16S rRNA Gene Sequencing Analysis (Stats and Visualization)

Jing Liu

2023-03-27

Load required packages and prepare your workspace.

```r
library(gplots) #visualization
library(ggplot2)
library(Seurat)
library(ggalluvial)
library(ggThemeAssist)
library(ggpubr)
library(RColorBrewer)
library(scales)
library(grid)
library(lattice)
library(rgl)
library(pheatmap)
library(corrplot)
library(circlize)
library(ggsci)
#library(ComplexHeatmap)

library(plyr) #Data Manipulation
library(dplyr)
library(tidyr)
library(tidyverse)
library(reshape2)

library(vegan) #Data Analysis
library(phyloseq)
library(metagenomeSeq)
library(car)
library(Rmisc)
library(psych)
#library(metagMisc)
library(RVAideMemoire)
library(DESeq2)

library(survival)
library(survminer)

# Get sig letters
library(emmeans)
library(multcomp)
library(multcompView)
library(rcompanion)

sem <- function(x) sd(x)/sqrt(length(x))

# lefse
library(data.table)
library("Maaslin2")
library(lefser)
library(microbiomeMarker)
library(knitr)
# Set working directory.
getwd()
```

```
## [1] "/Users/jingliu1/Desktop/phD/AR_Markus/Markus"
```

```
# setwd("/Users/jingliu/Desktop/Ak_data")

# remove history
rm(list=ls())
```

