cex=2 )

grid.draw (grob.list)

venn.data**<-list**()

venn.data$ Depleted.W <-rownames(Depleted.W)

venn.data$ Depleted.R <-rownames(Depleted.R)

venn.data$ Depleted.C <- rownames(Depleted.C)

grob.list<-venn.diagram(x=venn.data, filename=NULL, lwd=2, cex=1.2, cat.cex=2 )

grid.draw (grob.list)