

W4 Correlation study

Bora Kim

March 5, 2021

This is meant for sharing dataset and script of the publication “Strigolactone structural specificity in microbiome recruitment in rice, 2021”. This markdown contains the process of correlation study between the relative abundance of ASV/genus/EC/pathway (that were obtained from W2 and W3) and level of strigolactones (SLs). All correlation study was performed using negative binomial generalized linear model together with P value validation using permutation. Goodness of model fit was assessed using shapiro test on residual of predicted model and checking outliers by cook’s distance. Note that, results of permutation test (Perm.P) could be changed every run because it is based on random sampling. Permutation, especially with EC table, takes long time depending on your computer system. For this reason, loading image file “W4_correlation_study_image_full.Rdata” is recommended as it contains all intermediate and final results so that no need to run permutation, although the image file “W4_correlation_study_image.Rdata” is provided too for starting from beginning.

1. Getting started

Load required dataset: all datasets here were generated from W2 and W3.

```
load("W4_correlation_study_image_full.Rdata")
```

Load required packages for analysis.

```
library(ggplot2)
library(MASS)
library(dplyr)
library(tibble)
library(tidyr)
library(stringr)
```

2. Prepare some datasets

```
tot_tax<-rbind(bac_tax,fun_tax)
```

#ASVs

```
RS_ASV<-cbind(bac.FoRS_ASV, fun.FoRS_ASV[14:143])
bac.FoRT_ASV_rc<-rownames_to_column(bac.FoRT_ASV)
fun.FoRT_ASV_rc<-rownames_to_column(fun.FoRT_ASV)
RT_ASV<-right_join(bac.FoRT_ASV_rc, fun.FoRT_ASV_rc[,c(1,15:98)], by="rowname")
rownames(RT_ASV)<-RT_ASV$rowname
RT_ASV<-RT_ASV[, -1]
```

#Genus

```
RS_genus<-cbind(RS_ASV[1:13], bac.FoRS_genus, fun.FoRS_genus2[2:30])
bac.FoRT_genus2<-cbind(bac.FoRT_ASV_rc[1:14], bac.FoRT_genus)
RT_genus<-right_join(bac.FoRT_genus2, fun.FoRT_genus2, by="rowname")
```

```

rownames(RT_genus)<-RT_genus$rowname
RT_genus<-RT_genus[,-1]

#EC
RSEC_col<-str_replace(colnames(FoRS_EC),":",".")
colnames(FoRS_EC)<-RSEC_col
RTEC_col<-str_replace(colnames(FoRT_EC),":",".")
colnames(FoRT_EC)<-RTEC_col

#path
path_des_x<-str_replace(path_des$rowname,"X3_HYDROXYPHENYLACETATE_DEGRADATION_PWY","3_HYDROXYPHENYLACETATE_DEGRADATION_PWY")
path_des$rowname<-path_des_x

```

3. Negative binomial generalized linear model

3.1. ASVs

3.1.1. Fo_RS

```

data=RS_ASV
nasv = dim(data)[2]-13 # number of variables
fourdo_est <- 0*1:nasv
meo5ds_est <- 0*1:nasv
orb_est <- 0*1:nasv
fourdo_std.error <- 0*1:nasv
meo5ds_std.error <- 0*1:nasv
orb_std.error <- 0*1:nasv
fourdo_Zvalue <- 0*1:nasv
meo5ds_Zvalue <- 0*1:nasv
orb_Zvalue <- 0*1:nasv
fourdo_p <- 0*1:nasv
meo5ds_p <- 0*1:nasv
orb_p <- 0*1:nasv
fourdo_shapiro <- 0*1:nasv
meo5ds_shapiro <- 0*1:nasv
orb_shapiro <- 0*1:nasv
fourdo_cd<- 0*1:nasv
meo5ds_cd <- 0*1:nasv
orb_cd <- 0*1:nasv

for(i in 1:nasv) {
  fourdo.g <-glm.nb(data[,13+i]~X4D0_pmol_g, data=data)
  fourdo_est[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 1]
  fourdo_std.error[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 3]
  fourdo_Zvalue[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 4]
  fourdo_p[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 4]
  fourdo_shapiro[i]<-shapiro.test(resid(fourdo.g))[2]$p.value
  fourdo_cd[i]<-sum(cooks.distance(fourdo.g)>0.3)

  meo5ds.g <-glm.nb(data[,13+i]~Me05DS_pmol_g, data=data)
  meo5ds_est[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 1]
  meo5ds_std.error[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 3]
  meo5ds_Zvalue[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 4]

```

```

meo5ds_p[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 4]
meo5ds_shapiro[i]<-shapiro.test(resid(meo5ds.g))[2]$p.value
meo5ds_cd[i]<-sum(cooks.distance(meo5ds.g)>0.3)

orb.g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
orb_est[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 1]
orb_std.error[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 2]
orb_Zvalue[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 3]
orb_p[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 4]
orb_shapiro[i]<-shapiro.test(resid(orb.g))[2]$p.value
orb_cd[i]<-sum(cooks.distance(orb.g)>0.3)
}

RS_ASV_res<-as.data.frame(cbind(
  fourdo_est, fourdo_std.error, fourdo_Zvalue,fourdo_p, fourdo_shapiro,fourdo_cd,
  meo5ds_est, meo5ds_std.error, meo5ds_Zvalue, meo5ds_p, meo5ds_shapiro,meo5ds_cd,
  orb_est, orb_std.error, orb_Zvalue, orb_p, orb_shapiro,orb_cd))

rownames(RS_ASV_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normally distributed residual & no outliers and validate its p value using permutation.

