###treatment effects by linear mixed model

library(lme4)

library(car)

data=read.table("clipboard",sep='\t',header = T)

fm<-lmer(variable~tillage\*warming+(1|time),data=data)

summary(fm)

car::Anova(fm,type=2)

###cohen'd

library(effsize)

set.seed(45)

x <- rnorm(10, 10, 1)

y <- rnorm(10, 5, 5)

cohen.d(x,y)

---

" Random-forest model detects bacterial taxa that accurately predict warming and management effects."

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

# Load setting and functions

source("../stat\_plot\_functions.R")

# Set output directory

output\_dir="./"

otutab <- read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)

design = read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)

#design$group = design$group2

if (TRUE){

sub\_design = subset(design, group %in% c("TW","NW","TN","NN"))

sub\_design$group = factor(sub\_design$group, levels=c("TW","NW","TN","NN"))

}

idx = rownames(sub\_design) %in% colnames(otutab)

sub\_design = sub\_design[idx,]

sub\_otutab = otutab[,rownames(sub\_design)]

head(sub\_otutab)[,1:10]

otutab\_t = as.data.frame(t(sub\_otutab))

otutab\_t$group = factor(design[rownames(otutab\_t),]$group, levels= c("TW","NW","TN","NN"))

# set random seed for reproducible

set.seed(315)

library(randomForest)

otutab\_t.rf= randomForest(group ~ ., data=otutab\_t, importance=TRUE, proximity=TRUE)

print(otutab\_t.rf)

# MeanDecreaseAccuracy

write.table(otutab\_t.rf$confusion, file = " otu\_confusion.txt", sep = "\t", quote = F, row.names = T, col.names = T)

imp = as.data.frame(round(importance(otutab\_t.rf), 2))

imp=imp[order(imp$MeanDecreaseAccuracy,decreasing = F),]

write.table(imp, file = " family\_imp.txt", sep = "\t", quote = F, row.names = T, col.names = T)

n = ncol(otutab\_t)-1

myotutab\_t= otutab\_t[1:n]

set.seed(315)

result= rfcv(myotutab\_t, otutab\_t$group, cv.fold=5, scale = "log", step = 0.9)

# result$n.var

# length(result$n.var)

with(result, plot(n.var, error.cv, log="x", type="o", lwd=2))

result1 = result

error.cv = data.frame(num = result$n.var, error.1 = result$error.cv)

for (i in 316:(315+4)){

print(i)

set.seed(i)

result= rfcv(myotutab\_t, otutab\_t$group, cv.fold=5, scale = "log", step = 0.9)

error.cv = cbind(error.cv, result$error.cv)

}

n.var = error.cv$num

error.cv = error.cv[,2:6]

colnames(error.cv) = paste('err',1:5,sep='.')

err.mean = apply(error.cv,1,mean)

allerr = data.frame(num=n.var,err.mean=err.mean,error.cv)

# number of features selected

optimal =3

write.table(allerr, file = " family\_rfcv.txt", sep = "\t", quote = F, row.names = T, col.names = T)

p = ggplot() +

geom\_line(aes(x = allerr$num, y = allerr$err.1), colour = 'grey') +

geom\_line(aes(x = allerr$num, y = allerr$err.2), colour = 'grey') +

geom\_line(aes(x = allerr$num, y = allerr$err.3), colour = 'grey') +

geom\_line(aes(x = allerr$num, y = allerr$err.4), colour = 'grey') +

geom\_line(aes(x = allerr$num, y = allerr$err.5), colour = 'grey') +

geom\_line(aes(x = allerr$num, y = allerr$err.mean), colour = 'black') +

geom\_vline(xintercept = optimal, colour='black', lwd=0.36, linetype="dashed") +

# geom\_hline(yintercept = min(allerr$err.mean), colour='black', lwd=0.36, linetype="dashed") +

coord\_trans(x = "log2") +

scale\_x\_continuous(breaks = c(1, 2, 5, 10, 20, 30, 50, 100, 200)) + # , max(allerr$num)

labs(title=paste('Training set (n = ', dim(otutab\_t)[1],')', sep = ''),

x='Number of families ', y='Cross-validation error rate') +

annotate("text", x = optimal, y = max(allerr$err.mean), label=paste("optimal = ", optimal, sep=""))

p

ggsave(p, file = "family\_rfcv.pdf", width = 89, height = 50, unit = 'mm')

p

### Features barplot with taxonomy in phylum

imp = read.table("family\_imp.txt", header=T, row.names= 1, sep="\t")

imp = tail(imp, n = optimal)

imp$group = factor(rownames(imp), levels = rownames(imp))

