MasterPip

Chunk 1 - Load libraries for further analyses

```
library(ggplot2)
library(phyloseq)
library(plyr)
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-2
```

```
library("biomformat")
library(reshape2)
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
library(ggvegan)
library(ggordiplots)
```

```
## Loading required package: formatR
```

```
setwd("~/PIP2018/")

set.seed(8675309)

timecolors=c("#551A8B", "#FF4500", "#E69F00", "#E69F00")
boolcolors=c("salmon", "turquoise4")
```

Chunk 2: Generation of PIP figures and tables

```
#Piece of code for loading taxonomy.tsv / modules.pcl / pathways.pcl and calculating bad samples.
#In our final version, let's calculate E. coli IQR for each age group, and filter samples with > 1.5 IQ
NZGL_taxonomy<-import_qiime_sample_data("~/PIP2018/primary_data/taxonomy.tsv")
  # the imported taxonomy data should have each sample as a row and each variable or taxonomy as a colu
Taxonomy_filter_file<-NZGL_taxonomy # make a copy
  #First make a plot of unfiltered taxonomy data, showing E coli abundance for each age group.
NZGL_taxonomy$time<-as.factor(NZGL_taxonomy$time) # to separate boxplot by different age category, type
### SEE PLOT 1: Supplemental 1: Abundance of E. coli x stratified by age in unfiltered data

# Then filter those samples out of all data, and use these data for every downstream analysis.

# select the useful part and find the interquartile range for E. coli, filter out samples that E. coli
Taxonomy_filter_file$E_coli<-NZGL_taxonomy$k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o__Ente
# Split the dataset by time/age
E_coli_abundance_AtBirth<-subset(Taxonomy_filter_file, time==0)
E_coli_abundance_3_month<-subset(Taxonomy_filter_file, time==3)
E_coli_abundance_12_month<-subset(Taxonomy_filter_file, time==12)
E_coli_abundance_24_month<-subset(Taxonomy_filter_file, time==24)
# Calculate IQR by each time
E_coli_abundance_IQR_AtBirth<-IQR(E_coli_abundance_AtBirth$E_coli)
```

```
E_coli_abundance_IQR_3_month<-IQR(E_coli_abundance_3_month$E_coli)
E_coli_abundance_IQR_12_month<-IQR(E_coli_abundance_12_month$E_coli)
E_coli_abundance_IQR_24_month<-IQR(E_coli_abundance_24_month$E_coli)
# Filter the whole dataset at each time on E.coli > 1.5IQR
Taxonomy_filtered_AtBirth<-subset(E_coli_abundance_AtBirth, E_coli<=(1.5*E_coli_abundance_IOR_AtBirth))
Taxonomy_filtered_3_month<-subset(E_coli_abundance_3_month, E_coli<=(1.5*E_coli_abundance_IQR_3_month))
Taxonomy_filtered_12_month<-subset(E_coli_abundance_12_month, E_coli<=(1.5*E_coli_abundance_IQR_12_montl
Taxonomy_filtered_24_month<-subset(E_coli_abundance_24_month, E_coli<=(1.5*E_coli_abundance_IQR_24_montl
# a fully filtered data from each timepoint combined into one dataset
Taxonomy_filtered<-rbind(Taxonomy_filtered_AtBirth,Taxonomy_filtered_3_month,Taxonomy_filtered_12_month
#write.csv(Taxonomy_filtered, "~/pip-resubmit/derived-data/taxonomy-filtered.csv")
```

Chunk3: Plots regarding the relationships between E. coli abundance & age, E. coli abundance & time at room temperature, and time of storage of samples (Sup 1)

```
NZGL_taxonomy$time<-as.factor(NZGL_taxonomy$time) # to separate boxplot by different age category, type
#Plot the Abundance of Escherichia at different time points
a<-ggplot(NZGL_taxonomy, aes(time, NZGL_taxonomy$k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o
Metadata<-read.csv("~/PIP2018/primary_data/metadata.csv", header = TRUE) # load csv file

#Plot the Duration of storage of study fecal samples at room temperature before freezing
Molten_Meta<-melt(Metadata, id.vars = "Studyid", measure.vars = c("ftime_0", "ftime_3", "ftime_12", "ft
colnames(Molten_Meta)[2]<-"time"
Molten_Meta$time<-as.character(Molten_Meta$time)
Molten_Meta$time[Molten_Meta$time == "ftime_0"] <- "0"
Molten_Meta$time[Molten_Meta$time == "ftime_3"] <- "3"
Molten_Meta$time[Molten_Meta$time == "ftime_12"] <- "12"
Molten_Meta$time[Molten_Meta$time == "ftime_24"] <- "24"
Molten_Meta$time<-as.factor(Molten_Meta$time)

IDs<-read.table("~/PIP2018/primary_data/ids.txt", header = TRUE)
colnames(IDs)[2]<-"Studyid"
colnames(IDs)[3]<-"time"

Taxonomy<-import_qiime_sample_data("~/PIP2018/primary_data/taxonomy.tsv")
Taxonomy<-Taxonomy[,c(-1)]
select.var<-c("time", "Studyid", "k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o__Enterobacteri
Escherichia<-Taxonomy[,select.var]
Escherichia<-as.data.frame(Escherichia) # converting columns into rows

Escherichia$Otago.ID<-row.names(Escherichia) # assign otago.id to the dataset
Escherichia_ID<-merge(Escherichia, IDs, by=c("Otago.ID","Studyid","time"))
summary(Escherichia_ID)
```

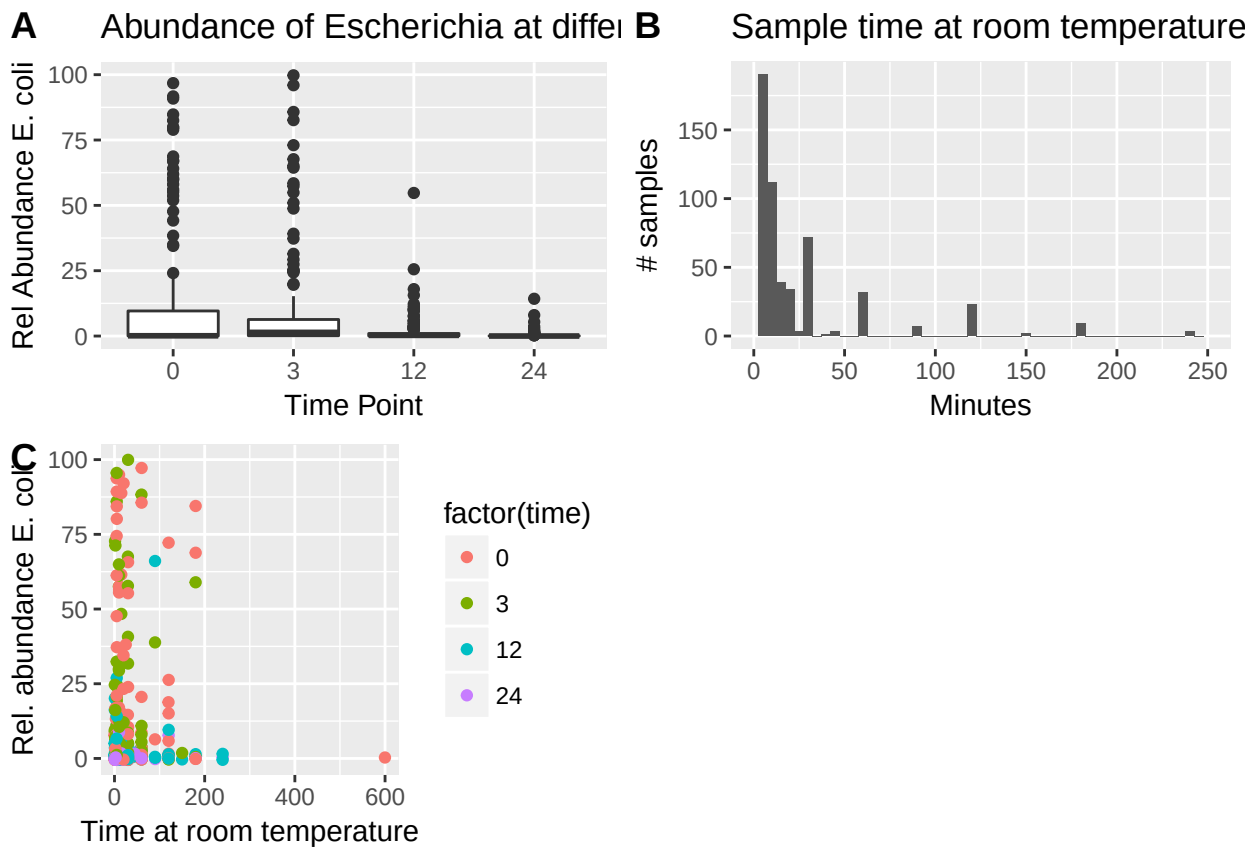
```
##    Otago.ID           Studyid         time
##  Length:645         P085   :  5   Min.   : 0.000
##  Class :character   P166   :  5   1st Qu.: 0.000
##  Mode  :character   P651   :  5   Median : 3.000
##                     P006   :  4   Mean   : 8.828
##                     P007   :  4   3rd Qu.:12.000
##                     P012   :  4   Max.   :24.000
##                     (Other):618
##  k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o__Enterobacteriales.f__Enterobacteriaceae.g__
##  Min.   : 0.00000
```

```
##  1st Qu.: 0.02118
##  Median : 0.28640
##  Mean   : 6.58298
##  3rd Qu.: 2.98484
##  Max.   :99.74987
##
```

```
Escherichia_Meta<-merge(Escherichia_ID, Molten_Meta, by=c("Studyid","time"))
colnames(Escherichia_Meta)[4]<-"Escherichia_growth"
colnames(Escherichia_Meta)[5]<-"Measurement_of_time"

b<-ggplot(Escherichia_Meta, aes(Measurement_of_time))+geom_histogram(stat = "bin", binwidth=5)+xlim(0,25

c<-ggplot(Escherichia_Meta, aes(color=factor(time), x=Measurement_of_time, y=Escherichia_growth)) +  geo

ggarrange(a, b, c, labels=c("A", "B", "C", "D"), ncol=2, nrow=2)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



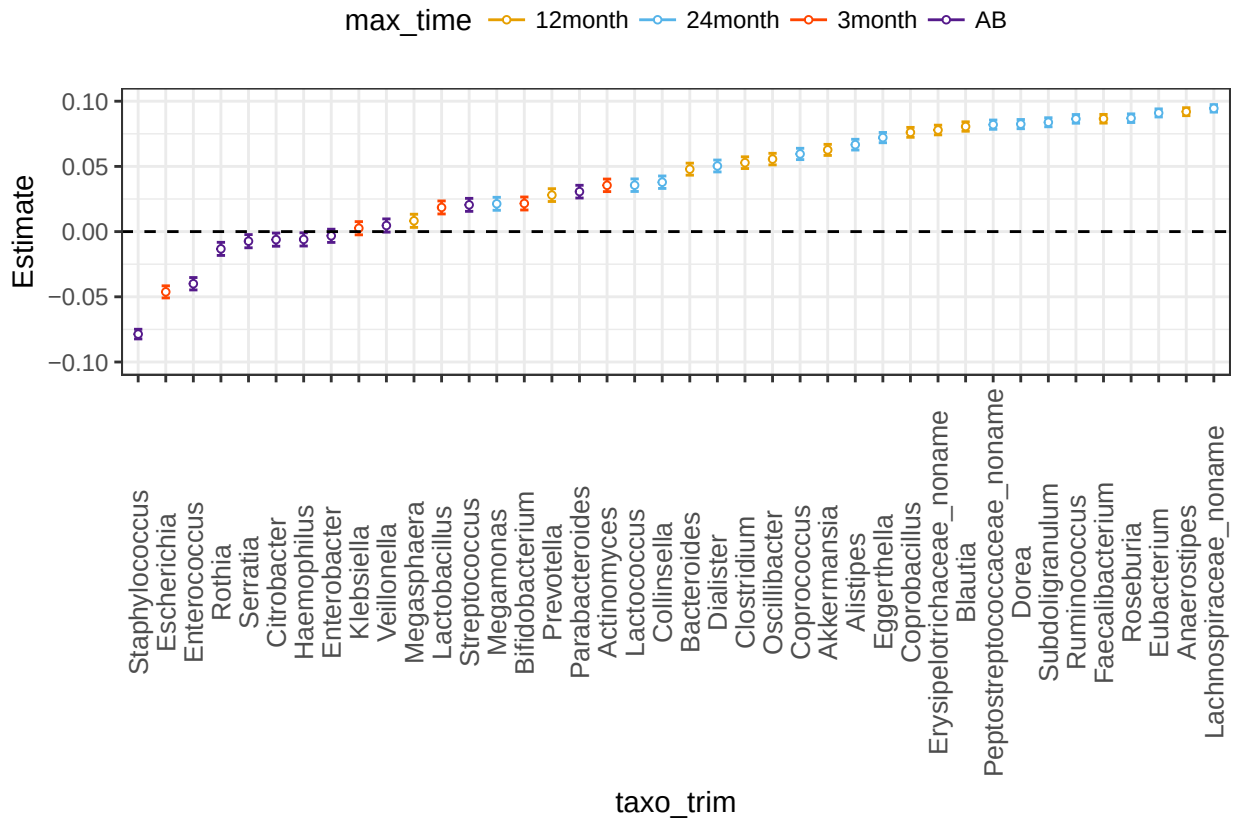