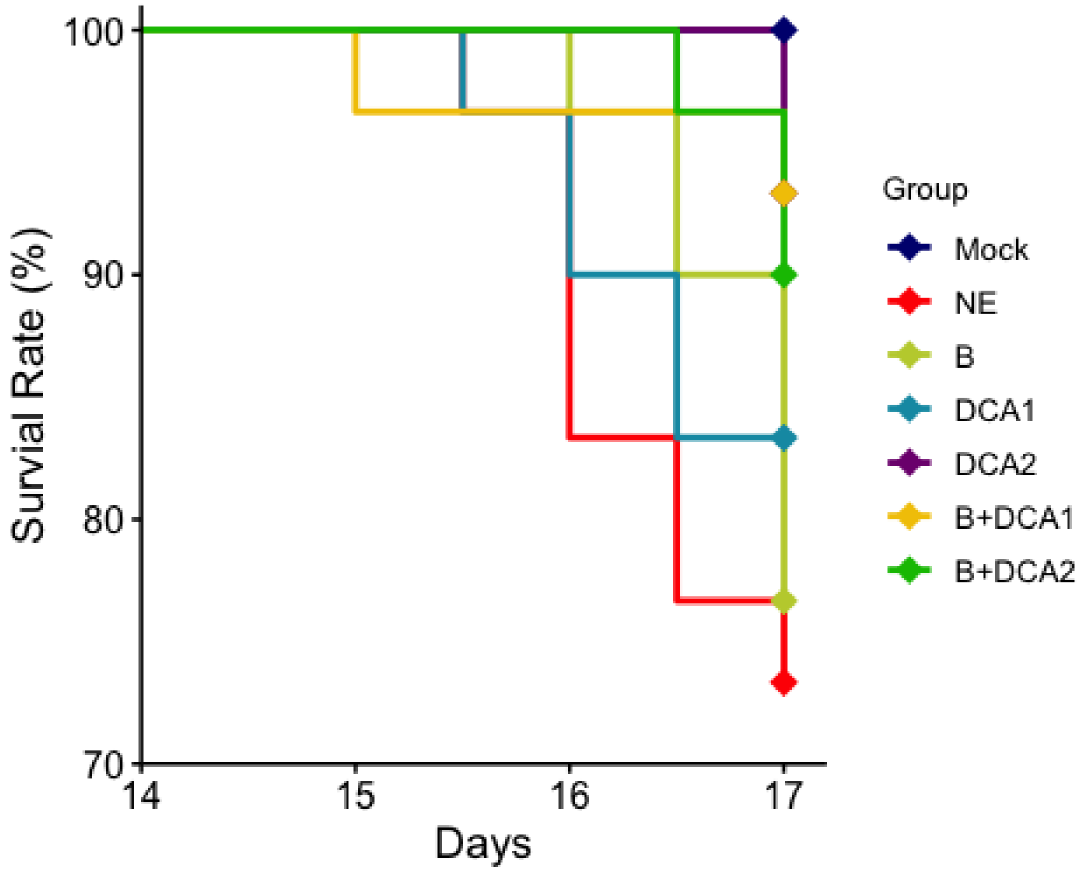
Performance Data

Survival Curve

```
Sur <- read.csv("Raw_data/Survival.csv")
Sur$Group <- factor(Sur$Group, levels = c("Mock", "NE", "B",
"DCA1","DCA2","B+DCA1", "B+DCA2"))

M.Surv <- Surv(time =Sur$Days, event = Sur$Status)
NEKM <- survfit(M.Surv ~ Group, data = Sur, type="kaplan-meier")

# log-rank test
sur.Diff <- survdiff(M.Surv ~ Group, data = Sur)
print(sur.Diff, digits = 4)
```

```
## Call:
## survdiff(formula = M.Surv ~ Group, data = Sur)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## Group=Mock     30        0    4.049    4.0488     4.928
## Group=NE       30        8    3.477    5.8823     6.984
## Group=B        30        7    3.858    2.5580     3.086
## Group=DCA1     30        5    3.651    0.4985     0.596
## Group=DCA2     30        2    4.049    1.0368     1.262
## Group=B+DCA1   30        2    3.919    0.9394     1.137
## Group=B+DCA2   30        3    3.997    0.2487     0.302
```

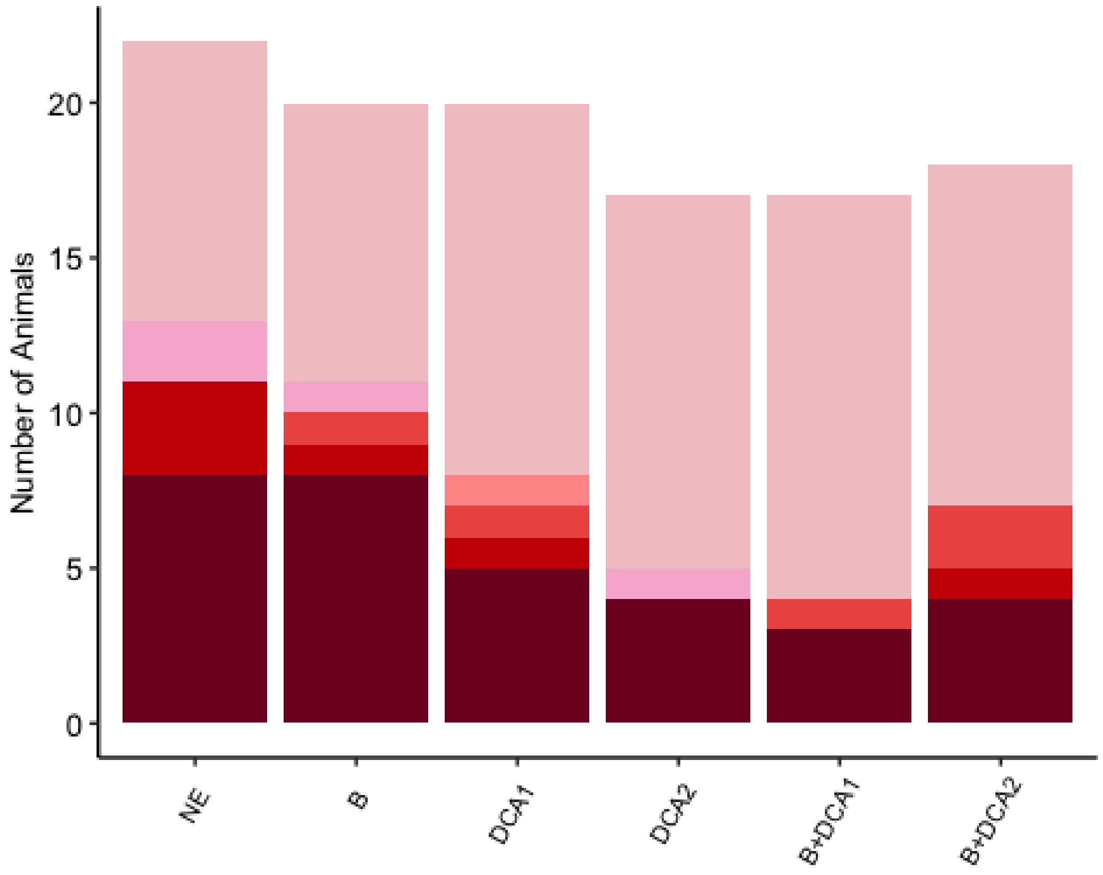
```
##
##  Chisq= 15.7  on 6 degrees of freedom, p= 0.015

# Pairwise test
restest <- pairwise_survdiff(Surv(Days,Status)~Group,Sur)

# Plotting
clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")
myplot <- ggsurvplot(NEKM, conf.int = FALSE, pval = TRUE, risk.table = FALSE,
    legend.labs = c("Mock", "NE", "B", "DCA1","DCA2","B+DCA1", "B+DCA2"),
    legend = "right",legend.title = "Group",
   # legend = "none",
    xlim = c(14, 17.2),
    break.time.by = 1,
    axes.offset = TRUE,
    surv.scale = "percent",
    censor = TRUE,censor.shape = 18, censor.size = 4.0,
    palette= clr,
    xlab = "Days", ylab = "Survial Rate (%)")

myplot$plot <- myplot$plot +
                  scale_x_continuous(expand = c(0, 0),
                                     breaks = c(14,15,16,17),
                                     labels = c("14","15","16","17")) +
  scale_y_continuous(expand = c(0, 0),limits = c(0.7,1.01),breaks =
c(0.7,0.8,0.9,1),labels = c("70","80","90","100")) +
   theme(
        panel.background = element_rect(fill = NA),
        text = element_text(size = 5, family = 'Arial'))

myplot$plot
```



```
ggsave('Figures/Survival.png',
       height = 2.5,
       width = 4.0,
       unit = 'in',
       dpi = 300)
```