```

# 4do
RS_ASV_4do_res<-(subset(RS_ASV_res, fourdo_p<0.05&fourdo_shapiro>0.05&fourdo_cd<3))[1:6]
RS_ASV_4do_res_name<-rownames(RS_ASV_4do_res)
RS_ASV_4do<-RS_ASV[RS_ASV_4do_res_name]

data=RS_ASV_4do
res=RS_ASV_4do_res
test=RS_ASV$X4DO_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_ASV_4do_res$Perm.p<-Perm.p

```

```

RS_ASV_4do_res_rc<-rownames_to_column(subset(RS_ASV_4do_res, Perm.p<0.05))
RS_ASV_4do_res_tax<-left_join(RS_ASV_4do_res_rc,tot_tax, by="rowname")

# Me05DS
RS_ASV_meo5ds_res<-(subset(RS_ASV_res, meo5ds_p<0.05&meo5ds_shapiro>0.05&meo5ds_cd<3))[7:12]
RS_ASV_meo5ds_res_name<-rownames(RS_ASV_meo5ds_res)
RS_ASV_meo5ds<-RS_ASV[RS_ASV_meo5ds_res_name]

data=RS_ASV_meo5ds
res=RS_ASV_meo5ds_res
test=RS_ASV$Me05DS_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[,j]<res[j,4])/nPerm
}

RS_ASV_meo5ds_res$Perm.p<-Perm.p
RS_ASV_meo5ds_res_rc<-rownames_to_column(subset(RS_ASV_meo5ds_res, Perm.p<0.05))
RS_ASV_meo5ds_res_tax<-left_join(RS_ASV_meo5ds_res_rc,tot_tax, by="rowname")

# orobanchol
RS_ASV_orb_res<-(subset(RS_ASV_res, orb_p<0.05&orb_shapiro>0.05&orb_cd<3))[13:18]
RS_ASV_orb_res_name<-rownames(RS_ASV_orb_res)
RS_ASV_orb<-RS_ASV[RS_ASV_orb_res_name]

data=RS_ASV_orb
res=RS_ASV_orb_res
test=RS_ASV$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){

```

```

    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_ASV_orb_res$Perm.p<-Perm.p
RS_ASV_orb_res_rc<-rownames_to_column(subset(RS_ASV_orb_res, Perm.p<0.05))
RS_ASV_orb_res_tax<-left_join(RS_ASV_orb_res_rc,tot_tax, by="rowname")

```

Combining final results in one table

```

RS_ASV_4do_res_tax<-RS_ASV_4do_res_tax%>%mutate(tested_SL="4D0")
RS_ASV_meo5ds_res_tax<-RS_ASV_meo5ds_res_tax%>%mutate(tested_SL="Me05DS")
RS_ASV_orb_res_tax<-RS_ASV_orb_res_tax%>%mutate(tested_SL="Orobanchol")

force_bind = function(df1, df2, df3) {
  colnames(df2) = colnames(df1) = colnames(df3)
  bind_rows(df1, df2, df3)
}

RS_ASV_res_final<-force_bind(RS_ASV_4do_res_tax, RS_ASV_meo5ds_res_tax,RS_ASV_orb_res_tax)

new_col<-c("tested_ASV","est","std.error","Zvalue","glm.P","shapiro.p","no.outliers","Perm.P",
           "Kingdom","Phylum","Class","Order","Family","Genus","Species","tested_SL")

colnames(RS_ASV_res_final)<-new_col
RS_ASV_res_final<-RS_ASV_res_final[,c(16,1:15)]

```

3.1.2. Fo_RT

```

data=RT_ASV
nasv = dim(data)[2]-13 # number of variables
fourdo_est <- 0*1:nasv
meo5ds_est <- 0*1:nasv
orb_est <- 0*1:nasv
fourdo_std.error <- 0*1:nasv
meo5ds_std.error <- 0*1:nasv
orb_std.error <- 0*1:nasv
fourdo_Zvalue <- 0*1:nasv
meo5ds_Zvalue <- 0*1:nasv
orb_Zvalue <- 0*1:nasv
fourdo_p <- 0*1:nasv
meo5ds_p <- 0*1:nasv
orb_p <- 0*1:nasv
fourdo_shapiro <- 0*1:nasv
meo5ds_shapiro <- 0*1:nasv
orb_shapiro <- 0*1:nasv

```

```

fourdo_cd<- 0*1:nasv
meo5ds_cd <- 0*1:nasv
orb_cd <- 0*1:nasv

for(i in 1:nasv) {
  fourdo.g <-glm.nb(data[,13+i]~X4D0_pmol_g, data=data)
  fourdo_est[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 1]
  fourdo_std.error[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 2]
  fourdo_Zvalue[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 3]
  fourdo_p[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 4]
  fourdo_shapiro[i]<-shapiro.test(resid(fourdo.g))[2]$p.value
  fourdo_cd[i]<-sum(cooks.distance(fourdo.g)>0.3)

  meo5ds.g <-glm.nb(data[,13+i]~Me05DS_pmol_g, data=data)
  meo5ds_est[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 1]
  meo5ds_std.error[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 2]
  meo5ds_Zvalue[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 3]
  meo5ds_p[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 4]
  meo5ds_shapiro[i]<-shapiro.test(resid(meo5ds.g))[2]$p.value
  meo5ds_cd[i]<-sum(cooks.distance(meo5ds.g)>0.3)

  orb.g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
  orb_est[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 1]
  orb_std.error[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 2]
  orb_Zvalue[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 3]
  orb_p[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 4]
  orb_shapiro[i]<-shapiro.test(resid(orb.g))[2]$p.value
  orb_cd[i]<-sum(cooks.distance(orb.g)>0.3)
}

