MasterPip - the unfiltered version

XCM & Cecilia Wang

June 2018

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

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

```
set.seed(8675309)

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

1. Generation of PIP figures and tables

```
#Piece of code for loading taxonomy.tsv / modules.pcl / pathways.pcl
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
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)
Taxonomy_unfiltered<-rbind(E_coli_abundance_AtBirth,E_coli_abundance_3_month,E_coli_abundance_12_month,
```

Plots: Regarding the relationships between E. coli abundance & age, E. coli abundance & time at room temperature, and time of storage of samples (Sup 1) This figure does not change in unfiltered version

```
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

```

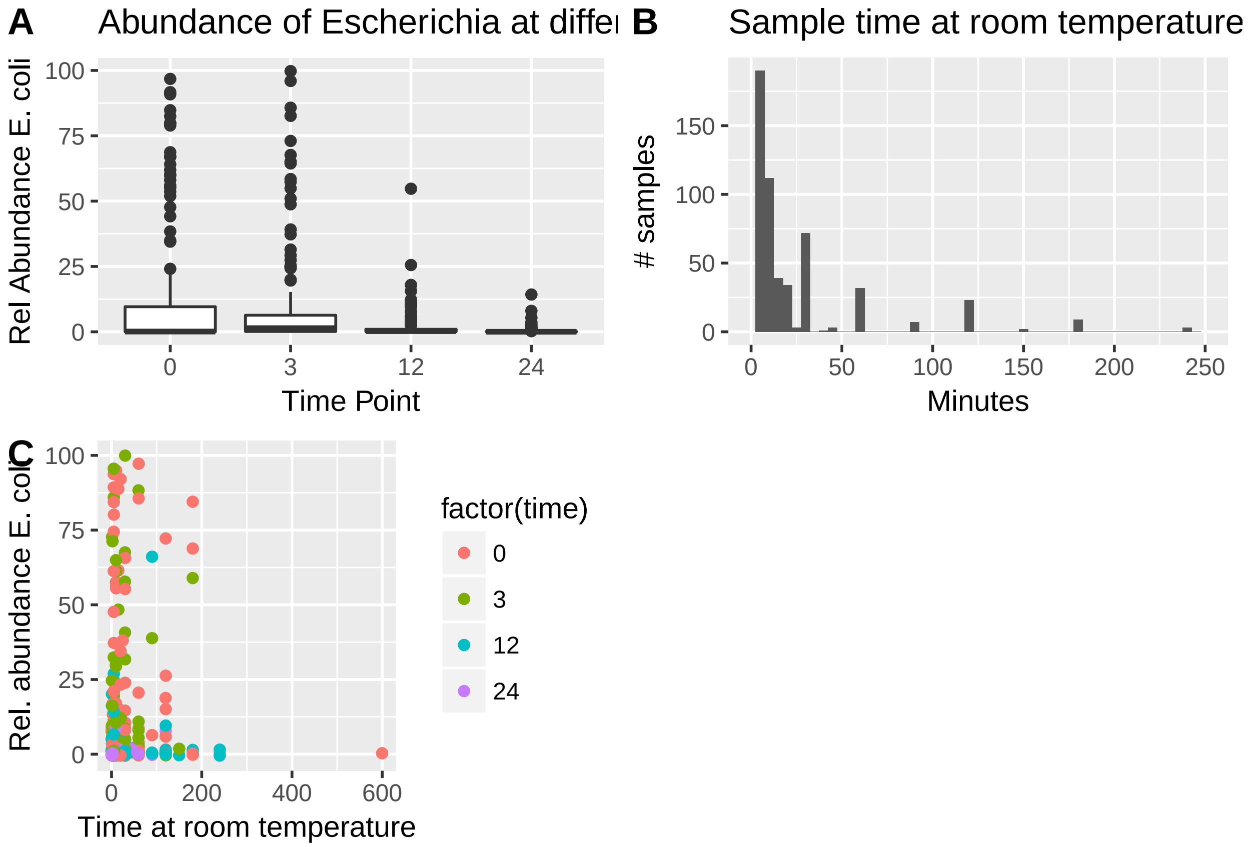
```r
#Plot the Duration of storage of study fecal samples at room temperature before freezing
Molten_Meta<-melt(Metadata, id.vars = "Studyid", measure.vars = c("ftime_0", "ftime_3", "ftime_12", "fti
colnames(Molten_Meta)[2]<-"time"
Molten_Meta$time<-as.character(Molten_Meta$time)
Molten_Meta$time[Molten_Meta$time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

```r
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(0,2

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

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).
```



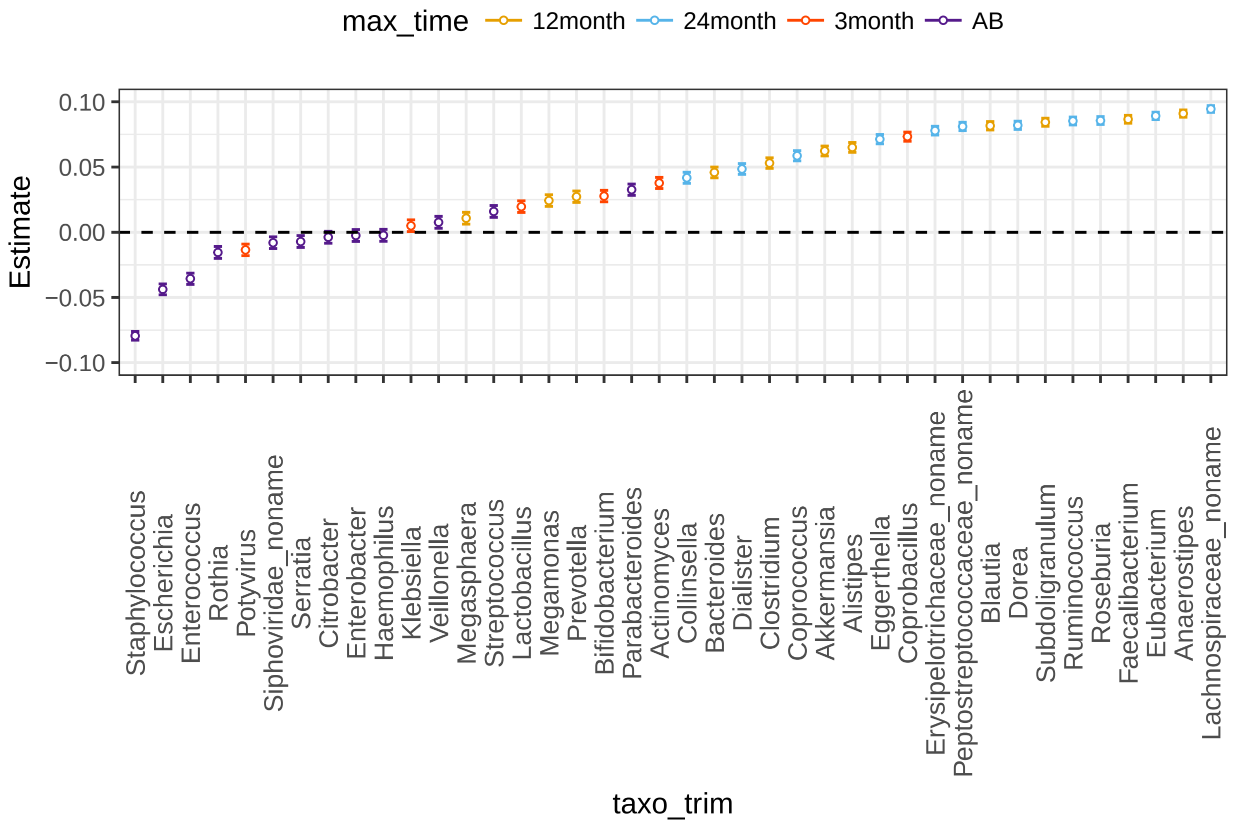
```
ggsave("~/PIP2018/results/unfiltered-SupFig1.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

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)

Generate Figure 2:

```
###======PLEASE RUN THE FIRST THREE CHUNKS BEFORE RUNNING THIS SESSION=====###
#Probably: Figure 2 of paper

#Concept: Visualize genera in a way that shows their correlation with time

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


# 1) log10 and normalise the taxonomy abundance
Log10_Taxonomy_unfiltered_num<-sapply(Taxonomy_unfiltered_num, function(x) log10(x))
row.names(Log10_Taxonomy_unfiltered_num)<-row.names(Taxonomy_unfiltered_num)
Norm_log10_abundance<-as.data.frame(scale(Log10_Taxonomy_unfiltered_num))