Lesion Score

```
clr <- c("#F3C7CB","#F7B6D2FF","#FF9896FF","#ED5851", "#CB0002","#800026")
Lesion_S <- read.csv("Raw_data/LS.csv")
Lesion_S$Group <- factor(Lesion_S$Group, levels = c("NE", "B",
"DCA1","DCA2","B+DCA1", "B+DCA2"))
Lesion_S$LS <- as.factor(Lesion_S$LS)

#FF9896FF"#DE858A",
# Plotting
ggplot(Lesion_S, aes(x = reorder(Group, LS >='3',sum), fill = LS)) +
  geom_bar(stat='count') +
  scale_fill_manual(name=NULL,
                    values = clr) +
  #scale_y_continuous(expand = c(0, 0),limits = c(0,40))+
  labs(title=NULL,
       x=NULL,
       y="Number of Animals") +
   theme_classic() +
   guides(fill=guide_legend(title="LS")) +
   theme(axis.title = element_text(size = 10,family = 'Arial', colour = 
"black"),
        axis.text = element_text(size = 10,family = 'Arial', color = 
"black"),
        axis.text.x = element_text(size = 8, vjust = 0.8, hjust = 0.8, angle 
= 60),
        legend.position = "none")

## Warning in Ops.factor(LS, "3"): '>=' not meaningful for factors
```