```
ggsave("~/PIP2018/results/SupFig1.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

Color code for the four chosen colors are: 12_month: yellow (#E69F00) 24_month: light blue(#56B4E9) 3_month: bright orange(#FF4500) At Birth(AB): dark purple (#551A8B)

Chunk4: Generate Figure 2:

```
  # remove the metadata part and left only taxonomy abundance data
Taxonomy_filtered_num<-Taxonomy_filtered[,c(-1:-28)]
  # to solve the -infinity problem when logging, add a small value to all datapoint that is 0
Taxonomy_filtered_num[Taxonomy_filtered_num==0]<-10e-8

# 1) log10 and normalise the taxonomy abundance
Log10_Taxonomy_filtered_num<-sapply(Taxonomy_filtered_num, function(x) log10(as.numeric(as.character(x))
row.names(Log10_Taxonomy_filtered_num)<-row.names(Taxonomy_filtered_num)
Norm_log10_abundance<-as.data.frame(scale(Log10_Taxonomy_filtered_num))


# 2) Glom to genera
  ## select any taxo names that the taxo has reached genus level
Norm_filtered_taxonomy_abundance_select<-Norm_log10_abundance[,grep("g__",colnames(Norm_log10_abundance)
  ## select any taxo names that has reached species level
NZGL_taxonomy_select_t_col<-colnames(Norm_log10_abundance[,grep("s__",colnames(Norm_log10_abundance))])
  ## select rows that has reached genus level but not species level
Norm_filtered_taxonomy_g<-Norm_filtered_taxonomy_abundance_select[,setdiff(colnames(Norm_filtered_taxono
  ## Only select genera that have data
Genera_sum<-as.data.frame(apply(Norm_filtered_taxonomy_g, 2, sum))
colnames(Genera_sum)<-"sum"
Genera_sum<-subset(Genera_sum, Genera_sum$sum!=0)
Genera<-rownames(Genera_sum)


# 3) Fit each genus to the linear model model<-lm(bug~time,data=bugdata)
  ## assign time for linear model
Norm_filtered_taxonomy_g$time<-Taxonomy_filtered$time[match(rownames(Norm_filtered_taxonomy_g), Taxonomy
  ## create an empty dataframe for saving the estimates and p-values
temp<-NULL
T1<-list()
  ## Linear model for each genus, this only apply to genus has meaningful data (Not 0)
for (a in Genera) {
  T<-summary(lm(Norm_filtered_taxonomy_g[,a]~Norm_filtered_taxonomy_g$time))
  T2<-as.data.frame(t(T[[4]][2,]))
  T2$taxo<-colnames(Norm_filtered_taxonomy_g[a])
  T1[[a]]<-T2
  temp<-do.call(rbind, T1)
}
  ## Reduce the length of taxo names to leave only genera names
temp$taxo_trim<-gsub("k__\\D+.p__\\D+.c\\D+.o\\D+.f__\\D+.g__(\\D+)", "\\1", temp$taxo)
# sort taxo column by the correspondance estimate values to make figure visually vetter
temp$taxo_trim<-factor(temp$taxo_trim, levels = temp$taxo_trim[order(temp$Estimate)])



# 4) For each bug genus x time, calculate its mean
  ## Figure out the most abundant genera
  ## Select any taxo names that has reached genus level
taxonomy_abundance_select<-Taxonomy_filtered_num[,grep("g__",colnames(Taxonomy_filtered_num))]
  ## select any taxo names that has reached species level
taxonomy_select_t_col<-Taxonomy_filtered_num[,grep("s__",colnames(Taxonomy_filtered_num))]
```

```
  ## substract taxonomy_select_t_col from taxonomy_abundance_select
taxonomy_genera<-taxonomy_abundance_select[,setdiff(colnames(taxonomy_abundance_select),colnames(taxonom
genera<-taxonomy_genera
  ## summarise dataset to get mean abundance for each genus
taxonomy_genera_sum1<-as.data.frame(sort(-apply(taxonomy_genera, 2, mean)))
  ## choose taxa based on the top 40 by mean
top_abundant_40_dataset<-temp[match(row.names(taxonomy_genera_sum1)[1:40], temp$taxo),]
top_abundant_40_dataset$taxo_trim<-factor(top_abundant_40_dataset$taxo_trim, levels = top_abundant_40_da

  ## save the genera names for further use
T40_genera<-row.names(top_abundant_40_dataset)

  ## find out which timepoint the taxa is most abundant for the top 40 genera
  ## make a copy of the dataset need for the analyses
test<-Taxonomy_filtered
  # separate the dataset by timepoint
test_AB<-subset(test, time==0)
test_3m<-subset(test, time==3)
test_12m<-subset(test, time==12)
test_24m<-subset(test, time==24)
  ## find out the mean abundance for genera at each time point
test_AB_mean<-as.data.frame(-apply(test_AB[,c(-1:-28)], 2, mean))
colnames(test_AB_mean)<-"AB"
test_3m_mean<-as.data.frame(-apply(test_3m[,c(-1:-28)], 2, mean))
colnames(test_3m_mean)<-"3month"
test_12m_mean<-as.data.frame(-apply(test_12m[,c(-1:-28)], 2, mean))
colnames(test_12m_mean)<-"12month"
test_24m_mean<-as.data.frame(-apply(test_24m[,c(-1:-28)], 2, mean))
colnames(test_24m_mean)<-"24month"
  ## find out which taxa is most abundant for the 40 genera
  ## combine dataset for comparison
test_mean_alltime<-cbind(test_AB_mean,test_3m_mean,test_12m_mean,test_24m_mean)
test_mean_alltime<-(-test_mean_alltime) # get rid of the minus sign I added before
  ## compare and pick up the time point with maximun mean for each genus (for coding, that means for eac
##======This piece of code should be used very carefully, due to the ties.method
  test_mean_alltime$max_time_randome<-colnames(test_mean_alltime)[max.col(test_mean_alltime)]
  test_mean_alltime$max_time_first<-colnames(test_mean_alltime[,1:4])[max.col(test_mean_alltime[,1:4], t
  test_mean_alltime$max_time_last<-colnames(test_mean_alltime[,1:4])[max.col(test_mean_alltime[,1:4], ti
  ##==== Had a look and using all three methods gave the same result, passed the checking
  # choose the 40 genera we are interested and assign this to top_abundant_40_dataset(data for figure)
top_abundant_40_dataset$max_time<-test_mean_alltime$max_time_randome[match(row.names(top_abundant_40_dat
top_abundant_40_dataset$max_time<-as.factor(top_abundant_40_dataset$max_time)

ggplot(top_abundant_40_dataset, aes(taxo_trim, Estimate, color=max_time)) + geom_errorbar(aes(ymin=top_a
```




```
ggsave("~/PIP2018/results/Fig2A.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

Chunk5: Generate Figure 2B

```
Module<-import_qiime_sample_data("~/PIP2018/primary_data/modules.pcl")
Module<-as.data.frame(t(Module))
```

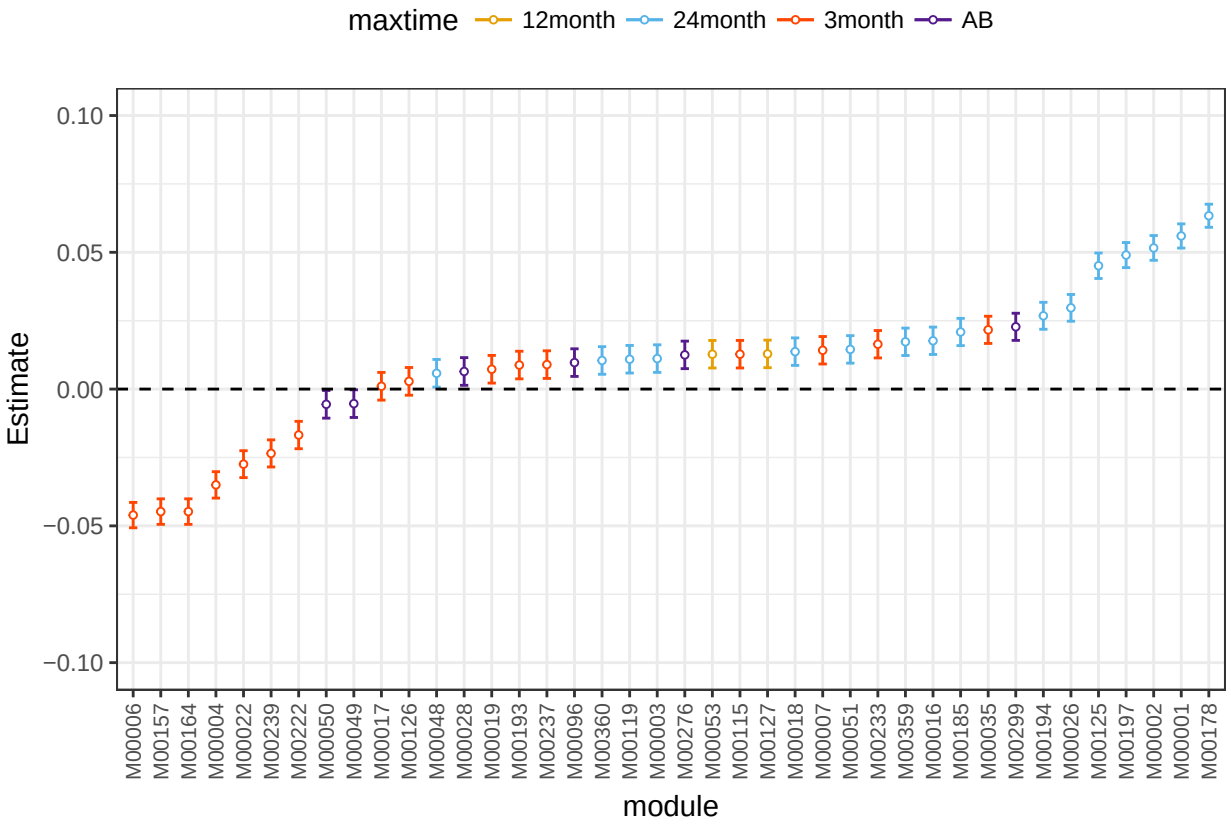
```
## Warning in class(X) <- NULL: Setting class(x) to NULL; result will no
## longer be an S4 object
```

```
# filter out the samples that have E.coli>1.5 IQR based on the the filtered taxonomy file
Module_filtered<-Module[rownames(Module)%in%Taxonomy_filtered$Sample,]
Module_filtered_num<-Module_filtered[,c(-1:-27)]
# now all the data are factors need to change them to numbers
Module_filtered_num[]<- lapply(Module_filtered_num, function(x){as.numeric(as.character(x))})
# add a small number to data where 0 could cause error for analyses
Module_filtered_num[Module_filtered_num==0]<-10e-8
# log and normalise data
Log10_Module_filtered_num<-sapply(Module_filtered_num, function(x) log10(x))
row.names(Log10_Module_filtered_num)<-row.names(Module_filtered_num)
Norm_log10_Module_abundance<-as.data.frame(scale(Log10_Module_filtered_num))


# select the modules that contain data
M_names<-as.data.frame(apply(Norm_log10_Module_abundance, 2, sum))
colnames(M_names)<-"sum"
```

```
M_names<-subset(M_names, M_names$sum!=0)
M_names<-rownames(M_names)

