Rmarkdown: Dietary carotenoid supplementation has long-term and community-wide effects on the amphibian skin microbiome

Alice Risely

29//03/23

Contents

Load packages

```
library(phyloseq)
library(ggplot2)
library(vegan)
library(dplyr)
library(scales)
library(grid)
library(reshape2)
library(ape)
library(gridExtra)
library(ade4)
library(plyr)
library(tidyr)
library(data.table)
library(stringr)
library(ggrepel)
library(r2glmm)
library(ggvenn)
library(viridis)
library(ggord)


library(GGally)
library(sjPlot)
library(performance)

library(ggpubr)
library(microbiome)
library(RColorBrewer)
library(gllvm)
library(EnvStats)
```

Prevalence function

Load data

```
setwd("C:/Users/risel/Dropbox/Academic projects/Frog microbiome UOW/Frogs_UOW/Diet treatment project/Ana

frog_biom <-phyloseq::import_biom("frog.biom")
frog_map <-phyloseq::import_qiime_sample_data ('frog_metadata.txt')
tree<-phyloseq::read_tree("frog.tree")
frog_ps <-phyloseq::merge_phyloseq(frog_biom, frog_map, tree)
frog_ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6093 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 6093 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 6093 tips and 6093 internal nodes ]
```

```
## fix metadata

sample_data(frog_ps)$Mass<-as.numeric(gsub(',','.',sample_data(frog_ps)$Mass))

frogs_sex <- read.csv("Data and code/frogs_sex.csv")

frogs_sex<-frogs_sex[,1:5]

names(frogs_sex)[1]<-"frog_id"
frogs_sex<-frogs_sex[,c(1,5)]

sample_data(frog_ps)$Sex<-expss::vlookup(sample_data(frog_ps)$frog_id, frogs_sex, lookup_column = "frog_

sample_data(frog_ps)$Sex<-factor(sample_data(frog_ps)$Sex, levels = c("F", "M"))
sample_data(frog_ps)$date_cat<-ifelse(sample_data(frog_ps)$date=="10.04.2018", "Date1", "Date2")

sample_data(frog_ps)$date <-as.Date( sample_data(frog_ps)$date, "%d.%m.%Y")

write.csv(data.frame(sample_data(frog_ps)), "frog_metadata_clean.csv")
```

Filter taxa

```
colnames(tax_table(frog_ps)) <- c("Kingdom", "Phylum", "Class",
                                     "Order", "Family", "Genus", "Species")

###get rid of the D_1 etc characters infront of tax names

taxonomy<-data.frame(tax_table(frog_ps))


taxonomy$Kingdom<-substr(taxonomy$Kingdom, 6, 60)
taxonomy$Phylum<-substr(taxonomy$Phylum, 6, 60)
taxonomy$Class<-substr(taxonomy$Class, 6, 60)
```

```
taxonomy$Order<-substr(taxonomy$Order, 6, 60)
taxonomy$Family<-substr(taxonomy$Family, 6, 60)
taxonomy$Genus<-substr(taxonomy$Genus, 6, 60)
taxonomy$Species<-substr(taxonomy$Species, 6, 60)



taxonomy1<-tax_table(as.matrix(taxonomy))
tax_table(frog_ps)<-taxonomy1


##filter taxa that are not bacteria or not assigned at phylum level

frog_filtered <- frog_ps %>%
  subset_taxa(
    Kingdom == "Bacteria")
frog_filtered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6089 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 6089 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 6089 tips and 6088 internal nodes ]
```

```
frog_filtered <- frog_filtered %>%
  subset_taxa(
    Phylum != "NA")
frog_filtered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6001 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 6001 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 6001 tips and 6000 internal nodes ]
```

```
frog_filtered <- frog_filtered %>%
  subset_taxa(
    Family != "Mitochondria")
frog_filtered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 5776 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 5776 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 5776 tips and 5775 internal nodes ]
```

```
frog_filtered <- frog_filtered %>%
  subset_taxa(
    Class != "Chloroplast")

frog_filtered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 5776 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 5776 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 5776 tips and 5775 internal nodes ]
```

```
frog_ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6093 taxa and 119 samples ]
## sample_data() Sample Data:       [ 119 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 6093 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 6093 tips and 6093 internal nodes ]
```

```
########## filter samples ############

frog_filtered<-prune_taxa(taxa_sums(frog_filtered)>0, frog_filtered)

1-(sum(sample_sums(frog_filtered))/sum(sample_sums(frog_ps)))
```

```
## [1] 0.001823955
```

```
# add sequencing depth to metadata

sample_data(frog_filtered)$Seq_depth <-sample_sums(frog_filtered)
summary(sample_data(frog_filtered)$Seq_depth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6708   52545   82871   98668  120237  505712
```

Replicates

```
## replicates

replicate_ps<-subset_samples(frog_filtered, Run == "BYLN3" | Run == "BYLLT")
length(unique(sample_data(replicate_ps)$frog_id))
```

```
## [1] 14
```

```
summary(sample_sums(replicate_ps))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18421   33970   49129   53273   69315  113671
```

```
replicate_rare<- rarefy_even_depth(replicate_ps, sample.size = 18000)

sample_data(replicate_rare)$Seq_run <-ifelse(sample_data(replicate_rare)$Run == "BYLLT", "Run 1", "Run
```

```r
set.seed(1)

## bray curtis


rep_bray <- ordinate(
  physeq = replicate_rare,
  method = "NMDS",
  distance = "bray"
)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.09977775
## Run 1 stress 0.09977766
## ... New best solution
## ... Procrustes: rmse 0.0004943518  max resid 0.001084299
## ... Similar to previous best
## Run 2 stress 0.09977741
## ... New best solution
## ... Procrustes: rmse 0.0001660374  max resid 0.0003792743
## ... Similar to previous best
## Run 3 stress 0.1039217
## Run 4 stress 0.09941244
## ... New best solution
## ... Procrustes: rmse 0.01826839  max resid 0.05915333
## Run 5 stress 0.1002943
## Run 6 stress 0.09941214
## ... New best solution
## ... Procrustes: rmse 0.0001284115  max resid 0.000316008
## ... Similar to previous best
## Run 7 stress 0.09941215
## ... Procrustes: rmse 0.00018313  max resid 0.000426331
## ... Similar to previous best
## Run 8 stress 0.1002944
## Run 9 stress 0.09941248
## ... Procrustes: rmse 0.00020962  max resid 0.0004901966
## ... Similar to previous best
## Run 10 stress 0.09977741
## ... Procrustes: rmse 0.01830784  max resid 0.05956637
## Run 11 stress 0.09941214
## ... New best solution
## ... Procrustes: rmse 3.990017e-05  max resid 9.279465e-05
## ... Similar to previous best
## Run 12 stress 0.09977749
## ... Procrustes: rmse 0.0183194  max resid 0.05956366
## Run 13 stress 0.1002944
## Run 14 stress 0.09941233
## ... Procrustes: rmse 0.0001466418  max resid 0.0003231208
## ... Similar to previous best
## Run 15 stress 0.09941227
## ... Procrustes: rmse 0.000141746  max resid 0.0003195454
## ... Similar to previous best
## Run 16 stress 0.1002944
## Run 17 stress 0.1002943
## Run 18 stress 0.1039217
## Run 19 stress 0.09941227
## ... Procrustes: rmse 0.0001317684  max resid 0.0003074905
## ... Similar to previous best
## Run 20 stress 0.09977755
## ... Procrustes: rmse 0.01830559  max resid 0.0595305
## *** Solution reached
```

```
ord_df<-data.frame(rep_bray$points)
rep_metadata<-data.frame(sample_data(replicate_rare))
head(rep_metadata)
```

```
##       feature.id frog_id   Run Treatment       date Mass Replicate Sex date_cat
## 115          115     115 BYLN3        UU 2018-04-10 2.95         N   F    Date1
## 115-2      115-2     115 BYLLT        UU 2018-04-10 2.95         Y   F    Date1
## 117          117     117 BYLN3        UU 2018-04-10 3.25         N   F    Date1
## 117-2      117-2     117 BYLLT        UU 2018-04-10 3.25         Y   F    Date1
## 119          119     119 BYLN3        CC 2018-04-10 3.25         N   F    Date1
## 119-2      119-2     119 BYLLT        CC 2018-04-10 3.25         Y   F    Date1
##       Seq_depth Seq_run
## 115       41790   Run 2
## 115-2     65468   Run 1
## 117       51839   Run 2
## 117-2     79929   Run 1
## 119       39556   Run 2
## 119-2     59527   Run 1
```

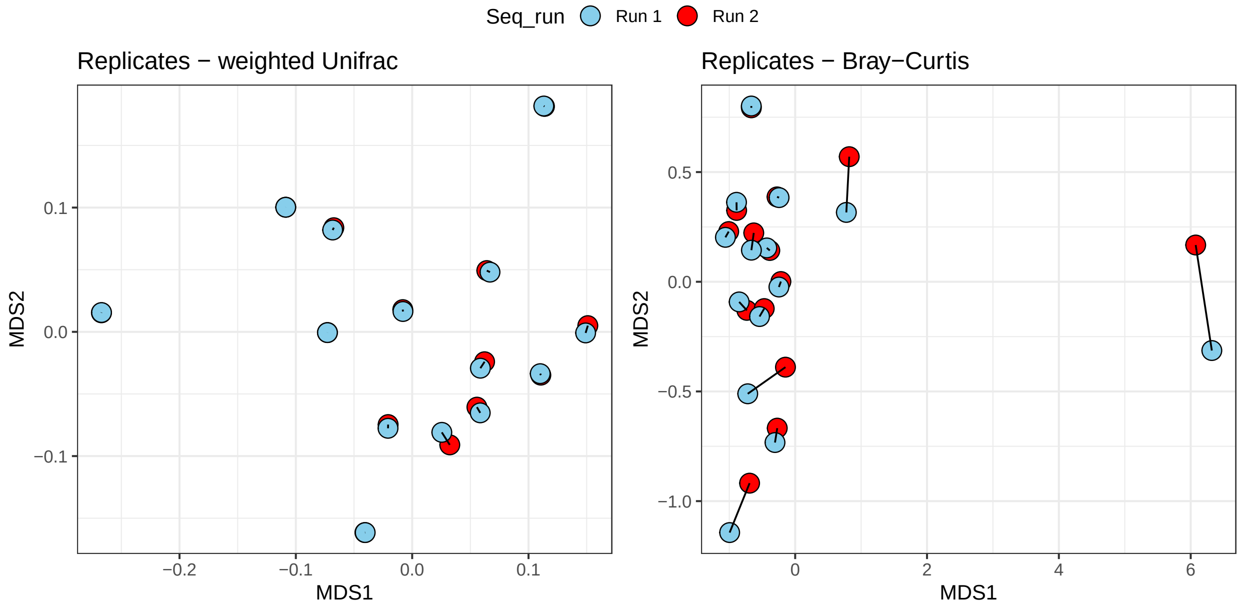
```
rep_bray_df<-merge(rep_metadata, ord_df, by = 0)

rep1<-ggplot(rep_bray_df, aes(x = MDS1, y = MDS2, fill = Seq_run, group = frog_id))+
  geom_point(pch = 21, size = 5)+
  geom_line()+
  theme_bw(base_size = 12)+
  ggtitle("Replicates - Bray-Curtis")+
  scale_fill_manual(values = c("skyblue", "red"))

## weighted unifrac

rep_wunifrac <- ordinate(
  physeq = replicate_rare,
  method = "NMDS",
  distance = "wunifrac"
)
```

```
## Run 0 stress 0.1292629
## Run 1 stress 0.1344655
## Run 2 stress 0.1344658
## Run 3 stress 0.159361
## Run 4 stress 0.1292629
## ... New best solution
## ... Procrustes: rmse 2.776196e-05  max resid 7.056753e-05
## ... Similar to previous best
## Run 5 stress 0.1991938
## Run 6 stress 0.1344662
## Run 7 stress 0.1292629
## ... New best solution
## ... Procrustes: rmse 2.23359e-05  max resid 5.557226e-05
## ... Similar to previous best
## Run 8 stress 0.1480542
## Run 9 stress 0.1991938
## Run 10 stress 0.2278001
## Run 11 stress 0.1797407
## Run 12 stress 0.1292629
## ... New best solution
## ... Procrustes: rmse 2.115137e-05  max resid 5.794563e-05
## ... Similar to previous best
## Run 13 stress 0.1767195
## Run 14 stress 0.1767443
## Run 15 stress 0.1344662
## Run 16 stress 0.1987553
## Run 17 stress 0.1292629
## ... Procrustes: rmse 3.016346e-05  max resid 8.497623e-05
## ... Similar to previous best
## Run 18 stress 0.1344661
## Run 19 stress 0.1344656
## Run 20 stress 0.1758584
## *** Solution reached
```

```
ord_df<-data.frame(rep_wunifrac$points)
rep_metadata<-data.frame(sample_data(replicate_rare))
head(rep_metadata)
```

```
##       feature.id frog_id   Run Treatment       date Mass Replicate Sex date_cat
## 115          115     115 BYLN3        UU 2018-04-10 2.95         N   F    Date1
## 115-2      115-2     115 BYLLT        UU 2018-04-10 2.95         Y   F    Date1
## 117          117     117 BYLN3        UU 2018-04-10 3.25         N   F    Date1
## 117-2      117-2     117 BYLLT        UU 2018-04-10 3.25         Y   F    Date1
## 119          119     119 BYLN3        CC 2018-04-10 3.25         N   F    Date1
## 119-2      119-2     119 BYLLT        CC 2018-04-10 3.25         Y   F    Date1
##       Seq_depth Seq_run
## 115       41790   Run 2
## 115-2     65468   Run 1
## 117       51839   Run 2
## 117-2     79929   Run 1
## 119       39556   Run 2
## 119-2     59527   Run 1
```

```
rep_wunifrac_df<-merge(rep_metadata, ord_df, by = 0)

rep2<-ggplot(rep_wunifrac_df, aes(x = MDS1, y = MDS2, fill = Seq_run, group = frog_id))+
  geom_point(pch = 21, size = 5)+
  geom_line()+
  theme_bw(base_size = 12)+
  ggtitle("Replicates - weighted Unifrac")+
  scale_fill_manual(values = c("skyblue", "red"))

ggarrange(rep2, rep1, ncol = 2, common.legend = T)
```



```
## exclude Run BYLN3 as these are replicates of run BYLLT and have fewer reads per sample


frog_filtered <- frog_filtered %>%
  subset_samples(
    Run!= "BYLN3")
```

Change ASV names

- change names to something informative

Data exploration

Create transformed datasets

```
frog_rare<-rarefy_even_depth(frog_filtered, sample.size = 6700)
```