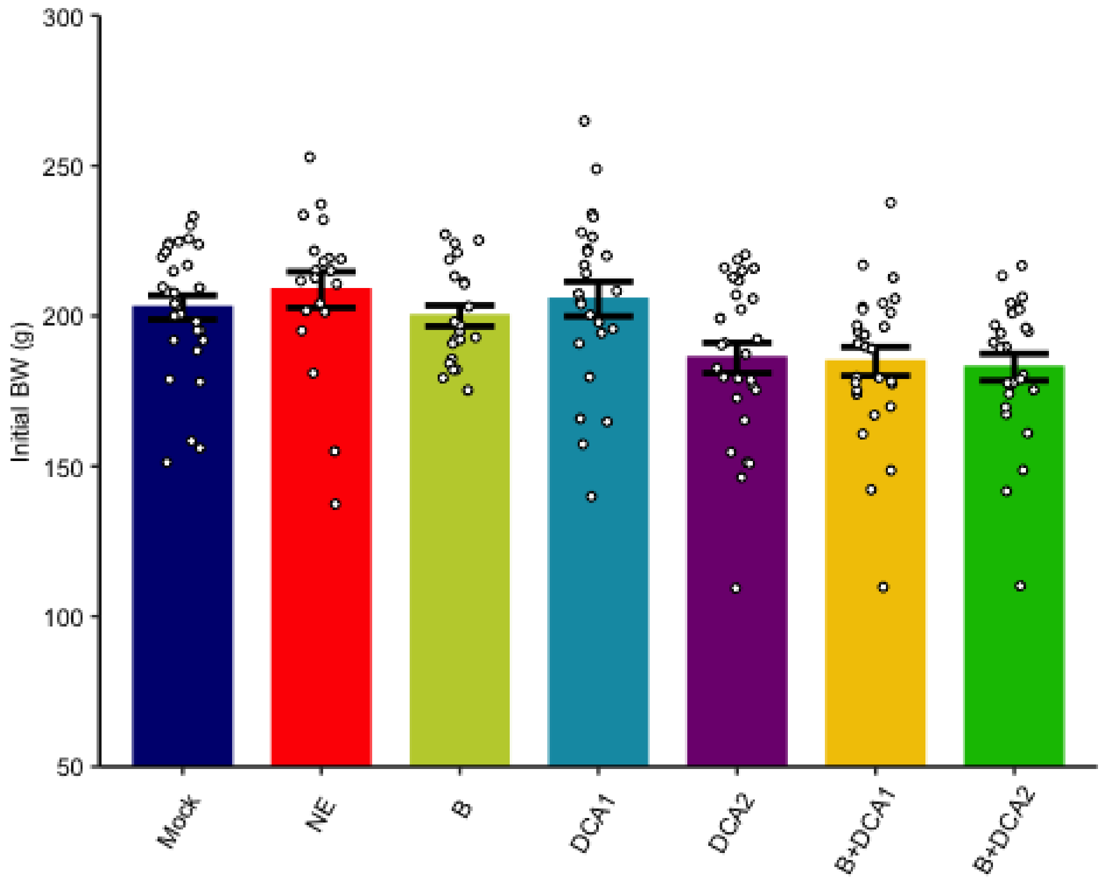
```
ggsave('Figures/Lesion_Score.png',
       height = 2.2,
       width = 2.0,
       unit = 'in',
       dpi = 300)
```