# assign time
Norm_log10_Module_abundance$time<-Taxonomy_filtered$time[match(rownames(Norm_log10_Module_abundance), Ta
# create an empty file for saving the results later
Module_rainbow<-NULL
T1<-list()
  ## Linear model for each genus, this only apply to genus has meaningful data (Not 0)
for (a in M_names) {
  T<-summary(lm(Norm_log10_Module_abundance[,a]~Norm_log10_Module_abundance$time))
  T2<-as.data.frame(t(T[[4]][2,]))
  T2$module<-colnames(Norm_log10_Module_abundance[a])
  T1[[a]]<-T2
  Module_rainbow<-do.call(rbind, T1)
}

# rainbow version of module*time, ordered by Estimate value
Module_rainbow$module<-factor(Module_rainbow$module, levels = Module_rainbow$module[order(Module_rainbou


# module*time, ordered by Estimate value and colored by most abundant timepoint/age
  # Find out the for each module, the max mean abundance timpoint/age
    # Note that I used the original value instead of the log normalised value
module_AB<-subset(Module_filtered, time==0)
module_AB<-module_AB[,c(-1:-27)]
module_AB[]<-lapply(module_AB,  function(x){as.numeric(as.character(x))})

module_3m<-subset(Module_filtered, time==3)
module_3m<-module_3m[,c(-1:-27)]
module_3m[]<-lapply(module_3m, function(x){as.numeric(as.character(x))})
module_12m<-subset(Module_filtered, time==12)
module_12m<-module_12m[,c(-1:-27)]
module_12m[]<-lapply(module_12m, function(x){as.numeric(as.character(x))})
module_24m<-subset(Module_filtered, time==24)
module_24m<-module_24m[,c(-1:-27)]
module_24m[]<-lapply(module_24m, function(x){as.numeric(as.character(x))})
  ## find out the mean abundance for genera at each time point
module_AB_mean<-as.data.frame(apply(module_AB, 2, mean))
colnames(module_AB_mean)<-"AB"
module_3m_mean<-as.data.frame(apply(module_3m, 2, mean))
colnames(module_3m_mean)<-"3month"
module_12m_mean<-as.data.frame(apply(module_12m, 2, mean))
colnames(module_12m_mean)<-"12month"
module_24m_mean<-as.data.frame(apply(module_24m, 2, mean))
colnames(module_24m_mean)<-"24month"
module_all_time<-cbind(module_AB_mean, module_3m_mean, module_12m_mean, module_24m_mean)
module_all_time$maxtime<-colnames(module_all_time)[apply(module_all_time,1,which.max)]

# assign the maxitime to Module_rainbow
Module_rainbow$maxtime<-module_all_time$maxtime[match(rownames(Module_rainbow),rownames(module_all_time

# for Module_filtered_num file before adding the fake 1e-7,
```

```
  # calculate module presence in all samples
Module_filtered_num1<-Module_filtered[,c(-1:-27)]
Module_filtered_num1[]<- lapply(Module_filtered_num1, function(x){as.numeric(as.character(x))})
module_presence<-NULL
for (i in 1:ncol(Module_filtered_num1)) {
  # create a temp file. For each column/module, calculate the module presence
  temp<-length(Module_filtered_num1[Module_filtered_num1[,i]>0,i])/nrow(Module_filtered_num1)
  module_presence<-rbind(module_presence, temp)
}
  module_presence<-as.data.frame(module_presence)
  colnames(module_presence)<-"Module_presence"
  module_presence$module<-colnames(Module_filtered_num1)
  rownames(module_presence)<-NULL
# select the modules that have presence higher than 10%
Abundant_module_presence<-module_presence[module_presence$Module_presence>=0.1,]
Abundant_presence_module_filtered<-Module_filtered_num1[,Abundant_module_presence$module]# 100 modules
# calculate and select the top 40 abundant modules from the module*time figure made for all modules
Top_40_abundant_module_names<-as.data.frame(sort(apply(Abundant_presence_module_filtered, 2, mean), dec
Top_40_abundant_module_names$module<-rownames(Top_40_abundant_module_names)
Top_40_abundant_module_names<-as.data.frame(Top_40_abundant_module_names[1:40,])
Top_40_abundant_modules<-Module_rainbow[Module_rainbow$module%in%c(rownames(Top_40_abundant_module_name

# plot the top 40 modules
ggplot(Top_40_abundant_modules, aes(module, Estimate,colour=maxtime))+geom_line()+geom_errorbar(aes(ymi
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```




```
ggsave("~/PIP2018/results/Fig2B.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```

```
#write.csv(Module_filtered, "~/pip-resubmit/derived-data/modules-filtered.csv")
```

Chunk6: Filter pathways the same way modules & taxa were filtered

```
Pathways<-import_qiime_sample_data("~/PIP2018/primary_data/pathways.pcl")
Pathways<-as.data.frame(t(Pathways))
```

```
## Warning in class(X) <- NULL: Setting class(x) to NULL; result will no
## longer be an S4 object
```

```
# filter out the samples that have E.coli>1.5 IQR based on the the filtered taxonomy file
Pathways_filtered<-Pathways[rownames(Pathways)%in%Taxonomy_filtered$Sample,]
#write.csv(Pathways_filtered, "~/pip-resubmit/derived-data/pathways-filtered.csv")
```

Chunk7: Make Figure 1 - Time & c-section stratified by time

```
# Create genera with no pseudocounts

# remove the metadata part and left only taxonomy abundance data
Taxonomy_filtered_num<-Taxonomy_filtered[,c(-1:-28)]
g1<-Taxonomy_filtered_num[,grep("g__",colnames(Taxonomy_filtered_num))]
  ## select any taxo names that has reached species level
```

```
g2<-colnames(Taxonomy_filtered_num[,grep("s__",colnames(Taxonomy_filtered_num))])
  ## select rows that has reached genus level but not species level
my_genera<-Taxonomy_filtered_num[,setdiff(colnames(g1),g2)]


#Are c-section, time, eczema, studygroup significant contributors to beta diversity? (in full data)
taxonomy_genera<-my_genera
meta<-Taxonomy_filtered[,1:28]
meta<-as.data.frame(as.matrix(meta))
#Overall permanova effects
foo<-adonis(taxonomy_genera~time + caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months

print(foo$aov.tab)
```

```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                               Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## time                           3    29.750  9.9167  48.809 0.21908  0.001
## caesar                         1     1.961  1.9608   9.651 0.01444  0.001
## eczema_by_2_years              1     0.334  0.3336   1.642 0.00246  0.099
## studygroup                     2     0.348  0.1738   0.856 0.00256  0.580
## Antibiotics_before_3_months    1     0.113  0.1134   0.558 0.00084  0.808
## Any_smoking_during_pregnancy   1     0.146  0.1464   0.721 0.00108  0.653
## Any_pet_at_birth               1     0.139  0.1386   0.682 0.00102  0.692
## Residuals                    507   103.008  0.2032         0.75854
## Total                        517   135.799                 1.00000
##
## time                          ***
## caesar                        ***
## eczema_by_2_years             .
## studygroup
## Antibiotics_before_3_months
## Any_smoking_during_pregnancy
## Any_pet_at_birth
## Residuals
## Total
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#With time as strata
foo<-adonis(taxonomy_genera~time + caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months

print(foo$aov.tab)
```

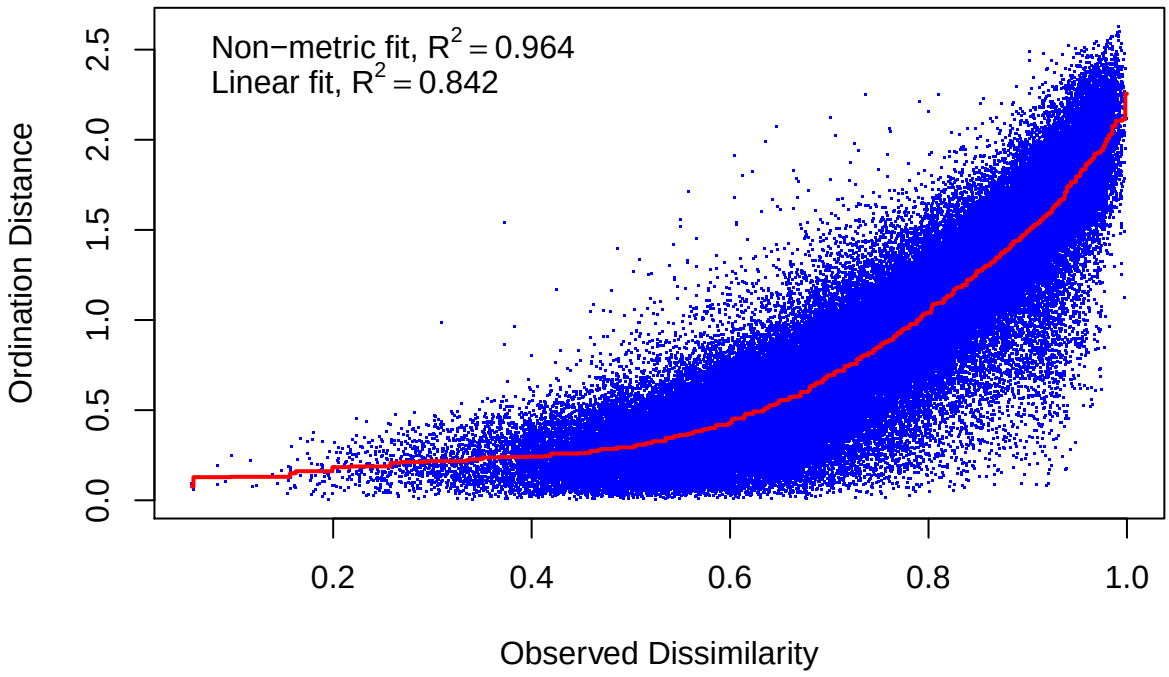
```
## Blocks:  strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                               Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
```

```
## time                           3    29.750  9.9167  48.809 0.21908  0.001
## caesar                         1     1.961  1.9608   9.651 0.01444  0.001
## eczema_by_2_years              1     0.334  0.3336   1.642 0.00246  0.113
## studygroup                     2     0.348  0.1738   0.856 0.00256  0.603
## Antibiotics_before_3_months    1     0.113  0.1134   0.558 0.00084  0.817
## Any_smoking_during_pregnancy   1     0.146  0.1464   0.721 0.00108  0.635
## Any_pet_at_birth               1     0.139  0.1386   0.682 0.00102  0.695
## Residuals                    507   103.008  0.2032         0.75854
## Total                        517   135.799                 1.00000
##
## time                         ***
## caesar                       ***
## eczema_by_2_years
## studygroup
## Antibiotics_before_3_months
## Any_smoking_during_pregnancy
## Any_pet_at_birth
## Residuals
## Total
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Make figure 1
test<-otu_table(taxonomy_genera, taxa_are_rows = FALSE)
mds<-metaMDS(test, dist="bray", k=2)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1905192
## Run 1 stress 0.2126407
## Run 2 stress 0.2120662
## Run 3 stress 0.2119449
## Run 4 stress 0.2033334
## Run 5 stress 0.2503196
## Run 6 stress 0.2054828
## Run 7 stress 0.2133659
## Run 8 stress 0.2034053
## Run 9 stress 0.2055301
## Run 10 stress 0.1987549
## Run 11 stress 0.2160786
## Run 12 stress 0.2143338
## Run 13 stress 0.1993714
## Run 14 stress 0.2000275
## Run 15 stress 0.2104548
## Run 16 stress 0.2027441
## Run 17 stress 0.2449327
## Run 18 stress 0.194369
## Run 19 stress 0.227156
## Run 20 stress 0.2054319
## *** No convergence -- monoMDS stopping criteria:
##     19: stress ratio > sratmax
##      1: scale factor of the gradient < sfgrmin
```

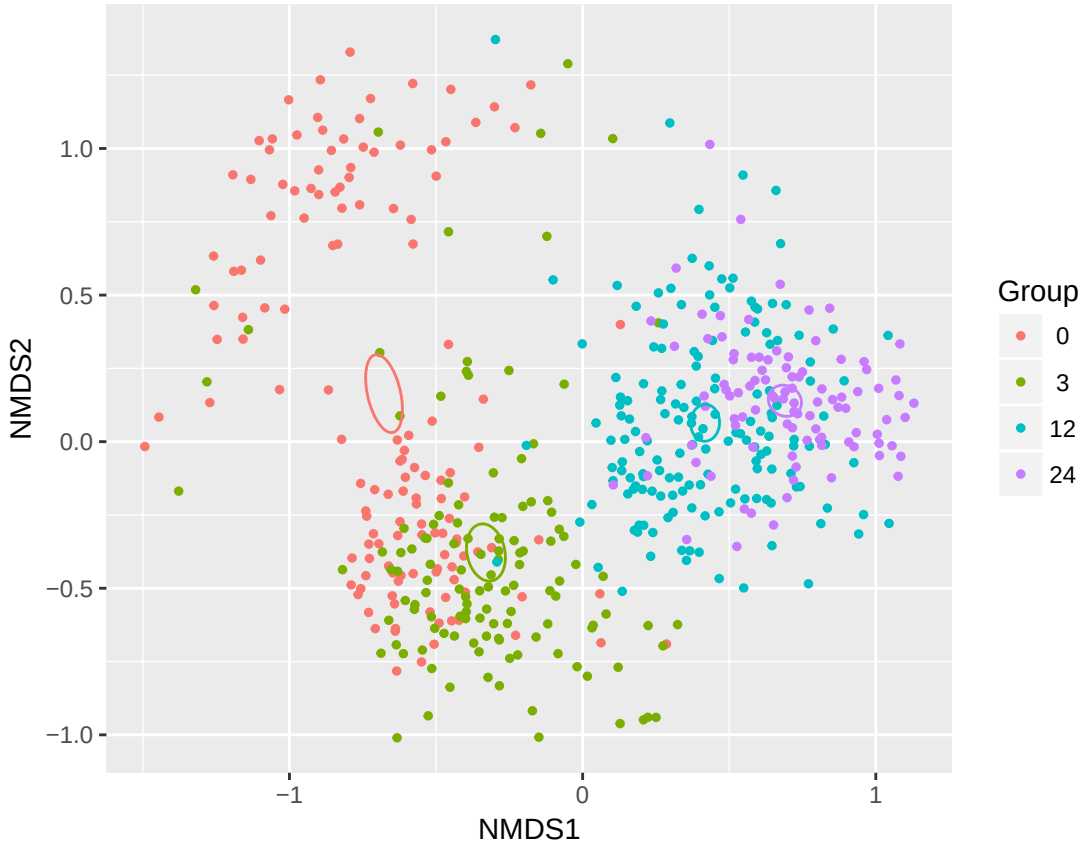
```
stressplot(mds)
```