Top taxa summary

```r
frog_comp<-microbiome::transform(frog_filtered, "compositional")

# phylum level
top_taxa_phylum<-aggregate_top_taxa(frog_comp, top = 10, level = "Phylum")
prev_df_phylum<-prevalence(top_taxa_phylum)%>% head()
prev_df_phylum %>% arrange(-MeanAbundance)%>% head()
```

```
##   Prevalence TotalAbundance MeanAbundance MedianAbundance          Phylum
## 1        104     22.5637596   0.216959227    0.2195824556   Bacteroidetes
## 2        102      4.8476797   0.046612305    0.0353077029  Actinobacteria
## 3        104      3.8452962   0.036974001    0.0285538591      Firmicutes
## 4        101      0.6030541   0.005798597    0.0048333153 Armatimonadetes
## 5         95      0.1912738   0.001839172    0.0008511468   Cyanobacteria
## 6         93      0.1193785   0.001147870    0.0007883345   Acidobacteria
##             unique
## 1   Bacteroidetes
## 2  Actinobacteria
## 3      Firmicutes
## 4 Armatimonadetes
## 5   Cyanobacteria
## 6   Acidobacteria
```

```r
# order level
top_taxa_order<-aggregate_top_taxa(frog_comp, top = 10, level = "Order")
prev_df_order<-prevalence(top_taxa_order)%>% head()
prev_df_order %>% arrange(-MeanAbundance)%>% head()
```

```
##   Prevalence TotalAbundance MeanAbundance MedianAbundance                 Order
## 1        104      14.710090    0.14144317     0.089660245     Enterobacteriales
## 2        104      12.376601    0.11900578     0.096795580      Flavobacteriales
## 3        104      10.131145    0.09741486     0.098694784 Betaproteobacteriales
## 4        104       8.726014    0.08390398     0.017711699         Aeromonadales
## 5        102       1.737204    0.01670389     0.011089966         Micrococcales
## 6        102       1.691951    0.01626876     0.006206775         Bacteroidales
##                   unique
## 1     Enterobacteriales
## 2      Flavobacteriales
## 3 Betaproteobacteriales
## 4         Aeromonadales
## 5         Micrococcales
## 6         Bacteroidales
```

```r
# family level
top_taxa_family<-aggregate_top_taxa(frog_comp, top = 10, level = "Family")
prev_df_family<-prevalence(top_taxa_family)%>% head()
prev_df_family %>% arrange(-MeanAbundance)%>% head()
```

```
##   Prevalence TotalAbundance MeanAbundance MedianAbundance             Family
## 1        104      14.710090    0.14144317      0.08966025 Enterobacteriaceae
## 2        104      10.566411    0.10160010      0.07672830  Flavobacteriaceae
## 3        104       9.977440    0.09593692      0.09661509   Burkholderiaceae
## 4        104       8.726014    0.08390398      0.01771170     Aeromonadaceae
## 5        104       8.304507    0.07985103      0.05738294      Moraxellaceae
## 6        103       1.738502    0.01671637      0.01303112   Beijerinckiaceae
##               unique
## 1 Enterobacteriaceae
## 2  Flavobacteriaceae
## 3   Burkholderiaceae
## 4     Aeromonadaceae
## 5      Moraxellaceae
## 6   Beijerinckiaceae
```

```
# genus level
top_taxa_genus<-aggregate_top_taxa(frog_filtered, top = 20, level = "Genus")
prev_df_genus<-prevalence(top_taxa_genus)
prev_df_genus$Prev<- prev_df_genus$Prevalence/104
prev_df_genus$RelAbund<- (prev_df_genus$TotalAbundance/sum(taxa_sums(frog_filtered)))
prev_df_genus %>% arrange(-RelAbund)%>% head()
```

```
##   Prevalence TotalAbundance MeanAbundance MedianAbundance          Genus
## 1        104        3347045     32183.125         24556.0          Other
## 2        104        1402856     13489.000          8268.0    Pseudomonas
## 3        104        1026905      9874.087          6688.5 Flavobacterium
## 4        104         819056      7875.538          4131.0  Acinetobacter
## 5        104         807076      7760.346          1382.0      Aeromonas
## 6        104         684720      6583.846          4857.0        Unknown
##           unique Prev   RelAbund
## 1          Other    1 0.30119258
## 2    Pseudomonas    1 0.12623966
## 3 Flavobacterium    1 0.09240873
## 4  Acinetobacter    1 0.07370489
## 5      Aeromonas    1 0.07262684
## 6        Unknown    1 0.06161632
```

```
prev_df_genus %>% arrange(-Prevalence) %>% head()
```

```
##   Prevalence TotalAbundance MeanAbundance MedianAbundance          Genus
## 1        104         819056      7875.538          4131.0  Acinetobacter
## 2        104         807076      7760.346          1382.0      Aeromonas
## 3        104        1026905      9874.087          6688.5 Flavobacterium
## 4        104         116227      1117.567           543.5     Morganella
## 5        104        3347045     32183.125         24556.0          Other
## 6        104        1402856     13489.000          8268.0    Pseudomonas
##           unique Prev   RelAbund
## 1  Acinetobacter    1 0.07370489
## 2      Aeromonas    1 0.07262684
## 3 Flavobacterium    1 0.09240873
## 4     Morganella    1 0.01045899
## 5          Other    1 0.30119258
## 6    Pseudomonas    1 0.12623966
```

Bar plots

```
frog_phylum<- tax_glom(frog_filtered, taxrank = "Phylum")
frog_phylum<- aggregate_top_taxa(frog_filtered, top = 10, level = "Phylum")
frog_phylum<- microbiome::transform(frog_phylum, "compositional")


### order

frog_order<- tax_glom(frog_filtered, taxrank = "Order")

frog_order<- microbiome::transform(frog_order, "compositional")
frog_order<-microbiome::core(frog_order, detection = 0, prevalence = 0.9)


##############################


ps1.com.fam <- microbiome::aggregate_top_taxa(frog_filtered, "Order", top = 22)

ps1.com.fam<- microbiome::transform(ps1.com.fam, "compositional")

plot.composition.relAbun <- microbiome::plot_composition(ps1.com.fam,
                                             sample.sort = "Treatment",
                                             x.label = "frog_id",
                                             group_by = "Treatment")


data.com <- plot.composition.relAbun$data
colnames(data.com)
```

```
## [1] "Tax"       "Sample"    "Abundance" "Group"     "xlabel"
```

```
data.com$Tax<-ifelse(data.com$Tax == "Fimbriimonadales" | data.com$Tax == "Propionibacteriales" |data.c


unique(data.com$Tax)
```

```
##  [1] "Aeromonadales"         "Azospirillales"        "Bacillales"
##  [4] "Bacteroidales"         "Betaproteobacteriales" "Chitinophagales"
##  [7] "Clostridiales"         "Corynebacteriales"     "Cytophagales"
## [10] "Enterobacteriales"     "Other"                 "Flavobacteriales"
## [13] "Fusobacteriales"       "Lactobacillales"       "Micrococcales"
## [16] "Pseudomonadales"       "Rhizobiales"           "Sphingobacteriales"
## [19] "Sphingomonadales"      "Xanthomonadales"
```

```
data.com$Tax<-factor(data.com$Tax, level = c("Flavobacteriales",
                                                     "Bacteroidales",
                                                     "Sphingobacteriales",
                                                     "Chitinophagales",
                                                     "Cytophagales",
                                                     "Enterobacteriales",
                                                     "Pseudomonadales",
                                                     "Aeromonadales",
                                                     "Xanthomonadales",
                                                     "Betaproteobacteriales",
                                                     "Rhizobiales",
                                                    # "Cardiobacteriales",
                                                     "Azospirillales",
                                                     "Sphingomonadales",
                                                     #"Desulfovibrionales",

                                                     "Corynebacteriales",
                                                    # "Pseudonocardiales",
                                                     "Micrococcales",

                                                     "Clostridiales",
                                                     "Bacillales",
                                                     "Lactobacillales",

                                                     "Fusobacteriales",
                                                     "Other"))


colors <- c("lightskyblue","skyblue4", "royalblue","darkslategray4", "cyan",

            "#E7E1EF", "#D4B9DA", "#C994C7", "#DF65B0" ,"#E7298A" ,"#CE1256", "#980043", "#67001F",

            "forestgreen","olivedrab2",

            "yellow","chocolate","bisque",

            "navy",

            "white")

barplot_order<-ggplot(data.com, aes(x = Sample, y = Abundance, fill = Tax))+
  geom_bar(position = "stack", stat = "identity", width = 1, col = "black")+
  scale_x_discrete(labels = data.com$xlabel, breaks = data.com$Sample)+
  facet_grid(~Group, scales = "free") + theme_bw()+
  scale_fill_manual(values = colors)+
   theme_bw(base_size = 14)+
  theme(axis.text.x = element_blank())+
  labs(fill = "Bacterial Order")+
  ylab("Relative abundance")+
    theme(legend.key.size = unit(0.4, 'cm'))+
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1))

#########################
#########################
#########################
#########################
#########################
```

```
plot.composition.relAbun <- microbiome::plot_composition(ps1.com.fam,
                                              sample.sort = "Sex",
                                              x.label = "frog_id",
                                              group_by = "Sex")


data.com <- plot.composition.relAbun$data
colnames(data.com)
```

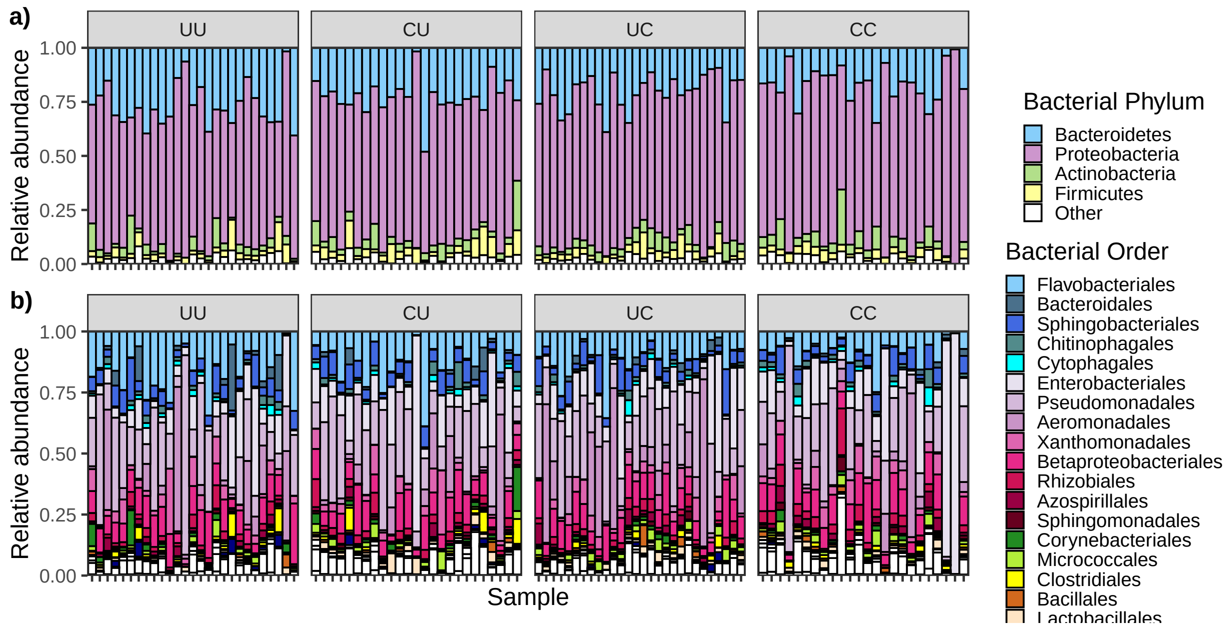
```
## [1] "Tax"       "Sample"    "Abundance" "Group"     "xlabel"
```

```
data.com$Tax<-ifelse(data.com$Tax == "Fimbriimonadales" | data.com$Tax == "Propionibacteriales" |data.c


unique(data.com$Tax)
```

```
##  [1] "Aeromonadales"         "Azospirillales"        "Bacillales"
##  [4] "Bacteroidales"         "Betaproteobacteriales" "Chitinophagales"
##  [7] "Clostridiales"         "Corynebacteriales"     "Cytophagales"
## [10] "Enterobacteriales"     "Other"                 "Flavobacteriales"
## [13] "Fusobacteriales"       "Lactobacillales"       "Micrococcales"
## [16] "Pseudomonadales"       "Rhizobiales"           "Sphingobacteriales"
## [19] "Sphingomonadales"      "Xanthomonadales"
```

```
data.com$Tax<-factor(data.com$Tax, level = c("Flavobacteriales",
                                                      "Bacteroidales",
                                                      "Sphingobacteriales",
                                                      "Chitinophagales",
                                                      "Cytophagales",

                                                      "Enterobacteriales",
                                                      "Pseudomonadales",
                                                      "Aeromonadales",
                                                      "Xanthomonadales",
                                                      "Betaproteobacteriales",
                                                      "Rhizobiales",
                                                     # "Cardiobacteriales",
                                                      "Azospirillales",
                                                      "Sphingomonadales",
                                                      #"Desulfovibrionales",

                                                      "Corynebacteriales",
                                                     # "Pseudonocardiales",
                                                      "Micrococcales",

                                                      "Clostridiales",
                                                      "Bacillales",
                                                      "Lactobacillales",

                                                      "Fusobacteriales",
                                                      "Other"))
```

```r
# phylum level


ps1.com.phy <- microbiome::aggregate_top_taxa(frog_filtered, "Phylum", top = 4)

ps1.com.phy<- microbiome::transform(ps1.com.phy, "compositional")

plot.composition.relAbun <- microbiome::plot_composition(ps1.com.phy,
                                            sample.sort = "Treatment",
                                            x.label = "frog_id",
                                            group_by = "Treatment")


data.com.phy <- plot.composition.relAbun$data


data.com.phy$Tax<-factor(data.com.phy$Tax, levels = c( "Bacteroidetes","Proteobacteria", "Actinobacteri


pal1<-brewer.pal(12,"Paired")
#show_col(pal1)
pal1<-pal1[c(1,2,3,4,5)]
pal1[5]<-"white"

pal1<-c("lightskyblue", "palevioletred" , "#33A02C" , "bisque", "white")
pal1<-c("lightskyblue", "plum3" , "#B2DF8A" , "#FFFF99", "white")


barplot_phylum<-ggplot(data.com.phy, aes(x = Sample, y = Abundance, fill = Tax))+
  geom_bar(position = "stack", stat = "identity", width = 1, col = "black")+
  scale_x_discrete(labels = data.com$xlabel, breaks = data.com$Sample)+
  facet_grid(~Group, scales = "free") + theme_bw()+
  scale_fill_manual(values = pal1)+
   theme_bw(base_size = 14)+
  theme(axis.text.x = element_blank())+
  theme(axis.title.x = element_blank())+
  #theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  labs(fill = "Bacterial Phylum")+
  ylab("Relative abundance")+
    theme(legend.key.size = unit(0.4, 'cm'))+
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1))


ggarrange(barplot_phylum, barplot_order, ncol = 1, align = "v", heights = c(1,1.2) , labels = c("a)", "
```

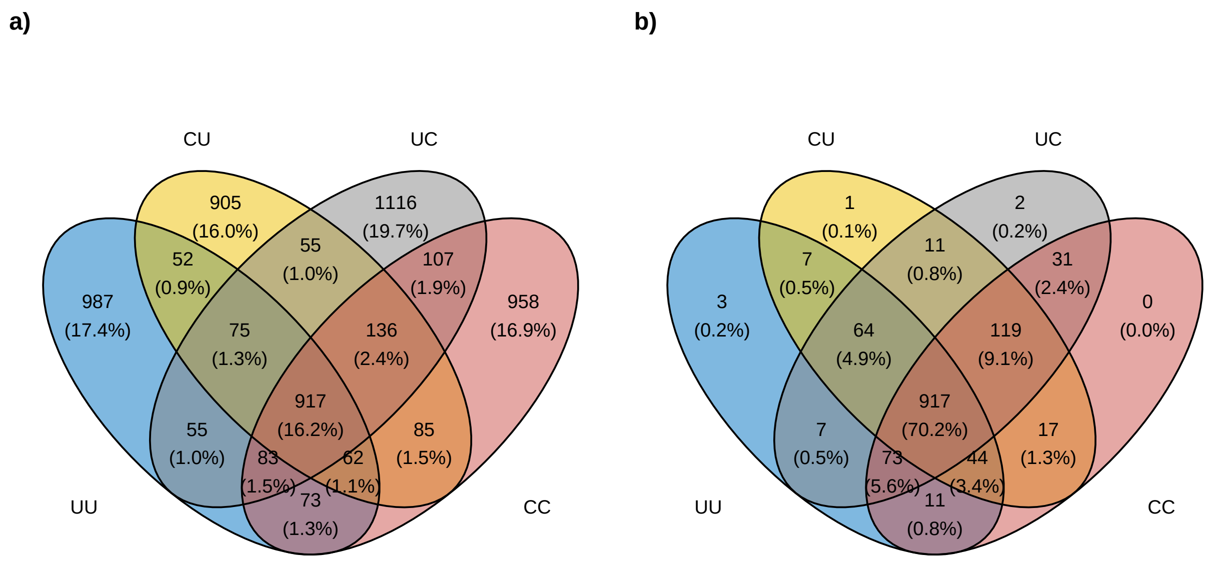


```
#ggsave("Figures/fig2.pdf")

#Saving 13.6 x 7.68 in image
```