#phylum <- read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)

phylum = read.table("exudate\_notes.txt",row.names = 1, header=T, sep="\t")

imp= merge(imp, phylum, by = "group")

p = ggplot(imp, aes(x = group, y = MeanDecreaseAccuracy, fill = exudates)) +

geom\_bar(stat = "identity") +

coord\_flip()

###“Circular heatmap”

library(ComplexHeatmap)

mat1<- read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)

head(mat1)

length(split) == nrow(mat1)

split<- as.factor(mat1[,1])

NN <- mat1[,"NN", drop = FALSE]

col1 <- setNames(c("#35A585", "#006FB0", "#B29043", "#F2C661", "#837E72",

"#FFB199", "#A6808C", "#7E102C"), levels(split))

circos.clear()

col\_fun1 = colorRamp2(c(1.9, 5.4, 8.9), c("#B7E4FA", "white", "#FF7302"))

circos.heatmap(NN, split = split, col = col\_fun1,rownames.side = "outside",bg.border = "black",bg.lwd=1,track.height = 0.1,

show.sector.labels = TRUE)

circos.heatmap(mat1$NW, split = split, col = col\_fun1,rownames.side = "outside",bg.border = "black",cluster = TRUE,track.height = 0.1,

show.sector.labels = TRUE)

# row\_mean = rowMeans(mat1[, 1:3])

row\_mean = mat1$P

circos.track(ylim = range(row\_mean), panel.fun = function(x, y) {

y = row\_mean[CELL\_META$subset]

y = y[CELL\_META$row\_order]

circos.lines(CELL\_META$cell.xlim, c(10, 1), lty = 1, col = "grey")

circos.points(seq\_along(y) - 0.5, y,pch=1, cex=0.8,col = ifelse(y > -log10(0.05), "#D95F04", "#1C9E77"))

}, cell.padding = c(0.02, 0, 0.02, 0),

track.height = 0.1)

circos.heatmap(mat1$split, split = split, col = col1,rownames.side = "inside",cluster = TRUE,track.height = 0.07,

show.sector.labels = TRUE)

lgd\_links = Legend(at = c(2, 4, 6, 8, 9), col\_fun = col\_fun1,

title\_position = "topleft", title = "Links", direction = "horizontal")

draw(lgd\_links, x = unit(1, "npc") - unit(2, "mm"), y = unit(4, "mm"),

just = c("right", "bottom"))

dev.off()

“KEGG enrichment analysis”

p <- enrichmentNetwork(enrich.go,

colorBy = 'pvalue',

#colorType = c("pval"),

nodeSize = 'Count',

fontSize = 4,

drawEllipses = F,

pCutoff = -10,

verbose = TRUE) +

scale\_color\_continuous\_c4a\_seq('yl\_gn\_bu', reverse = T)

p

###“Sankey analysis ”

data <- read.xlsx("specise.xlsx", sheet = "Sheet13")

mydata1 <- melt(

data,

id.vars = c("Group"),

variable.name = "variable",

value.name = "value"

)

df\_clean <- subset(mydata1, !value %in% 0)

data <- df\_clean

###

#group<-unique(data$group)