RT_ASV_res<-as.data.frame(cbind(
  fourdo_est, fourdo_std.error, fourdo_Zvalue,fourdo_p, fourdo_shapiro,fourdo_cd,
  meo5ds_est, meo5ds_std.error, meo5ds_Zvalue, meo5ds_p, meo5ds_shapiro,meo5ds_cd,
  orb_est, orb_std.error, orb_Zvalue, orb_p, orb_shapiro,orb_cd))

rownames(RT_ASV_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normaly distributed residual & no outliers and validate its p value using permutation.

```

# 4do
RT_ASV_4do_res<-(subset(RT_ASV_res, fourdo_p<0.05&fourdo_shapiro>0.05&fourdo_cd<3))[1:6]
RT_ASV_4do_res_name<-rownames(RT_ASV_4do_res)
RT_ASV_4do<-RT_ASV[RT_ASV_4do_res_name]

data=RT_ASV_4do
res=RT_ASV_4do_res
test=RT_ASV_4do$X4D0_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

```

```

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:naSV){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:naSV
for(j in 1:naSV){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_ASV_4do_res$Perm.p<-Perm.p
RT_ASV_4do_res_rc<-rownames_to_column(subset(RT_ASV_4do_res, Perm.p<0.05))
RT_ASV_4do_res_tax<-left_join(RT_ASV_4do_res_rc,tot_tax, by="rowname")

# MeO5DS
RT_ASV_meo5ds_res<-(subset(RT_ASV_res, meo5ds_p<0.05&meo5ds_shapiro>0.05&meo5ds_cd<3))[7:12]
RT_ASV_meo5ds_res_name<-rownames(RT_ASV_meo5ds_res)
RT_ASV_meo5ds<-RT_ASV[RT_ASV_meo5ds_res_name]

data=RT_ASV_meo5ds
res=RT_ASV_meo5ds_res
test=RT_ASV$MeO5DS_pmol_g

naSV = dim(data)[2] # number of variables
p <- 0*1:naSV
nPerm<-1000
Perm.table<-data.frame(matrix(NA,naSV,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:naSV){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:naSV
for(j in 1:naSV){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_ASV_meo5ds_res$Perm.p<-Perm.p
RT_ASV_meo5ds_res_rc<-rownames_to_column(subset(RT_ASV_meo5ds_res, Perm.p<0.05))
RT_ASV_meo5ds_res_tax<-left_join(RT_ASV_meo5ds_res_rc,tot_tax, by="rowname")

```

```

# orobanchol
RT_ASV_orb_res<-(subset(RT_ASV_res, orb_p<0.05&orb_shapiro>0.05&orb_cd<3))[13:18]
RT_ASV_orb_res_name<-rownames(RT_ASV_orb_res)
RT_ASV_orb<-RT_ASV[RT_ASV_orb_res_name]

data=RT_ASV_orb
res=RT_ASV_orb_res
test=RT_ASV$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grep1("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_ASV_orb_res$Perm.p<-Perm.p
RT_ASV_orb_res_rc<-rownames_to_column(subset(RT_ASV_orb_res, Perm.p<0.05))
RT_ASV_orb_res_tax<-left_join(RT_ASV_orb_res_rc,tot_tax, by="rowname")

```

Combining final results in one table

```

RT_ASV_4do_res_tax<-RT_ASV_4do_res_tax%>%mutate(tested_SL="4D0")
RT_ASV_meo5ds_res_tax<-RT_ASV_meo5ds_res_tax%>%mutate(tested_SL="MeO5DS")
RT_ASV_orb_res_tax<-RT_ASV_orb_res_tax%>%mutate(tested_SL="Orobanchol")

RT_ASV_res_final<-force_bind(RT_ASV_4do_res_tax, RT_ASV_meo5ds_res_tax,RT_ASV_orb_res_tax)

colnames(RT_ASV_res_final)<-new_col
RT_ASV_res_final<-RT_ASV_res_final[,c(16,1:15)]

```

3.1.3. Final results table

```

RS_ASV_res_final<-RS_ASV_res_final%>%mutate(dataset="Rhizosphere")
RT_ASV_res_final<-RT_ASV_res_final%>%mutate(dataset="Roots")

ASV_res_final<-rbind(RS_ASV_res_final,RT_ASV_res_final)
ASV_res_final<-ASV_res_final[,c(17,1:16)]
ASV_res_final$Perm.P[ASV_res_final$Perm.P<0.001] <- "p<0.001"
ASV_res_final[1:5,1:10]

```



```
ASV_res_final[1:5,1:10]
```

```
##      dataset tested_SL tested_ASV      est  std.error  Zvalue
## 1 Rhizosphere      4D0      bASV71  0.01943393 0.008271721  2.349441
## 2 Rhizosphere      4D0      fASV6  -0.04744957 0.019660556 -2.413440
## 3 Rhizosphere      4D0      fASV103 -0.10362707 0.043344399 -2.390783
## 4 Rhizosphere      Me05DS      fASV8  -0.01556487 0.005426705 -2.868199
## 5 Rhizosphere      Me05DS      fASV34  0.03135082 0.012422409  2.523731
##      glm.P shapiro.p no.outliers Perm.P
## 1 0.018801598 0.6571607          0 0.026
## 2 0.015802732 0.4872214          0 0.043
## 3 0.016812468 0.3274588          1 0.038
## 4 0.004128155 0.7852315          1 0.011
## 5 0.011611677 0.2298358          1 0.018
```