```
print(mds$stress)
```

```
## [1] 0.1905192
```

```
fig1A<-gg_ordiplot(mds, groups=meta$time, scaling = 1, choices = c(1, 2), kind = "se", conf = 0.95, shou
```




```
meta$time = as.numeric(as.character(meta$time))
# Fig 1B
taxo_g0 <-subset(taxonomy_genera, Taxonomy_filtered$time == 0)
meta0<-subset(meta, meta$time== 0)
foo<-adonis(taxo_g0~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smoking_
print(foo$aov.tab)
```

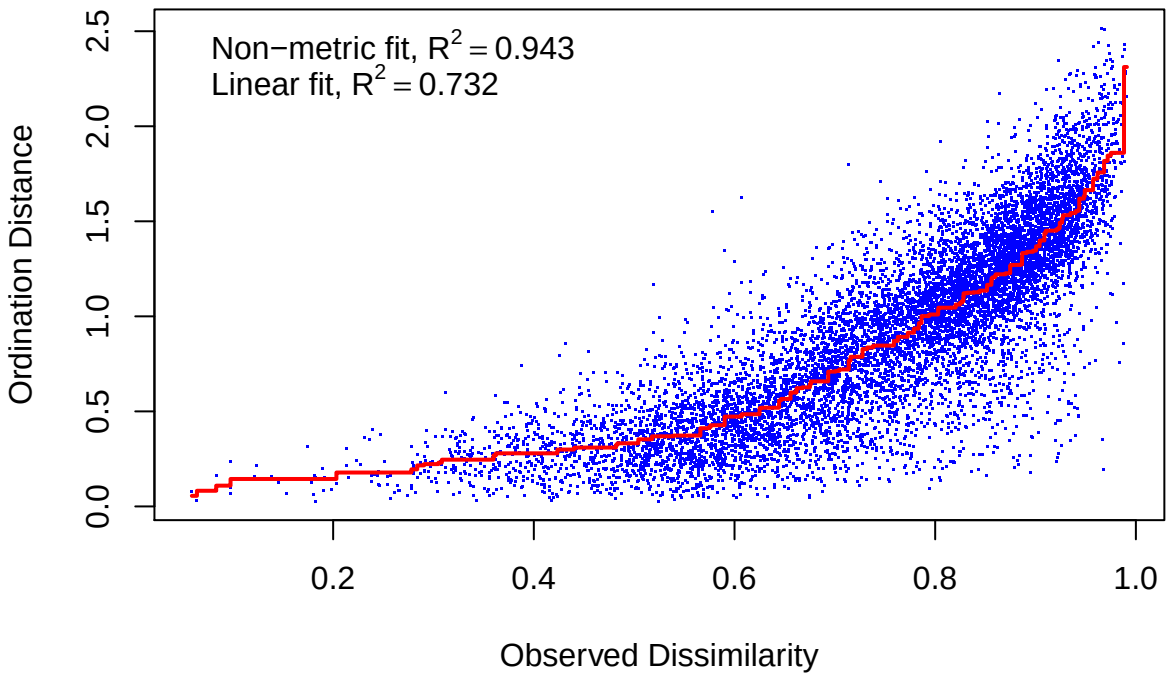
```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## caesar                          1     4.557  4.5572 16.7826 0.11050  0.001
## eczema_by_2_years               1     0.207  0.2072  0.7629 0.00502  0.531
## studygroup                      2     0.393  0.1966  0.7240 0.00953  0.681
## Antibiotics_before_3_months     1     0.073  0.0734  0.2703 0.00178  0.967
## Any_smoking_during_pregnancy    1     0.375  0.3753  1.3823 0.00910  0.161
## Any_pet_at_birth                1     0.334  0.3336  1.2285 0.00809  0.261
## Residuals                     130    35.300  0.2715         0.85597
## Total                         137    41.240                 1.00000
##
## caesar                        ***
## eczema_by_2_years
## studygroup
## Antibiotics_before_3_months
```

```
## Any_smoking_during_pregnancy
## Any_pet_at_birth
## Residuals
## Total
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mds<-metaMDS(taxo_g0, dist="bray", k=2)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2378948
## Run 1 stress 0.2504637
## Run 2 stress 0.2506716
## Run 3 stress 0.2402094
## Run 4 stress 0.2479262
## Run 5 stress 0.2549275
## Run 6 stress 0.2512069
## Run 7 stress 0.2470475
## Run 8 stress 0.2501983
## Run 9 stress 0.2442254
## Run 10 stress 0.2469749
## Run 11 stress 0.2508618
## Run 12 stress 0.2526858
## Run 13 stress 0.2560065
## Run 14 stress 0.2496005
## Run 15 stress 0.2410821
## Run 16 stress 0.2450733
## Run 17 stress 0.2515838
## Run 18 stress 0.2521267
## Run 19 stress 0.244556
## Run 20 stress 0.2541569
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

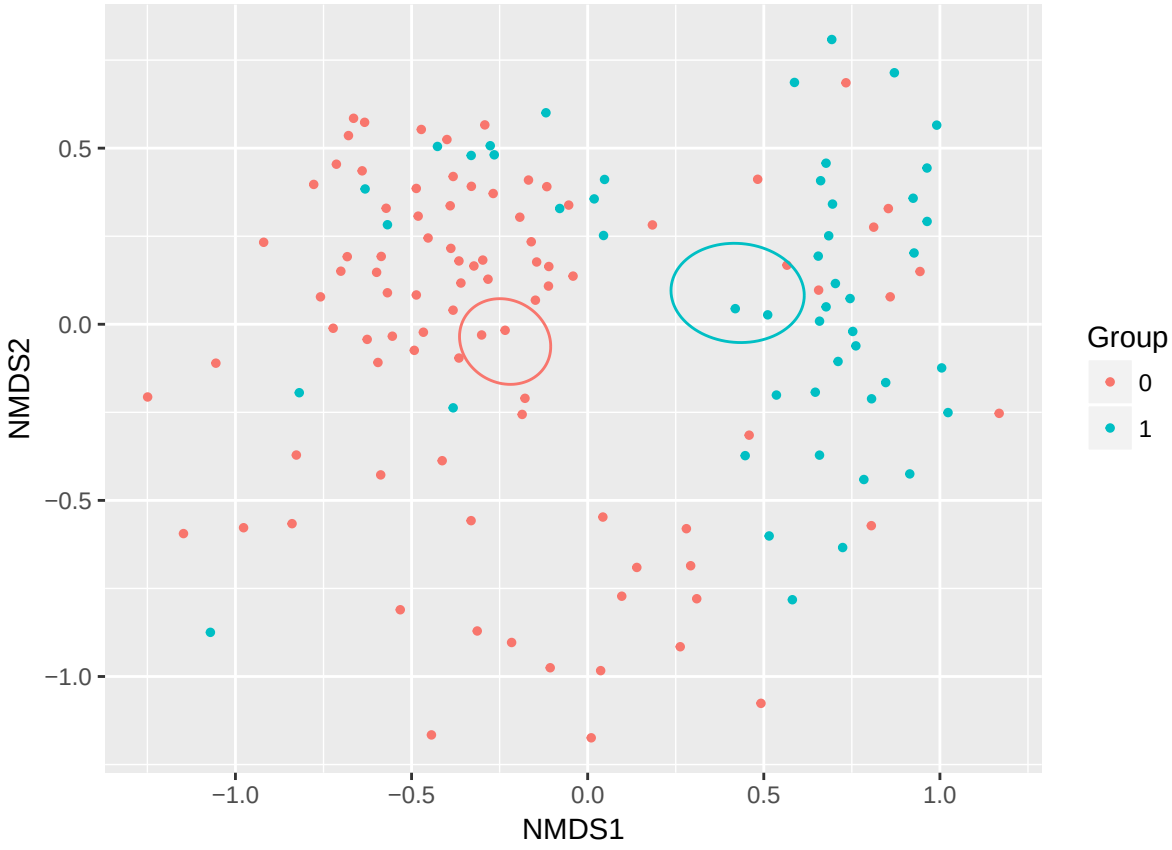
```
stressplot(mds)
```




```
print(mds$stress)
```

```
## [1] 0.2378948
```

```
fig1B<-gg_ordiplot(mds, groups=meta0$caesar, scaling = 1, choices = c(1, 2), kind = "se", conf = 0.95, s
```




```
# Fig 1c
taxo_g3 <-subset(taxonomy_genera, Taxonomy_filtered$time == 3)
meta3<-subset(meta, meta$time == 3)
foo<-adonis(taxo_g3~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smoking
print(foo$aov.tab)
```