# 2) Glom to genera
```

```
  ## select any taxo names that the taxo has reached genus level
Norm_unfiltered_taxonomy_abundance_select<-Norm_log10_abundance[,grep("g__",colnames(Norm_log10_abundan
  ## 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_unfiltered_taxonomy_g<-Norm_unfiltered_taxonomy_abundance_select[,setdiff(colnames(Norm_unfiltered_
  ## Only select genera that have data
Genera_sum<-as.data.frame(apply(Norm_unfiltered_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_unfiltered_taxonomy_g$time<-Taxonomy_unfiltered$time[match(rownames(Norm_unfiltered_taxonomy_g), Ta
  ## 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_unfiltered_taxonomy_g[,a]~Norm_unfiltered_taxonomy_g$time))
  T2<-as.data.frame(t(T[[4]][2,]))
  T2$taxo<-colnames(Norm_unfiltered_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)])

keep<-subset(temp, temp$`Pr(>|t|)` <= 0.05)
write.csv(keep, file="~/PIP2018/results/suptable2-unfiltered-bugs.csv")

# 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_unfiltered_num[,grep("g__",colnames(Taxonomy_unfiltered_num))]
  ## select any taxo names that has reached species level
taxonomy_select_t_col<-Taxonomy_unfiltered_num[,grep("s__",colnames(Taxonomy_unfiltered_num))]
  ## substract taxonomy_select_t_col from taxonomy_abundance_select
taxonomy_genera<-taxonomy_abundance_select[,setdiff(colnames(taxonomy_abundance_select),colnames(taxonom
  ## 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_unfiltered
  # 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/unfiltered-Fig2A.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```

Generate Figure 2B

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

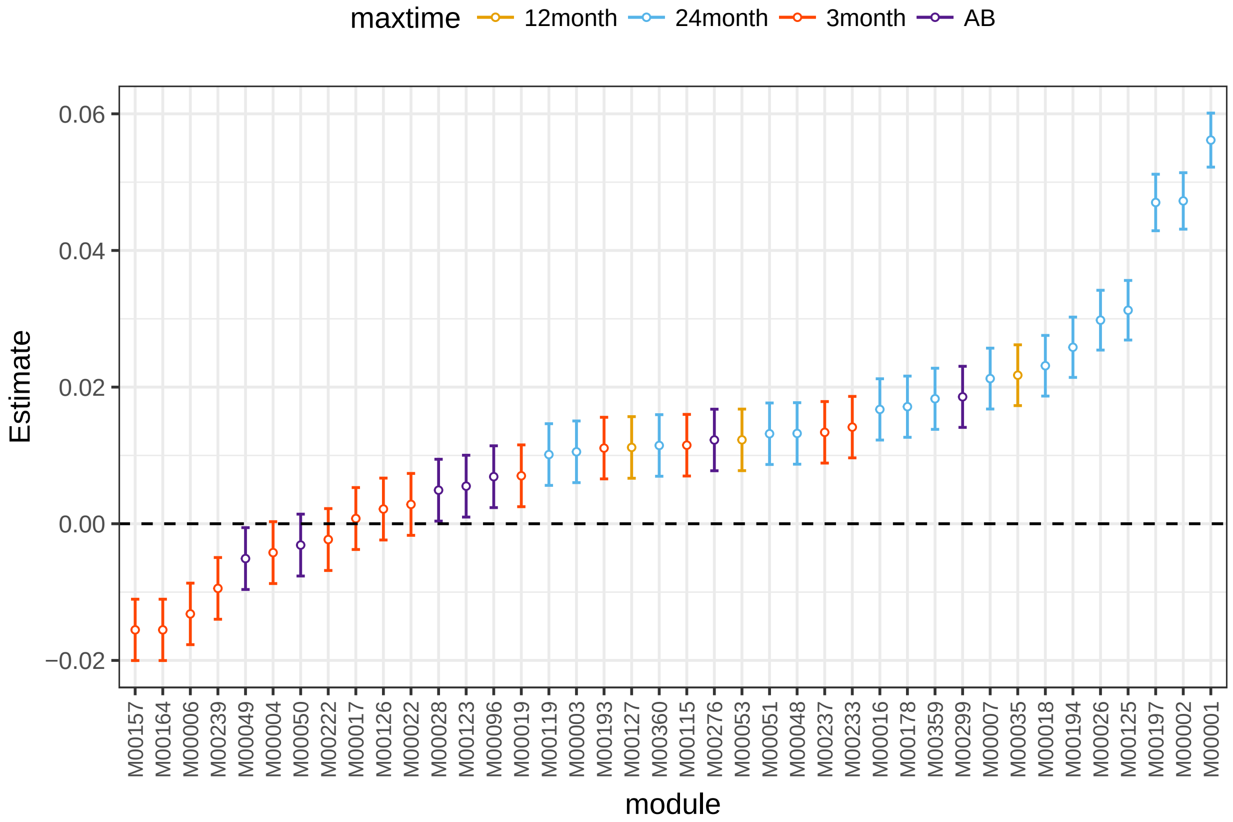
```
## Warning in class(X) <- NULL: Setting class(x) to NULL; result will no
## longer be an S4 object
```

```
# 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_unfiltered$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$time<-Taxonomy_unfiltered$time[match(rownames(Norm_log10_Module_abundance),
# 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_rainbo
# ggplot(Module_rainbow, aes(module, Estimate,colour=module))+geom_line()+geom_errorbar(ymin=Module_rai

# 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

# Showing the estimation of all modules with timpoint groups indicated.
```

```r
# ggplot(Module_rainbow, aes(module, Estimate,colour=maxtime))+geom_line()+geom_errorbar(aes(ymin=Modul

# 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$module]# 100 modules
# calculate and select the top 40 abundant modules from the module*time figure made for all modules
Top_40_abundant_module_names<-as.data.frame(sort(apply(Abundant_presence_module_filtered, 2, mean), deci
Top_40_abundant_module_names$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_names
# plot the top 40 modules
ggplot(Top_40_abundant_modules, aes(module, Estimate,colour=maxtime))+geom_line()+geom_errorbar(aes(ymin
```

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



```
ggsave("~/PIP2018/results/Fig2B-unfiltered.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?
```

```
#keep<-subset(Module_rainbow, Module_rainbow$`Pr(>|t|)` <= 0.05)
#write.csv(keep, file="~/PIP2018/results/suptable2-unfiltered-mods.csv")

#write.csv(Module_filtered, "~/PIP2018/derived-data/modules-filtered.csv")
```

Filter pathways the same way modules & taxa were filtered

```
Pathways<-import_qiime_sample_data("~/PIP/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_unfiltered$Sample,]
#write.csv(Pathways_filtered, "~/PIP2018/derived-data/pathways-filtered.csv")
```