3.2. Genus

3.2.1. Fo_RS

```
data=RS_genus
nasv = dim(data)[2]-13 # number of variables
fourdo_est <- 0*1:nasv
meo5ds_est <- 0*1:nasv
orb_est <- 0*1:nasv
fourdo_std.error <- 0*1:nasv
meo5ds_std.error <- 0*1:nasv
orb_std.error <- 0*1:nasv
fourdo_Zvalue <- 0*1:nasv
meo5ds_Zvalue <- 0*1:nasv
orb_Zvalue <- 0*1:nasv
fourdo_p <- 0*1:nasv
meo5ds_p <- 0*1:nasv
orb_p <- 0*1:nasv
fourdo_shapiro <- 0*1:nasv
meo5ds_shapiro <- 0*1:nasv
orb_shapiro <- 0*1:nasv
fourdo_cd <- 0*1:nasv
meo5ds_cd <- 0*1:nasv
orb_cd <- 0*1:nasv

for(i in 1:nasv) {
  fourdo.g <-glm.nb(data[,13+i]~X4D0_pmol_g, data=data)
  fourdo_est[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 1]
  fourdo_std.error[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 3]
  fourdo_Zvalue[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 4]
  fourdo_p[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g)))), 4]
  fourdo_shapiro[i]<-shapiro.test(resid(fourdo.g))[2]$p.value
  fourdo_cd[i]<-sum(cooks.distance(fourdo.g)>0.3)

  meo5ds.g <-glm.nb(data[,13+i]~Me05DS_pmol_g, data=data)
  meo5ds_est[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 1]
  meo5ds_std.error[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 3]
  meo5ds_Zvalue[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 4]
  meo5ds_p[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g)))), 4]
```

```

meo5ds_shapiro[i]<-shapiro.test(resid(meo5ds.g))[2]$p.value
meo5ds_cd[i]<-sum(cooks.distance(meo5ds.g)>0.3)

orb.g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
orb_est[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 1]
orb_std.error[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 2]
orb_Zvalue[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 3]
orb_p[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g)))), 4]
orb_shapiro[i]<-shapiro.test(resid(orb.g))[2]$p.value
orb_cd[i]<-sum(cooks.distance(orb.g)>0.3)
}

RS_genus_res<-as.data.frame(cbind(
  fourdo_est, fourdo_std.error, fourdo_Zvalue, fourdo_p, fourdo_shapiro, fourdo_cd,
  meo5ds_est, meo5ds_std.error, meo5ds_Zvalue, meo5ds_p, meo5ds_shapiro, meo5ds_cd,
  orb_est, orb_std.error, orb_Zvalue, orb_p, orb_shapiro, orb_cd))

rownames(RS_genus_res)<-colnames(data[,14:dim(data)[2]])

Now, select significant model that has normaly distributed residual & no outliers and validate its p value
using permutation.

# 4do
RS_genus_4do_res<-(subset(RS_genus_res, fourdo_p<0.05&fourdo_shapiro>0.05&fourdo_cd<3))[1:6]
RS_genus_4do_res_name<-rownames(RS_genus_4do_res)
RS_genus_4do<-RS_genus[RS_genus_4do_res_name]

data=RS_genus_4do
res=RS_genus_4do_res
test=RS_genus_4do$X4DO_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_genus_4do_res$Perm.p<-Perm.p
RS_genus_4do_res_final<-subset(RS_genus_4do_res, Perm.p<0.05)

```

```

# MeO5DS
RS_genus_meo5ds_res<-(subset(RS_genus_res, meo5ds_p<0.05&meo5ds_shapiro>0.05&meo5ds_cd<3))[7:12]
RS_genus_meo5ds_res_name<-rownames(RS_genus_meo5ds_res)
RS_genus_meo5ds<-RS_genus[RS_genus_meo5ds_res_name]

data=RS_genus_meo5ds
res=RS_genus_meo5ds_res
test=RS_genus$MeO5DS_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_genus_meo5ds_res$Perm.p<-Perm.p
RS_genus_meo5ds_res_final<-subset(RS_genus_meo5ds_res, Perm.p<0.05)

# orobanchol
RS_genus_orb_res<-(subset(RS_genus_res, orb_p<0.05&orb_shapiro>0.05&orb_cd<3))[13:18]
RS_genus_orb_res_name<-rownames(RS_genus_orb_res)
RS_genus_orb<-RS_genus[RS_genus_orb_res_name]

data=RS_genus_orb
res=RS_genus_orb_res
test=RS_genus$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
}

```

```

  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_genus_orb_res$Perm.p<-Perm.p
RS_genus_orb_res_final<-subset(RS_genus_orb_res, Perm.p<0.05)

```

Combining final results in one table

```

RS_genus_4do_res_final$tested_genus<-rownames(RS_genus_4do_res_final)
RS_genus_meo5ds_res_final$tested_genus<-rownames(RS_genus_meo5ds_res_final)
RS_genus_orb_res_final$tested_genus<-rownames(RS_genus_orb_res_final)
RS_genus_4do_res_final<-RS_genus_4do_res_final%>%mutate(tested_SL="4D0")
RS_genus_meo5ds_res_final<-RS_genus_meo5ds_res_final%>%mutate(tested_SL="Me05DS")
RS_genus_orb_res_final<-RS_genus_orb_res_final%>%mutate(tested_SL="Orobanchol")

RS_genus_res_final<-force_bind(RS_genus_4do_res_final, RS_genus_meo5ds_res_final, RS_genus_orb_res_final)
new_col2<-c("est", "std.error", "Zvalue", "glm.P", "shapiro.p", "no.outliers", "Perm.P", "tested_genus", "tested_genus")
colnames(RS_genus_res_final)<-new_col2

RS_genus_res_final<-RS_genus_res_final[,c(9,8,1:7)]

