**Correlation matrix**

mydata<- read.csv("genus\_lefse.csv",header = T,row.names = 1)

str(mydata)

library(Hmisc)

res2 <- rcorr(as.matrix(mydata),type = "spearman")

res2

library(corrplot)

# Insignificant correlation are crossed

corrplot(res2$r, type="upper", order="original",

 p.mat = res2$P, sig.level = 0.05,tl.pos = "td",tl.cex=0.55,

 tl.col = c("#FF6666","#FF6666",

 "#009933","#009933","#009933","#009933","#009933"

 ,"#009933","#009933","#009933"),cl.length = 5,cl.cex = 0.6,cl.ratio=0.2,cl.align.text="r", cl.offset = -0.03)

**PCoA**

otu <- read.delim('F-M.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

otu <- data.frame(t(otu))

group <- read.delim('F-M\_group.txt', sep = '\t', stringsAsFactors = FALSE)

library(vegan)

distance<- vegdist(otu,method='bray')

pcoa <- cmdscale(distance, k = (nrow(otu) - 1), eig = TRUE)

ordiplot(scores(pcoa)[ ,c(1, 2)], type = 't')

summary(pcoa)

pcoa$eig

point <- data.frame(pcoa$point)

pcoa\_eig <- (pcoa$eig)[1:2] / sum(pcoa$eig)

sample\_site <- data.frame({pcoa$point})[1:2]

sample\_site$ID <- rownames(sample\_site)

names(sample\_site)[1:2] <- c('PCoA1', 'PCoA2')

sample\_site <- merge(sample\_site, group, by = 'ID', all.x = TRUE)

#sample\_site$Group <- factor(sample\_site$Group, levels = c("YES","NO"))

library(plyr)

group\_border <- ddply(sample\_site, 'Group', function(df) df[chull(df[[2]], df[[3]]), ])

library(ggplot2)

pcoa\_plot <- ggplot(sample\_site, aes(PCoA1, PCoA2, group = Group)) +

 theme(panel.grid = element\_line(color = 'gray', linetype = 2, size = 0.1), panel.background = element\_rect(color = 'black', fill = 'transparent'), legend.key = element\_rect(fill = 'transparent')) +

 geom\_vline(xintercept = 0, color = 'gray', size = 0.4) +

 geom\_hline(yintercept = 0, color = 'gray', size = 0.4) +

 geom\_polygon(data = group\_border, aes(fill = Group)) +

 geom\_point(aes(color = Group, shape = Group), size = 2.5, alpha = 0.8) +

 scale\_shape\_manual(values = c(16, 16)) +

 scale\_color\_manual(values = c('#33CCCC', '#FF3300')) +

 scale\_fill\_manual(values = alpha(c("#00FFFF", "#FF0066"), .2)) +

 guides(fill = guide\_legend(order = 1), shape = guide\_legend(order = 2), color = guide\_legend(order = 3)) +

 labs(x = paste('PCoA axis1: ', round(100 \* pcoa\_eig[1], 2), '%'), y = paste('PCoA axis2: ', round(100 \* pcoa\_eig[2], 2), '%'))

pcoa\_plot

**Hierarchical Clustering**

library(vegan)

rowdata<-read.delim("test.txt",row.names = 1,header = T)

dis\_bray<- vegdist(rowdata,method="bray")

clust\_single <- hclust(dis\_bray, method = 'complete')

plot(clust\_single,hang = -1, cex = 0.6)

library(dendextend)

color<-read.csv("color.csv",header = T)

color<-as.vector.factor(color$G)

color<-c(color)

dend <- as.dendrogram(clust\_single)

dend %>% set("leaves\_pch", 19) %>% # node point type

 set("leaves\_cex", 1.0) %>% # node point size

 set("leaves\_col", color) %>% #node point color

 set("labels\_cex", 0.6)%>%

 plot(main = "Leaves points",cex = 0.4)

**AUCRF**

mydata<- read.csv("RF2.csv", header = T, row.names = 1)

str(mydata)

mydata$Group<- as.factor(mydata$Group)

library(AUCRF)

set.seed(03)

mymodel<- AUCRF(Group~.,data=mydata)

mymodelcv<- AUCRFcv(mymodel,nCV=10,M=10)

summary(mymodelcv)

plot(mymodelcv)

attributes(mymodelcv)

**RF model construction with cross-validation**

mydata<- read.csv("1\_2\_TOP4.csv", header = T,row.names = 1)

str(mydata)

mydata$Group<- as.factor(mydata$Group)

library(caret)

library(randomForest)

set.seed(04)

splitRule<- trainControl(method = "repeatedcv", number = 10,repeats = 10,

 classProbs = T, summaryFunction =twoClassSummary)

model <- train(

 Group ~., data = mydata, method = "rf",

 trControl = splitRule,

 importance = TRUE,metric="ROC")

model$bestTune

modelf<-model$finalModel

modelf

importance(modelf)

attributes(modelf)

predicted.classes <- predict(modelf, newdata=mydata,type="prob")

predicted.classes

varImpPlot(model$finalModel, type = 2)

varImp(model)

**RF model test**

testdata<-read.csv("1\_2\_TOP4.csv", header = T,row.names = 1)

testdata$Group<- as.factor(testdata$Group)

predicted.classes <- predict(modelf, newdata=testdata,type="prob")

predicted.classes