Group<-unique(data$Group)

variable<-unique(data$variable)

nodes<-data.frame(name=c( Group,variable))

links<-data%>%

bind\_rows(

data%>%select(Group,variable,value)%>%

rename(source=Group,target=variable,value=value),

)%>%

group\_by(source,target)%>%

summarise(value=sum(value),.groups='drop')%>%

ungroup()

links$IDsource<-match(links$source,nodes$name)-1

links$IDtarget<-match(links$target,nodes$name)-1

colourScale<-sprintf('d3.scaleOrdinal().range(["%s"])',

paste(mycol2,collapse='","'))

nodes$ID <- 0:(nrow(nodes) - 1)

links <- links %>%

left\_join(nodes, by = c("source" = "name")) %>%

rename(IDsource = ID) %>%

left\_join(nodes, by = c("target" = "name")) %>%

rename(IDtarget = ID)

write.csv(links,"links2.csv")

o <- sankeyNetwork(Links=links,

Nodes=nodes,

Source="IDsource",

Target="IDtarget",

Value="value",

NodeID="name",

units="T",

fontSize=20,

fontFamily="Times New Roman",

nodeWidth=15,

height=600,

width=800,

iterations=0,

nodePadding=10,

colourScale=colourScale)

o

saveNetwork(p,"sankey1.html")

###"RDA\_analysis"

rm(list=ls())

library(pacman)

library(vegan)

library(ggplot2)

data <- read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)

head(data,n=3)

env=read.table(pipe("pbpaste"),sep = "\t",header = T,row.names = 1)head(env,n=3)

print(decorana(t(data)))

B.rda=rda(t(data),env[-1],scale = T)

B.rda.data=data.frame(B.rda$CCA$u[,1:2],

env$Treat)

colnames(B.rda.data)=c("RDA1","RDA2","group")

head(B.rda.data,n=3)

B.rda.spe=data.frame(B.rda$CCA$v[,1:2])

B.rda.spe=as.data.frame(B.rda.spe)

B.rda.spe$Species<-rownames(B.rda.spe)

head(B.rda.spe,n=3)

B.rda.env <- B.rda$CCA$biplot[,1:2]

B.rda.env <- as.data.frame(B.rda.env)

head(B.rda.env,n=3)

yanse<-c("#487297","#a579a1","#3a9c84","#d07527")

p1=ggplot(data=B.rda.data,aes(RDA1,RDA2))+

geom\_point(aes(color=group,fill=group),alpha=0.6,size=12)+

labs(title = "B RDA plot",

x=paste("RDA1",round(B.rda$CCA$eig[1]/sum(B.rda$CCA$eig)\*100,2)," %"),

y=paste("RDA2",round(B.rda$CCA$eig[2]/sum(B.rda$CCA$eig)\*100,2)," %"))+

geom\_hline(yintercept = 0,lty=3)+

geom\_vline(xintercept = 0,lty=3)+

geom\_segment(data=B.rda.env,aes(x=0,y=0,xend=B.rda.env[,1]\*0.5,yend=B.rda.env[,2]\*0.5),

colour="gray",size=1,arrow=arrow(angle = 45,length=unit(0.3,"cm")))+

geom\_text(data=B.rda.env,aes(x=B.rda.env[,1]\*0.5,y=B.rda.env[,2]\*0.5,

label=rownames(B.rda.env)),size=5,colour="black",

hjust=(1-sign(B.rda.env[,1]))/2,angle=(180/pi)\*atan(B.rda.env[,2]/B.rda.env[,1]))+

stat\_ellipse(aes(RDA1,RDA2,color=group,fill=B.rda.data$group), geom ="polygon",

alpha=0.34, linetype = 0,level=0.85,show.legend = F)+

# stat\_ellipse(aes(RDA1,RDA2,color=group), geom ="polygon",alpha=0.56,

# linetype = 1,level=0.8,show.legend = F)+

scale\_color\_manual(values=yanse)+

# scale\_shape\_manual(values = c(21,22,43))+

scale\_fill\_manual(values = yanse)+

# geom\_point(data=B.rda.spe,aes(RDA1,RDA2),pch=8,size=5)+

# geom\_text(data=B.rda.spe,aes(x=B.rda.spe[,1],y=B.rda.spe[,2],label=Species),

# size=5.5,colour="black",hjust=0.5,vjust=1)

B.sum=summary(B.rda)