```

3.2.2. Fo_RT

```

data=RT_genus
nasv = dim(data)[2]-13 # number of variables
fourdo_est <- 0*1:nasv
meo5ds_est <- 0*1:nasv
orb_est <- 0*1:nasv
fourdo_std.error <- 0*1:nasv
meo5ds_std.error <- 0*1:nasv
orb_std.error <- 0*1:nasv
fourdo_Zvalue <- 0*1:nasv
meo5ds_Zvalue <- 0*1:nasv
orb_Zvalue <- 0*1:nasv
fourdo_p <- 0*1:nasv
meo5ds_p <- 0*1:nasv
orb_p <- 0*1:nasv
fourdo_shapiro <- 0*1:nasv
meo5ds_shapiro <- 0*1:nasv
orb_shapiro <- 0*1:nasv
fourdo_cd<- 0*1:nasv
meo5ds_cd <- 0*1:nasv
orb_cd <- 0*1:nasv

for(i in 1:nasv) {
  fourdo.g <-glm.nb(data[,13+i]~X4D0_pmol_g, data=data)
  fourdo_est[i]<-coef(summary(fourdo.g))[grep1("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 1]

```

```

fourdo_std.error[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 1]
fourdo_Zvalue[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 3]
fourdo_p[i]<-coef(summary(fourdo.g))[grepl("X4D0_pmol_g$",row.names(coef(summary(fourdo.g))))], 4]
fourdo_shapiro[i]<-shapiro.test(resid(fourdo.g))[2]$p.value
fourdo_cd[i]<-sum(cooks.distance(fourdo.g)>0.3)

meo5ds.g <-glm.nb(data[,13+i]~Me05DS_pmol_g, data=data)
meo5ds_est[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 1]
meo5ds_std.error[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 2]
meo5ds_Zvalue[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 3]
meo5ds_p[i]<-coef(summary(meo5ds.g))[grepl("Me05DS_pmol_g$",row.names(coef(summary(meo5ds.g))))], 4]
meo5ds_shapiro[i]<-shapiro.test(resid(meo5ds.g))[2]$p.value
meo5ds_cd[i]<-sum(cooks.distance(meo5ds.g)>0.3)

orb.g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
orb_est[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 1]
orb_std.error[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 2]
orb_Zvalue[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 3]
orb_p[i]<-coef(summary(orb.g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(orb.g))))], 4]
orb_shapiro[i]<-shapiro.test(resid(orb.g))[2]$p.value
orb_cd[i]<-sum(cooks.distance(orb.g)>0.3)
}

RT_genus_res<-as.data.frame(cbind(
  fourdo_est, fourdo_std.error, fourdo_Zvalue,fourdo_p,fourdo_shapiro,fourdo_cd,
  meo5ds_est, meo5ds_std.error, meo5ds_Zvalue, meo5ds_p, meo5ds_shapiro,meo5ds_cd,
  orb_est, orb_std.error, orb_Zvalue, orb_p, orb_shapiro,orb_cd))

rownames(RT_genus_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normally distributed residual & no outliers and validate its p value using permutation.

```

# 4do
RT_genus_4do_res<-(subset(RT_genus_res, fourdo_p<0.05&fourdo_shapiro>0.05&fourdo_cd<3))[1:6]
RT_genus_4do_res_name<-rownames(RT_genus_4do_res)
RT_genus_4do<-RT_genus[RT_genus_4do_res_name]

data=RT_genus_4do
res=RT_genus_4do_res
test=RT_genus$X4D0_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
}

```

```

  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_genus_4do_res$Perm.p<-Perm.p
RT_genus_4do_res_final<-subset(RT_genus_4do_res, Perm.p<0.05)

# Me05DS
RT_genus_meo5ds_res<-(subset(RT_genus_res, meo5ds_p<0.05&meo5ds_shapiro>0.05&meo5ds_cd<3))[7:12]
RT_genus_meo5ds_res_name<-rownames(RT_genus_meo5ds_res)
RT_genus_meo5ds<-RT_genus[RT_genus_meo5ds_res_name]

data=RT_genus_meo5ds
res=RT_genus_meo5ds_res
test=RT_genus$Me05DS_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grep1("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_genus_meo5ds_res$Perm.p<-Perm.p
RT_genus_meo5ds_res_final<-subset(RT_genus_meo5ds_res, Perm.p<0.05)

# orobanchol
RT_genus_orb_res<-(subset(RT_genus_res, orb_p<0.05&orb_shapiro>0.05&orb_cd<3))[13:18]
RT_genus_orb_res_name<-rownames(RT_genus_orb_res)
RT_genus_orb<-RT_genus[RT_genus_orb_res_name]

data=RT_genus_orb
res=RT_genus_orb_res
test=RT_genus$orobanchol_pmol_g

```

```

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grep1("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_genus_orb_res$Perm.p<-Perm.p
RT_genus_orb_res_final<-subset(RT_genus_orb_res, Perm.p<0.05)

```

Combining final results in one table

```

RT_genus_4do_res_final$tested_genus<-rownames(RT_genus_4do_res_final)
RT_genus_meo5ds_res_final$tested_genus<-rownames(RT_genus_meo5ds_res_final)
RT_genus_orb_res_final$tested_genus<-rownames(RT_genus_orb_res_final)
RT_genus_4do_res_final<-RT_genus_4do_res_final%>%mutate(tested_SL="4D0")
RT_genus_meo5ds_res_final<-RT_genus_meo5ds_res_final%>%mutate(tested_SL="Me05DS")
RT_genus_orb_res_final<-RT_genus_orb_res_final%>%mutate(tested_SL="Orobanchol")

RT_genus_res_final<-force_bind(RT_genus_4do_res_final, RT_genus_meo5ds_res_final, RT_genus_orb_res_final)
colnames(RT_genus_res_final)<-new_col2