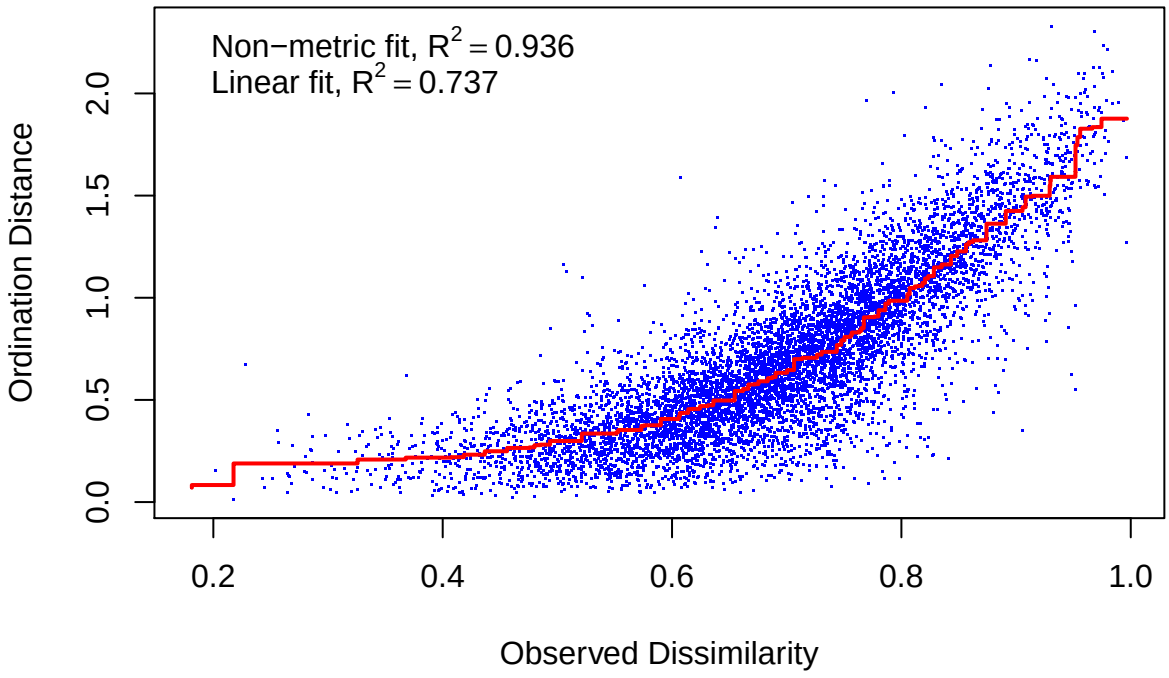
```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                               Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## caesar                         1    0.4397 0.43975  3.3013 0.02680  0.014
## eczema_by_2_years              1    0.1623 0.16231  1.2185 0.00989  0.232
## studygroup                     2    0.2288 0.11441  0.8589 0.01395  0.515
## Antibiotics_before_3_months    1    0.1628 0.16281  1.2223 0.00992  0.234
## Any_smoking_during_pregnancy   1    0.0284 0.02844  0.2135 0.00173  0.930
## Any_pet_at_birth               1    0.0670 0.06701  0.5031 0.00408  0.834
## Residuals                    115   15.3185 0.13320         0.93362
## Total                        122   16.4076                  1.00000
##
## caesar                        *
## eczema_by_2_years
## studygroup
## Antibiotics_before_3_months
## Any_smoking_during_pregnancy
```

```
## Any_pet_at_birth
## Residuals
## Total
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mds<-metaMDS(taxo_g3, dist="bray", k=2)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.254122
## Run 1 stress 0.2536562
## ... New best solution
## ... Procrustes: rmse 0.01108077  max resid 0.1085426
## Run 2 stress 0.2701901
## Run 3 stress 0.2539722
## ... Procrustes: rmse 0.009498835  max resid 0.08654073
## Run 4 stress 0.254228
## Run 5 stress 0.2541479
## ... Procrustes: rmse 0.03410487  max resid 0.3168217
## Run 6 stress 0.2656143
## Run 7 stress 0.269582
## Run 8 stress 0.2564003
## Run 9 stress 0.2548542
## Run 10 stress 0.2547803
## Run 11 stress 0.2549068
## Run 12 stress 0.2539105
## ... Procrustes: rmse 0.03242984  max resid 0.2879045
## Run 13 stress 0.2696898
## Run 14 stress 0.2538287
## ... Procrustes: rmse 0.005329744  max resid 0.03030608
## Run 15 stress 0.2540489
## ... Procrustes: rmse 0.01401671  max resid 0.1450969
## Run 16 stress 0.2549386
## Run 17 stress 0.2670914
## Run 18 stress 0.2587041
## Run 19 stress 0.2549403
## Run 20 stress 0.2543551
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

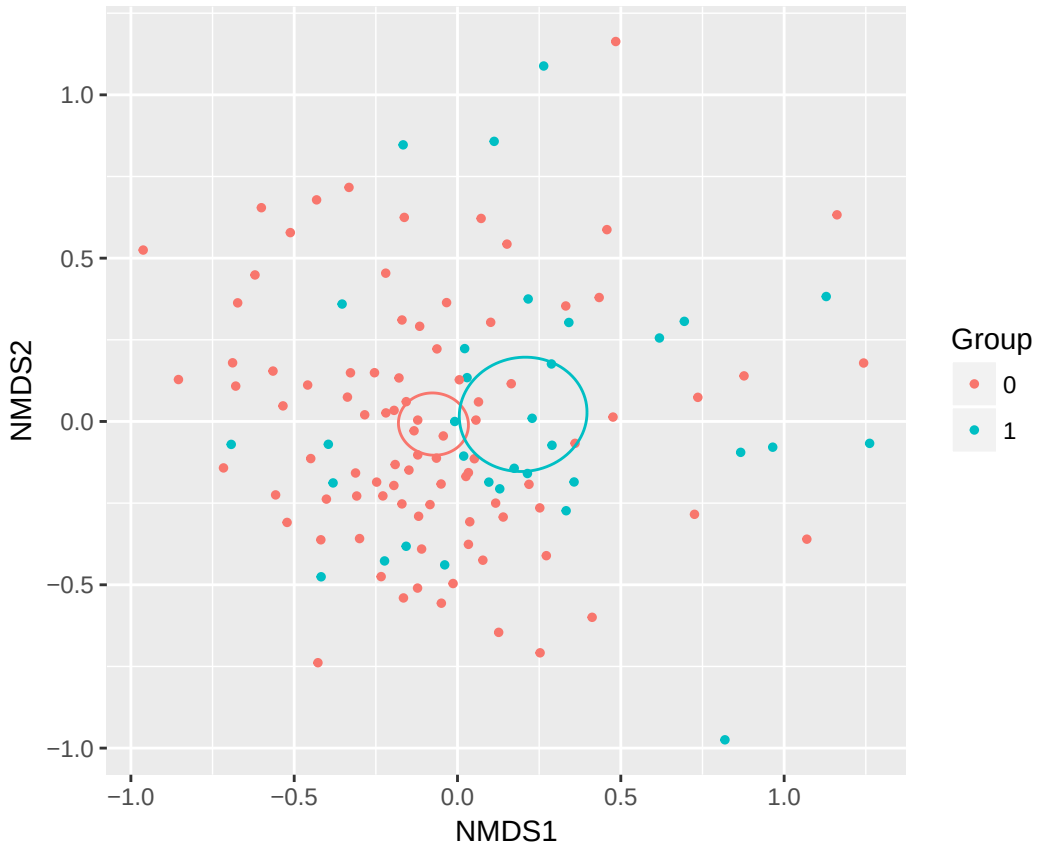
```
stressplot(mds)
```




```
print(mds$stress)
```

```
## [1] 0.2536562
```

```
fig1C<-gg_ordiplot(mds, groups=meta3$caesar, scaling = 1, choices = c(1, 2), kind = "se", conf = 0.95, s
```




```
# Fig 1D
taxo_g12 <-subset(taxonomy_genera, Taxonomy_filtered$time == 12)
meta12<-subset(meta, meta$time == 12)
foo<-adonis(taxo_g12~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smokin
print(foo$aov.tab)
```

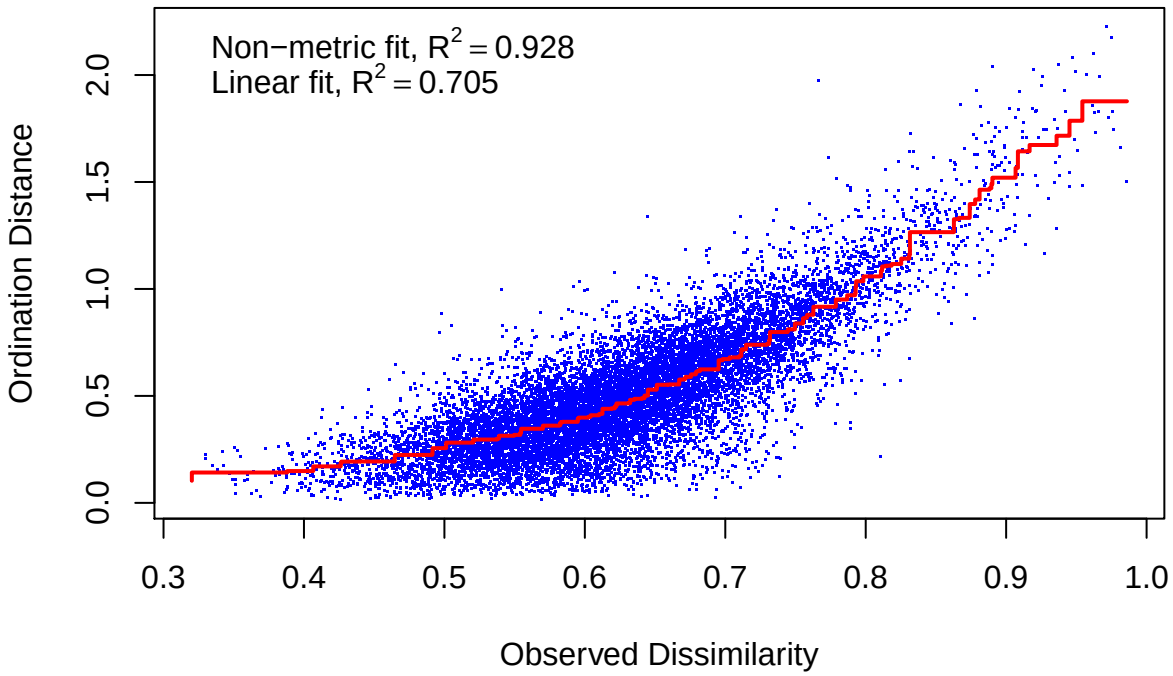
```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## caesar                          1     0.321 0.32063 1.58140 0.01006  0.126
## eczema_by_2_years               1     0.199 0.19851 0.97907 0.00623  0.407
## studygroup                      2     0.210 0.10502 0.51800 0.00659  0.925
## Antibiotics_before_3_months     1     0.106 0.10644 0.52499 0.00334  0.839
## Any_smoking_during_pregnancy    1     0.076 0.07646 0.37710 0.00240  0.926
## Any_pet_at_birth                1     0.343 0.34329 1.69318 0.01077  0.107
## Residuals                     151    30.615 0.20275         0.96061
## Total                         158    31.871                 1.00000
```

```
mds<-metaMDS(taxo_g12, dist="bray", k=2)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2677589
## Run 1 stress 0.2843149
```

```
## Run 2 stress 0.2686563
## Run 3 stress 0.2882062
## Run 4 stress 0.2700374
## Run 5 stress 0.2725519
## Run 6 stress 0.2842737
## Run 7 stress 0.2751686
## Run 8 stress 0.2844893
## Run 9 stress 0.2686282
## Run 10 stress 0.2677638
## ... Procrustes: rmse 0.00353414  max resid 0.01997208
## Run 11 stress 0.2686507
## Run 12 stress 0.2683043
## Run 13 stress 0.2755584
## Run 14 stress 0.27425
## Run 15 stress 0.2731212
## Run 16 stress 0.2678109
## ... Procrustes: rmse 0.004205508  max resid 0.03094318
## Run 17 stress 0.2763228
## Run 18 stress 0.2846621
## Run 19 stress 0.2727286
## Run 20 stress 0.2725033
## *** No convergence -- monoMDS stopping criteria:
##      3: no. of iterations >= maxit
##     17: stress ratio > sratmax
```

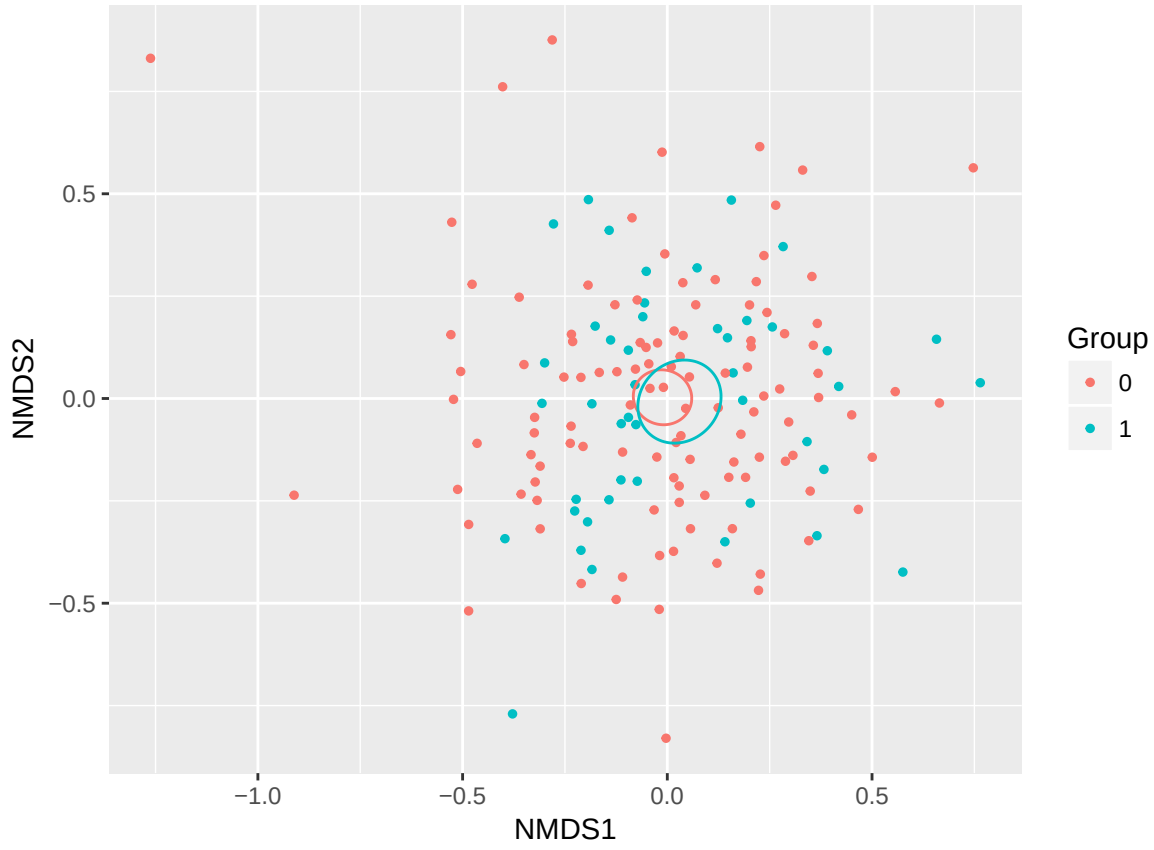
```
stressplot(mds)
```