Body Weight

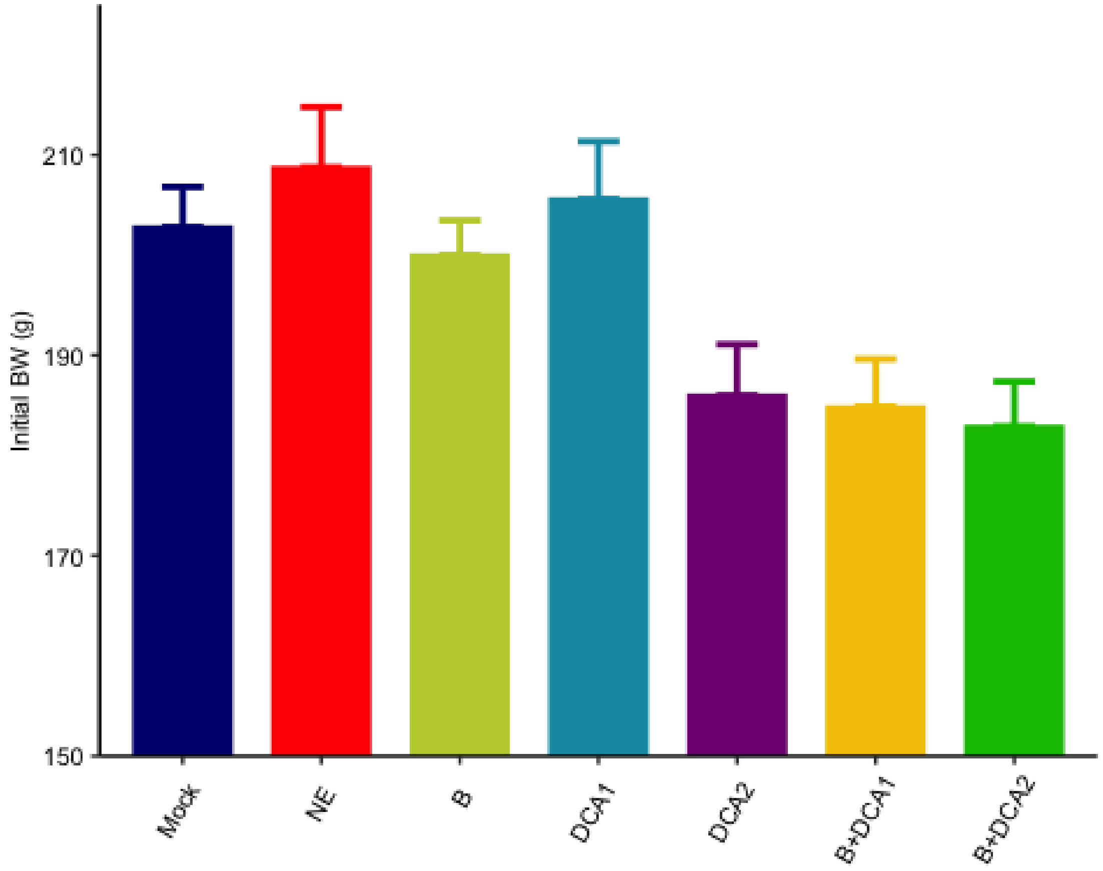
```
BW <- read.csv("Raw_data/BW.csv")
BW$Group <- factor(BW$Group, levels = c("Mock", "NE", "B",
"DCA1","DCA2","B+DCA1", "B+DCA2"))

# Calculate significant differences among groups.
kruskal.test(BW10 ~ Group,BW) # p-value < 0.001**

##
##  Kruskal-Wallis rank sum test
##
## data:  BW10 by Group
## Kruskal-Wallis chi-squared = 31.441, df = 6, p-value = 2.088e-05

pairwise.wilcox.test(BW$BW10, BW$Group,p.adjust.method = "none")
##  Pairwise comparisons using Wilcoxon rank sum test with continuity
correction
##
## data:  BW$BW10 and BW$Group
##
##        Mock    NE      B       DCA1    DCA2    B+DCA1
## NE     0.27154 -       -       -       -       -
## B      0.33982 0.06773 -       -       -       -
## DCA1   0.66635 0.61525 0.23243 -       -       -
## DCA2   0.00872 0.00329 0.05661 0.01082 -       -
## B+DCA1 0.00280 0.00060 0.02335 0.00448 0.67597 -
## B+DCA2 0.00094 0.00020 0.01066 0.00185 0.50592 0.82665
##
## P value adjustment method: none

# Data Summary
# After imported the data, we’ll plot the mean value of BWL in each group.
The standard deviation is used to draw the error bars on the graph.To achieve
this, we first have to create a data set containing the mean and standard
errors by group. We can use the aggregate function as shown below:
BW_summary.10 <- aggregate(BW10 ~ Group, BW,       # Create summary data
                          function(x) c(BW10 = mean(x),
                                        SEM = sd(x) / sqrt(length(x))))
BW_summary.10 <- data.frame(Group = BW_summary.10[ , 1], BW_summary.10$BW10)
```

Plotting-Barplot

```
clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")
update_geom_defaults("point", list(size = 1.0))

ggplot(BW, aes(x=Group, y=BW10,fill = Group,shape =Group )) +
  geom_bar(stat = "identity", data = BW_summary.10,
           fill = clr, width=0.70, color = clr) +
  geom_errorbar(
    aes(ymin = BW10-SEM, ymax = BW10+SEM),
    data = BW_summary.10, width = 0.5, linewidth = 1) +
    geom_jitter(position = position_jitter(0.15), color = "black") +
    scale_fill_manual(name=NULL,
                    values=c("white", "white", "white", "white",
"white","white","white")) +
  scale_shape_manual(values=c(21,21,21,21,21,21,21)) +
  scale_size_manual(values=c(0.1,0.1,0.1,0.1,0.1,0.1,0.1)) +
 coord_cartesian(ylim = c(50, 300))+scale_y_continuous(expand = c(0,0))+
  labs(title=NULL,
       x=NULL,
       y="Initial BW (g)") +
 theme_classic()+
  theme(axis.title = element_text(size = 8,family = 'Arial', colour =
"black"),
        axis.text = element_text(size = 8,family = 'Arial', color = "black"),
        legend.position = "none",
        axis.text.x = element_text(size = 8, hjust = 0.8,vjust = 0.8, angle =
60))
```



```
ggplot(BW_summary.10, aes(x=Group, y=BW10, fill=Group)) +
  geom_bar(stat="identity", colour= clr, position=position_dodge(0.6),width =
0.7) +
  geom_errorbar(aes(ymin=BW10, ymax=BW10+SEM),size = 1, width=.3,
                position=position_dodge(.6),colour = clr) +
   scale_fill_manual(name=NULL,
                    values=clr)+
  scale_y_continuous(expand = c(0, 0))+
  coord_cartesian(ylim = c(150, 225))+scale_y_continuous(expand = c(0,0))+
  labs(title=NULL,
       x=NULL,
       y="Initial BW (g)") +
   theme_classic()+
   theme(axis.title = element_text(size = 8,family = 'Arial', colour =
"black"),
        axis.text = element_text(size = 8,family = 'Arial', color = "black"),
        legend.position = "none",
        axis.text.x = element_text(size = 8, hjust = 0.8,vjust = 0.8, angle =
60))
```



```
ggsave('Figures/Phenotype/BW_D10_2.png',
       height = 2.2,
       width = 2.2,
       unit = 'in',
       dpi = 300)
```