RT_genus_res_final<-RT_genus_res_final[,c(9,8,1:7)]

```

3.2.3. Final results table

```

RS_genus_res_final<-RS_genus_res_final%>%mutate(dataset="Rhizosphere")
RT_genus_res_final<-RT_genus_res_final%>%mutate(dataset="Roots")

genus_res_final<-rbind(RS_genus_res_final,RT_genus_res_final)
genus_res_final<-genus_res_final[,c(10,1:9)]
genus_res_final$Perm.P[genus_res_final$Perm.P<0.001] <- "p<0.001"
genus_res_final

```

```
genus_res_final
```

##	dataset	tested_SL	tested_genus
## 1	Rhizosphere	4D0	Clonostachys
## 2	Rhizosphere	Me05DS	Mortierella
## 3	Rhizosphere	Me05DS	Archaeosporales
## 4	Rhizosphere	Orobanchol	Burkholderia-Caballeronia-Paraburkholderia

## 5	Rhizosphere	Orobanchol				Granulicella
## 6	Rhizosphere	Orobanchol				Mortierella
## 7	Rhizosphere	Orobanchol				Nadsonia
## 8	Rhizosphere	Orobanchol				Saitozyma
## 9	Rhizosphere	Orobanchol				Solicoccozyma
## 10	Rhizosphere	Orobanchol				Umbelopsis
## 11	Rhizosphere	Orobanchol				Archaeosporales
## 12	Roots	4D0				Dyella
## 13	Roots	4D0				Trichoderma
## 14	Roots	4D0				Umbelopsis
## 15	Roots	Me05DS				Umbelopsis
## 16	Roots	Orobanchol				Chujaibacter
## 17	Roots	Orobanchol				Acaulospora
## 18	Roots	Orobanchol				Discosia
## 19	Roots	Orobanchol				Hyaloscypha
## 20	Roots	Orobanchol				Archaeosporales
## 21	Roots	Orobanchol				Diversisporales
##	est	std.error	Zvalue	glm.P	shapiro.p	no.outliers
## 1	-0.04744957	0.019660556	-2.413440	1.580273e-02	0.48722145	0
## 2	-0.01134350	0.004636392	-2.446622	1.442021e-02	0.05823702	1
## 3	0.02832734	0.012609046	2.246589	2.466631e-02	0.71648569	1
## 4	1.84528750	0.588184766	3.137258	1.705359e-03	0.14118739	1
## 5	1.98829250	0.766770149	2.593075	9.512203e-03	0.94562754	1
## 6	-2.12222610	0.380251096	-5.581118	2.389779e-08	0.75528638	1
## 7	-2.09689095	0.698729908	-3.001004	2.690914e-03	0.05625160	0
## 8	-1.71577327	0.525772316	-3.263339	1.101078e-03	0.50327019	0
## 9	-1.73465491	0.560969892	-3.092242	1.986505e-03	0.59358881	0
## 10	-3.22161753	1.101492819	-2.924774	3.447065e-03	0.11035075	0
## 11	3.40012824	1.239573864	2.742982	6.088411e-03	0.49392454	0
## 12	0.01493292	0.006859983	2.176816	2.949429e-02	0.90533818	0
## 13	-0.05276056	0.021838252	-2.415970	1.569336e-02	0.10933067	1
## 14	0.04490453	0.014516458	3.093353	1.979085e-03	0.96974818	1
## 15	0.02986105	0.012114540	2.464893	1.370540e-02	0.81456424	1
## 16	-1.20030161	0.562903399	-2.132340	3.297888e-02	0.06474408	0
## 17	4.42161088	2.049016627	2.157918	3.093417e-02	0.55851615	0
## 18	-6.18175863	1.811113823	-3.413236	6.419636e-04	0.18120634	0
## 19	-2.47342403	0.930032613	-2.659502	7.825616e-03	0.18038178	0
## 20	3.84717895	1.536629243	2.503648	1.229202e-02	0.30472999	0
## 21	4.42161088	2.049016627	2.157918	3.093417e-02	0.55851615	0
##	Perm.P					
## 1	0.039					
## 2	0.026					
## 3	0.046					
## 4	0.004					
## 5	0.016					
## 6	p<0.001					
## 7	0.005					
## 8	0.002					
## 9	0.007					
## 10	0.007					
## 11	0.007					
## 12	0.039					
## 13	0.038					
## 14	0.022					


```
## 15 0.047
## 16 0.038
## 17 0.043
## 18 0.001
## 19 0.019
## 20 0.024
## 21 0.043
```

3.3. EC with only orobanchol

3.3.1. Fo_RS

```
data=ForS_EC
nasv = dim(data)[2]-13 # number of variables

est <- 0*1:nasv
std.error <- 0*1:nasv
Zvalue <- 0*1:nasv
glm.p <- 0*1:nasv
shapiro.p <- 0*1:nasv
no.outliers <- 0*1:nasv

for(i in 1:nasv) {
  g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
  est[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 1]
  std.error[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 2]
  Zvalue[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 3]
  glm.p[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 4]
  shapiro.p[i]<-shapiro.test(resid(g))[2]$p.value
  no.outliers[i]<-sum(cooks.distance(g)>0.3)
}

RS_ec_res<-as.data.frame(cbind(est, std.error, Zvalue, glm.p, shapiro.p, no.outliers))

rownames(RS_ec_res)<-colnames(data[,14:dim(data)[2]])
```

Now, select significant model that has normaly distributed residual & no outliers and validate its p value using permutation.