```
print(mds$stress)
```

```
## [1] 0.2677589
```

```
fig1D<-gg_ordiplot(mds, groups=meta12$caesar, scaling = 1, choices = c(1, 2), kind = "se", conf = 0.95,
```



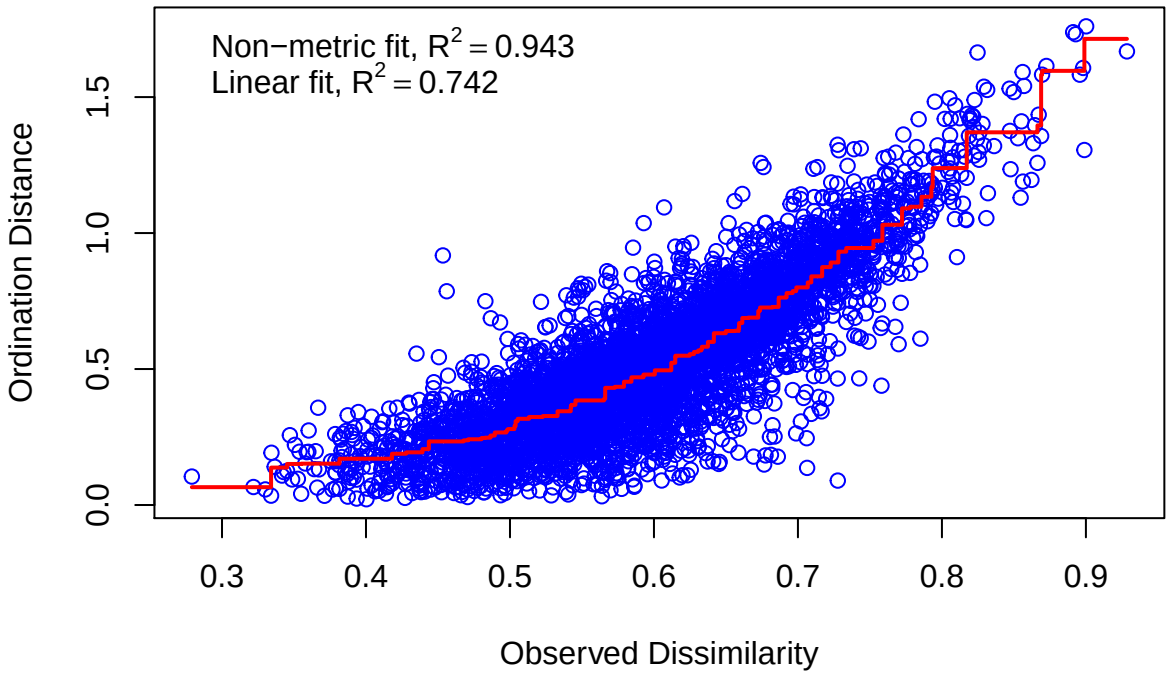

```
# Fig 1E
taxo_g24 <-subset(taxonomy_genera, Taxonomy_filtered$time == 24)
meta24<-subset(meta, meta$time == 24)
foo<-adonis(taxo_g24~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smokin
print(foo$aov.tab)
```

```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                               Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## caesar                         1    0.1803 0.180296 1.04906 0.01091  0.356
## eczema_by_2_years              1    0.1728 0.172808 1.00550 0.01045  0.388
## studygroup                     2    0.2870 0.143491 0.83491 0.01736  0.646
## Antibiotics_before_3_months    1    0.0518 0.051799 0.30140 0.00313  0.972
## Any_smoking_during_pregnancy   1    0.2627 0.262720 1.52865 0.01589  0.134
## Any_pet_at_birth               1    0.1077 0.107709 0.62671 0.00652  0.765
## Residuals                     90   15.4677 0.171864         0.93573
## Total                         97   16.5301                  1.00000
```

```r
mds<-metaMDS(taxo_g24, dist="bray", k=2)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2383588
## Run 1 stress 0.24023
## Run 2 stress 0.2497051
## Run 3 stress 0.2383957
## ... Procrustes: rmse 0.02874707  max resid 0.189865
## Run 4 stress 0.2429204
## Run 5 stress 0.2462832
## Run 6 stress 0.2527173
## Run 7 stress 0.2431869
## Run 8 stress 0.2467362
## Run 9 stress 0.2499124
## Run 10 stress 0.250396
## Run 11 stress 0.4117939
## Run 12 stress 0.2444736
## Run 13 stress 0.2579634
## Run 14 stress 0.2435176
## Run 15 stress 0.2469969
## Run 16 stress 0.2484539
## Run 17 stress 0.2661387
## Run 18 stress 0.2387027
## ... Procrustes: rmse 0.04060223  max resid 0.187845
## Run 19 stress 0.2499912
## Run 20 stress 0.2442356
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

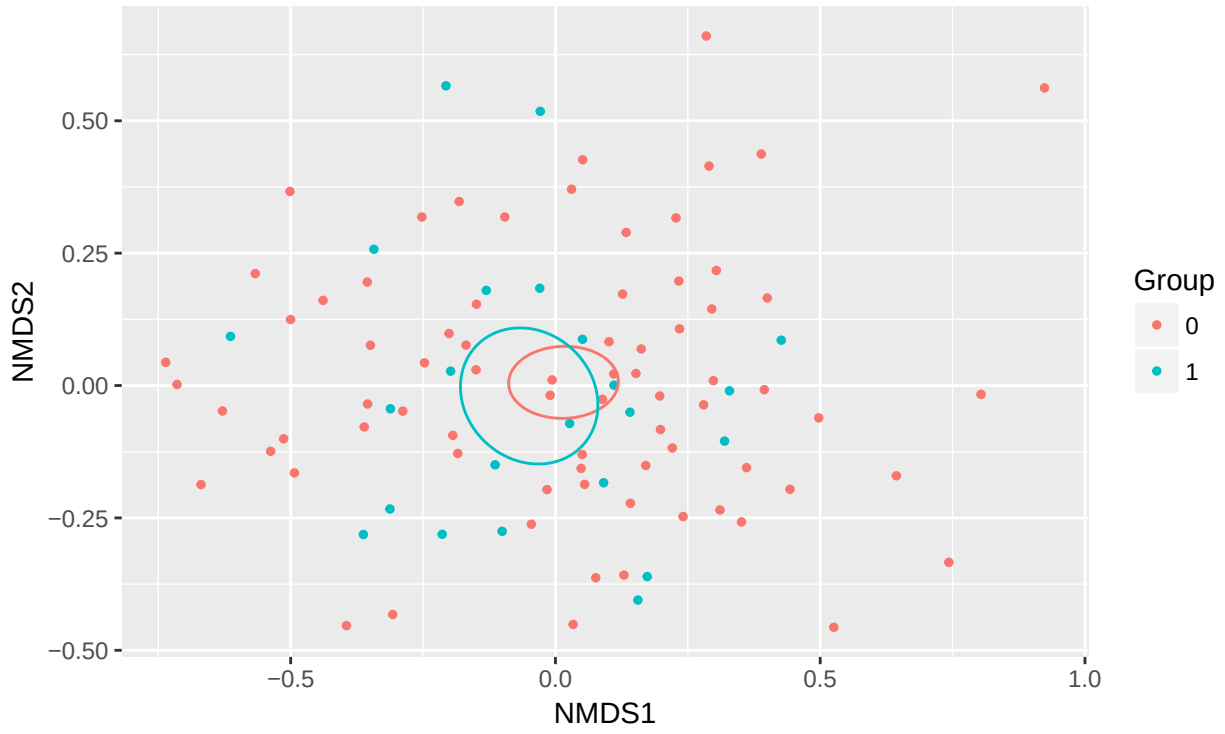
```r
stressplot(mds)
```




```
print(mds$stress)
```

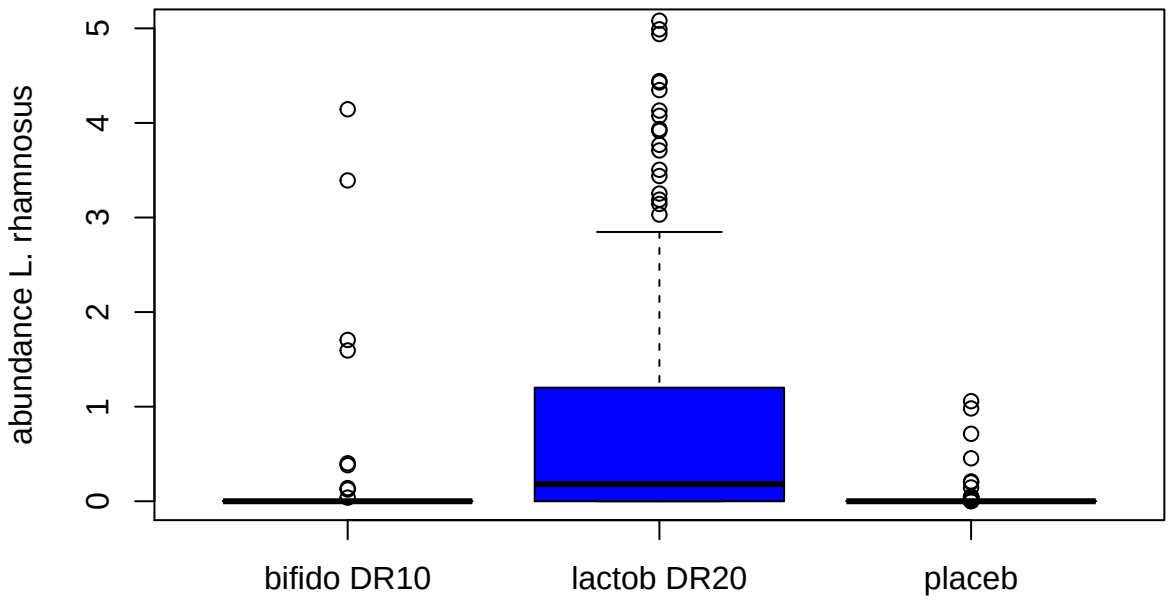
```
## [1] 0.2383588
```

```
fig1E<-gg_ordiplot(mds, groups=meta24$caesar, scaling = 1, choices = c(1, 2), kind = "se", conf = 0.95,
```




```
A<-fig1A$plot

t1<-fig1B$plot + coord_cartesian(xlim = c(-1.25, 1.25), ylim=c(-1.25, 1.25))
t2<-fig1C$plot + coord_cartesian(xlim = c(-1.25, 1.25), ylim=c(-1.25, 1.25))
t3<-fig1D$plot + coord_cartesian(xlim = c(-1.25, 1.25), ylim=c(-1.25, 1.25))
t4<-fig1E$plot + coord_cartesian(xlim = c(-1.25, 1.25), ylim=c(-1.25, 1.25))
foo<-ggarrange(t1, t2, t3, t4, ncol=2, nrow=2, common.legend=TRUE, widths=c(1, 1), heights=c(1, 1), lab
bar<-ggarrange(A, foo, ncol=2, labels=c("A", "B"), legend=c("bottom"), widths=c(1, 1.5))


ggsave("~/PIP2018/results/Fig1.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)

# This plot is post-processed in Inkscape to add stress, recenter B row 1 labels, and shade centroids
```

```
boxplot(Taxonomy_filtered[,541] ~ Taxonomy_filtered$studygroup, ylim = c(0, 5), ylab="abundance L. rham
```




```
kruskal.test(Taxonomy_filtered[,166] ~ Taxonomy_filtered$studygroup)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Taxonomy_filtered[, 166] by Taxonomy_filtered$studygroup
## Kruskal-Wallis chi-squared = 58.324, df = 2, p-value = 2.164e-13
```

```
boxplot(Taxonomy_filtered[,166] ~ Taxonomy_filtered$studygroup, ylim = c(0, 1), ylab="abundance B. anima

kruskal.test(Taxonomy_filtered[,541] ~ Taxonomy_filtered$studygroup)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Taxonomy_filtered[, 541] by Taxonomy_filtered$studygroup
## Kruskal-Wallis chi-squared = 144.37, df = 2, p-value < 2.2e-16
```

```
biff<-cbind(Taxonomy_filtered$time, as.character(Taxonomy_filtered$studygroup), Taxonomy_filtered[,166]
biff<-as.data.frame(biff)
colnames(biff) = c("time", "studygroup", "b.animalis", "l.rhamnosus")

levels(biff$time) = c(0, 3, 12, 24)
biff$studygroup = factor(biff$studygroup)


biff$b.animalis = as.numeric(as.character(biff$b.animalis))
```

```r
biff$l.rhamnosus = as.numeric(as.character(biff$l.rhamnosus))
biff$studygroup = gsub("bifido DR10", "b.lactis HN019", biff$studygroup)
biff$studygroup = gsub("lactob DR20", "l.rhamnosus HN001", biff$studygroup)
biff$studygroup = gsub("placeb", "placebo", biff$studygroup)

c<-ggplot(biff, aes(y=log(b.animalis), x=studygroup, color=studygroup))
c<- c + geom_boxplot() + geom_jitter(width=0.25) + theme(axis.text.x = element_text(angle = 45, hjust =
d<-ggplot(biff, aes(y=log(l.rhamnosus), x=studygroup, color=studygroup))

d<-d + geom_boxplot() + geom_jitter(width=0.25) + theme(axis.text.x = element_text(angle = 45, hjust = 
ggarrange(c, d, labels=c("A", "B"), common.legend=TRUE)
```

```
## Warning: Removed 418 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 418 rows containing non-finite values
## (stat_compare_means).
```

```
## Warning: Computation failed in `stat_compare_means()`:
## Can't find specified reference group: 3. Allowed values include one of: 1
```