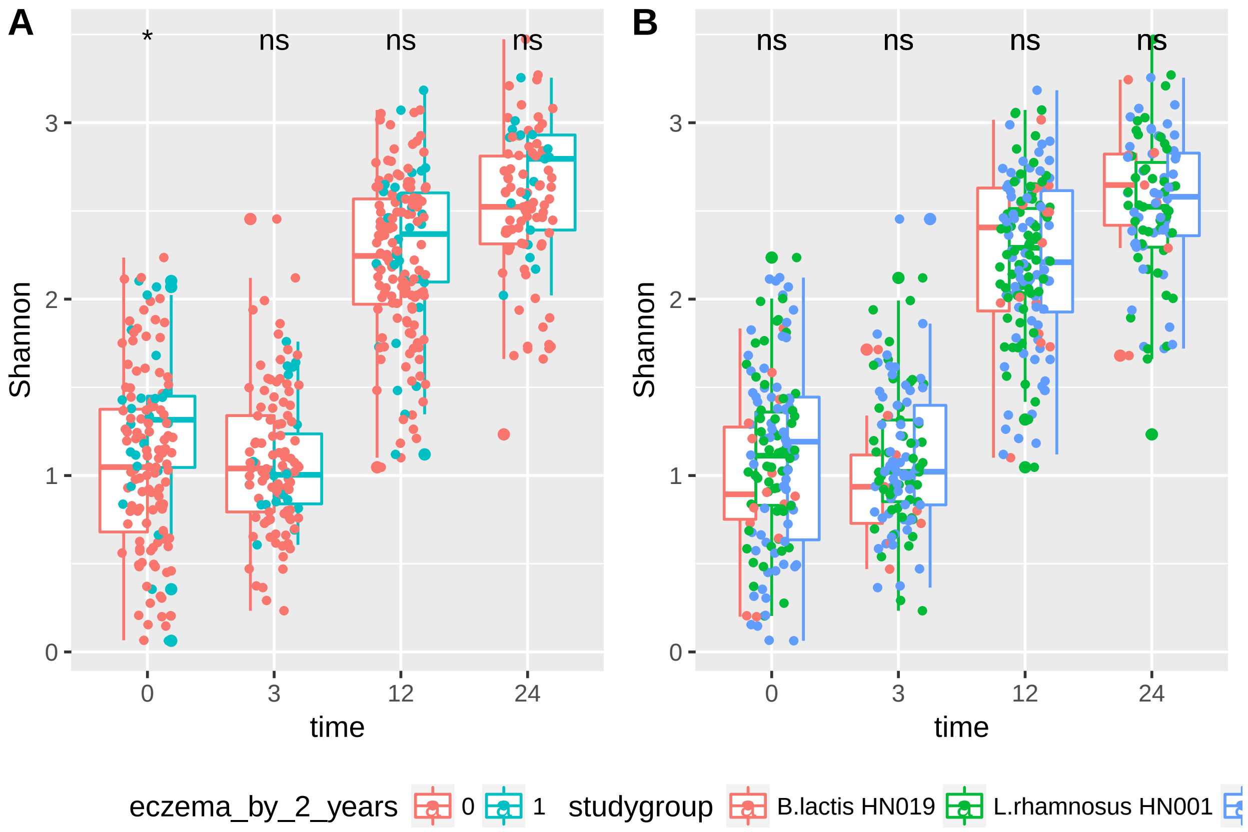
Supplemental Plot: Understanding alpha diversity in the dataset

```
#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]
# by commenting out this line, use all data, not filtered data only
#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_unfiltered[,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'), summari

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=

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



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

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

```
# Create genera with no pseudocounts

# remove the metadata part and left only taxonomy abundance data
# we use initial data, not filtered data here
Taxonomy_filtered_num<-NZGL_taxonomy[,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)]
```

```r
#Are c-section, time, eczema, studygroup significant contributors to beta diversity? (in full data)
taxonomy_genera<-my_genera
# change this line of code to use unfiltered data
meta<-NZGL_taxonomy[,1:28]
meta<-as.data.frame(as.matrix(meta))
```

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

```r
#Overall permanova effects
foo<-adonis(taxonomy_genera~time + caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months
```

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

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

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

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

```r
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    34.474 11.4914  49.894 0.18635  0.001
## caesar                         1     2.330  2.3296  10.115 0.01259  0.001
## eczema_by_2_years              1     0.290  0.2901   1.259 0.00157  0.235
## studygroup                     2     0.706  0.3530   1.532 0.00382  0.095
## Antibiotics_before_3_months    1     0.373  0.3730   1.619 0.00202  0.120
## Any_smoking_during_pregnancy   1     0.378  0.3779   1.641 0.00204  0.108
## Any_pet_at_birth               1     0.191  0.1907   0.828 0.00103  0.521
## Residuals                    635   146.251  0.2303         0.79058
## Total                        645   184.993                 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
```

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

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

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

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

```
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    34.474 11.4914  49.894 0.18635  0.001
## caesar                         1     2.330  2.3296  10.115 0.01259  0.001
## eczema_by_2_years              1     0.290  0.2901   1.259 0.00157  0.243
## studygroup                     2     0.706  0.3530   1.532 0.00382  0.111
## Antibiotics_before_3_months    1     0.373  0.3730   1.619 0.00202  0.102
## Any_smoking_during_pregnancy   1     0.378  0.3779   1.641 0.00204  0.129
## Any_pet_at_birth               1     0.191  0.1907   0.828 0.00103  0.564
## Residuals                    635   146.251  0.2303         0.79058
## Total                        645   184.993                 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
```

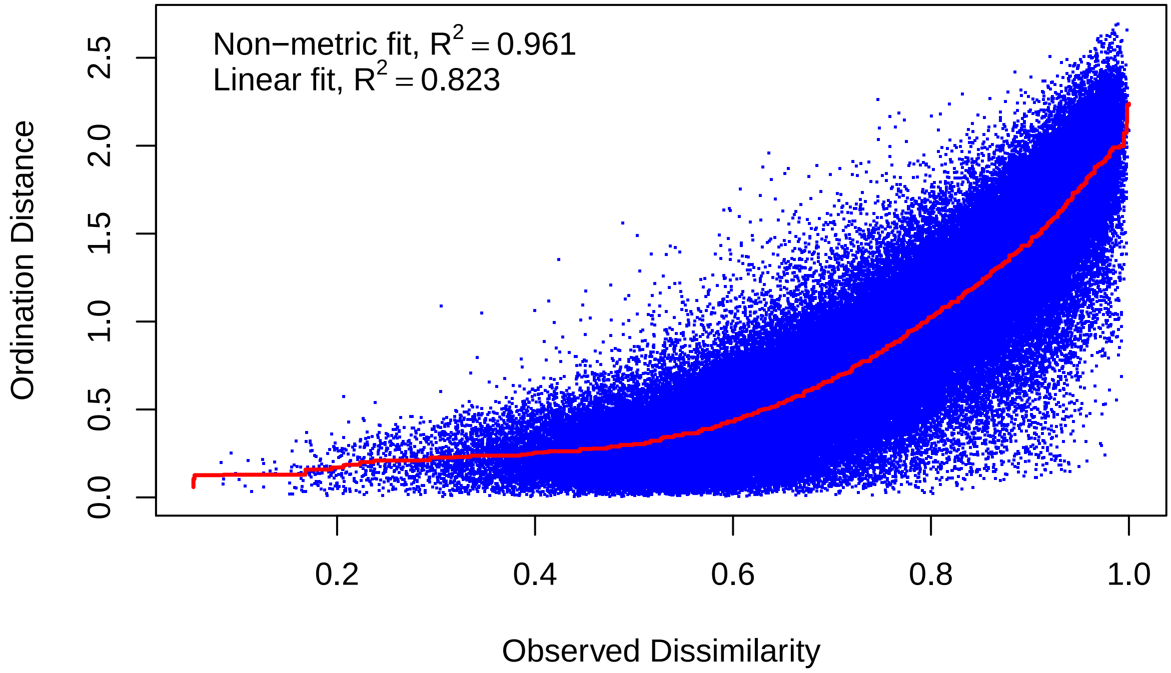
```
# Make figure 1
test<-otu_table(taxonomy_genera, taxa_are_rows = FALSE)
```

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

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

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1984467
## Run 1 stress 0.2140767
## Run 2 stress 0.2112492
## Run 3 stress 0.2198976
## Run 4 stress 0.2106339
## Run 5 stress 0.2090107
## Run 6 stress 0.2038647
## Run 7 stress 0.2089369
## Run 8 stress 0.2176731
## Run 9 stress 0.4198666
## Run 10 stress 0.2055612
## Run 11 stress 0.2100283
## Run 12 stress 0.2126074
## Run 13 stress 0.2123933
## Run 14 stress 0.2067599
## Run 15 stress 0.2158852
## Run 16 stress 0.2012339
## Run 17 stress 0.2193652
## Run 18 stress 0.4198746
## Run 19 stress 0.2106233
## Run 20 stress 0.206813
## *** No convergence -- monoMDS stopping criteria:
##      1: no. of iterations >= maxit
##     19: stress ratio > sratmax
```