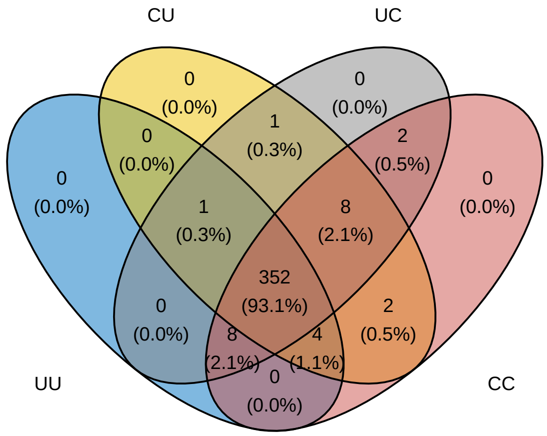
Venn diagram

```
frog_uu<-subset_samples(frog_filtered, Treatment == "UU")
frog_uu<-prune_taxa(taxa_sums(frog_uu)>0, frog_uu)

frog_cc<-subset_samples(frog_filtered, Treatment == "CC")
frog_cc<-prune_taxa(taxa_sums(frog_cc)>0, frog_cc)

frog_cu<-subset_samples(frog_filtered, Treatment == "CU")
frog_cu<-prune_taxa(taxa_sums(frog_cu)>0, frog_cu)

frog_uc<-subset_samples(frog_filtered, Treatment == "UC")
frog_uc<-prune_taxa(taxa_sums(frog_uc)>0, frog_uc)

uu_asvs<-taxa_names(frog_uu)
cc_asvs<-taxa_names(frog_cc)
uc_asvs<-taxa_names(frog_uc)
cu_asvs<-taxa_names(frog_cu)

x <- list(
  UU = uu_asvs,
  CU = cu_asvs,
  UC = uc_asvs,
  CC = cc_asvs
  )
```

```r
p1<- ggvenn(
  x,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 4)


###

frog_filtered5pc<-core(frog_filtered, detection = 0, prevalence = 0.03)

frog_uu<-subset_samples(frog_filtered5pc, Treatment == "UU")
frog_uu<-prune_taxa(taxa_sums(frog_uu)>0, frog_uu)

frog_cc<-subset_samples(frog_filtered5pc, Treatment == "CC")
frog_cc<-prune_taxa(taxa_sums(frog_cc)>0, frog_cc)

frog_cu<-subset_samples(frog_filtered5pc, Treatment == "CU")
frog_cu<-prune_taxa(taxa_sums(frog_cu)>0, frog_cu)

frog_uc<-subset_samples(frog_filtered5pc, Treatment == "UC")
frog_uc<-prune_taxa(taxa_sums(frog_uc)>0, frog_uc)

uu_asvs<-taxa_names(frog_uu)
cc_asvs<-taxa_names(frog_cc)
uc_asvs<-taxa_names(frog_uc)
cu_asvs<-taxa_names(frog_cu)

y <- list(
  UU = uu_asvs,
  CU = cu_asvs,
  UC = uc_asvs,
  CC = cc_asvs
  )


p2<-ggvenn(
  y,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 4)

ggarrange(p1, p2, labels = c("a)", "b)"))
```



```
##### genus level

frog_genus<-tax_glom(frog_filtered, taxrank = "Genus")
frog_genus<-core(frog_genus, detection = 0, prevalence = 0.05)

frog_uu<-subset_samples(frog_genus, Treatment == "UU")
frog_uu<-prune_taxa(taxa_sums(frog_uu)>0, frog_uu)

frog_cc<-subset_samples(frog_genus, Treatment == "CC")
frog_cc<-prune_taxa(taxa_sums(frog_cc)>0, frog_cc)

frog_cu<-subset_samples(frog_genus, Treatment == "CU")
frog_cu<-prune_taxa(taxa_sums(frog_cu)>0, frog_cu)

frog_uc<-subset_samples(frog_genus, Treatment == "UC")
frog_uc<-prune_taxa(taxa_sums(frog_uc)>0, frog_uc)

uu_asvs<-taxa_names(frog_uu)
cc_asvs<-taxa_names(frog_cc)
uc_asvs<-taxa_names(frog_uc)
cu_asvs<-taxa_names(frog_cu)

x <- list(
  UU = uu_asvs,
  CU = cu_asvs,
  UC = uc_asvs,
  CC = cc_asvs
  )


ggvenn(
  x,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 4)
```



Heat map

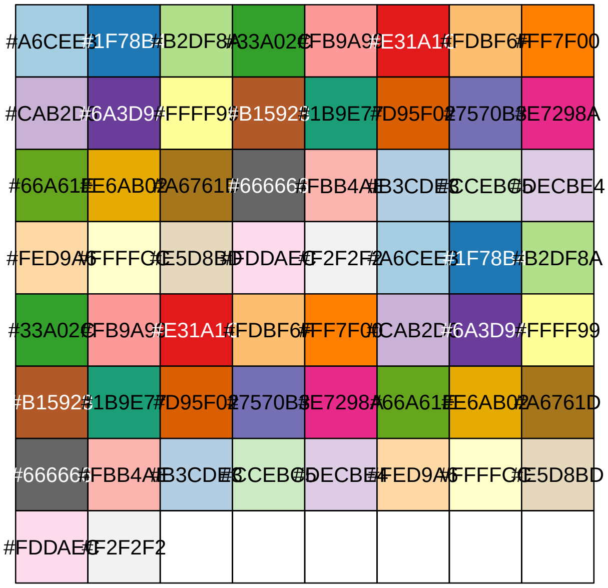
Taxa levels changes: Relative abundance

```
frog_genus<- microbiome::aggregate_top_taxa(frog_rare, "Genus", top = 20)

taxtable<-data.frame(tax_table(frog_genus))
taxa_names(frog_genus)<-taxtable$Genus
taxa_names(frog_genus)
```

```
##  [1] "Acinetobacter"        "Aeromonas"            "Citrobacter"
##  [4] "Comamonas"            "Delftia"              "Enterobacter"
##  [7] "Flavobacterium"       "Janthinobacterium"    "Methylobacterium"
## [10] "Myroides"             "Other"                "Pedobacter"
## [13] "Proteus"              "Providencia"          "Pseudomonas"
## [16] "Rhodococcus"          "Serratia"             "Sphingobacterium"
## [19] "Stenotrophomonas"     "uncultured bacterium" "Unknown"
```

```
#### table

UU_top<-subset_samples(frog_genus, Treatment == "UU") %>% transform("compositional")
UU_prev<-prevalence(UU_top)
UU_prev$Treatment<-"UU"


CU_top<-subset_samples(frog_genus, Treatment == "CU")%>% transform("compositional")
```

```
CU_prev<-prevalence(CU_top)
CU_prev$Treatment<-"CU"


UC_top<-subset_samples(frog_genus, Treatment == "UC") %>% transform("compositional")
UC_prev<-prevalence(UC_top)
UC_prev$Treatment<-"UC"


CC_top<-subset_samples(frog_genus, Treatment == "CC") %>% transform("compositional")
CC_prev<-prevalence(CC_top)
CC_prev$Treatment<-"CC"

prev_df<-rbind(UU_prev, UC_prev, CU_prev, CC_prev)
prev_df$Treatment <- factor(prev_df$Treatment, levels = c("UU", "CU", "UC", "CC"))

palx<-brewer.pal(12,"Paired")
paly<-brewer.pal(12,"Dark2")
pali<-brewer.pal(12,"Pastel1")

palz<-c(palx,paly, pali,palx,paly, pali)
scales::show_col(palz)
```

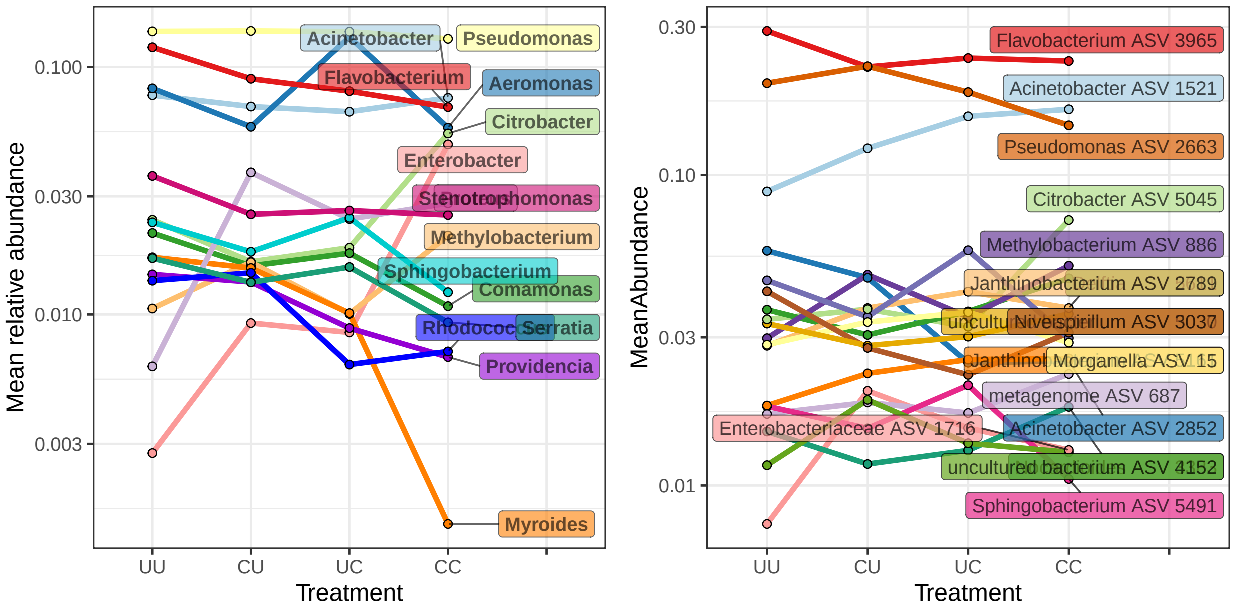


```
palz[12]<-"blue"
palz[14]<-"cyan3"
palz[15]<-"deeppink3"
palz[10]<-"darkviolet"

prev_df<-prev_df %>%  mutate(label = ifelse(Treatment == "CC", as.character(Genus), NA))
levels(prev_df$Treatment) <- c(levels(prev_df$Treatment),'') # add blank level

df<-subset(prev_df, Genus != "Other" & Genus !="Unknown" & Genus !="uncultured bacterium")
length(unique(df$Genus))
```

```
## [1] 18
```

```
p1<-ggplot(subset(prev_df, Genus != "Other" & Genus !="Unknown" & Genus !="uncultured bacterium" & Genu

  geom_line(aes(col = Genus), size = 1.5)+
  geom_point(aes(fill = Genus), size = 2, col = "black", pch = 21)+
  scale_color_manual(values = palz)+
  scale_fill_manual(values = palz)+
  scale_y_log10()+
  theme_bw(base_size = 14)+
  theme(legend.position = "none")+
  scale_x_discrete(drop=FALSE)+

  geom_label_repel(aes(label = label, fill = Genus),
                   alpha = 0.6,nudge_x = 1, fontface="bold",
                   na.rm = TRUE)+
  ylab("Mean relative abundance")




## ASV level ###
## ASV level ###
## ASV level ###
## ASV level ###


frog_asv<- core(frog_rare, detection = 0, prevalence = 0.94)


#### table

UU_top<-subset_samples(frog_asv, Treatment == "UU") %>% transform("compositional")
UU_prev<-prevalence(UU_top)
UU_prev$Treatment<-"UU"
UU_prev$ASV<-row.names(UU_prev)


CU_top<-subset_samples(frog_asv, Treatment == "CU")%>% transform("compositional")
CU_prev<-prevalence(CU_top)
CU_prev$Treatment<-"CU"
CU_prev$ASV<-row.names(CU_prev)



UC_top<-subset_samples(frog_asv, Treatment == "UC") %>% transform("compositional")
UC_prev<-prevalence(UC_top)
UC_prev$Treatment<-"UC"
UC_prev$ASV<-row.names(UC_prev)
```

```r
CC_top<-subset_samples(frog_asv, Treatment == "CC") %>% transform("compositional")
CC_prev<-prevalence(CC_top)
CC_prev$Treatment<-"CC"
CC_prev$ASV<-row.names(CC_prev)



prev_df_asv<-rbind(UU_prev, UC_prev, CU_prev, CC_prev)
prev_df_asv$Treatment <- factor(prev_df_asv$Treatment, levels = c("UU", "CU", "UC", "CC"))

palx<-brewer.pal(12,"Paired")
paly<-brewer.pal(12,"Dark2")

palz<-c(palx,paly)

prev_df_asv<-prev_df_asv %>%  mutate(label = ifelse(Treatment == "CC", as.character(ASV), NA))
levels(prev_df_asv$Treatment) <- c(levels(prev_df_asv$Treatment),'') # add blank level

p2<-ggplot(prev_df_asv, aes(y = MeanAbundance, x = Treatment, group = ASV))+

  geom_line(aes(col = ASV), size = 1.5)+
  geom_point(aes(fill = ASV), size = 2, col = "black", pch = 21)+
  scale_color_manual(values = palz)+
  scale_fill_manual(values = palz)+
  scale_y_log10()+
  theme_bw(base_size = 14)+
  theme(legend.position = "none")+
  scale_x_discrete(drop=FALSE)+

  geom_label_repel(aes(label = label, fill = ASV),
                  alpha = 0.7,nudge_x = 1,
                  na.rm = TRUE)


ggarrange(p1, p2, ncol = 2)
```



Alpha diversity

```
alpha_df<-data.frame(estimate_richness(frog_rare))

alpha_df$Faiths_PD<-as.numeric(metagMisc::phyloseq_phylo_div(frog_rare, "PD")$PD)
row.names(alpha_df)<-sample_names(frog_filtered)

metadata<-data.frame(sample_data(frog_filtered))
metadata<-merge(metadata, alpha_df, by = 0)
summary(metadata$Observed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    29.0   221.2   286.0   279.2   349.2   468.0
```

```
# ACE = Fisher = Observed = Chao1 (=) ACE.se

metadata$Treatment <- factor(metadata$Treatment, levels = c("UU", "CU", "UC", "CC"))

# scale variables

metadata$Mass_scaled<-as.numeric(scale(metadata$Mass))
metadata$Seq_depth_transformed<-as.numeric(scale(sqrt(metadata$Seq_depth)))
```

Model/distribution selection

```
## choose best distributions for 3 measures of alpha diversity


EnvStats::distChoose(metadata$Observed)$decision
```

```
## [1] "Normal"
```

```
EnvStats::distChoose(metadata$InvSimpson)$decision
```

```
## [1] "Gamma"
```

```
EnvStats::distChoose(metadata$Shannon)$decision
```

```
## [1] "Nonparametric"
```

```
#### observed

observed_model<-glm(Observed~Treatment + Seq_depth_transformed+Sex+Mass_scaled, family= Gamma(link = "l

observed_model2<-glm(Observed~Treatment + Seq_depth_transformed+Sex+Mass_scaled, family= Gamma(link = "

observed_model3<-lm(Observed~Treatment + Seq_depth_transformed+Sex+Mass_scaled,  data = metadata) # bes


AIC(observed_model, observed_model2, observed_model3)
```

```
##                 df      AIC
## observed_model   8 1251.853
## observed_model2  8 1249.690
## observed_model3  8 1210.330
```

```
######## invnormal ###
######## invnormal ###
######## invnormal ###

invsimp_model<-glm(InvSimpson~Treatment + Seq_depth_transformed+Sex, family= Gamma(link = "log"), data


invsimp_model2<-glm(InvSimpson~Treatment + Seq_depth_transformed+Sex, family= Gamma(link = "identity"),


invsimp_model3<-lm(InvSimpson~Treatment + Seq_depth_transformed+Sex,  data = metadata)


AIC(invsimp_model, invsimp_model2, invsimp_model3)
```

```
##                df      AIC
## invsimp_model   7 829.1156
## invsimp_model2  7 828.9828
## invsimp_model3  7 850.3511
```

```
##### shannon ########
##### shannon ########
##### shannon ########
##### shannon ########
```

```
shannon_model<-glm(Shannon~Treatment + Seq_depth_transformed+Sex+Mass_scaled, family= Gamma(link = "log


shannon_model2<-glm(Shannon~Treatment + Seq_depth_transformed+Sex+Mass_scaled, family= Gamma(link = "ide


shannon_model3<-lm(Shannon~Treatment + Seq_depth_transformed+Sex+Mass_scaled,  data = metadata) # best

AIC(shannon_model, shannon_model2, shannon_model3)
```

```
##                df      AIC
## shannon_model   8 273.3078
## shannon_model2  8 273.1418
## shannon_model3  8 241.9552
```