```
RS_ec_res_sig<-subset(RS_ec_res, glm.p<0.01&shapiro.p>0.05&no.outliers<3)
RS_ec_res_sig_name<-rownames(RS_ec_res_sig)
RS_ec_res_sig_ec<-ForS_EC[RS_ec_res_sig_name]

data=RS_ec_res_sig_ec
res=RS_ec_res_sig
test=ForS_EC$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
```

```

for(j in 1:nasv){
  g <-glm.nb(data[,j]~sp)
  p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
}
Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RS_ec_res_sig$Perm.p<-Perm.p
RS_ec_res_sig_rc<-rownames_to_column(subset(RS_ec_res_sig, Perm.p<0.05))
RS_ec_res_des<-left_join(RS_ec_res_sig_rc,ec_des, by="rowname")
RS_ec_res_des<-RS_ec_res_des%>%mutate(dataset="Rhizosphere")

```

3.3.2. Fo_RT

```

data=FoRT_EC
nasv = dim(data)[2]-13 # number of variables
est <- 0*1:nasv
std.error <- 0*1:nasv
Zvalue <- 0*1:nasv
glm.p <- 0*1:nasv
shapiro.p <- 0*1:nasv
no.outliers <- 0*1:nasv

for(i in 1:nasv) {
  g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
  est[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g))))], 1]
  std.error[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g))))], 2]
  Zvalue[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g))))], 3]
  glm.p[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g))))], 4]
  shapiro.p[i]<-shapiro.test(resid(g))[2]$p.value
  no.outliers[i]<-sum(cooks.distance(g)>0.3)
}

RT_ec_res<-as.data.frame(cbind(est, std.error, Zvalue, glm.p, shapiro.p, no.outliers))
rownames(RT_ec_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normally distributed residual & no outliers and validate its p value using permutation.

```

RT_ec_res_sig<-subset(RT_ec_res, glm.p<0.01&shapiro.p>0.05&no.outliers<3)
RT_ec_res_sig_name<-rownames(RT_ec_res_sig)
RT_ec_res_sig_ec<-FoRT_EC[RT_ec_res_sig_name]

data=RT_ec_res_sig_ec
res=RT_ec_res_sig
test=FoRT_EC$orobanchol_pmol_g

```

```

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g))))], 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_ec_res_sig$Perm.p<-Perm.p
RT_ec_res_sig_rc<-rownames_to_column(subset(RT_ec_res_sig, Perm.p<0.05))
RT_ec_res_des<-left_join(RT_ec_res_sig_rc,ec_des, by="rowname")
RT_ec_res_des<-RT_ec_res_des%>%mutate(dataset="roots")

```

3.3.3. Final results in one table

```

ec_res_final<-rbind(RS_ec_res_des, RT_ec_res_des)
ec_res_final<-ec_res_final[,c(10,1,9,2:8)]
ec_res_final<-ec_res_final %>% rename( tested_EC = rowname)
ec_res_final$Perm.p[ec_res_final$Perm.p<0.001] <- "p<0.001"
ec_res_final[1:5,1:10]

```

```
ec_res_final[1:5,1:10]
```

##	dataset	tested_EC	description	est	std.error
## 1	Rhizosphere	EC.1.1.1.11	D-arabinitol 4-dehydrogenase	1.5588457	0.4510438
## 2	Rhizosphere	EC.1.1.1.122	D-threo-aldose 1-dehydrogenase	1.3481292	0.3531108
## 3	Rhizosphere	EC.1.1.1.169	2-dehydropantoate 2-reductase	0.5116984	0.1849320
## 4	Rhizosphere	EC.1.1.1.173	L-rhamnose 1-dehydrogenase	1.3296184	0.4990864
## 5	Rhizosphere	EC.1.1.1.18	Inositol 2-dehydrogenase	1.3180703	0.4902519
##	Zvalue	glm.p	shapiro.p	no.outliers	Perm.p
## 1	3.456085	0.0005480816	0.8704957	0	0.004
## 2	3.817864	0.0001346119	0.2225518	0	p<0.001
## 3	2.766955	0.0056582591	0.3964631	0	0.006
## 4	2.664105	0.0077193551	0.6265064	0	0.009
## 5	2.688557	0.0071761506	0.6609961	0	0.014

3.4. pathway with only orobanchol

3.4.1. Fo_RS

```

data=FoRS_PW
nasv = dim(data)[2]-13 # number of variables

```

```

est <- 0*1:nasv
std.error <- 0*1:nasv
Zvalue <- 0*1:nasv
glm.p <- 0*1:nasv
shapiro.p <- 0*1:nasv
no.outliers <- 0*1:nasv

for(i in 1:nasv) {
  g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
  est[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 1]
  std.error[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 2]
  Zvalue[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 3]
  glm.p[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 4]
  shapiro.p[i]<-shapiro.test(resid(g))[2]$p.value
  no.outliers[i]<-sum(cooks.distance(g)>0.3)
}

RS_pw_res<-as.data.frame(cbind(est, std.error, Zvalue, glm.p, shapiro.p, no.outliers))

rownames(RS_pw_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normally distributed residual & no outliers and validate its p value using permutation.