B.sum$constr.chi/B.sum$tot.chi #constrainedB.sum$unconst.chi/B.sum$tot.chi#unconstrained

cor\_data=data.frame(row.names = c("Explained","Unexplained"),

B=c(B.sum$constr.chi/B.sum$tot.chi,B.sum$unconst.chi/B.sum$tot.chi))

cor\_data$group=rownames(cor\_data)

head(cor\_data,n=3)

cor\_data <- data.frame(cor\_data)

cor\_data=arrange(cor\_data,B)

head(cor\_data,n=3)

labs<-paste0(cor\_data$group,"\n(",round(cor\_data$B/sum(cor\_data$B)\*100,2),"%)")

pp <- pie(cor\_data$B,labels=labs,init.angle = 90,col=brewer.pal(nrow(cor\_data),"Reds"))

####“random Forest”

library(randomForest)

library(ade4)

library(ggplot2)

library(RColorBrewer)

library(rfUtilities)

rm(list=ls())

allindice <- read.table(pipe("pbpaste"),row.names = 1,sep = "\t",header = T)

allindice <- scale(allindice,center = TRUE, scale = TRUE)

set.seed(123)

enzyme\_forest <- randomForest(Denitrification~., data = allindice,

importance = TRUE, ntree = 1500, nPerm = 1000)

enzyme\_forest

library(rfPermute)

set.seed(100)

indice\_rfP <- rfPermute(Denitrification~., data = allindice, importance = TRUE, ntree = 2000, nrep = 1000, num.cores = 3)

indice\_rfP

importance\_indice.scale <- data.frame(importance(indice\_rfP, scale = TRUE), check.names = FALSE)

importance\_indice.scale

# summary(indice\_rfP)

importance\_indice.scale.pval <- (indice\_rfP$pval)[ , , 2]

importance\_indice.scale.pval

plot(importance(indice\_rfP, scale = TRUE))

importance\_indice.scale <- importance\_indice.scale[order(importance\_indice.scale$'%IncMSE', decreasing = TRUE), ]

library(ggplot2)

importance\_indice.scale$indice\_name <- rownames(importance\_indice.scale)

importance\_indice.scale$indice\_name <- factor(importance\_indice.scale$indice\_name, levels = importance\_indice.scale$indice\_name)

p <- ggplot(importance\_indice.scale, aes(indice\_name, `%IncMSE`)) +

geom\_col(width = 0.7, fill = '#7CA9C8', color = NA) +

labs(title = NULL, x = NULL, y = 'Increase in MSE (%)', fill = NULL) +

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

axis.line = element\_line(colour = 'black', size = 0.2)) +

theme(axis.text.x = element\_text(angle =45, hjust = 1)) +

scale\_y\_continuous(expand = c(0, 0), limit = c(0, 30))+#???????????Ƿ???0??ʼ

theme(axis.text.x = element\_text (colour="black", size=18),

axis.text.y = element\_text(colour="black", size=18),

axis.title.y = element\_text( colour="black", size=22),

axis.ticks = element\_line(colour = "black", size = 0.5),

panel.border = element\_rect(colour = "black",fill = "transparent",size =1))

#7CA9C8��ɫ F6AC62??ɫ 88CBC1??ɫ A7D174????

p

for (indice in rownames(importance\_indice.scale)) {

importance\_indice.scale[indice,'%IncMSE.pval'] <- importance\_indice.scale.pval[indice,'%IncMSE']

if (importance\_indice.scale[indice,'%IncMSE.pval'] >= 0.05) importance\_indice.scale[indice,'%IncMSE.sig'] <- ''

else if (importance\_indice.scale[indice,'%IncMSE.pval'] >= 0.01 & importance\_indice.scale[indice,'%IncMSE.pval'] < 0.05) importance\_indice.scale[indice,'%IncMSE.sig'] <- '\*'

else if (importance\_indice.scale[indice,'%IncMSE.pval'] >= 0.001 & importance\_indice.scale[indice,'%IncMSE.pval'] < 0.01) importance\_indice.scale[indice,'%IncMSE.sig'] <- '\*\*'

else if (importance\_indice.scale[indice,'%IncMSE.pval'] < 0.001) importance\_indice.scale[indice,'%IncMSE.sig'] <- '\*\*\*'