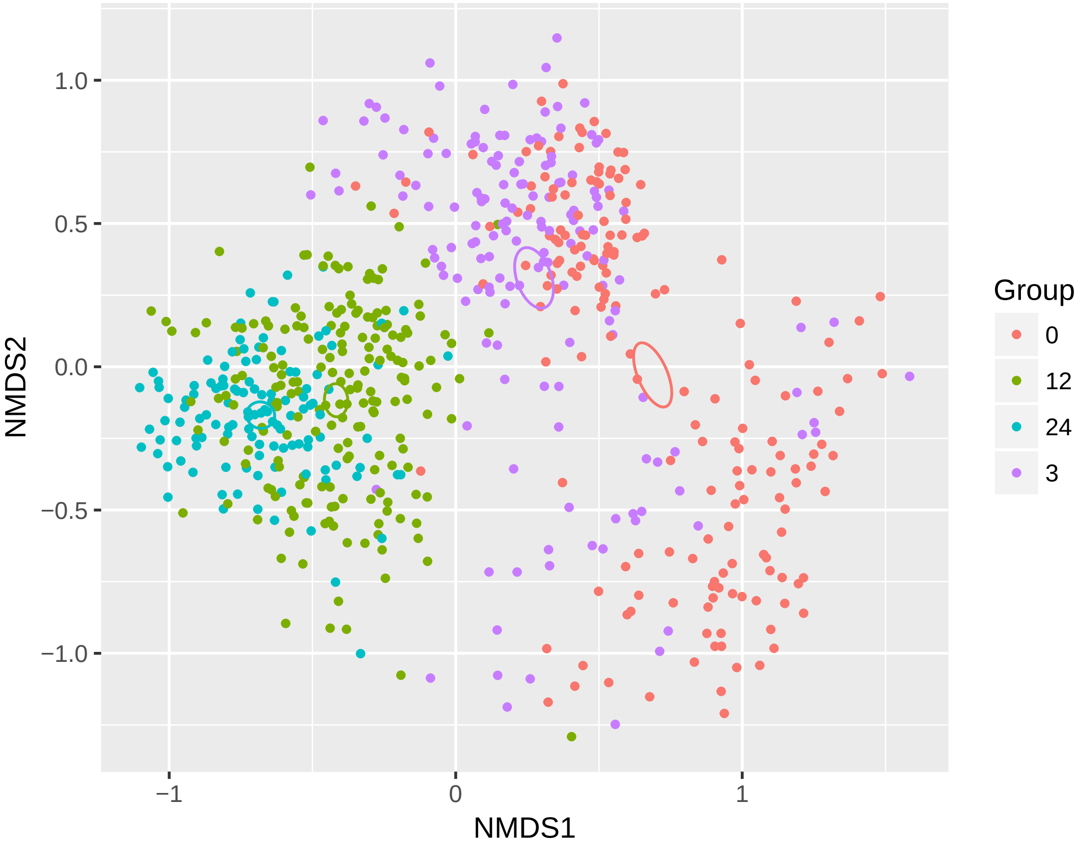
```
stressplot(mds)
```



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

```
## [1] 0.1984467
```

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



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

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

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

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

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

```
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     4.695  4.6952 15.7288 0.08354  0.001
## eczema_by_2_years              1     0.117  0.1174  0.3933 0.00209  0.889
## studygroup                     2     0.535  0.2675  0.8960 0.00952  0.510
## Antibiotics_before_3_months    1     0.179  0.1794  0.6009 0.00319  0.699
## Any_smoking_during_pregnancy   1     0.337  0.3365  1.1274 0.00599  0.324
## Any_pet_at_birth               1     0.193  0.1926  0.6453 0.00343  0.654
## Residuals                    168    50.149  0.2985         0.89225
## Total                        175    56.205                 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
```

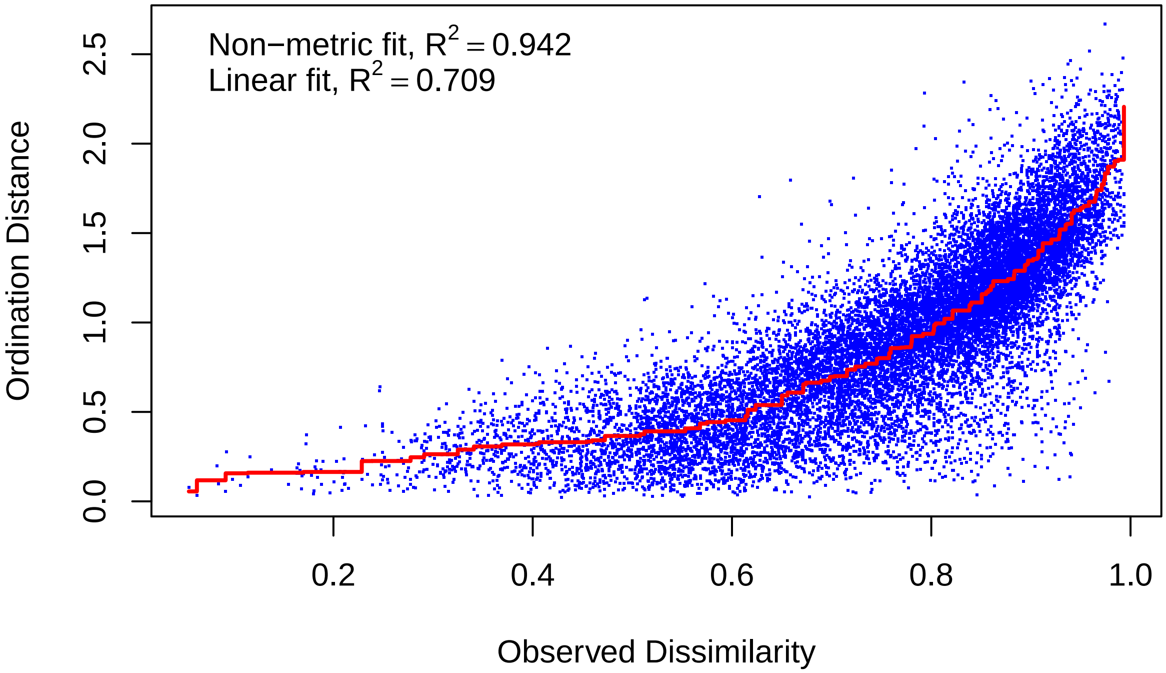
```
mds<-metaMDS(taxo_g0, dist="bray", k=2)
```

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

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

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.240766
## Run 1 stress 0.2545213
## Run 2 stress 0.2424785
## Run 3 stress 0.2619181
## Run 4 stress 0.2695917
## Run 5 stress 0.2650409
## Run 6 stress 0.2743378
## Run 7 stress 0.2453508
## Run 8 stress 0.2556785
## Run 9 stress 0.2428531
## Run 10 stress 0.2638116
## Run 11 stress 0.2593198
## Run 12 stress 0.2520143
## Run 13 stress 0.2757661
## Run 14 stress 0.2671207
## Run 15 stress 0.2766051
## Run 16 stress 0.2523408
## Run 17 stress 0.2600899
## Run 18 stress 0.2781436
## Run 19 stress 0.2406149
## ... New best solution
## ... Procrustes: rmse 0.01394258  max resid 0.118188
## Run 20 stress 0.2570309
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

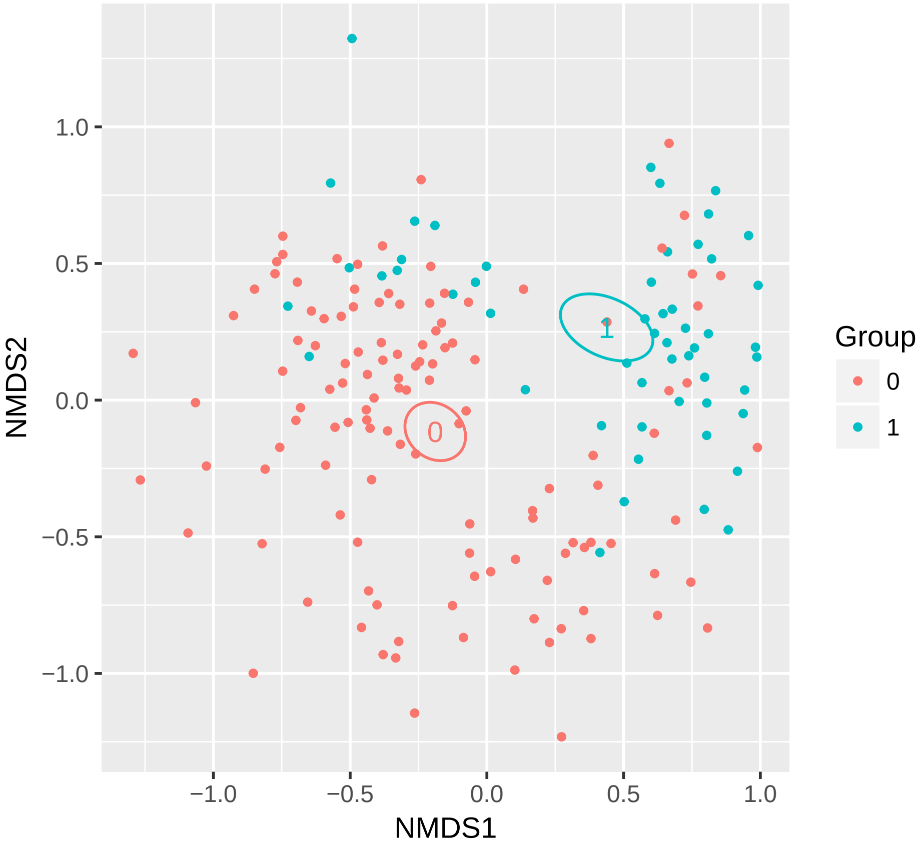
```
stressplot(mds)
```



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

```
## [1] 0.2406149
```

```
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, meta$time == 3)
meta3<-subset(meta, meta$time == 3)
foo<-adonis(taxo_g3~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smoking_
```