16S Downstream Analysis

Bacterial diversity

The taxonomy, feature, and metadata tables were loaded into R and placed in a phyloseq object.

```r
#Feature Table
otumat <- read.csv("QIIME2/feature_table/feature_table.txt", sep="")
rownames(otumat) <- otumat[,1]
otumat <- otumat[,-c(1)]
#Sample Data
sampledata <- read.csv("Raw_data/sample_info.txt", sep="", header=T)
sampledata$Group<-as.factor(sampledata$Group)
sampledata$Site<-as.factor(sampledata$Site)

#Taxonomy table
taxmat <- read.csv(file = 'Processed_data/taxonomy_diversity.csv', header =
T, sep = ",", stringsAsFactors = F) # delete percent coloums in excel

rownames(taxmat) <- taxmat[,1]
taxmat <- taxmat[,-1]
stopifnot(rownames(otumat) == rownames(taxmat))
#Convert taxonomy and OTU table to matrix
taxmat <- as.matrix(taxmat)
TAX <- tax_table(taxmat)
otumat <- data.matrix(otumat)
OTU <- otu_table(otumat, taxa_are_rows = T)
sample.data <- sample_data(sampledata)
rownames(sample.data) <- sample.data$SampleID
#colnames(otumat)
physeq <- phyloseq(TAX, OTU, sample.data)

# Import tree file
mytree_rooted <- read_tree("QIIME2/rooted_tree/tree.nwk",errorIfNULL=FALSE)
physeq <- merge_phyloseq(physeq,mytree_rooted)

colnames(tax_table(physeq)) <- c("Kingdom", "Phylum", "Class",
                                     "Order", "Family", "Genus")
sample_data(physeq)$Group <- factor(sample_data(physeq)$Group, levels =
c('Mock', 'NE',"NaB","DCA1",'DCA2','NaB+DCA1','NaB+DCA2'))
sample_data(physeq)$Site <- factor(sample_data(physeq)$Site, levels =
c('Ileum', 'Cecum'))

# Subset sample based on site and trial
physeq_Ile <- subset_samples(physeq, Site == "Ileum")

# Remove ASVs present less than 5% of the sample
physeq <- physeq_Ile %>%
  prune_taxa(rowSums(otu_table(physeq_Ile) != 0) >= 4, .) # 64*0.05
```