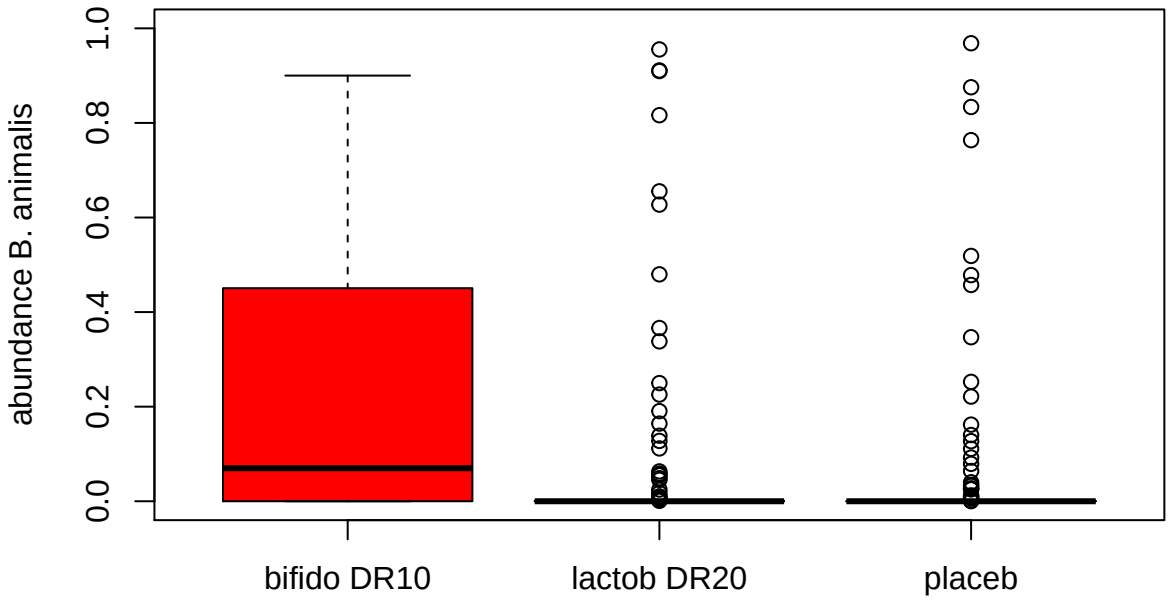
```
## Warning: Removed 418 rows containing non-finite values (stat_boxplot).
```

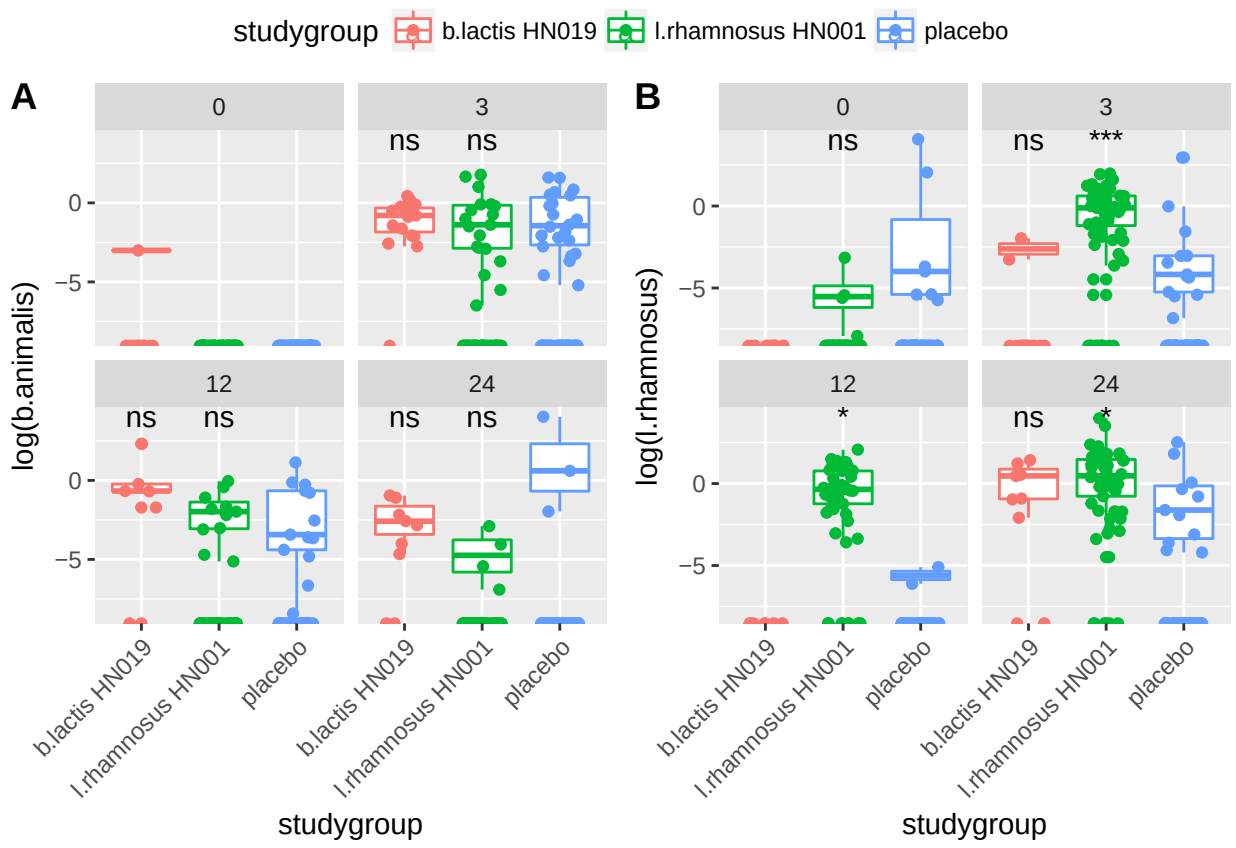
```
## Warning: Removed 418 rows containing non-finite values
## (stat_compare_means).
```

```
## Warning: Computation failed in `stat_compare_means()`:
## Can't find specified reference group: 3. Allowed values include one of: 1
```

```
## Warning: Removed 339 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 339 rows containing non-finite values
## (stat_compare_means).
```


abundance B. animalis
0.0
0.2
0.4
0.6
0.8
1.0
bifido DR10
lactob DR20
placeb




```
ggsave("~/PIP2018/results/SupFig3.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)


# calculate p values
pairwise.wilcox.test(biff$l.rhamnosus, interaction(biff$studygroup,biff$time), p.adj = "BH")
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
```

```
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  biff$l.rhamnosus and interaction(biff$studygroup, biff$time)
##
##                       b.lactis HN019.0 l.rhamnosus HN001.0 placebo.0
## l.rhamnosus HN001.0  0.43022          -                   -
## placebo.0            0.31776          0.51918             -
## b.lactis HN019.3     0.30234          0.51918             0.79983
## l.rhamnosus HN001.3  3.2e-06          9.9e-16             5.3e-15
## placebo.3            0.16147          0.10758             0.37008
## b.lactis HN019.12    -                0.56198             0.47349
## l.rhamnosus HN001.12 2.4e-05          6.9e-13             6.5e-12
## placebo.12           0.53136          0.61897             0.30234
## b.lactis HN019.24    0.00041          5.9e-08             1.6e-06
## l.rhamnosus HN001.24 1.2e-06          4.5e-16             1.6e-15
## placebo.24           0.16142          0.09707             0.32343
##                       b.lactis HN019.3 l.rhamnosus HN001.3 placebo.3
## l.rhamnosus HN001.0  -                -                   -
## placebo.0            -                -                   -
## b.lactis HN019.3     -                -                   -
## l.rhamnosus HN001.3  3.3e-06          -                   -
## placebo.3            0.75337          1.6e-15             -
## b.lactis HN019.12    0.47349          0.00078             0.33708
## l.rhamnosus HN001.12 3.2e-05          0.52238             7.0e-12
## placebo.12           0.31874          1.7e-14             0.05039
## b.lactis HN019.24    0.00118          0.87188             9.9e-06
## l.rhamnosus HN001.24 1.0e-06          0.05039             4.5e-16
```

```
## placebo.24           0.65031          3.0e-12             0.79983
##                      b.lactis HN019.12 l.rhamnosus HN001.12 placebo.12
## l.rhamnosus HN001.0  -                 -                    -
## placebo.0            -                 -                    -
## b.lactis HN019.3     -                 -                    -
## l.rhamnosus HN001.3  -                 -                    -
## placebo.3            -                 -                    -
## b.lactis HN019.12    -                 -                    -
## l.rhamnosus HN001.12 0.00218           -                    -
## placebo.12           0.66385           6.5e-12              -
## b.lactis HN019.24    0.01040           0.74269              4.7e-08
## l.rhamnosus HN001.24 0.00040           0.02478              4.7e-15
## placebo.24           0.33708           3.0e-09              0.04620
##                      b.lactis HN019.24 l.rhamnosus HN001.24
## l.rhamnosus HN001.0  -                 -
## placebo.0            -                 -
## b.lactis HN019.3     -                 -
## l.rhamnosus HN001.3  -                 -
## placebo.3            -                 -
## b.lactis HN019.12    -                 -
## l.rhamnosus HN001.12 -                 -
## placebo.12           -                 -
## b.lactis HN019.24    -                 -
## l.rhamnosus HN001.24 0.47032           -
## placebo.24           9.5e-05           2.9e-13
##
## P value adjustment method: BH
```

```r
# calculate p values
pairwise.wilcox.test(biff$b.animalis, interaction(biff$studygroup,biff$time), p.adj = "BH")
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  biff$b.animalis and interaction(biff$studygroup, biff$time)
##
##                      b.lactis HN019.0 l.rhamnosus HN001.0 placebo.0
## l.rhamnosus HN001.0  0.06157          -                   -
## placebo.0            0.04493          -                   -
## b.lactis HN019.3     1.7e-05          1.5e-14             6.1e-16
## l.rhamnosus HN001.3  0.10787          3.8e-05             1.3e-05
## placebo.3            0.10089          2.8e-05             9.9e-06
## b.lactis HN019.12    0.00375          3.0e-10             2.2e-11
## l.rhamnosus HN001.12 0.17055          0.00014             4.8e-05
## placebo.12           0.15833          7.4e-05             2.7e-05
## b.lactis HN019.24    0.00174          2.2e-11             2.1e-12
## l.rhamnosus HN001.24 0.97332          0.04422             0.03131
## placebo.24           0.82088          0.11961             0.10010
##                      b.lactis HN019.3 l.rhamnosus HN001.3 placebo.3
## l.rhamnosus HN001.0  -                -                   -
## placebo.0            -                -                   -
## b.lactis HN019.3     -                -                   -
## l.rhamnosus HN001.3  7.4e-06          -                   -
## placebo.3            9.9e-06          0.94478             -
## b.lactis HN019.12    0.97332          0.01521             0.01527
## l.rhamnosus HN001.12 1.5e-06          0.69217             0.63394
## placebo.12           2.1e-06          0.76509             0.66566
## b.lactis HN019.24    0.01054          0.02263             0.03109
## l.rhamnosus HN001.24 2.2e-11          0.00606             0.00395
## placebo.24           6.9e-12          0.00138             0.00095
##                      b.lactis HN019.12 l.rhamnosus HN001.12 placebo.12
## l.rhamnosus HN001.0  -                 -                    -
## placebo.0            -                 -                    -
## b.lactis HN019.3     -                 -                    -
```

```
## l.rhamnosus HN001.3  -                 -                     -
## placebo.3            -                 -                     -
## b.lactis HN019.12    -                 -                     -
## l.rhamnosus HN001.12 0.00633           -                     -
## placebo.12           0.01011           0.97332               -
## b.lactis HN019.24    0.19578           0.01066               0.00992
## l.rhamnosus HN001.24 1.2e-05           0.01922               0.01416
## placebo.24           1.5e-06           0.00621               0.00363
##                      b.lactis HN019.24 l.rhamnosus HN001.24
## l.rhamnosus HN001.0  -                 -
## placebo.0            -                 -
## b.lactis HN019.3     -                 -
## l.rhamnosus HN001.3  -                 -
## placebo.3            -                 -
## b.lactis HN019.12    -                 -
## l.rhamnosus HN001.12 -                 -
## placebo.12           -                 -
## b.lactis HN019.24    -                 -
## l.rhamnosus HN001.24 1.0e-06           -
## placebo.24           1.5e-07           0.63394
##
## P value adjustment method: BH
```