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

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

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

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

```
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     1.123 1.12275  5.6366 0.03485  0.001
## eczema_by_2_years                1     0.126 0.12566  0.6309 0.00390  0.643
## studygroup                     2     0.740 0.37018  1.8584 0.02298  0.063
## Antibiotics_before_3_months    1     0.424 0.42359  2.1266 0.01315  0.076
## Any_smoking_during_pregnancy   1     0.432 0.43209  2.1692 0.01341  0.069
## Any_pet_at_birth               1     0.094 0.09400  0.4719 0.00292  0.801
## Residuals                    147    29.281 0.19919         0.90880
## Total                        154    32.219                 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
```

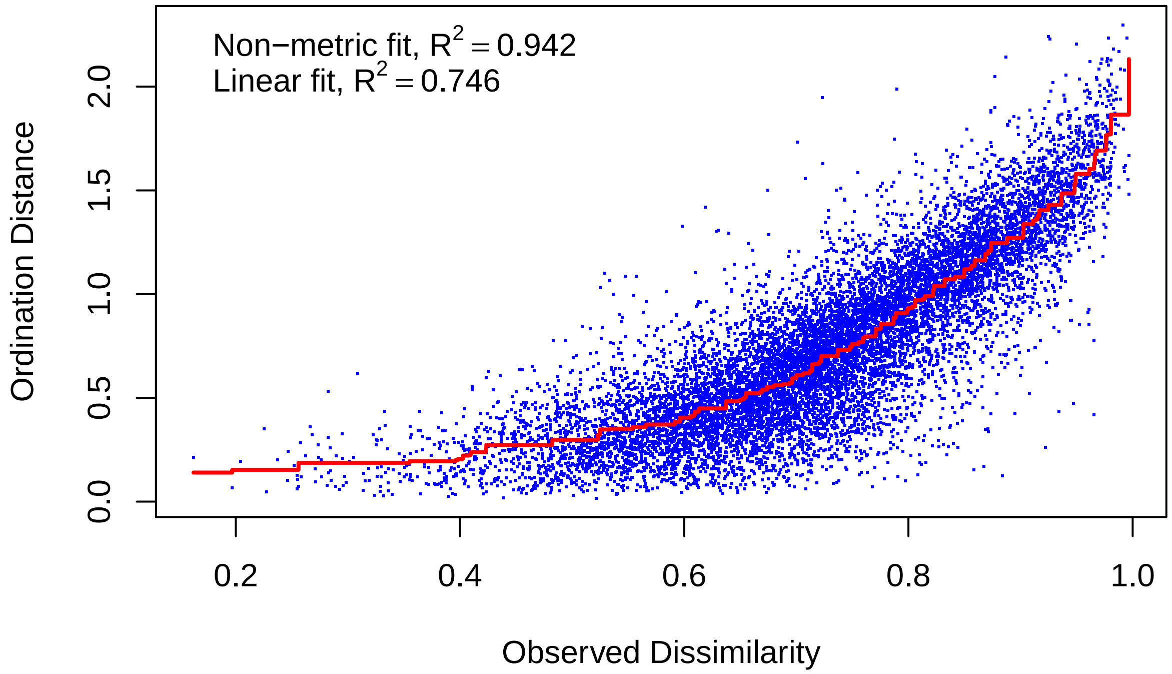
```
mds<-metaMDS(taxo_g3, dist="bray", k=2)
```

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

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

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2428796
## Run 1 stress 0.2562052
## Run 2 stress 0.2437331
## Run 3 stress 0.2441582
## Run 4 stress 0.2500696
## Run 5 stress 0.2599483
## Run 6 stress 0.2467636
## Run 7 stress 0.2504782
## Run 8 stress 0.415355
## Run 9 stress 0.2412351
## ... New best solution
## ... Procrustes: rmse 0.03140129  max resid 0.2421646
## Run 10 stress 0.2542495
## Run 11 stress 0.2529525
## Run 12 stress 0.242025
## Run 13 stress 0.2456651
## Run 14 stress 0.2468717
## Run 15 stress 0.2521026
## Run 16 stress 0.2482103
## Run 17 stress 0.2421731
## Run 18 stress 0.2450527
## Run 19 stress 0.2522095
## Run 20 stress 0.2480045
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

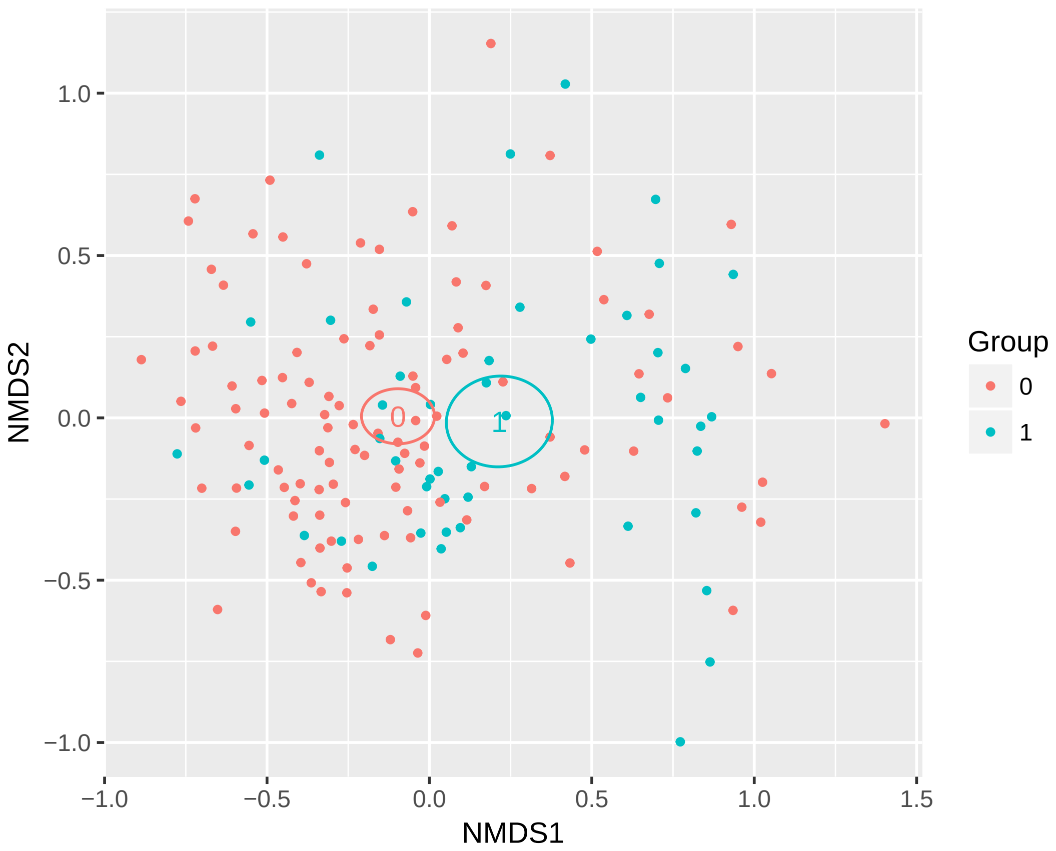
```
stressplot(mds)
```



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

```
## [1] 0.2412351
```

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



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

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

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

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

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

```r
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     0.252 0.25212 1.17806 0.00613  0.276
## eczema_by_2_years               1     0.208 0.20778 0.97090 0.00505  0.453
## studygroup                     2     0.228 0.11392 0.53229 0.00554  0.930
## Antibiotics_before_3_months    1     0.145 0.14465 0.67590 0.00352  0.674
## Any_smoking_during_pregnancy   1     0.062 0.06215 0.29039 0.00151  0.979
## Any_pet_at_birth               1     0.447 0.44653 2.08644 0.01085  0.057
## Residuals                    186    39.807 0.21401         0.96741
## Total                        193    41.148                 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
```