Data normalization

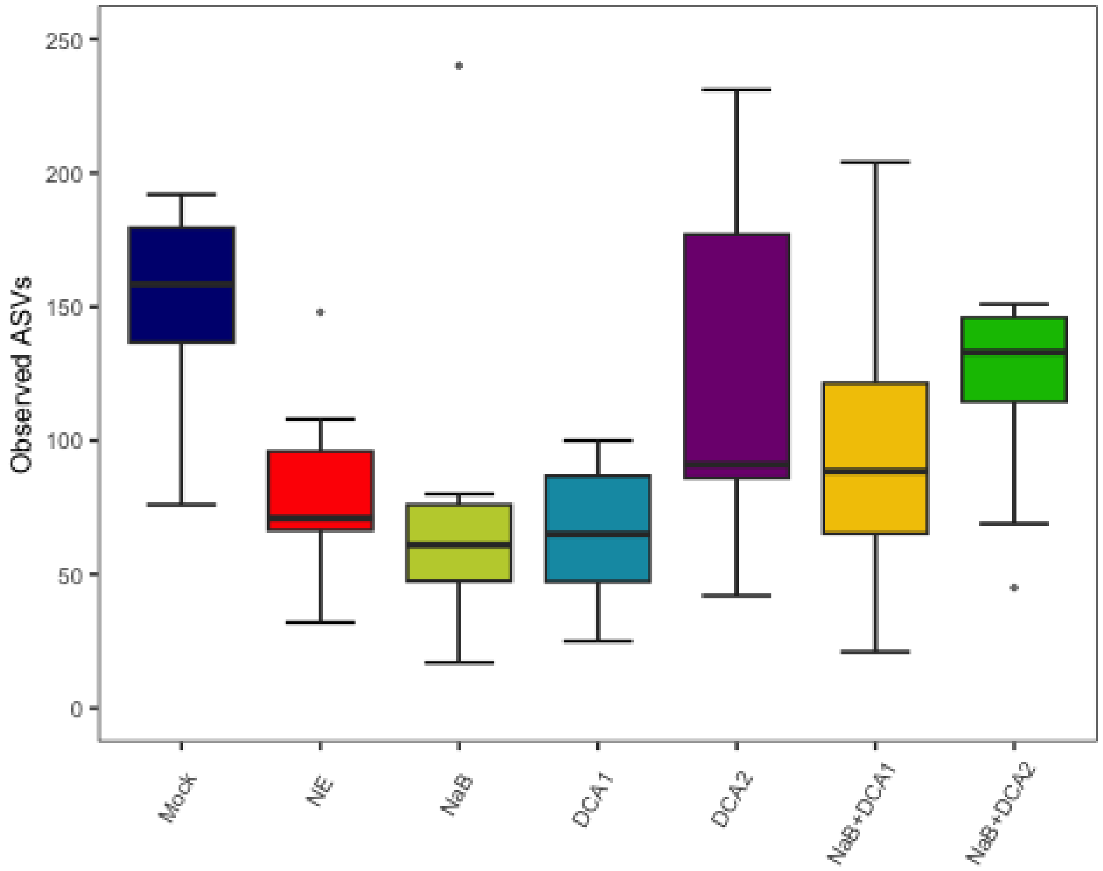
```
#Normalize data using CSS method
# Convert phyloseq object to metageomeSeq object
metaSeq <- phyloseq_to_metagenomeSeq(physeq)
# Calculate pth quantile
p <- cumNormStatFast(metaSeq)
# CSS normalization
meta_sub <- cumNorm(metaSeq, p = p)
# Pull OTU table out of metagenomeSeq
otu_data <- as.data.frame(MRcounts(meta_sub, norm = T, log = F))
bacteria <- physeq
otu_table(bacteria) <- otu_table(otu_data, taxa_are_rows = T)
bacteria <- subset_samples(bacteria, SampleID != "IC.90" & SampleID !=
"ID15.701" & SampleID != "ID15.585" & SampleID != "ID75.574")
bacteria <- bacteria %>%
prune_taxa(taxa_sums(.) > 0, .)
```

Alpha Diversity

```
data <- as.data.frame(otu_table(bacteria))
data[,1:60] <- sapply(data[,1:60], as.integer)
otu_table(bacteria) <- otu_table(data, taxa_are_rows = T)

alpha <- estimate_richness(bacteria)
alpha$Sample <- rownames(alpha)

# Add Metadata
s <- data.frame(sample_data(bacteria))
s$SampleID  <- as.character(s$SampleID)
alpha$Sample <- as.character(alpha$Sample)
meta_alpha <- inner_join(alpha, s, by =c('Sample'='SampleID'))
meta_alpha$Group<- as.factor(meta_alpha$Group)
```

Observed Features

```
clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")
#Statistical analyzing
kruskal <-kruskal.test(Observed ~ Group, meta_alpha)
p <- pairwise.wilcox.test(meta_alpha$Observed, meta_alpha$Group,
p.adjust.method = 'none')

# Plot Observed OTUs
ggplot(meta_alpha, aes(x = Group, y = Observed,fill=Group)) +
  stat_boxplot(geom = 'errorbar', width =0.5) +
  geom_boxplot(fill=clr,outlier.alpha = 0.6, outlier.size = 0.5) +
  scale_y_continuous(limits=c(0,250))+
  ylab('Observed ASVs') +
  xlab(NULL) +
  theme_bw()+
   theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8),
        text = element_text(size = 9, family = 'Arial'))
```



```
ggsave('Figures/Ileum/Diversity/Sobs.png',
       height = 2.4,
       width = 1.9,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Evenness