Fitting final models

```
# final model

observed_model<-lm(Observed~Treatment +Sex +  Seq_depth_transformed, data = metadata)

summary(observed_model)
```

```
##
## Call:
## lm(formula = Observed ~ Treatment + Sex + Seq_depth_transformed,
##     data = metadata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -221.04  -48.87   19.11   52.47  196.20
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            259.222     16.602  15.614  < 2e-16 ***
## TreatmentCU             34.300     21.917   1.565   0.1208
## TreatmentUC             52.927     21.419   2.471   0.0152 *
## TreatmentCC             52.499     22.042   2.382   0.0192 *
## SexM                   -26.234     15.622  -1.679   0.0963 .
## Seq_depth_transformed   40.844      7.762   5.262  8.4e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 77.92 on 98 degrees of freedom
## Multiple R-squared:  0.2971, Adjusted R-squared:  0.2613
## F-statistic: 8.285 on 5 and 98 DF,  p-value: 1.463e-06
```

```
r2beta(observed_model, partial = TRUE, method = "sgv", data = metadata)
```

```
##                   Effect   Rsq upper.CL lower.CL
## 1                 Model 0.297    0.456    0.186
## 6 Seq_depth_transformed 0.220    0.365    0.098
## 3           TreatmentUC 0.059    0.174    0.003
## 4           TreatmentCC 0.055    0.168    0.002
## 5                  SexM 0.028    0.123    0.000
## 2           TreatmentCU 0.024    0.117    0.000
```

```
#### GET CONFIDENCE INTERVALS
summary<-data.frame(confint(observed_model))
summaryest<-summary(observed_model) # add estimate as column
summaryest<-data.frame(summaryest$coefficients)

summary$Estimate<-summaryest$Estimate
summary<-summary[c(1:4),] #keep just first 4 rows

summary[2:4,]<-summary[2:4,]+summary[1,3] # add intercept to estimates for forest and organic
row.names(summary)<-c("UU", "CU", "UC", "CC")
names(summary)<-c("CI_lower", "CI_upper", "Est")
summary$Treatment<-row.names(summary)
summary$Treatment<-factor(summary$Treatment, levels = c(c("UU", "CU", "UC", "CC")))


# standard errors

summary_coef1<-summary(observed_model)
summary_coef1$coefficients
```

```
##                         Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)           259.22246  16.602024 15.613907 2.468740e-28
## TreatmentCU            34.30007  21.916668  1.565022 1.208013e-01
## TreatmentUC            52.92729  21.418638  2.471086 1.519612e-02
## TreatmentCC            52.49873  22.041531  2.381810 1.915884e-02
## SexM                  -26.23399  15.622443 -1.679250 9.628900e-02
## Seq_depth_transformed  40.84367   7.762255  5.261831 8.401860e-07
```

```
summary_coef1<- data.frame(summary_coef1$coefficients)
names(summary_coef1)
```

```
## [1] "Estimate"   "Std..Error" "t.value"    "Pr...t.."
```

```
summary$SE_lower<-summary$Est-summary_coef1$Std..Error[1:4]
summary$SE_upper<-summary$Est+summary_coef1$Std..Error[1:4]

summary1<-summary

### ESIMATES AND CONFIDENCE INTERBALS ARE BACKTRANSFORMED IN PLOT USING EXP()

p1<-ggplot(summary1, aes( x =Treatment, y = Est))+
  geom_errorbar(aes(ymin = CI_lower, ymax = CI_upper), width = 0, col = "grey", size = 1.5)+
  geom_errorbar(aes(ymin = SE_lower, ymax = SE_upper), width = 0,  size = 1.5)+
   geom_point( size = 4, pch = 21, fill = "black", col = "grey")+
  theme_light(base_size = 14)+
  ylab("Observed ASV richness")+
  theme(legend.position = "none")+
  xlab("")

############ inverse simpson  ##################
############ inverse simpson  ##################
############ inverse simpson  ##################
############ inverse simpson  ##################


# final model
invsimp_model<-glm(InvSimpson~Treatment +Sex+ Seq_depth_transformed, family= Gamma(link = "log"), data =


summary(invsimp_model)
```

```
##
## Call:
## glm(formula = InvSimpson ~ Treatment + Sex + Seq_depth_transformed,
##     family = Gamma(link = "log"), data = metadata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9428  -0.5499  -0.1357   0.2500   1.6519
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.04602    0.14242  21.388   <2e-16 ***
## TreatmentCU             0.17573    0.18801   0.935   0.3523
## TreatmentUC             0.21582    0.18374   1.175   0.2430
## TreatmentCC             0.40665    0.18908   2.151   0.0340 *
## SexM                   -0.29936    0.13402  -2.234   0.0278 *
## Seq_depth_transformed  0.09662    0.06659   1.451   0.1499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.4468416)
##
##     Null deviance: 54.438  on 103  degrees of freedom
## Residual deviance: 48.919  on  98  degrees of freedom
## AIC: 829.12
##
## Number of Fisher Scoring iterations: 6
```

```
r2beta(invsimp_model, partial = TRUE, method = "sgv", data = metadata)
```

```
##                  Effect   Rsq upper.CL lower.CL
## 1                 Model 0.109    0.272    0.047
## 5                  SexM 0.048    0.158    0.001
## 4           TreatmentCC 0.045    0.153    0.001
## 6 Seq_depth_transformed 0.021    0.110    0.000
## 3           TreatmentUC 0.014    0.094    0.000
## 2           TreatmentCU 0.009    0.081    0.000
```

```
#### GET CONFIDENCE INTERVALS FROM GAMMA

summary2<-data.frame(confint(invsimp_model))
summary2est<-summary(invsimp_model) # add estimate as column
summary2est<-data.frame(summary2est$coefficients)

summary2$Estimate<-summary2est$Estimate
summary2<-summary2[c(1:4),] #keep just first 4 rows

summary2[2:4,]<-summary2[2:4,]+summary2[1,3] # add intercept to estimates for forest and organic
row.names(summary2)<-c("UU", "CU", "UC", "CC")
names(summary2)<-c("CI_lower", "CI_upper", "Est")
summary2$Treatment<-row.names(summary2)
summary2$Treatment<-factor(summary2$Treatment, levels = c(c("UU", "CU", "UC", "CC")))


# standard errors

summary2_coef1<-summary(invsimp_model)
summary2_coef1$coefficients
```

```
##                          Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)            3.04602450 0.14241933 21.3877179 1.070322e-38
## TreatmentCU            0.17572669 0.18801064  0.9346635 3.522591e-01
## TreatmentUC            0.21581778 0.18373832  1.1745932 2.430038e-01
## TreatmentCC            0.40665037 0.18908177  2.1506588 3.396255e-02
## SexM                  -0.29935957 0.13401606 -2.2337588 2.777237e-02
## Seq_depth_transformed  0.09662485 0.06658797  1.4510856 1.499496e-01
```

```
summary2_coef1<- data.frame(summary2_coef1$coefficients)
names(summary2_coef1)
```

```
## [1] "Estimate"   "Std..Error" "t.value"    "Pr...t.."
```

```
summary2$SE_lower<-summary2$Est-summary2_coef1$Std..Error[1:4]
summary2$SE_upper<-summary2$Est+summary2_coef1$Std..Error[1:4]

### ESIMATES AND CONFIDENCE INTERBALS ARE BACKTRANSFORMED IN PLOT USING EXP()

p2<-ggplot(summary2, aes( x =Treatment, y = exp(Est)))+
  geom_errorbar(aes(ymin = exp(CI_lower), ymax = exp(CI_upper)), width = 0, col = "grey", size = 1.5)+
  geom_errorbar(aes(ymin = exp(SE_lower), ymax = exp(SE_upper)), width = 0,  size = 1.5)+
   geom_point( size = 4, pch = 21, fill = "black", col = "grey")+
  theme_light(base_size = 14)+
  ylab("Inverse Simpson Index")+
  theme(legend.position = "none")


################ shannon #################
################ shannon #################
################ shannon #################
################ shannon #################
```

```
# final model

shannon_model<-lm(Shannon~Treatment +Sex+ Seq_depth_transformed,  data = metadata)
summary(shannon_model)
```

```
##
## Call:
## lm(formula = Shannon ~ Treatment + Sex + Seq_depth_transformed,
##     data = metadata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6941 -0.3258  0.1338  0.4674  1.1924
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.76768    0.15738  23.940   <2e-16 ***
## TreatmentCU            0.29870    0.20776   1.438   0.1537
## TreatmentUC            0.31370    0.20304   1.545   0.1256
## TreatmentCC            0.29754    0.20894   1.424   0.1576
## SexM                  -0.31149    0.14809  -2.103   0.0380 *
## Seq_depth_transformed  0.20196    0.07358   2.745   0.0072 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7387 on 98 degrees of freedom
## Multiple R-squared:  0.1403, Adjusted R-squared:  0.09649
## F-statistic:   3.2 on 5 and 98 DF,  p-value: 0.01018
```

```
r2beta(shannon_model, partial = TRUE, method = "sgv", data = metadata)
```

```
##                  Effect   Rsq upper.CL lower.CL
## 1                 Model 0.140    0.306    0.066
## 6 Seq_depth_transformed 0.071    0.192    0.006
## 5                  SexM 0.043    0.150    0.001
## 3           TreatmentUC 0.024    0.115    0.000
## 2           TreatmentCU 0.021    0.109    0.000
## 4           TreatmentCC 0.020    0.108    0.000
```

```
#### GET CONFIDENCE INTERVALS FROM GAMMA

summary3<-data.frame(confint(shannon_model))
summary3est<-summary(shannon_model) # add estimate as column
summary3est<-data.frame(summary3est$coefficients)

summary3$Estimate<-summary3est$Estimate
summary3<-summary3[c(1:4),] #keep just first 4 rows

summary3[2:4,]<-summary3[2:4,]+summary3[1,3] # add intercept to estimates for forest and organic
row.names(summary3)<-c("UU", "CU", "UC", "CC")
names(summary3)<-c("CI_lower", "CI_upper", "Est")
summary3$Treatment<-row.names(summary3)
```

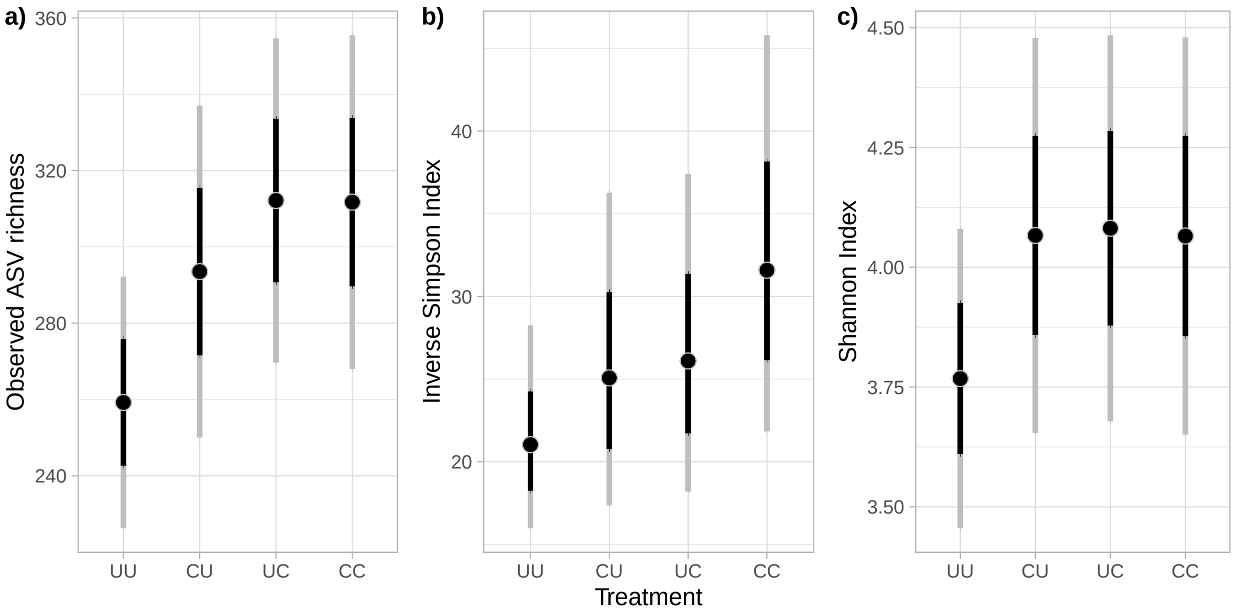
```
summary3$Treatment<-factor(summary3$Treatment, levels = c(c("UU", "CU", "UC", "CC")))


# standard errors

summary3_coef1<-summary(shannon_model)
summary3_coef1$coefficients
```

```
##                          Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)             3.7676773  0.1573776 23.940365 9.884541e-43
## TreatmentCU             0.2986959  0.2077574  1.437715 1.537004e-01
## TreatmentUC             0.3137031  0.2030363  1.545059 1.255552e-01
## TreatmentCC             0.2975433  0.2089410  1.424054 1.576072e-01
## SexM                   -0.3114881  0.1480917 -2.103345 3.799668e-02
## Seq_depth_transformed  0.2019643  0.0735817  2.744762 7.203664e-03
```

```
summary3_coef1<- data.frame(summary3_coef1$coefficients)
names(summary3_coef1)
```

```
## [1] "Estimate"   "Std..Error" "t.value"    "Pr...t.."
```

```
summary3$SE_lower<-summary3$Est-summary3_coef1$Std..Error[1:4]
summary3$SE_upper<-summary3$Est+summary3_coef1$Std..Error[1:4]

### ESIMATES AND CONFIDENCE INTERBALS ARE BACKTRANSFORMED IN PLOT USING EXP()

p3<-ggplot(summary3, aes( x =Treatment, y = Est))+
  geom_errorbar(aes(ymin = CI_lower, ymax = CI_upper), width = 0, col = "grey", size = 1.5)+
  geom_errorbar(aes(ymin = SE_lower, ymax = SE_upper), width = 0,  size = 1.5)+
   geom_point( size = 4, pch = 21, fill = "black", col = "grey")+
  theme_light(base_size = 14)+
  ylab("Shannon Index")+
  theme(legend.position = "none")+
  xlab("")



ggpubr::ggarrange(p1, p2, p3, ncol = 3, labels = c("a)", "b)", "c)"))
```



```
tab_model(observed_model, invsimp_model, shannon_model,  pred.labels = c("Intercept [UU]", "Treatment [
```

a) Observed ASV richness

  b) Inverse Simpson

    c) Shannon

Predictors
Estimates
CI
p
Estimates
CI
p
Estimates
CI
p
Intercept [UU]
259.22
226.28 – 292.17
<0.001
21.03
15.98 – 28.25
<0.001
3.77
3.46 – 4.08
<0.001
Treatment [CU]
34.30
-9.19 – 77.79
0.121
1.19

0.83 – 1.72
0.350
0.30
-0.11 – 0.71
0.154
Treatment [UC]
52.93
10.42 – 95.43
0.015
1.24
0.86 – 1.78
0.240
0.31
-0.09 – 0.72
0.126
Treatment [CC]
52.50
8.76 – 96.24
0.019
1.50
1.04 – 2.18
0.032
0.30
-0.12 – 0.71
0.158
Sex [Male]
-26.23
-57.24 – 4.77
0.096
0.74
0.57 – 0.96
0.025
-0.31
-0.61 – -0.02
0.038
Sequencing depth
40.84
25.44 – 56.25
<0.001
1.10
0.97 – 1.26
0.147
0.20
0.06 – 0.35
0.007
Observations
104
104
104
R2 / R2 adjusted
0.297 / 0.261
0.127
0.140 / 0.096

```
#ggsave("Figures/Fig3.pdf")
```