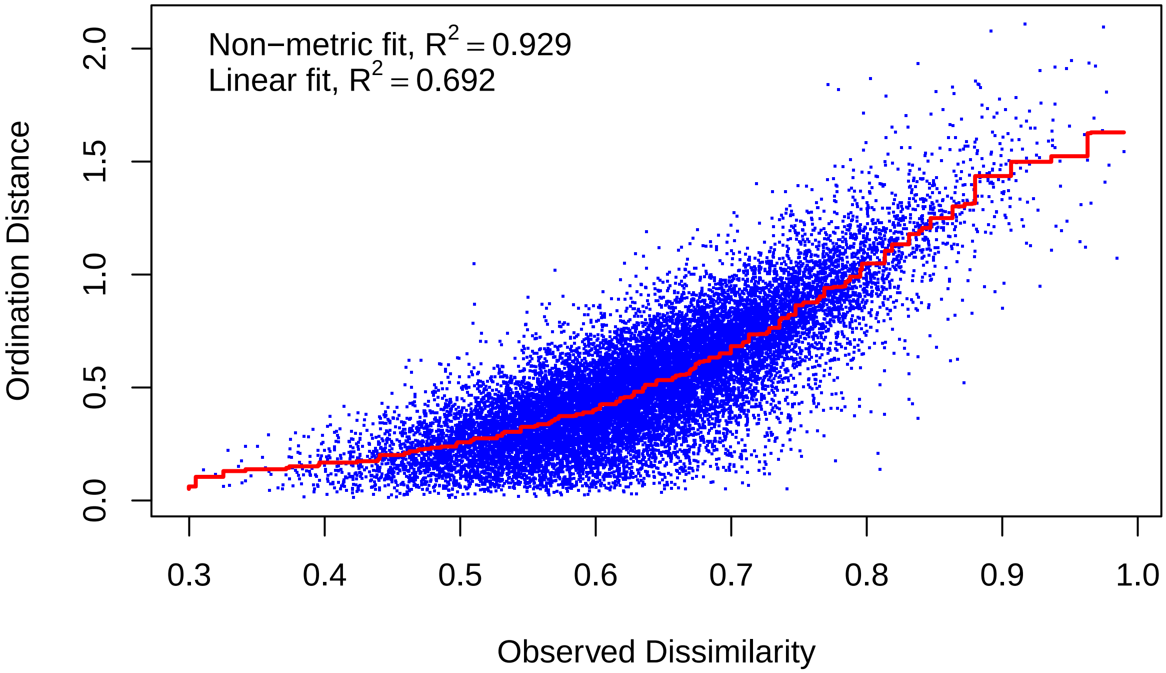
```
mds<-metaMDS(taxo_g12, dist="bray", k=2)
```

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

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

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2666865
## Run 1 stress 0.267441
## Run 2 stress 0.2688729
## Run 3 stress 0.2669704
## ... Procrustes: rmse 0.02303378  max resid 0.2797917
## Run 4 stress 0.2853484
## Run 5 stress 0.2750307
## Run 6 stress 0.2666884
## ... Procrustes: rmse 0.001836167  max resid 0.01621896
## Run 7 stress 0.2716019
## Run 8 stress 0.2741275
## Run 9 stress 0.2725327
## Run 10 stress 0.2815153
## Run 11 stress 0.2701165
## Run 12 stress 0.2717924
## Run 13 stress 0.270131
## Run 14 stress 0.27389
## Run 15 stress 0.2769053
## Run 16 stress 0.2713787
## Run 17 stress 0.2686142
## Run 18 stress 0.2689046
## Run 19 stress 0.2727697
## Run 20 stress 0.2765624
## *** No convergence -- monoMDS stopping criteria:
##      1: no. of iterations >= maxit
##     19: stress ratio > sratmax
```

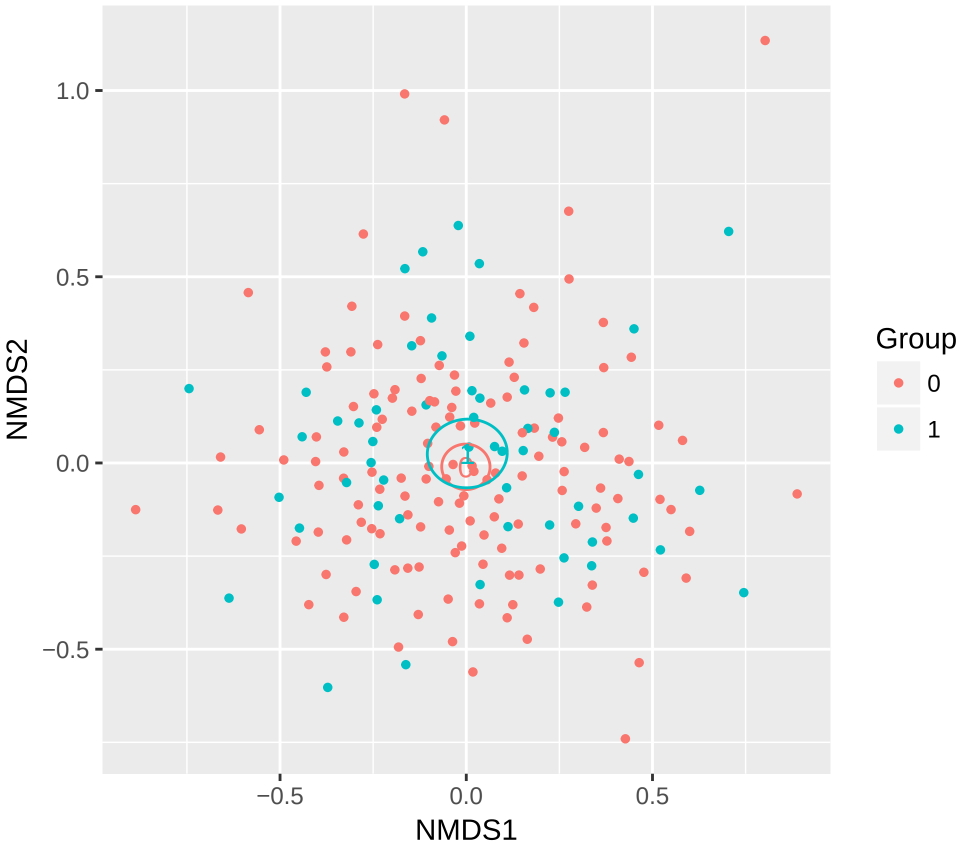
```
stressplot(mds)
```



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

```
## [1] 0.2666865
```

```
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, meta$time == 24)
meta24<-subset(meta, meta$time == 24)
foo<-adonis(taxo_g24~caesar + eczema_by_2_years + studygroup + Antibiotics_before_3_months + Any_smokin
```

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

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

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

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