Supplementary Figure 4

```
#collect objects for ggplotting
modules<-read.table("~/PIP2018/derived_data/filtered_modules.tsv", header=TRUE, sep="\t")
tax<-read.table("~/PIP2018/derived_data/filtered_taxonomy.tsv", header=TRUE, sep="\t")

pathways<-read.table("~/PIP2018/derived_data/filtered_pathways.tsv", header=TRUE, sep="\t")

myvars3 = c("Sample", "ko00531", "ko00240", "ko04141")
pwys<-pathways[myvars3]

# match colnames taxonomy filtered
myvars <- c("Sample", "time", "studygroup", "eczema_by_2_years", "M00198", "M00277")
mods <- modules[myvars]
myvars2 <- c("Sample", "k__Bacteria.p__Actinobacteria.c__Actinobacteria.o__Bifidobacteriales.f__Bifidoba
tx <- tax[myvars2]
m1 <- merge(x=mods,y=tx,by.x = c("Sample"),by.y = c("Sample"),all.y = TRUE)
colnames(m1)[7] = "B.animalis"
colnames(m1)[8] = "L.rhamnosus"
m1<-merge(x=m1, y=pwys,  by.x=c("Sample"), by.y=c("Sample"), all.y=TRUE)
m1$eczema_by_2_years = factor(m1$eczema_by_2_years)
m1$time = factor(m1$time)
m1$studygroup = gsub("bifido DR10", "b.lactis HN019", biff$studygroup)
m1$studygroup = gsub("lactob DR20", "l.rhamnosus HN001", biff$studygroup)
m1$studygroup = gsub("placeb", "placebo", biff$studygroup)


# Make plots
#S4A
a<-ggplot(data=m1, aes(x=L.rhamnosus, y=M00198, colour=studygroup)) + geom_point() + facet_wrap(~studygr
    theme(legend.position="bottom")
# S4B
```

```
b1<-ggplot(data=m1, aes(x=time, y=M00277, colour=eczema_by_2_years)) + geom_boxplot() + stat_compare_mea
b2<-ggplot(data=m1, aes(x=time, y=log(ko00531), colour=eczema_by_2_years)) + geom_boxplot() +   stat_cor
b3<-ggplot(data=m1, aes(x=time, y=log(ko00240), colour=eczema_by_2_years)) + geom_boxplot() +   stat_cor
b4<-ggplot(data=m1, aes(x=time, y=log(ko04141), colour=eczema_by_2_years)) + geom_boxplot() +   stat_cor
panelb<-ggarrange(b1, b2, b3, b4, common.legend = TRUE, legend="bottom", labels=c("B"))
```

```
## Warning: Removed 17 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 17 rows containing non-finite values (stat_compare_means).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_compare_means).
```

```
c<-ggplot(data=m1, aes(x=time, y=log(L.rhamnosus), colour=eczema_by_2_years)) + geom_boxplot() + stat_c



leftpanel<-ggarrange(a, c, nrow=2, labels=c("A", "C"))
```

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 6 rows containing non-finite values (stat_cor).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Warning: Removed 339 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 339 rows containing non-finite values
## (stat_compare_means).
```

```
all<-ggarrange(leftpanel, panelb, ncol=2)
ggsave("~/PIP2018/results/SupFig4.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

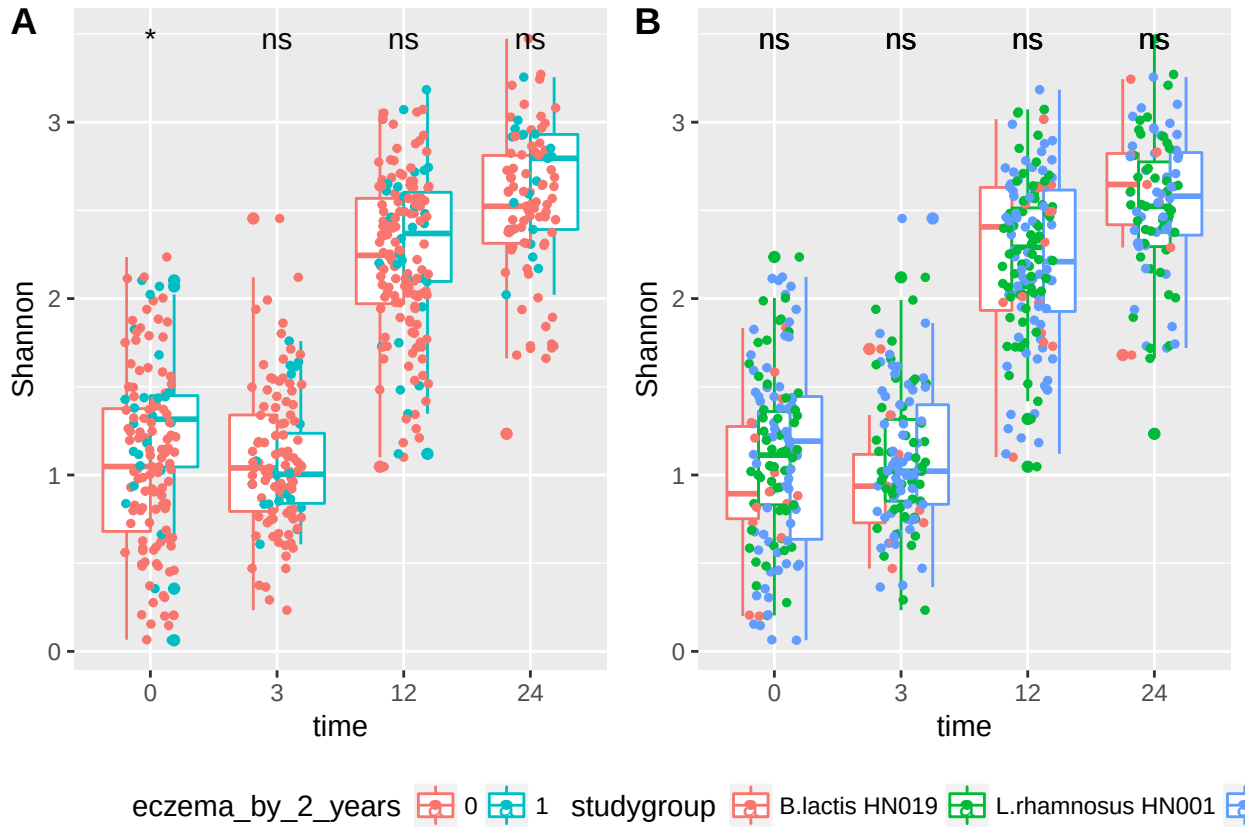
Supplemental Plot: Understanding alpha diversity in the dataset

```
#File of total reads per sample that went into MetaPhlAn / HUMANN
NZGL_taxonomy_SP_counts<-read.csv("~/PIP2018/derived_data/NZGL_taxonomy_count_table.csv", header = TRUE)
rownames(NZGL_taxonomy_SP_counts)<-NZGL_taxonomy_SP_counts$X
NZGL_taxonomy_SP_counts<-NZGL_taxonomy_SP_counts[,-1]
NZGL_taxonomy_SP_counts1<-NZGL_taxonomy_SP_counts[,colnames(NZGL_taxonomy_SP_counts)%in%c(as.character(



## use the following method to produce a biom file to make rarefraction curve.
# 1. open the NZGL_taxonomy_SP_counts.csv file in excel (the one in the new location) and create a new
# 2. move the taxonomy column to the every end and name it "taxonomy". Save the modified csv file to tx
# 3. convert it to josn biom use MacQiime: biom convert -i NZGL_taxonomy_SP_counts.txt -o NZGL_taxonomy



#============run this part after the biom file is made===========#
# I have put the biom file I made in the repository to let the analyses run. However, feel free to make
# A) Rarefaction curves: Mean shannon diversity (with SD error bars/ 95% CI) for each age group
NZGL_taxonomy_SP_counts1<-import_biom("~/PIP2018/derived_data/NZGL_taxonomy_SP_counts.biom")
Count_table<-NZGL_taxonomy_SP_counts1
```

```
Pipmeta<-as.data.frame(Taxonomy_filtered[,c(1:28)])
source("~/PIP2018/src/Rarefraction_functions.r", local = TRUE)
set.seed(42)

rarefaction_curve_data <- calculate_rarefaction_curves(Count_table, c('Observed',"Chao1", "Shannon"), r

# calculate mean shannon/any other mesure alpha diveristy for each sample at each depth.
rarefaction_curve_data_summary <- ddply(rarefaction_curve_data, c('Depth', 'Sample', 'Measure'), summar

rarefaction_curve_data_shannon<-subset(rarefaction_curve_data_summary, Measure == "Shannon")

# Pipmeta has been transposed so load a new set of metadata for selecting samples based on metadata cat
Pipmeta<-read.delim("~/PIP2018/primary_data/taxonomy.tsv", header = TRUE)
shannon_merge<-merge(rarefaction_curve_data_shannon, data.frame(Pipmeta), by.x = "Sample")
shannon_merge$time<-as.factor(shannon_merge$time)
shannon_merge_summary<-summarySE(shannon_merge, measurevar="Alpha_diversity_mean", c("Depth", "time"))

Sample_reads_sum<-as.data.frame(sample_sums(Count_table))

Shannon_calcualtion<-estimate_richness(Count_table, measures = "Shannon")
```

```
## Warning in estimate_richness(Count_table, measures = "Shannon"): The data you have provided does not
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
Shannon_calcualtion$Sample<-rownames(Shannon_calcualtion)
shannon_merge<-merge(Shannon_calcualtion, data.frame(Pipmeta), by.x = "Sample")
#add the total reads to metadata for correspondance samples
shannon_merge$Sample_reads_sum<-Sample_reads_sum$`sample_sums(Count_table)`[(match(shannon_merge$Sample
#convert time variable to factor instead of integer
shannon_merge$time<-as.factor(shannon_merge$time)
shannon_merge$eczema_by_2_years = factor(shannon_merge$eczema_by_2_years)


panel1<-ggplot(shannon_merge, aes(time, Shannon, color=eczema_by_2_years))+geom_boxplot() + geom_jitter

panel2<-ggplot(shannon_merge, aes(time, Shannon, color=studygroup))+geom_boxplot() + geom_jitter(width=0

ggarrange(panel1, panel2, labels=c("A", "B"), legend="bottom")
```




```
ggsave("~/PIP2018/results/SupFig2.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```