Beta diversity - constrained ordination

Weighted Unifrac

```
#############


wunifrac_dist<-distance(frog_rare, method = "wunifrac")
otutable<-data.frame(t(frog_rare@otu_table@.Data))
metadata <- data.frame(sample_data(frog_rare))


Treatment <- metadata$Treatment
Seq_depth <- as.numeric(scale(metadata$Seq_depth))
Mass <-metadata$Mass
Sex<-as.factor(metadata$Sex)
Date<-as.factor(metadata$date_cat)

metadata<- metadata %>% mutate(MassCat = case_when((Mass <2.45 ~ "Light"),
                                          (Mass > 3 ~ "Heavy")))

metadata$MassCat <-ifelse(is.na(metadata$MassCat), "Average", metadata$MassCat)

MassCat <-as.factor(metadata$MassCat)


final_model<-capscale(wunifrac_dist ~
                   Treatment+
                   Seq_depth+
                     Sex,
                env = metadata,
                comm = otutable)




# Note: including mass reduces effect of treatment - mechanism?

# weighted unifrac

anova_wunifrac<-anova.cca(final_model, by="terms")
anova_wunifrac
```

```
## Permutation test for capscale under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
```

```
## Model: capscale(formula = wunifrac_dist ~ Treatment + Seq_depth + Sex, comm = otutable, env = metada
##           Df SumOfSqs      F Pr(>F)
## Treatment  3  0.15612 1.7294  0.017 *
## Seq_depth  1  0.04207 1.3980  0.157
## Sex        1  0.06180 2.0538  0.038 *
## Residual  98  2.94895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
####### plot #####
####### plot #####
####### plot #####


## extract data from model

final_model_df<-scores(final_model)

# extract CAP scores

vectors_df<-data.frame(final_model_df$sites)
vectors_df$feature.id<-row.names(vectors_df)


# merge with info on dominant family

sample_metadata<-data.frame(sample_data(frog_rare))[,c("feature.id", "Treatment")]
vectors_df<-merge(vectors_df, sample_metadata, by = "feature.id")


#### add arrows #########
#### add arrows #########
#### add arrows #########
#### add arrows #########

centroids_df<-data.frame(final_model_df$centroids)
centroids_df<-centroids_df[1:6,]
centroids_df
```

```
##                     CAP1        CAP2
## TreatmentUU  0.57669504  0.04861942
## TreatmentCU  0.05663779 -0.11379604
## TreatmentUC -0.14513312  0.18877645
## TreatmentCC -0.53845764 -0.15639851
## SexF         0.05447384 -0.38786690
## SexM        -0.04320339  0.30761858
```

```
row.names(centroids_df)<-c("Treatment: UU", "Treatment: CU","Treatment: UC","Treatment: CC", "Sex:F", "



### add taxa scores ########
### add taxa scores ########
### add taxa scores ########
### add taxa scores ########



species_scores<-data.frame(final_model_df$species)

summary(species_scores$CAP1)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.2366392 -0.0001541 -0.0000168  0.0000000  0.0000466  0.4308569
```

```
summary(species_scores$CAP2)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0908923 -0.0001191 -0.0000125  0.0000000  0.0000383  0.4885048
```

```
species_scores<-subset(species_scores,  (CAP2 > 0.2 | CAP2 < -0.2) |  (CAP1 > 0.2 | CAP1 < -0.2))

species_scores$CAP1<-species_scores$CAP1 *4
species_scores$CAP2<-species_scores$CAP2 *4

###########
###########
###########
###########

vectors_wunifrac<-vectors_df
centroids_wunifrac<-centroids_df
species_wunifrac<-species_scores

# colour palette
pal<-pals::stepped3()[c(1,5,9,13)]
pal<-pals::tol()[c(1,3,4,12)]


plot_wunifrac<-ggplot(vectors_wunifrac, aes(x = CAP1, y = CAP2))+

   stat_ellipse(geom = "polygon", aes(fill = Treatment, col = Treatment), level = 0.9, alpha = 0.3, siz
  geom_point(aes(fill =Treatment), pch = 21, size = 3, alpha = 1, stroke = 1, col = "black")+

  theme_bw()+
 scale_fill_viridis(discrete = TRUE)+
 scale_color_viridis(discrete = TRUE)+

  # add arrows

  geom_segment(data=centroids_wunifrac[1:4,], aes(x = 0, y = 0, xend = CAP1*2, yend = CAP2*2),
    arrow = arrow(length = unit(0.5, "cm"), type = "closed"), lwd = 1,  col = "black")+
  ggrepel::geom_label_repel(data=centroids_wunifrac[1:4,],
    alpha = 0.9, col = "black", size = 4, fill = "yellow",
    #hjust = c(0,1),
    #vjust = c(1,1),
    aes(CAP1*2, CAP2*2, label = row.names(centroids_wunifrac[1:4,])))+

    # add species
 # geom_point(data = species_scores,  size = 6, col = "black", pch = 8, stroke = 0, alpha = 0)+
#  geom_label_repel(data=species_scores,
#    alpha = 0.9, col = "black", size = 3.5, fill = "skyblue",
#    aes( label = row.names(species_scores)))+


  theme_light(base_size = 14)+
  ggtitle("a) Treatment: WU")
```

Bray Curtis

```
#############

bray_dist<-distance(frog_rare, method = "bray")
otutable<-data.frame(t(frog_rare@otu_table@.Data))
metadata <- data.frame(sample_data(frog_rare))


Treatment <- metadata$Treatment
Seq_depth <- as.numeric(scale(metadata$Seq_depth))
Mass <-metadata$Mass
Sex<-as.factor(metadata$Sex)
Date<-as.factor(metadata$date_cat)

metadata<- metadata %>% mutate(MassCat = case_when((Mass <2.45 ~ "Light"),
                                          (Mass > 3 ~ "Heavy")))

metadata$MassCat <-ifelse(is.na(metadata$MassCat), "Average", metadata$MassCat)

MassCat <-as.factor(metadata$MassCat)



final_model<-capscale(bray_dist ~
                   Treatment+
                    Seq_depth+
                     Sex,

                env = metadata,
                comm = otutable)



# Note: including mass reduces effect of treatment - mechanism?

# weighted bray
```

```
anova_bray<-anova.cca(final_model, by="terms")
round(data.frame(anova_bray), 3)
```

```
##           Df SumOfSqs     F Pr..F.
## Treatment  3    1.058 1.398  0.012
## Seq_depth  1    0.361 1.431  0.087
## Sex        1    0.806 3.198  0.001
## Residual  98   24.709    NA     NA
```

```
round(data.frame(anova_wunifrac), 3)
```

```
##           Df SumOfSqs     F Pr..F.
## Treatment  3    0.156 1.729  0.017
## Seq_depth  1    0.042 1.398  0.157
## Sex        1    0.062 2.054  0.038
## Residual  98    2.949    NA     NA
```

```
####### plot #####
####### plot #####
####### plot #####


## extract data from model

final_model_df<-scores(final_model)


# extract CAP scores

vectors_df<-data.frame(final_model_df$sites)
vectors_df$feature.id<-row.names(vectors_df)


# merge with info on dominant family

sample_metadata<-data.frame(sample_data(frog_rare))[,c("feature.id", "Treatment")]
vectors_df<-merge(vectors_df, sample_metadata, by = "feature.id")



#### add arrows #########
#### add arrows #########
#### add arrows #########
#### add arrows #########

centroids_df<-data.frame(final_model_df$centroids)
centroids_df<-centroids_df[1:6,]
row.names(centroids_df)<-c("Treatment: UU", "Treatment: CU","Treatment: UC","Treatment: CC", "SexF", "Se



### add taxa scores ########
### add taxa scores ########


species_scores<-data.frame(final_model_df$species)

summary(species_scores$CAP1)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.9884724 -0.0000471  0.0000161  0.0000000  0.0002009  0.2321747
```

```
summary(species_scores$CAP2)
```

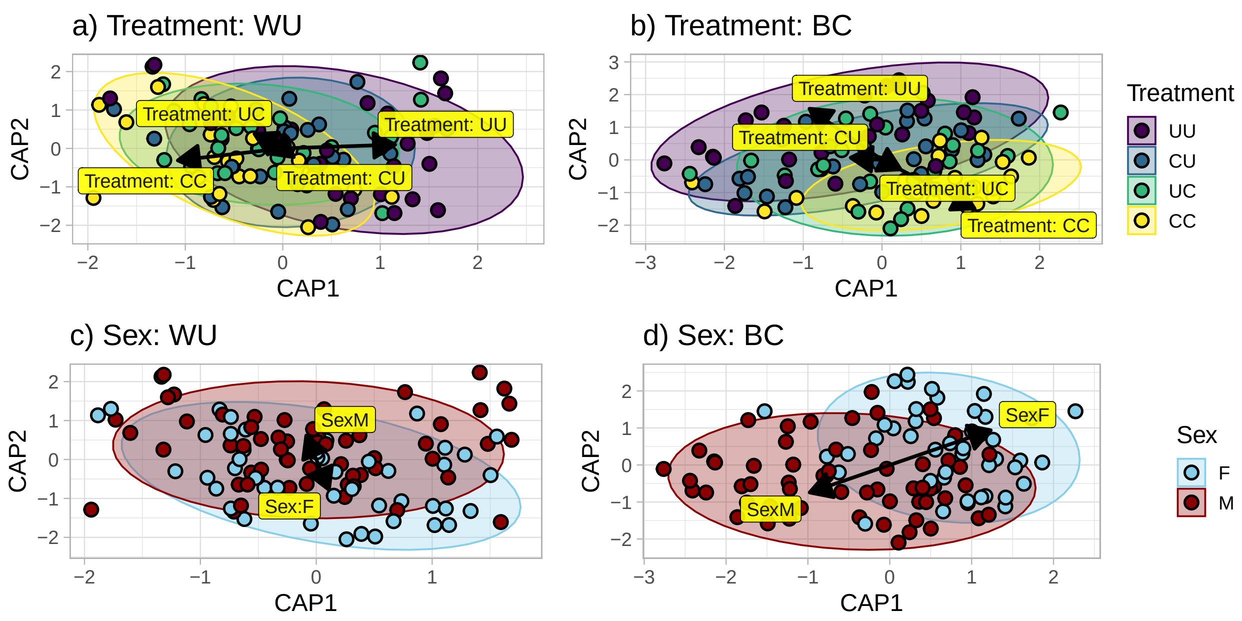
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.4944150 -0.0001981 -0.0000329  0.0000000  0.0000542  0.6013368
```

```
species_scores<-subset(species_scores,  (CAP2 > 0.2 | CAP2 < -0.2) |  (CAP1 > 0.2 | CAP1 < -0.2))

species_scores$CAP1<-species_scores$CAP1 *3
species_scores$CAP2<-species_scores$CAP2 *3

###########################
###########################
###########################
###########################

plot_bray<-ggplot(vectors_df, aes(x = CAP1, y = CAP2))+

    stat_ellipse(geom = "polygon", aes(fill = Treatment, col = Treatment), level = 0.9, alpha = 0.3, si
  geom_point(aes(fill =Treatment), pch = 21, size = 3, alpha = 1, stroke = 1, col = "black")+


 scale_fill_viridis(discrete = TRUE)+
 scale_color_viridis(discrete = TRUE)+

  # add arrows

  geom_segment(data=centroids_df[1:4,], aes(x = 0, y = 0, xend = CAP1*2, yend = CAP2*2),
    arrow = arrow(length = unit(0.5, "cm"), type = "closed"), lwd = 1,  col = "black")+
  ggrepel::geom_label_repel(data=centroids_df[1:4,],
    alpha = 0.9, col = "black", size = 4, fill = "yellow",

    aes(CAP1*2, CAP2*2, label = row.names(centroids_df[1:4,])))+

    # add species
 # geom_point(data = species_scores,  size = 6, col = "black", pch = 8, stroke = 0, alpha = 0)+
#  geom_label_repel(data=species_scores,
#    alpha = 0.9, col = "black", size = 3.5, fill = "skyblue",
#    aes( label = row.names(species_scores)))+


  theme_light(base_size = 14)+
  ggtitle("b) Treatment: BC")
```

```
### plot together

treatment_plots<-ggarrange(plot_wunifrac, plot_bray,  common.legend = T, legend = "right")
```

Group by sex

```
sex_wunifrac<-ggplot(vectors_wunifrac, aes(x = CAP1, y = CAP2))+

   stat_ellipse(geom = "polygon", aes(fill = Sex, col = Sex), level = 0.9, alpha = 0.3, size = 0.5)+
  geom_point(aes(fill =Sex), pch = 21, size = 3, alpha = 1, stroke = 1, col = "black")+

  theme_bw()+
 scale_fill_manual(values = c("skyblue", "darkred"))+
 scale_color_manual(values = c("skyblue", "darkred"))+

  # add arrows

  geom_segment(data=centroids_wunifrac[5:6,], aes(x = 0, y = 0, xend = CAP1*2, yend = CAP2*2),
    arrow = arrow(length = unit(0.5, "cm"), type = "closed"), lwd = 1,  col = "black")+
  ggrepel::geom_label_repel(data=centroids_wunifrac[5:6,],
    alpha = 0.9, col = "black", size = 4, fill = "yellow",
    #hjust = c(0,1),
    #vjust = c(1,1),
    aes(CAP1*2, CAP2*2, label = row.names(centroids_wunifrac[5:6,])))+
  theme_light(base_size = 14)+
  ggtitle("c) Sex: WU")+
  theme(plot.margin = margin(0.2,0.8,0.2,0.2, "cm"))

#################
#################
#################
#################


sex_bray<- ggplot(vectors_df, aes(x = CAP1, y = CAP2))+

    stat_ellipse(geom = "polygon", aes(fill = Sex, col = Sex), level = 0.9, alpha = 0.3, size = 0.5)+
  geom_point(aes(fill =Sex), pch = 21, size = 3, alpha = 1, stroke = 1, col = "black")+


 scale_fill_manual(values = c("skyblue", "darkred"))+
 scale_color_manual(values = c("skyblue", "darkred"))+

  # add arrows

  geom_segment(data=centroids_df[5:6,], aes(x = 0, y = 0, xend = CAP1*2, yend = CAP2*2),
    arrow = arrow(length = unit(0.5, "cm"), type = "closed"), lwd = 1,  col = "black")+
  ggrepel::geom_label_repel(data=centroids_df[5:6,],
    alpha = 0.9, col = "black", size = 4, fill = "yellow",
    aes(CAP1*2, CAP2*2, label = row.names(centroids_df[5:6,])))+


  theme_light(base_size = 14)+
  ggtitle("d) Sex: BC")+
  theme(plot.margin = margin(0.2,1.3,0.2,0, "cm"))

sex_plots<-ggarrange(sex_wunifrac, sex_bray,  common.legend = T, legend = "right")


ggarrange(treatment_plots, sex_plots,  common.legend = T, legend = "right", nrow = 2, align = "v")
```



```
#ggsave("Figures/Fig4.pdf")
```

Joint-species distribution modelling

JSDM: Genus level

```
# https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.13303

# final model

frog_genus<-tax_glom(frog_filtered, taxrank = "Genus")

frog_genus_core<-core(frog_genus, detection = 20, prevalence = 0.70)
#frog_genus_core<-microbiome::aggregate_top_taxa(frog_filtered, "Genus", top = 45)

frog_genus
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 661 taxa and 104 samples ]
## sample_data() Sample Data:       [ 104 samples by 10 sample variables ]
## tax_table()   Taxonomy Table:    [ 661 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 661 tips and 660 internal nodes ]
```

```
frog_genus_core
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 70 taxa and 104 samples ]
## sample_data() Sample Data:       [ 104 samples by 10 sample variables ]
## tax_table()   Taxonomy Table:    [ 70 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 70 tips and 69 internal nodes ]
```