```
print(foo$aov.tab)
```

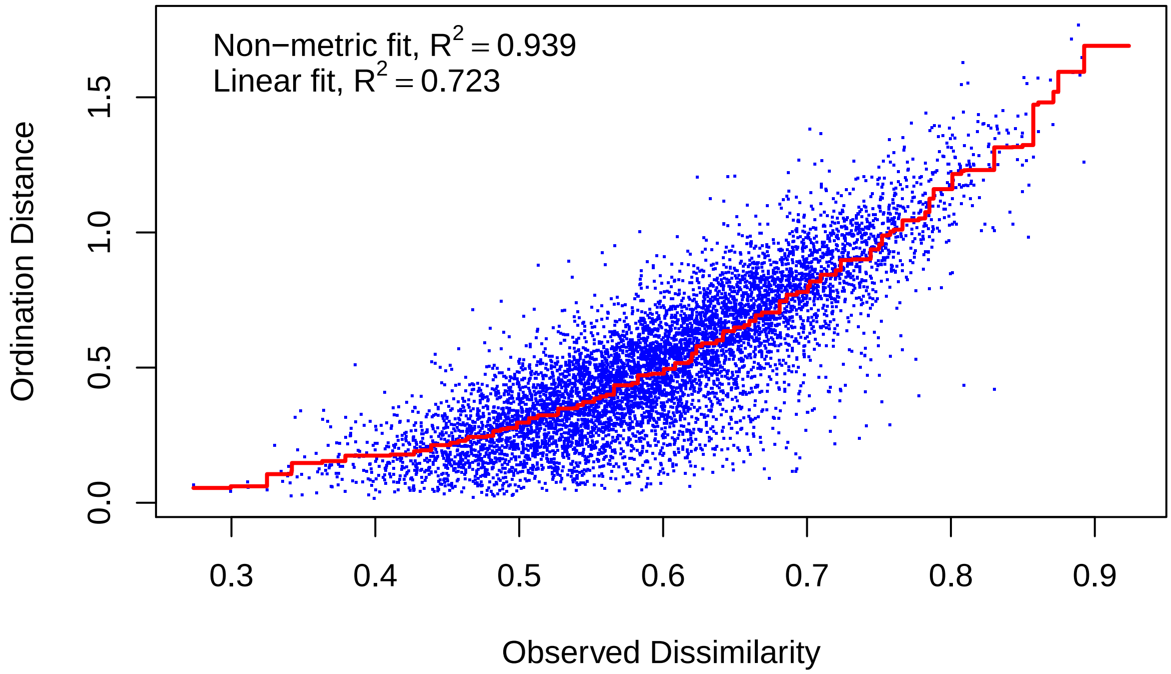
```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                               Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## caesar                         1    0.1584 0.158442 0.89839 0.00756  0.530
## eczema_by_2_years              1    0.1358 0.135797 0.76999 0.00648  0.635
## studygroup                     2    0.3107 0.155354 0.88088 0.01483  0.597
## Antibiotics_before_3_months    1    0.1220 0.122020 0.69187 0.00583  0.701
## Any_smoking_during_pregnancy   1    0.1909 0.190910 1.08249 0.00911  0.347
## Any_pet_at_birth               1    0.0995 0.099516 0.56427 0.00475  0.819
## Residuals                    113   19.9289 0.176362         0.95143
## Total                        120   20.9463                  1.00000
```

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

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

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

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2475038
## Run 1 stress 0.2560973
## Run 2 stress 0.2619714
## Run 3 stress 0.2518723
## Run 4 stress 0.2504158
## Run 5 stress 0.2642249
## Run 6 stress 0.2495326
## Run 7 stress 0.2497836
## Run 8 stress 0.2543287
## Run 9 stress 0.2484823
## Run 10 stress 0.2598734
## Run 11 stress 0.2496944
## Run 12 stress 0.255745
## Run 13 stress 0.2523392
## Run 14 stress 0.2584488
## Run 15 stress 0.2501223
## Run 16 stress 0.2598109
## Run 17 stress 0.2574176
## Run 18 stress 0.2518641
## Run 19 stress 0.2654111
## Run 20 stress 0.2503251
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

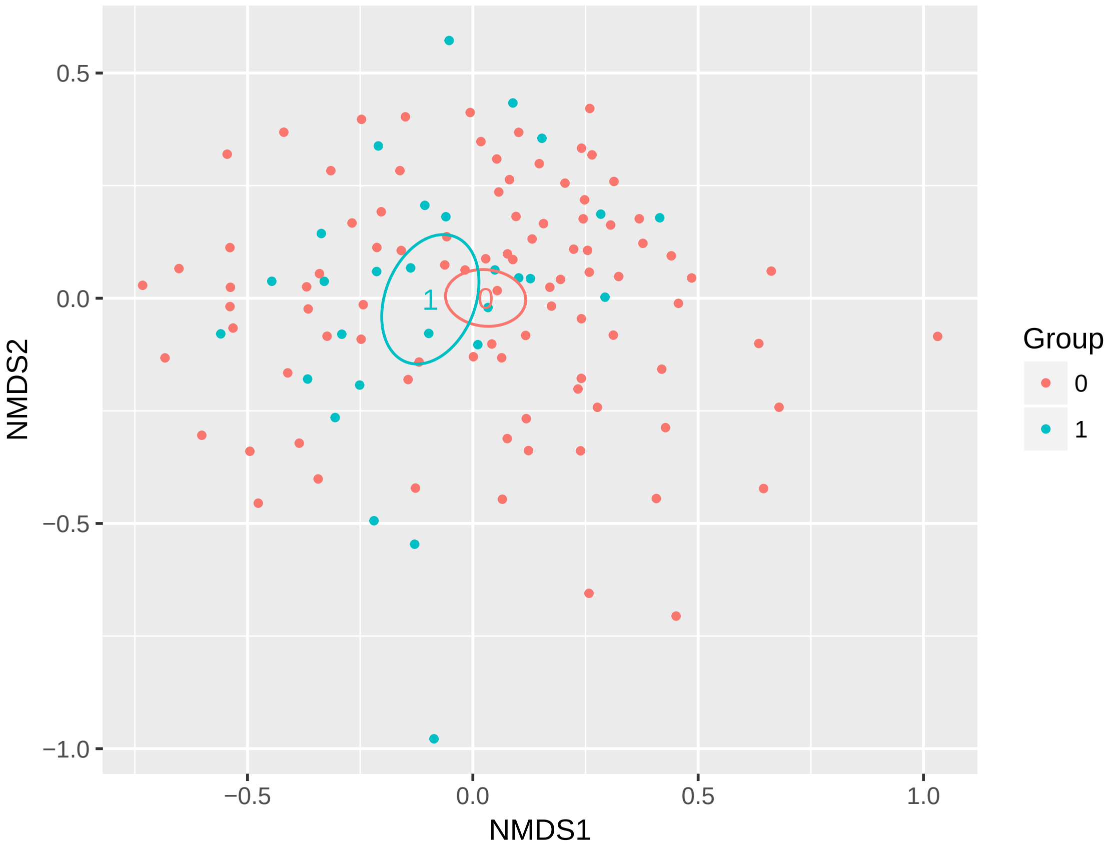
```
stressplot(mds)
```



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

```
## [1] 0.2475038
```

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

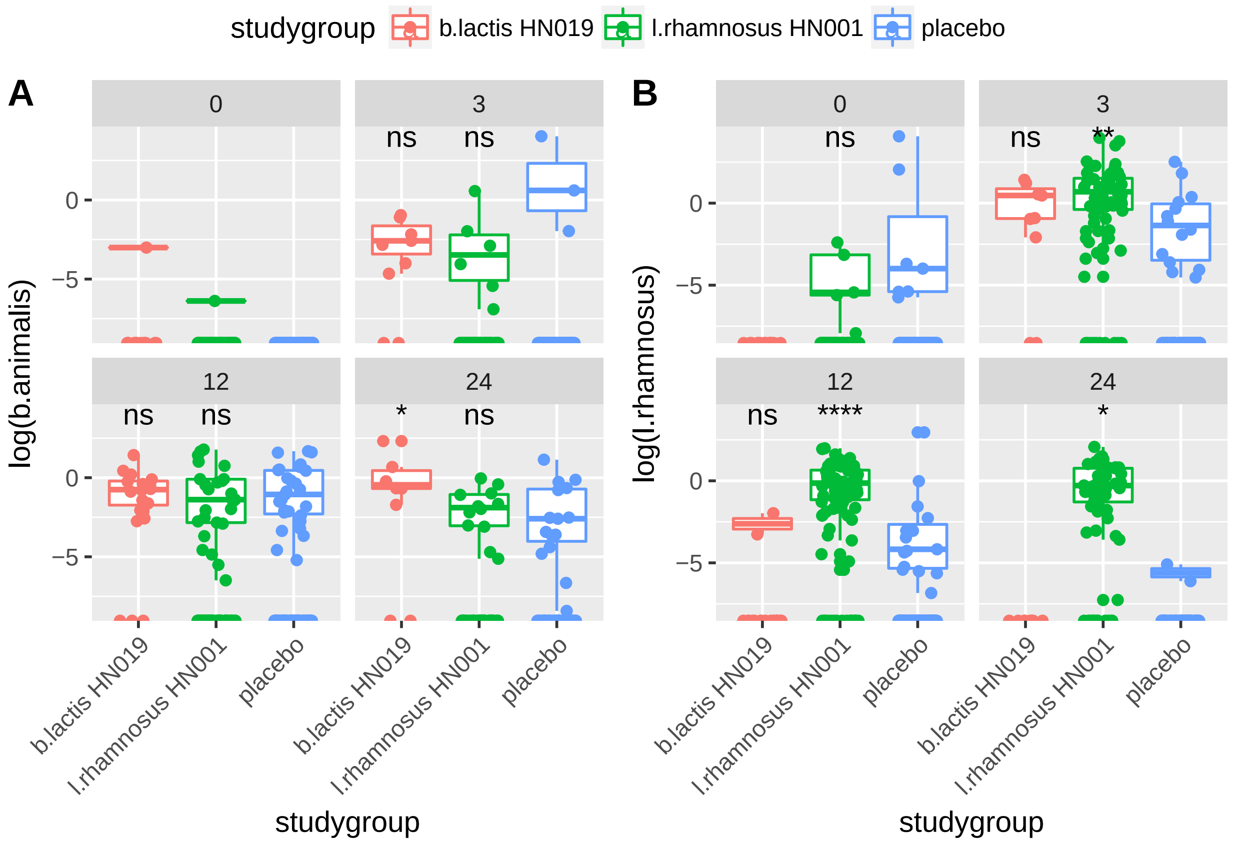