```

RS_pw_res_sig<-subset(RS_pw_res, glm.p<0.01&shapiro.p>0.05&no.outliers<3)
RS_pw_res_sig_name<-rownames(RS_pw_res_sig)
RS_pw_res_sig_path<-FoRS_PW[RS_pw_res_sig_name]

data=RS_pw_res_sig_path
res=RS_pw_res_sig
test=FoRS_PW$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
  }
  Perm.table[,i]<-p
}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

```

```

RS_pw_res_sig$Perm.p<-Perm.p
RS_pw_res_sig_rc<-rownames_to_column(subset(RS_pw_res_sig, Perm.p<0.05))
RS_pw_res_des<-left_join(RS_pw_res_sig_rc,path_des, by="rowname")
RS_pw_res_des<-RS_pw_res_des%>%mutate(dataset="Rhizosphere")

```

3.3.2. Fo_RT

```

data=FoRT_PW
nasv = dim(data)[2]-13 # number of variables
est <- 0*1:nasv
std.error <- 0*1:nasv
Zvalue <- 0*1:nasv
glm.p <- 0*1:nasv
shapiro.p <- 0*1:nasv
no.outliers <- 0*1:nasv

for(i in 1:nasv) {
  g <-glm.nb(data[,13+i]~orobanchol_pmol_g,data=data)
  est[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 1]
  std.error[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 2]
  Zvalue[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 3]
  glm.p[i]<-coef(summary(g))[grepl("orobanchol_pmol_g$",row.names(coef(summary(g)))), 4]
  shapiro.p[i]<-shapiro.test(resid(g))[2]$p.value
  no.outliers[i]<-sum(cooks.distance(g)>0.3)
}

RT_pw_res<-as.data.frame(cbind(est, std.error, Zvalue, glm.p, shapiro.p, no.outliers))
rownames(RT_pw_res)<-colnames(data[,14:dim(data)[2]])

```

Now, select significant model that has normaly distributed residual & no outliers and validate its p value using permutation.

```

RT_pw_res_sig<-subset(RT_pw_res, glm.p<0.01&shapiro.p>0.05&no.outliers<3)
RT_pw_res_sig_name<-rownames(RT_pw_res_sig)
RT_pw_res_sig_path<-FoRT_PW[RT_pw_res_sig_name]

data=RT_pw_res_sig_path
res=RT_pw_res_sig
test=FoRT_PW$orobanchol_pmol_g

nasv = dim(data)[2] # number of variables
p <- 0*1:nasv
nPerm<-1000
Perm.table<-data.frame(matrix(NA,nasv,nPerm))

for(i in 1:nPerm){
  sp<-sample(test)
  for(j in 1:nasv){
    g <-glm.nb(data[,j]~sp)
    p[j]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
  }
  Perm.table[,i]<-p
}

```

```

}
rownames(Perm.table)<-colnames(data)
Perm.p<-0*1:nasv
for(j in 1:nasv){
  Perm.p[j]<-sum(Perm.table[j,]<res[j,4])/nPerm
}

RT_pw_res_sig$Perm.p<-Perm.p
RT_pw_res_sig_rc<-rownames_to_column(subset(RT_pw_res_sig, Perm.p<0.05))
RT_pw_res_des<-left_join(RT_pw_res_sig_rc,path_des, by="rowname")
RT_pw_res_des<-RT_pw_res_des%>%mutate(dataset="roots")

```

3.4.3. Final results in one table

```

pw_res_final<-rbind(RS_pw_res_des, RT_pw_res_des)
pw_res_final<-pw_res_final[,c(10,1,9,2:8)]
pw_res_final<-pw_res_final %>% rename( tested_EC = rowname)
pw_res_final$Perm.p[pw_res_final$Perm.p<0.001] <- "p<0.001"
pw_res_final[1:5,1:10]

```

```
pw_res_final[1:5,1:10]
```

```

##      dataset                                tested_EC
## 1 Rhizosphere 3_HYDROXYPHENYLACETATE_DEGRADATION_PWY
## 2 Rhizosphere                                AST_PWY
## 3 Rhizosphere          BRANCHED_CHAIN_AA_SYN_PWY
## 4 Rhizosphere      CATECHOL_ORTHO_CLEAVAGE_PWY
## 5 Rhizosphere                                COA_PWY
##
##      description      est std.error
## 1      4-hydroxyphenylacetate degradation  2.2761924 0.78214477
## 2      L-arginine degradation II (AST pathway)  1.6538711 0.42374395
## 3 superpathway of branched amino acid biosynthesis -0.1704086 0.06101935
## 4      catechol degradation to &beta;-ketoadipate  1.5207408 0.50744485
## 5      coenzyme A biosynthesis I -0.2107005 0.07810090
##      Zvalue      glm.p shapiro.p no.outliers  Perm.p
## 1  2.910193 3.612054e-03 0.9964203          1  0.005
## 2  3.902996 9.500909e-05 0.9356856          0 p<0.001
## 3 -2.792698 5.227051e-03 0.7894962          0  0.006
## 4  2.996859 2.727766e-03 0.8531441          2  0.004
## 5 -2.697799 6.979964e-03 0.3994041          0  0.01

```

```
sessionInfo()
```

```

## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C

```

```

## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] stringr_1.4.0 tidyr_1.1.2  tibble_3.0.4 dplyr_1.0.2  MASS_7.3-53
## [6] ggplot2_3.3.2
##
## loaded via a namespace (and not attached):
## [1] knitr_1.31      magrittr_1.5      tidyselect_1.1.0 munsell_0.5.0
## [5] colorspace_1.4-1 R6_2.4.1          rlang_0.4.10     tools_4.0.3
## [9] grid_4.0.3      gtable_0.3.0      xfun_0.21        withr_2.3.0
## [13] htmltools_0.5.1.1 ellipsis_0.3.1    yaml_2.2.1       digest_0.6.25
## [17] lifecycle_0.2.0 crayon_1.3.4      purrr_0.3.4      vctrs_0.3.4
## [21] glue_1.4.2      evaluate_0.14     rmarkdown_2.7    stringi_1.5.3
## [25] compiler_4.0.3  pillar_1.4.6     generics_0.0.2   scales_1.1.1
## [29] pkgconfig_2.0.3

```