```
sum(sample_sums(frog_genus_core))/sum(sample_sums(frog_genus))
```

```
## [1] 0.8882152
```

```
taxtable<-data.frame(tax_table(frog_genus_core))

frog_genus_core<-subset_taxa( frog_genus_core, Genus != "uncultured" )
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "metagenome" )
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "uncultured bacterium" )
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "Other" )
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "Unknown" )
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "Allorhizobium-Neorhizobium-Pararhizobium-Rhizob
frog_genus_core<-subset_taxa( frog_genus_core, Genus != "CL500-29 marine group" )

taxa_keep<- taxa_names(frog_genus_core)

sum(sample_sums(frog_genus_core))/sum(sample_sums(frog_genus))
```

```
## [1] 0.8486052
```

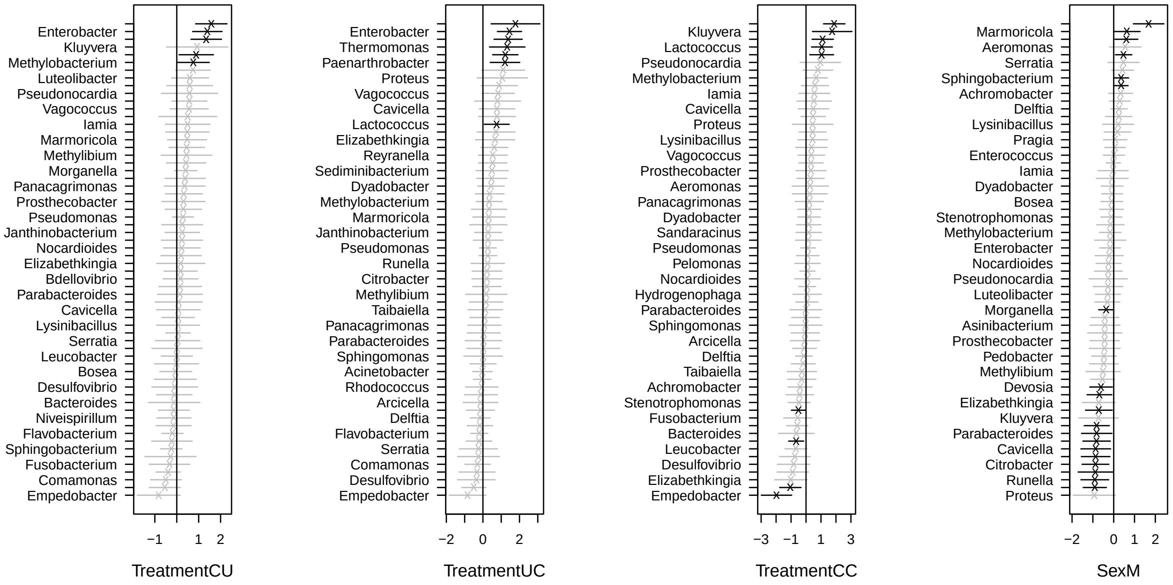
```
taxtable<-data.frame(tax_table(frog_genus_core))

taxa_names(frog_genus_core)<- taxtable$Genus


## Extract relevant data for model

y <- data.frame(t(otu_table(frog_genus_core)))
X<-data.frame(sample_data(frog_genus_core))


X$Mass_scaled<-as.numeric(scale(X$Mass))
X$Seq_depth_scaled<-as.numeric(scale(sqrt(X$Seq_depth)))

X<-X[,c("Treatment", "Mass_scaled", "Seq_depth_scaled", "Sex")]
```

Model comparison

```
fit_1LV <- gllvm(y, X,
             num.lv = 1,
            formula = ~ Treatment+ Sex  + Seq_depth_scaled,
             family = "negative.binomial")

fit_2LV <- gllvm(y, X,
             num.lv = 2,
            formula = ~ Treatment+ Sex + Seq_depth_scaled,
             family = "negative.binomial")


fit_3LV <- gllvm(y, X,
             num.lv = 3,
            formula = ~ Treatment+ Sex  + Seq_depth_scaled,
             family = "negative.binomial")




# compare AIC

AIC( fit_1LV, fit_2LV, fit_3LV)
```

Final model

```
fit <- gllvm(y, X,
             num.lv = 3,
            formula = ~ Treatment+ Sex + Seq_depth_scaled,
             family = "negative.binomial")

coefplot(fit, cex.ylab = 1, mar = c(5,12,2,1), which.Xcoef = c(1:4))
```



```
dev.off()
```

```
## null device
##           1
```

```
cr1<-getResidualCor(fit)

corrplot::corrplot(cr1, type = "lower", order = "hclust", tl.cex = 0.7)

dev.off()
```

```
## null device
##           1
```

Extract estimates

```
### extract estimates and CIs ###
### extract estimates and CIs ###


df<-coef(fit)
est_df<-data.frame(df$Intercept)

est_df2<-data.frame(df$Xcoef)[,1:3] # choose columns of interest

est_df3<-merge(est_df, est_df2, by = 0)

est_df3$CU_est<- est_df3$TreatmentCU+est_df3$df.Intercept
est_df3$UC_est<- est_df3$TreatmentUC+est_df3$df.Intercept
est_df3$CC_est<- est_df3$TreatmentCC+est_df3$df.Intercept
```

```
est_df3<-est_df3[,c(1,2,6:8)]

# order genera

row.names(est_df3)<-est_df3$Row.names
est_df3<-est_df3[taxa_names(frog_genus_core),]


############# extract confindence intervals  ####
############# extract confindence intervals  ####
############# extract confindence intervals  ####
############# extract confindence intervals  ####


confint_df<-data.frame(confint(fit))

confint_df$row<- 1:nrow(confint_df)

confint_df<-confint_df[190:437,] # manually choose rows that represent estimates for variable of interes
confint_df$row<- 1:nrow(confint_df)


# add a column with correct variable level

#confint_df$Treatment<-c(rep("UU", 40), rep("CU", 40), rep("UC", 40), rep("CC", 40))
confint_df$Treatment<-c(rep("UU", nrow(est_df)), rep("CU", nrow(est_df)), rep("UC", nrow(est_df)), rep(

# column with taxa names. Should be automatically in the correct order but double check
confint_df$Genus<-c(taxa_names(frog_genus_core), taxa_names(frog_genus_core), taxa_names(frog_genus_core



### extract standard errors #######
### extract standard errors #######
### extract standard errors #######
### extract standard errors #######

se_df<-se(fit)
se1<-data.frame(se_df$sd$Xcoef)[,1:3] # select relevant columns
se2<- data.frame(se_df$sd$beta0)
se_df<-merge(se1, se2, by = 0)
row.names(se_df)<-se_df$Row.names
se_df<-se_df[taxa_names(frog_genus_core),]
names(se_df)<-c("Genus", "CU", "UC", "CC", "UU")


# now have estimates, confidence intervals, and standard errors as seperate data frames, but they are i


### get estimates and confidence intervals for UU (reference group)

UU_ci<-subset(confint_df, Treatment == "UU")
```

```
UU_df<- data.frame(est_df3$df.Intercept, UU_ci$X2.5.., UU_ci$X97.5.., UU_ci$Genus, se_df$UU)
names(UU_df)<-c("Est", "Lower_CI", "Upper_CI", "Genus", "SE")
UU_df$Treatment <-"UU"

# get estimates and CIs for CU. This needs to be modifed

CU_ci<-subset(confint_df, Treatment == "CU")

CU_df<- data.frame(est_df3$CU_est, est_df3$df.Intercept, CU_ci$X2.5.., CU_ci$X97.5.., CU_ci$Genus, se_d

CU_df$CU_ci.X2.5..<-CU_df$est_df3.df.Intercept+ CU_df$CU_ci.X2.5..
CU_df$CU_ci.X97.5..<-CU_df$est_df3.df.Intercept+ CU_df$CU_ci.X97.5..
CU_df<-CU_df[,c(1,3:6)]

names(CU_df)<-c("Est", "Lower_CI", "Upper_CI", "Genus", "SE")
CU_df$Treatment <-"CU"


# get estimates and CIs for UC

UC_ci<-subset(confint_df, Treatment == "UC")

UC_df<- data.frame(est_df3$UC_est, est_df3$df.Intercept, UC_ci$X2.5.., UC_ci$X97.5.., UC_ci$Genus, se_d
UC_df$UC_ci.X2.5..<-UC_df$est_df3.df.Intercept+ UC_df$UC_ci.X2.5..
UC_df$UC_ci.X97.5..<-UC_df$est_df3.df.Intercept+ UC_df$UC_ci.X97.5..
UC_df<-UC_df[,c(1,3:6)]
names(UC_df)<-c("Est", "Lower_CI", "Upper_CI", "Genus", "SE")
UC_df$Treatment <-"UC"


# get estimates and CIs for CC

CC_ci<-subset(confint_df, Treatment == "CC")

CC_df<- data.frame(est_df3$CC_est, est_df3$df.Intercept, CC_ci$X2.5.., CC_ci$X97.5.., CC_ci$Genus, se_d
CC_df$CC_ci.X2.5..<-CC_df$est_df3.df.Intercept+ CC_df$CC_ci.X2.5..
CC_df$CC_ci.X97.5..<-CC_df$est_df3.df.Intercept+ CC_df$CC_ci.X97.5..
CC_df<-CC_df[,c(1,3:6)]
names(CC_df)<-c("Est", "Lower_CI", "Upper_CI", "Genus", "SE")
CC_df$Treatment <-"CC"


treatment_ci_df<-rbind(UU_df, CU_df, UC_df, CC_df)


treatment_ci_df$Treatment<-factor(treatment_ci_df$Treatment, levels = c("UU", "CU", "UC", "CC"))


## get p values


summary<-summary(fit)
summary_df<-data.frame(summary$Coef.tableX)
```

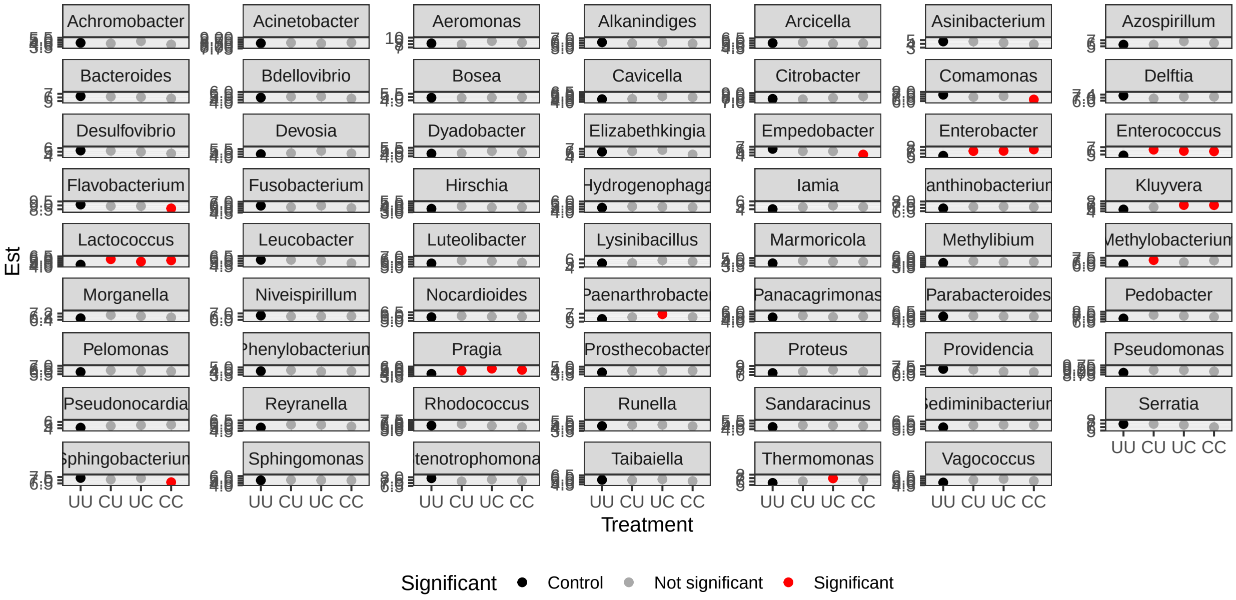
```r
summary_df$nrow <- 1:nrow(summary_df)
#View(summary_df)
summary_df<-summary_df[1:186,] # keep only treatment stats

treatment_ci_df$P_val<-NA
treatment_ci_df$Significant<-NA

dim(treatment_ci_df)
```

```
## [1] 248   8
```

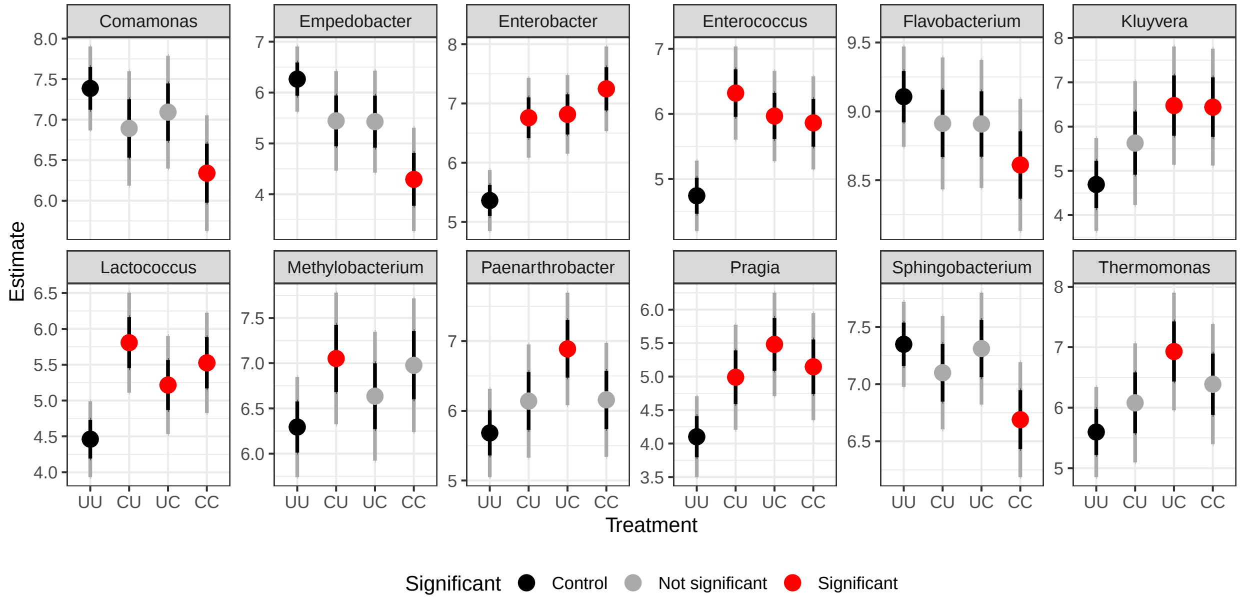
```r
treatment_ci_df$P_val[63:248]<- summary_df$Pr...z..

treatment_ci_df$Significant<-ifelse(treatment_ci_df$P_val <0.05, "Significant", "Not significant")
treatment_ci_df$Significant<-ifelse(is.na(treatment_ci_df$Significant), "Control", treatment_ci_df$Signi

treatment_ci_df$Treatment<-factor(treatment_ci_df$Treatment, levels = c("UU", "CU", "UC", "CC"))



# plot all genera

head(treatment_ci_df)
```

```
##        Est Lower_CI Upper_CI            Genus        SE Treatment P_val
## 1 7.349193 6.976958 7.721427 Sphingobacterium 0.1899189        UU    NA
## 2 7.121299 6.512760 7.729837       Pedobacter 0.3104845        UU    NA
## 3 6.349678 5.457898 7.241457      Bacteroides 0.4549977        UU    NA
## 4 5.480900 4.701014 6.260785  Parabacteroides 0.3979081        UU    NA
## 5 5.523642 4.793797 6.253487        Arcicella 0.3723766        UU    NA
## 6 4.462898 3.899924 5.025871      Dyadobacter 0.2872365        UU    NA
##   Significant
## 1     Control
## 2     Control
## 3     Control
## 4     Control
## 5     Control
## 6     Control
```