}

p <- p +

geom\_text(data = importance\_indice.scale, aes(x = indice\_name, y = `%IncMSE`, label = `%IncMSE.sig`),

nudge\_y = 1,size=13)

p

importance\_indice.scale

library(A3)

set.seed(123)

indice\_forest.pval <- a3(Denitrificationn~.,data = allindice, randomForest,

model.args = list(importance = TRUE, ntree = 500))

indice\_forest.pval

# set.seed(123)

# otu\_forest.pval <- a3(plant\_age~., data = otu, model.fn = randomForest, p.acc = 0.001, model.args = list(importance = TRUE, ntree = 500))

library(ggplot2)

library(ggpubr)

library(splines)

library(dplyr)

library(ggpmisc)

all <- read.table(pipe("pbpaste"),row.names = 1,sep = "\t",header = T)

{c<- ggplot(all, aes(TOC\_g, Denitrification))+

geom\_point(aes(colour=factor(group)),size=9)+

geom\_smooth(method = "lm",formula = y~x,color="#696969",size=3)+

# stat\_smooth(formula = y ~ x,method = "lm")+

stat\_poly\_eq(aes(label = paste(..eq.label.., ..rr.label..,..p.value.label..,sep = '~~~~')),

formula = y ~ x, parse = TRUE,size = 5, label.x = 0.1,

label.y = 0.95)+

scale\_colour\_manual(values =c("#56B4E9","#E69F00","#009E73","red"))+

# theme\_bw(base\_size = 15)+

theme(axis.title.x = element\_text(face="bold", colour="black", size=42),

axis.title.y = element\_text(face="bold", colour="black", size=42),

axis.text.x = element\_text(face="bold", colour="black", size=36),

axis.text.y = element\_text(face="bold", colour="black", size=36),

axis.line = element\_line(color="black",size = 0),

axis.ticks = element\_line(colour = "black", size = 3.5), panel.grid.major = element\_line(linetype = "dashed"),

panel.border = element\_rect(colour = "black",fill = "transparent",size =4),

plot.background = element\_blank(),panel.background = element\_blank())}

###violin+boxplot

library(ggplot2)

library(gghalves)

(p1<-ggplot(data,aes(x=treatment,y=value,color=treatment))+

#scale\_fill\_manual(values = c("red","blue"))+

scale\_colour\_manual(values = c("red","blue"))+

geom\_half\_violin(position=position\_nudge(x=0.3,y=0),

side='R',adjust=1,trim=F,alpha=0.5,width=0.5)+

geom\_boxplot(outlier.shape = NA, width =0.4,alpha=0.5,size=1)+

stat\_summary(fun="mean", geom="point", shape=20, size=2.5, color="black", fill="black",alpha=0.7)+

geom\_jitter(size =2.5,alpha =0.5,width =0.2)+

labs(x="",y="Value")+

scale\_y\_continuous(labels = scales::number\_format(accuracy = 0.1))+

theme\_bw()+ theme(panel.grid=element\_blank())+

theme(strip.text = element\_text(size = 15),

strip.background = element\_rect(fill = "gray95"))+

theme(panel.border = element\_blank()))

###linear fit

(p=ggplot(data =data, aes(x = time, y = velue,color=treatment)) +

geom\_point(size=3) + scale\_colour\_manual(values = c("red","blue"))+

geom\_smooth(method = 'glm', formula = y~x, se = TRUE, show.legend = FALSE) +

stat\_poly\_eq(aes(label = paste(after\_stat(rr.label), after\_stat(after\_stat(p.value.label)), sep = '~`,`~')),

formula = y~x, parse = TRUE, label.x.npc = 'left', label.y.npc = 'top', size = 4) +

theme\_bw()+theme(axis.text=element\_text(colour='black',size=12),

axis.title = element\_text(colour='black',size=15))+

theme(panel.grid = element\_blank(), panel.background = element\_rect(fill = 'transparent', color = 'gray10'))+

theme(strip.text = element\_text(size = 12),

strip.background = element\_rect(fill = "gray95")))