```
A<-fig1A$plot

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), labe
bar<-ggarrange(A, foo, ncol=2, labels=c("A", "B"), legend=c("bottom"), widths=c(1, 1.5))


ggsave("~/PIP2018/results/Fig1-unfiltered.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
```

```
# Add next line to switch all instances of Taxononmy_filtered to NZGL taxonomy (unfiltered)
Taxonomy_filtered<-NZGL_taxonomy
#boxplot(Taxonomy_filtered[,541] ~ Taxonomy_filtered$studygroup, ylim = c(0, 5), ylab="abundance L. rha

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


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

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

```r
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))
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 527 rows containing non-finite values (stat_boxplot).

## Warning: Removed 527 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, 2

## Warning: Removed 527 rows containing non-finite values (stat_boxplot).

## Warning: Removed 527 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, 2

## Warning: Removed 425 rows containing non-finite values (stat_boxplot).

## Warning: Removed 425 rows containing non-finite values
## (stat_compare_means).
```



```r
ggsave("~/PIP2018/results/SupFig3-unfiltered.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

##
```

```
##  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  0.42896          -                   -
## placebo.0            0.37779          0.72259             -
## b.lactis HN019.3     0.00040          3.9e-09             2.5e-08
## l.rhamnosus HN001.3  1.1e-06          < 2e-16             < 2e-16
## placebo.3            0.13859          0.03493             0.06926
## b.lactis HN019.12    0.33097          0.59948             0.71722
## l.rhamnosus HN001.12 2.9e-06          < 2e-16             < 2e-16
## placebo.12           0.16325          0.07738             0.14436
## b.lactis HN019.24    -                0.54046             0.49783
## l.rhamnosus HN001.24 1.9e-05          5.8e-16             6.4e-16
## placebo.24           0.57424          0.48584             0.33097
##                      b.lactis HN019.3 l.rhamnosus HN001.3 placebo.3
## l.rhamnosus HN001.0  -                -                   -
## placebo.0            -                -                   -
## b.lactis HN019.3     -                -                   -
## l.rhamnosus HN001.3  0.40326          -                   -
## placebo.3            6.7e-05          6.1e-16             -
## b.lactis HN019.12    0.00040          8.2e-08             0.47581
## l.rhamnosus HN001.12 0.80934          0.00737             3.6e-14
## placebo.12           3.5e-06          < 2e-16             0.61925
## b.lactis HN019.24    0.00601          0.00017             0.29115
## l.rhamnosus HN001.24 0.63961          0.00335             1.5e-10
## placebo.24           1.5e-09          < 2e-16             0.01041
##                      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 3.5e-07           -                    -
## placebo.12           0.61925           < 2e-16              -
## b.lactis HN019.24    0.47581           0.00034              0.31887
## l.rhamnosus HN001.24 4.7e-06           0.49783              3.6e-14
## placebo.24           0.30475           < 2e-16              0.02140
##                      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 0.00097           -
## placebo.24           0.65293           2.1e-14
##
## 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

##
##  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  0.22499          -                   -
## placebo.0            0.02158          0.32662             -
## b.lactis HN019.3     0.00150          1.4e-12             7.6e-16
## l.rhamnosus HN001.3  0.88411          0.05806             0.00926
## placebo.3            0.67051          0.34954             0.08696
## b.lactis HN019.12    5.7e-05          1.1e-15             < 2e-16
## l.rhamnosus HN001.12 0.08884          4.1e-06             2.5e-07
## placebo.12           0.08884          4.0e-06             2.5e-07
## b.lactis HN019.24    0.00161          7.9e-12             2.0e-15
## l.rhamnosus HN001.24 0.23959          0.00022             1.6e-05
## placebo.24           0.18265          5.3e-05             3.1e-06
##                      b.lactis HN019.3 l.rhamnosus HN001.3 placebo.3
## l.rhamnosus HN001.0  -                -                   -
## placebo.0            -                -                   -
## b.lactis HN019.3     -                -                   -
## l.rhamnosus HN001.3  4.7e-07          -                   -
## placebo.3            3.7e-09          0.32647             -
## b.lactis HN019.12    0.03490          1.7e-11             5.1e-13
## l.rhamnosus HN001.12 0.03092          0.00144             8.3e-05
## placebo.12           0.04476          0.00124             7.7e-05
## b.lactis HN019.24    0.11673          7.3e-07             7.5e-09
## l.rhamnosus HN001.24 0.00374          0.03906             0.00374
## placebo.24           0.00473          0.01469             0.00111
##                      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 2.7e-05            -                  -
## placebo.12           4.4e-05            0.93078            -
## b.lactis HN019.24    0.65240            0.00543            0.00545
## l.rhamnosus HN001.24 7.3e-07            0.30872            0.27021
## placebo.24           1.2e-06            0.43572            0.37494
##                      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 0.00061            -
## placebo.24           0.00133            0.80948
##
## P value adjustment method: BH
```

Supplementary Figure 4

```r
#Code corrected to use unfiltered data
#collect objects for ggplotting
#modules<-read.table("~/PIP2018/derived-data/filtered_modules.tsv", header=TRUE, sep="\t")
modules<-read.table("~/PIP2018/primary_data/modules.pcl", header=TRUE, row.names=1, sep="\t")
modules<-t(modules)
modules<-as.data.frame(modules)
modules$Sample = rownames(modules)
#tax<-read.table("~/PIP2018/derived-data/filtered_taxonomy.tsv", header=TRUE, sep="\t")
tax<-read.table("~/PIP2018/primary_data/taxonomy.tsv", header=TRUE, sep="\t")
#pathways<-read.table("~/PIP2018/derived-data/filtered_pathways.tsv", header=TRUE, sep="\t")

pathways<-read.table("~/PIP2018/primary_data/pathways.pcl", header=TRUE, sep="\t", row.names=1)
pathways<-t(pathways)
pathways<-as.data.frame(pathways)
pathways$Sample = rownames(pathways)

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"))
colnames(m1)[7] = "B.animalis"
colnames(m1)[8] = "L.rhamnosus"
m1<-merge(x=m1, y=pwys,  by.x=c("Sample"), by.y=c("Sample"))
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)
m1$studygroup = factor(m1$studygroup)
m1$M00198 = as.numeric(as.character(m1$M00198))
m1$M00277 = as.numeric(as.character(m1$M00277))
m1$ko00531 = as.numeric(as.character(m1$ko00531))
m1$ko00240  = as.numeric(as.character(m1$ko00240))
m1$ko04141   = as.numeric(as.character(m1$ko04141))


# Make plots
#S4A
a<-ggplot(data=m1, aes(x=L.rhamnosus, y=M00198, colour=studygroup)) + geom_point() + facet_wrap(~studyg
    theme(legend.position="bottom")
# S4B
b1<-ggplot(data=m1, aes(x=time, y=M00277, colour=eczema_by_2_years)) + geom_boxplot() + stat_compare_me
b2<-ggplot(data=m1, aes(x=time, y=log(ko00531), colour=eczema_by_2_years)) + geom_boxplot() +   stat_co
b3<-ggplot(data=m1, aes(x=time, y=log(ko00240), colour=eczema_by_2_years)) + geom_boxplot() +   stat_co
b4<-ggplot(data=m1, aes(x=time, y=log(ko04141), colour=eczema_by_2_years)) + geom_boxplot() +   stat_co
panelb<-ggarrange(b1, b2, b3, b4, common.legend = TRUE, legend="bottom", labels=c("B"))
```

```
## Warning: Removed 20 rows containing non-finite values (stat_boxplot).

## Warning: Removed 20 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 9 rows containing non-finite values (stat_smooth).

## Warning: Removed 9 rows containing non-finite values (stat_cor).

## Warning: Removed 9 rows containing missing values (geom_point).

## Warning: Removed 425 rows containing non-finite values (stat_boxplot).

## Warning: Removed 425 rows containing non-finite values
## (stat_compare_means).
```

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