```r
treatment_ci_df<-treatment_ci_df[!is.na(treatment_ci_df$Est),]


ggplot(treatment_ci_df, aes(y = Est, x = Treatment))+

  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0, col = "darkgrey", size = 1)+
  facet_wrap(~Genus, nrow = 10, scales = "free_y")+
   geom_errorbar(aes(ymin = Est-SE, ymax = Est+SE), width = 0, col = "black", size = 1)+
  theme_bw(base_size = 12)+
  scale_color_manual(values = c("black", "darkgrey", "red"))+
   geom_point(size = 2, aes(col = Significant))+
  theme(legend.position = "bottom")
```



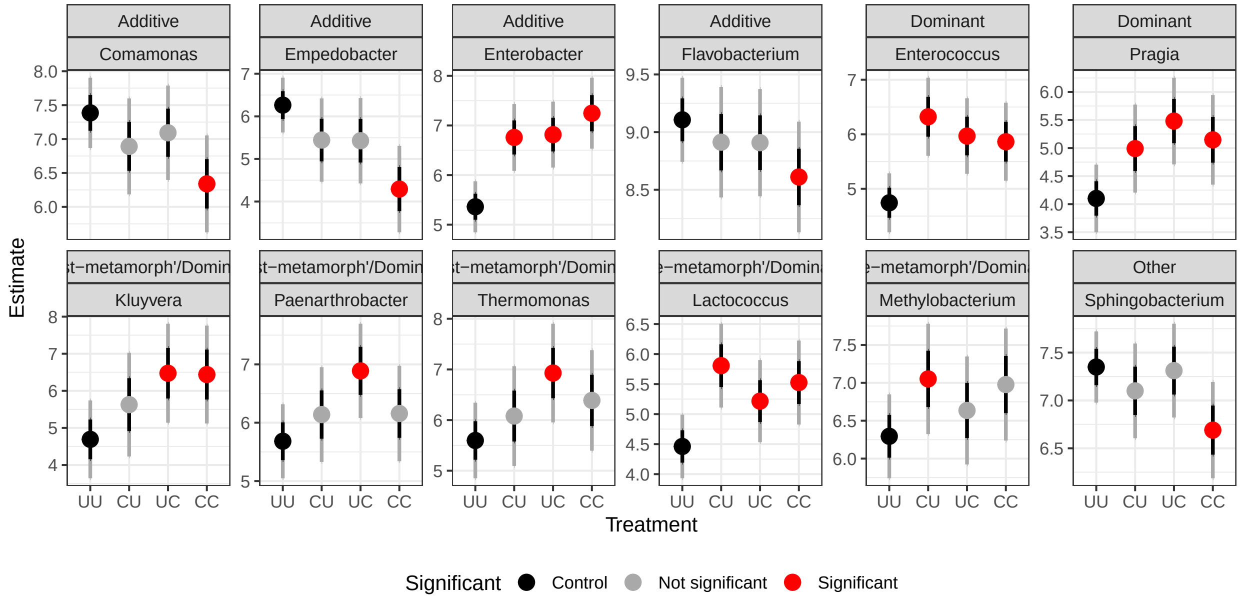
```
## plot only significant

estimates_sig<- subset(treatment_ci_df,  Genus == "Comamonas" |  Genus == "Empedobacter" |  Genus == "E

estimates_sig<-estimates_sig%>%arrange(Treatment, Genus)

ggplot(estimates_sig,
       aes(y = Est, x = Treatment))+
   geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0, col = "darkgrey", size = 1)+
  facet_wrap(~Genus, nrow = 2, scales = "free_y")+
   geom_errorbar(aes(ymin = Est-SE, ymax = Est+SE), width = 0, col = "black", size = 1)+
  theme_bw(base_size = 12)+
  scale_color_manual(values = c("black", "darkgrey", "red"))+
   geom_point(size = 4, aes(col = Significant))+
  theme(legend.position = "bottom")+
  ylab("Estimate")
```



```r
estimates_sig$Genus<-factor(estimates_sig$Genus)
unique(estimates_sig$Genus)
```

```
##  [1] Comamonas        Empedobacter     Enterobacter     Enterococcus
##  [5] Flavobacterium   Kluyvera         Lactococcus      Methylobacterium
##  [9] Paenarthrobacter Pragia           Sphingobacterium Thermomonas
## 12 Levels: Comamonas Empedobacter Enterobacter Enterococcus ... Thermomonas
```

```r
estimates_sig$Effect_type <- rep(c("Additive",
                                   "Additive",
                                   "Additive",
                                   "Dominant", #Enterococcus
                                   "Additive",  #Flavo
                                   "Post-metamorph'/Dominant", #Kluyvera
                                   "Pre-metamorph'/Dominant", #Lactococcus
                                   "Pre-metamorph'/Dominant", #Meth
                                   "Post-metamorph'/Dominant", #Paenar
                                   "Dominant", #Pragia
                                   "Other",
                                   "Post-metamorph'/Dominant"), 4)

estimates_sig$Effect_type <-factor(estimates_sig$Effect_type , levels = c("Additive", "Dominant", "Post

ggplot(estimates_sig,
       aes(y = Est, x = Treatment))+
   geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0, col = "darkgrey", size = 1)+
  facet_wrap(~Effect_type+Genus, nrow = 2, scales = "free_y")+
   geom_errorbar(aes(ymin = Est-SE, ymax = Est+SE), width = 0, col = "black", size = 1)+
  theme_bw(base_size = 12)+
  scale_color_manual(values = c("black", "darkgrey", "red"))+
   geom_point(size = 4, aes(col = Significant))+
  theme(legend.position = "bottom")+
  ylab("Estimate")
```



JSDM: ASV level

```
##### ASV level ##########
##### ASV level ##########

frog_core<-core(frog_filtered, detection = 30, prevalence = 0.70)
taxanames<-taxa_names(frog_core)

sum(taxa_sums(frog_core))/sum(taxa_sums(frog_filtered))
```

```
## [1] 0.619076
```

```
y <- data.frame(t(otu_table(frog_core)))
X<-data.frame(sample_data(frog_core))


X$Mass_scaled<-as.numeric(scale(X$Mass))
X$Seq_depth_scaled<-as.numeric(scale(sqrt(X$Seq_depth)))

X<-X[,c("Treatment","Sex", "Mass_scaled", "Seq_depth_scaled")]

X$Treatment<-factor(X$Treatment, levels = c("UU", "CU", "UC", "CC"))
```

Model comparison

```
fit <- gllvm(y, X,
             num.lv = 2,
            formula = ~ Treatment+  Sex +Seq_depth_scaled,
             family = "negative.binomial")
```

```r
#############

fit1 <- gllvm(y, X,
             num.lv = 2,
            formula = ~ Treatment+ Mass_scaled  +Seq_depth_scaled,
             family = "negative.binomial")

###################

fit2 <- gllvm(y, X,
             num.lv = 3,
            formula = ~ Treatment+  Sex +Seq_depth_scaled,
             family = "negative.binomial")

#############

fit3 <- gllvm(y, X,
             num.lv = 3,
            formula = ~ Treatment+ Mass_scaled  +Seq_depth_scaled,
             family = "negative.binomial")

###################


AIC(fit, fit1, fit2, fit3)
```

Final model

```r
fit <- gllvm(y, X,
             num.lv = 3,
            formula = ~ Treatment+ Sex  +Seq_depth_scaled,
             family = "negative.binomial")
dev.off()
```

```
## null device
##           1
```

```r
coefplot(fit, cex.ylab = 0.7, which.Xcoef = c(1:4), mar = c(5,14,2,1))
```

```
## Error in plot.new(): figure margins too large
```

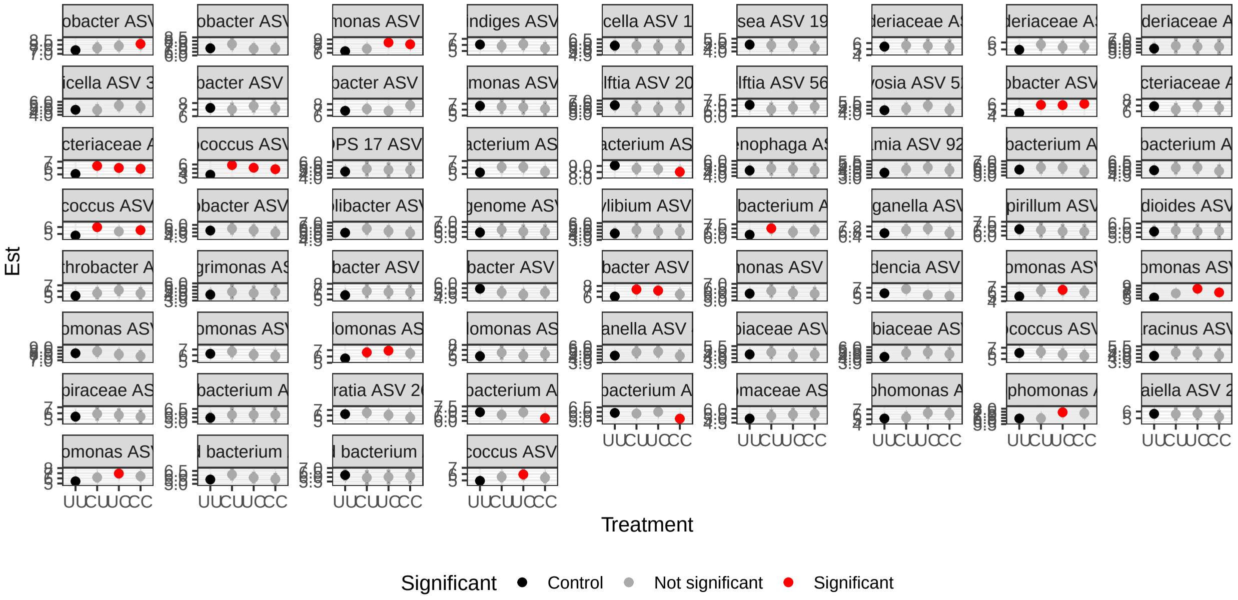
```r
dev.off()
```

```
## null device
##           1
```

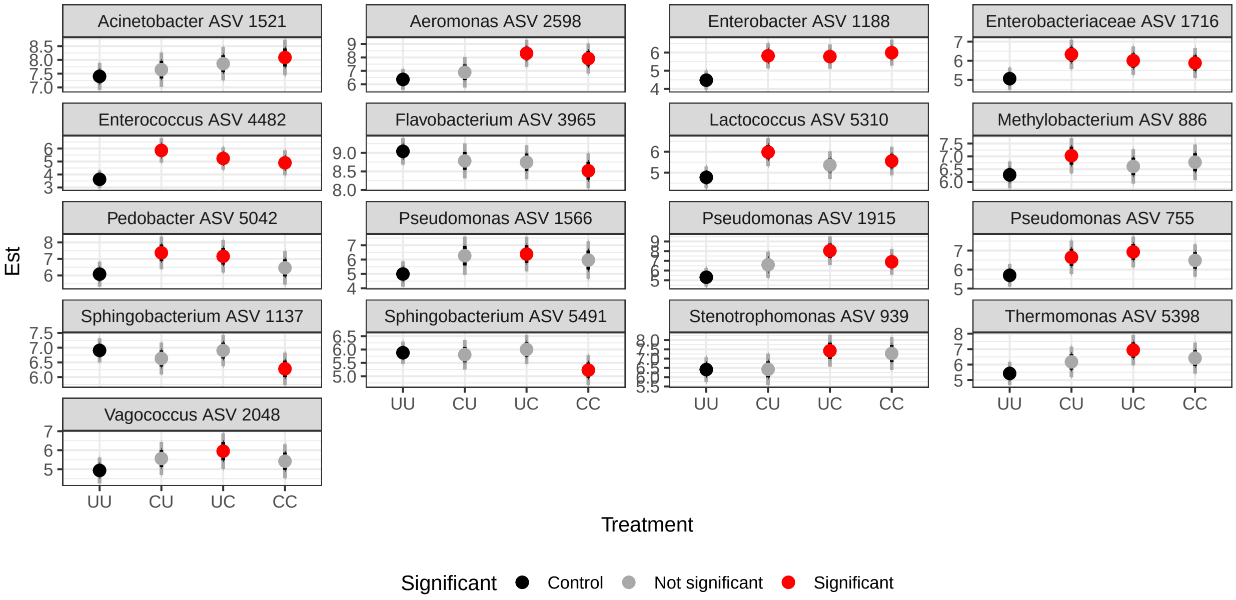
```r
### extract estimates and CIs ###
### extract estimates and CIs ###
### extract estimates and CIs ###
### extract estimates and CIs ###

df<-coef(fit)
est_df<-data.frame(df$Intercept)

est_df2<-data.frame(df$Xcoef)[,1:3] # choose columns of interest


est_df3<-merge(est_df, est_df2, by = 0)

est_df3$CU_est<- est_df3$TreatmentCU+est_df3$df.Intercept
est_df3$UC_est<- est_df3$TreatmentUC+est_df3$df.Intercept
est_df3$CC_est<- est_df3$TreatmentCC+est_df3$df.Intercept
est_df3<-est_df3[,c(1,2,6:8)] # change for zour data

# order genera

row.names(est_df3)<-est_df3$Row.names
est_df3<-est_df3[gsub(' ', '.', taxa_names(frog_core)),] # order taxa


############# extract confindence intervals  ####
############# extract confindence intervals  ####
############# extract confindence intervals  ####
############# extract confindence intervals  ####


confint_df<-data.frame(confint(fit))

confint_df$row<- 1:nrow(confint_df)

confint_df<-confint_df[205:472,] # manually choose rows that represent estimates for variable of intere
confint_df$row<- 1:nrow(confint_df)


# add a column with correct variable level


confint_df$Treatment<-c(rep("UU", nrow(est_df)), rep("CU", nrow(est_df)), rep("UC", nrow(est_df)), rep(

# column with taxa names. Should be automatically in the correct order but double check

confint_df$ASV<-c(taxa_names(frog_core), taxa_names(frog_core), taxa_names(frog_core), taxa_names(frog_


### extract standard errors #######
### extract standard errors #######
### extract standard errors #######
### extract standard errors #######

se_df<-se(fit)
se1<-data.frame(se_df$sd$Xcoef)[,1:3] # select relevant columns
se2<- data.frame(se_df$sd$beta0)
```

```r
se_df<-merge(se1, se2, by = 0)
row.names(se_df)<-se_df$Row.names
se_df<-se_df[gsub(' ', '.', taxa_names(frog_core)),]
names(se_df)<-c("ASV", "CU", "UC", "CC", "UU")


# now have estimates, confidence intervals, and standard errors as seperate data frames, but they are i


### get estimates and confidence intervals for UU (reference group)

UU_ci<-subset(confint_df, Treatment == "UU")

UU_df<- data.frame(est_df3$df.Intercept, UU_ci$X2.5.., UU_ci$X97.5.., UU_ci$ASV, se_df$UU)
names(UU_df)<-c("Est", "Lower_CI", "Upper_CI", "ASV", "SE")
UU_df$Treatment <-"UU"

# get estimates and CIs for CU. This needs extra lines to manually modify the CIs

CU_ci<-subset(confint_df, Treatment == "CU")

CU_df<- data.frame(est_df3$CU_est, est_df3$df.Intercept, CU_ci$X2.5.., CU_ci$X97.5.., CU_ci$ASV, se_df$C

CU_df$CU_ci.X2.5..<-CU_df$est_df3.df.Intercept+ CU_df$CU_ci.X2.5..
CU_df$CU_ci.X97.5..<-CU_df$est_df3.df.Intercept+ CU_df$CU_ci.X97.5..
CU_df<-CU_df[,c(1,3:6)]

names(CU_df)<-c("Est", "Lower_CI", "Upper_CI", "ASV", "SE")
CU_df$Treatment <-"CU"


# get estimates and CIs for UC

UC_ci<-subset(confint_df, Treatment == "UC")

UC_df<- data.frame(est_df3$UC_est, est_df3$df.Intercept, UC_ci$X2.5.., UC_ci$X97.5.., UC_ci$ASV, se_df$U
UC_df$UC_ci.X2.5..<-UC_df$est_df3.df.Intercept+ UC_df$UC_ci.X2.5..
UC_df$UC_ci.X97.5..<-UC_df$est_df3.df.Intercept+ UC_df$UC_ci.X97.5..
UC_df<-UC_df[,c(1,3:6)]
names(UC_df)<-c("Est", "Lower_CI", "Upper_CI", "ASV", "SE")
UC_df$Treatment <-"UC"


# get estimates and CIs for CC

CC_ci<-subset(confint_df, Treatment == "CC")

CC_df<- data.frame(est_df3$CC_est, est_df3$df.Intercept, CC_ci$X2.5.., CC_ci$X97.5.., CC_ci$ASV, se_df$C
CC_df$CC_ci.X2.5..<-CC_df$est_df3.df.Intercept+ CC_df$CC_ci.X2.5..
CC_df$CC_ci.X97.5..<-CC_df$est_df3.df.Intercept+ CC_df$CC_ci.X97.5..
CC_df<-CC_df[,c(1,3:6)]
names(CC_df)<-c("Est", "Lower_CI", "Upper_CI", "ASV", "SE")
CC_df$Treatment <-"CC"
```

```
treatment_ci_df<-rbind(UU_df, CU_df, UC_df, CC_df)
treatment_ci_df$Treatment<-factor(treatment_ci_df$Treatment, levels = c("UU", "CU", "UC", "CC"))

## get p values

summary<-summary(fit)
summary_df<-data.frame(summary$Coef.tableX)
summary_df$row<-1:nrow(summary_df)

summary_df<-summary_df[1:201,] # keep only treatment stats

treatment_ci_df$P_val<-NA
treatment_ci_df$Significant<-NA

treatment_ci_df$P_val[68:268]<- summary_df$Pr...z..

treatment_ci_df$Significant<-ifelse(treatment_ci_df$P_val <0.05, "Significant", "Not significant")
treatment_ci_df$Significant<-ifelse(is.na(treatment_ci_df$Significant), "Control", treatment_ci_df$Sign

treatment_ci_df$Treatment<-factor(treatment_ci_df$Treatment, levels = c("UU", "CU", "UC", "CC"))


# plot all genera
ggplot(treatment_ci_df, aes(y = Est, x = Treatment))+

  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0, col = "darkgrey", size = 1)+
  facet_wrap(~ASV, scales = "free_y")+
  geom_errorbar(aes(ymin = Est-SE, ymax = Est+SE), width = 0, col = "black", size = 1)+
  theme_bw(base_size = 12)+
  scale_color_manual(values = c("black", "darkgrey", "red"))+
  geom_point(size = 2, aes(col = Significant))+
  theme(legend.position = "bottom")
```



```
############
############
############

sig_asvs<-unique((subset(treatment_ci_df, Significant == "Significant"))$ASV)


treatment_ci_df$Keep<-treatment_ci_df$ASV %in% sig_asvs

estimates_sig<-subset(treatment_ci_df, Keep == TRUE)

ggplot(estimates_sig, aes(y = Est, x = Treatment))+

  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0, col = "darkgrey", size = 1)+
  facet_wrap(~ASV, scales = "free_y", ncol =4)+
  geom_errorbar(aes(ymin = Est-SE, ymax = Est+SE), width = 0, col = "black", size = 1)+
  theme_bw(base_size = 12)+
  scale_color_manual(values = c("black", "darkgrey", "red"))+
  geom_point(size = 3, aes(col = Significant))+
  theme(legend.position = "bottom")
```



BLAST

```
unique(estimates_sig$ASV)
```

```
##  [1] "Sphingobacterium ASV 5491"   "Sphingobacterium ASV 1137"
##  [3] "Pedobacter ASV 5042"         "Flavobacterium ASV 3965"
##  [5] "Methylobacterium ASV 886"    "Lactococcus ASV 5310"
##  [7] "Enterococcus ASV 4482"       "Vagococcus ASV 2048"
##  [9] "Stenotrophomonas ASV 939"    "Thermomonas ASV 5398"
## [11] "Pseudomonas ASV 1915"        "Pseudomonas ASV 755"
## [13] "Pseudomonas ASV 1566"        "Acinetobacter ASV 1521"
## [15] "Aeromonas ASV 2598"          "Enterobacter ASV 1188"
## [17] "Enterobacteriaceae ASV 1716"
```

```
head(taxtable_blast)
```

```
##                                   Kingdom        Phylum       Class
## af33d42b102c84e7c328b35fada3a910 Bacteria Bacteroidetes Bacteroidia
## 006197e83a7f1ae488ac7c83312f6de7 Bacteria Bacteroidetes Bacteroidia
## e177077cef9850a43a9ef18e3e2c286b Bacteria Bacteroidetes Bacteroidia
## a6dab9446affe3c98103d9eb20881f9a Bacteria Bacteroidetes Bacteroidia
## e09be3b78ebe96dc4d2c36a386ed6f0d Bacteria Bacteroidetes Bacteroidia
## 21549c143a3d0d841deb4fcd82d459ad Bacteria Bacteroidetes Bacteroidia
##                                              Order             Family
## af33d42b102c84e7c328b35fada3a910  Flavobacteriales Crocinitomicaceae
## 006197e83a7f1ae488ac7c83312f6de7  Flavobacteriales    Cryomorphaceae
## e177077cef9850a43a9ef18e3e2c286b  Flavobacteriales    Cryomorphaceae
## a6dab9446affe3c98103d9eb20881f9a  Flavobacteriales Crocinitomicaceae
## e09be3b78ebe96dc4d2c36a386ed6f0d  Flavobacteriales Crocinitomicaceae
## 21549c143a3d0d841deb4fcd82d459ad Sphingobacteriales           AKYH767
##                                             Genus              Species
## af33d42b102c84e7c328b35fada3a910     Fluviicola           metagenome
## 006197e83a7f1ae488ac7c83312f6de7 Cryomorphaceae                 <NA>
## e177077cef9850a43a9ef18e3e2c286b Cryomorphaceae uncultured bacterium
## a6dab9446affe3c98103d9eb20881f9a     Fluviicola                 <NA>
## e09be3b78ebe96dc4d2c36a386ed6f0d     Fluviicola                 <NA>
## 21549c143a3d0d841deb4fcd82d459ad        AKYH767                 <NA>
##                                                               Taxa  ASV
## af33d42b102c84e7c328b35fada3a910 af33d42b102c84e7c328b35fada3a910 3932
## 006197e83a7f1ae488ac7c83312f6de7 006197e83a7f1ae488ac7c83312f6de7    7
## e177077cef9850a43a9ef18e3e2c286b e177077cef9850a43a9ef18e3e2c286b 5105
## a6dab9446affe3c98103d9eb20881f9a a6dab9446affe3c98103d9eb20881f9a 3751
## e09be3b78ebe96dc4d2c36a386ed6f0d e09be3b78ebe96dc4d2c36a386ed6f0d 5083
## 21549c143a3d0d841deb4fcd82d459ad 21549c143a3d0d841deb4fcd82d459ad  748
##                                                 New_Taxa
## af33d42b102c84e7c328b35fada3a910     Fluviicola ASV 3932
## 006197e83a7f1ae488ac7c83312f6de7    Cryomorphaceae ASV 7
## e177077cef9850a43a9ef18e3e2c286b Cryomorphaceae ASV 5105
## a6dab9446affe3c98103d9eb20881f9a     Fluviicola ASV 3751
## e09be3b78ebe96dc4d2c36a386ed6f0d     Fluviicola ASV 5083
## 21549c143a3d0d841deb4fcd82d459ad         AKYH767 ASV 748
```

```
subset(taxtable_blast, New_Taxa == "Sphingobacterium ASV 5491")
```

```
##                                   Kingdom        Phylum       Class
## f304e4fa0483742c48aebf7456178c7d Bacteria Bacteroidetes Bacteroidia
##                                               Order              Family
## f304e4fa0483742c48aebf7456178c7d Sphingobacteriales Sphingobacteriaceae
##                                             Genus Species
## f304e4fa0483742c48aebf7456178c7d Sphingobacterium    <NA>
##                                                               Taxa  ASV
## f304e4fa0483742c48aebf7456178c7d f304e4fa0483742c48aebf7456178c7d 5491
##                                                   New_Taxa
## f304e4fa0483742c48aebf7456178c7d Sphingobacterium ASV 5491
```

```
subset(taxtable_blast, New_Taxa == "Sphingobacterium ASV 1137")
```

```
##                                   Kingdom        Phylum       Class
## 327e29bbd843799e4c5b5a437169c791 Bacteria Bacteroidetes Bacteroidia
##                                               Order              Family
## 327e29bbd843799e4c5b5a437169c791 Sphingobacteriales Sphingobacteriaceae
##                                             Genus Species
## 327e29bbd843799e4c5b5a437169c791 Sphingobacterium    <NA>
##                                                               Taxa  ASV
## 327e29bbd843799e4c5b5a437169c791 327e29bbd843799e4c5b5a437169c791 1137
##                                                   New_Taxa
## 327e29bbd843799e4c5b5a437169c791 Sphingobacterium ASV 1137
```

```
subset(taxtable_blast, New_Taxa == "Lactococcus ASV 5310") # Lactococcus garvae
```

```
##                                   Kingdom     Phylum   Class           Order
## eaa57f77edaae985a5784f41ece4ce33 Bacteria Firmicutes Bacilli Lactobacillales
##                                            Family       Genus
## eaa57f77edaae985a5784f41ece4ce33 Streptococcaceae Lactococcus
##                                                               Species
## eaa57f77edaae985a5784f41ece4ce33 Lactococcus garvieae subsp. garvieae
##                                                              Taxa  ASV
## eaa57f77edaae985a5784f41ece4ce33 eaa57f77edaae985a5784f41ece4ce33 5310
##                                              New_Taxa
## eaa57f77edaae985a5784f41ece4ce33 Lactococcus ASV 5310
```

```
subset(taxtable_blast, New_Taxa == "Enterococcus ASV 4482") #   Enterococcus faecalis
```

```
##                                   Kingdom     Phylum   Class           Order
## c6cdc71a1f7376db5a843ade152ed641 Bacteria Firmicutes Bacilli Lactobacillales
##                                           Family        Genus Species
## c6cdc71a1f7376db5a843ade152ed641 Enterococcaceae Enterococcus    <NA>
##                                                              Taxa  ASV
## c6cdc71a1f7376db5a843ade152ed641 c6cdc71a1f7376db5a843ade152ed641 4482
##                                               New_Taxa
## c6cdc71a1f7376db5a843ade152ed641 Enterococcus ASV 4482
```

```
subset(taxtable_blast, New_Taxa == "Pseudomonas ASV 1915") #    Pseudomonas alcaligenes?
```

```
##                                   Kingdom         Phylum               Class
## 56f47bc063e4ca61aa04b8708df6c900 Bacteria Proteobacteria Gammaproteobacteria
##                                            Order           Family       Genus
## 56f47bc063e4ca61aa04b8708df6c900 Pseudomonadales Pseudomonadaceae Pseudomonas
##                                  Species                             Taxa  ASV
## 56f47bc063e4ca61aa04b8708df6c900    <NA> 56f47bc063e4ca61aa04b8708df6c900 1915
##                                              New_Taxa
## 56f47bc063e4ca61aa04b8708df6c900 Pseudomonas ASV 1915
```

```
subset(taxtable_blast, New_Taxa == "Pseudomonas ASV 755") #
```

```
##                                   Kingdom         Phylum               Class
## 21797842f1e1946123b1398a316b0a51 Bacteria Proteobacteria Gammaproteobacteria
##                                            Order           Family       Genus
## 21797842f1e1946123b1398a316b0a51 Pseudomonadales Pseudomonadaceae Pseudomonas
##                                  Species                             Taxa ASV
## 21797842f1e1946123b1398a316b0a51    <NA> 21797842f1e1946123b1398a316b0a51 755
##                                             New_Taxa
## 21797842f1e1946123b1398a316b0a51 Pseudomonas ASV 755
```

```
subset(taxtable_blast, New_Taxa == "Pseudomonas ASV 1566") #    Pseudomonas putida?
```

```
##                                   Kingdom         Phylum               Class
## 457d8bcebcf3531ba295c479fced9f32 Bacteria Proteobacteria Gammaproteobacteria
##                                            Order           Family       Genus
## 457d8bcebcf3531ba295c479fced9f32 Pseudomonadales Pseudomonadaceae Pseudomonas
##                                  Species                             Taxa  ASV
## 457d8bcebcf3531ba295c479fced9f32    <NA> 457d8bcebcf3531ba295c479fced9f32 1566
##                                              New_Taxa
## 457d8bcebcf3531ba295c479fced9f32 Pseudomonas ASV 1566
```

```
subset(taxtable_blast, New_Taxa == "Methylobacterium ASV 886") #    Pseudomonas putida?
```

```
##                                   Kingdom         Phylum               Class
## 282395114a7d9b8fab4455122e573b5f Bacteria Proteobacteria Alphaproteobacteria
##                                        Order           Family            Genus
## 282395114a7d9b8fab4455122e573b5f Rhizobiales Beijerinckiaceae Methylobacterium
##                                  Species                             Taxa ASV
## 282395114a7d9b8fab4455122e573b5f    <NA> 282395114a7d9b8fab4455122e573b5f 886
##                                                  New_Taxa
## 282395114a7d9b8fab4455122e573b5f Methylobacterium ASV 886
```

```
subset(taxtable_blast, New_Taxa == "Enterobacter ASV 1188") #   Enterobacter cloacae
```

```
##                                   Kingdom         Phylum               Class
## 352d1b35082a8b1188393a1ae9a5abb7 Bacteria Proteobacteria Gammaproteobacteria
##                                              Order             Family
## 352d1b35082a8b1188393a1ae9a5abb7 Enterobacteriales Enterobacteriaceae
##                                         Genus Species
## 352d1b35082a8b1188393a1ae9a5abb7 Enterobacter    <NA>
##                                                              Taxa  ASV
## 352d1b35082a8b1188393a1ae9a5abb7 352d1b35082a8b1188393a1ae9a5abb7 1188
##                                               New_Taxa
## 352d1b35082a8b1188393a1ae9a5abb7 Enterobacter ASV 1188
```

```
subset(taxtable_blast, New_Taxa == "Aeromonas ASV 2598") #
```

```
##                                   Kingdom         Phylum               Class
## 740ba29c6c30bce2c57d055dde9938ca Bacteria Proteobacteria Gammaproteobacteria
##                                          Order         Family     Genus Species
## 740ba29c6c30bce2c57d055dde9938ca Aeromonadales Aeromonadaceae Aeromonas    <NA>
##                                                              Taxa  ASV
## 740ba29c6c30bce2c57d055dde9938ca 740ba29c6c30bce2c57d055dde9938ca 2598
##                                            New_Taxa
## 740ba29c6c30bce2c57d055dde9938ca Aeromonas ASV 2598
```

```
subset(taxtable_blast, New_Taxa == "Acinetobacter ASV 1521") #
```

```
##                                   Kingdom         Phylum               Class
## 43a45981af8e469a020ce725320e06ee Bacteria Proteobacteria Gammaproteobacteria
##                                            Order        Family         Genus
## 43a45981af8e469a020ce725320e06ee Pseudomonadales Moraxellaceae Acinetobacter
##                                  Species                             Taxa  ASV
## 43a45981af8e469a020ce725320e06ee    <NA> 43a45981af8e469a020ce725320e06ee 1521
##                                                New_Taxa
## 43a45981af8e469a020ce725320e06ee Acinetobacter ASV 1521
```

```
subset(taxtable_blast, New_Taxa == "Stenotrophomonas ASV 939") #
```

```
##                                   Kingdom         Phylum               Class
## 2a912300a70f985f2df786cb0dd145dd Bacteria Proteobacteria Gammaproteobacteria
##                                             Order           Family
## 2a912300a70f985f2df786cb0dd145dd Xanthomonadales Xanthomonadaceae
##                                              Genus Species
## 2a912300a70f985f2df786cb0dd145dd Stenotrophomonas    <NA>
##                                                               Taxa ASV
## 2a912300a70f985f2df786cb0dd145dd 2a912300a70f985f2df786cb0dd145dd 939
##                                                  New_Taxa
## 2a912300a70f985f2df786cb0dd145dd Stenotrophomonas ASV 939
```

```
subset(taxtable_blast, New_Taxa == "Aeromonas ASV 2598") #
```

```
##                                   Kingdom         Phylum               Class
## 740ba29c6c30bce2c57d055dde9938ca Bacteria Proteobacteria Gammaproteobacteria
##                                         Order         Family     Genus Species
## 740ba29c6c30bce2c57d055dde9938ca Aeromonadales Aeromonadaceae Aeromonas    <NA>
##                                                               Taxa  ASV
## 740ba29c6c30bce2c57d055dde9938ca 740ba29c6c30bce2c57d055dde9938ca 2598
##                                            New_Taxa
## 740ba29c6c30bce2c57d055dde9938ca Aeromonas ASV 2598
```

```
subset(taxtable_blast, New_Taxa == "Vagococcus ASV 2048") #     Vegococcus fluvialis (LAB)
```

```
##                                   Kingdom     Phylum   Class           Order
## 5cae3996656fb088cb7f7fd82c034ee2 Bacteria Firmicutes Bacilli Lactobacillales
##                                           Family      Genus Species
## 5cae3996656fb088cb7f7fd82c034ee2 Enterococcaceae Vagococcus    <NA>
##                                                               Taxa  ASV
## 5cae3996656fb088cb7f7fd82c034ee2 5cae3996656fb088cb7f7fd82c034ee2 2048
##                                             New_Taxa
## 5cae3996656fb088cb7f7fd82c034ee2 Vagococcus ASV 2048
```

```
subset(taxtable_blast, New_Taxa == "Flavobacterium ASV 3965") #
```

```
##                                   Kingdom        Phylum       Class
## b02fbacd2d2f96c360f72afb01761adc Bacteria Bacteroidetes Bacteroidia
##                                             Order            Family
## b02fbacd2d2f96c360f72afb01761adc Flavobacteriales Flavobacteriaceae
##                                           Genus       Species
## b02fbacd2d2f96c360f72afb01761adc Flavobacterium bacterium 7B4
##                                                               Taxa  ASV
## b02fbacd2d2f96c360f72afb01761adc b02fbacd2d2f96c360f72afb01761adc 3965
##                                                 New_Taxa
## b02fbacd2d2f96c360f72afb01761adc Flavobacterium ASV 3965
```