```
H <- meta_alpha$Shannon
S1 <- meta_alpha$Observed
S <- log(S1)
evenness <- H/S
meta_alpha$Evenness <- evenness

#Statistical analyzing

kruskal.test(Evenness ~ Group, meta_alpha)
```

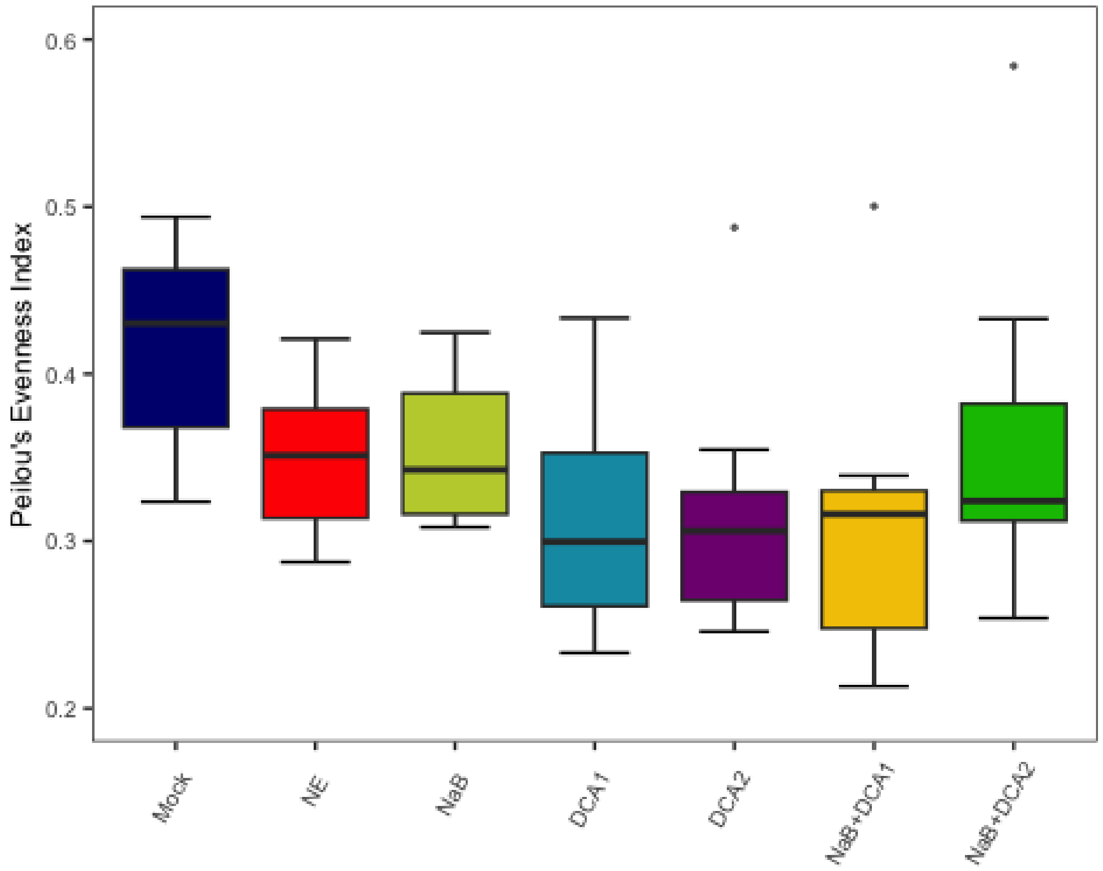
```
##
##  Kruskal-Wallis rank sum test
##
## data:  Evenness by Group
## Kruskal-Wallis chi-squared = 12.82, df = 6, p-value = 0.04598
```

```
pairwise.wilcox.test(meta_alpha$Evenness,meta_alpha$Group,p.adjust.method =
"none",paired = F)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  meta_alpha$Evenness and meta_alpha$Group
##
##          Mock   NE     NaB    DCA1   DCA2   NaB+DCA1
## NE       0.0360 -      -      -      -      -
## NaB      0.0311 0.8665 -      -      -      -
## DCA1     0.0111 0.2786 0.2810 -      -      -
## DCA2     0.0056 0.0927 0.0907 0.8148 -      -
## NaB+DCA1 0.1139 0.5737 0.5358 0.8785 0.4807 -
## NaB+DCA2 0.0381 0.8404 0.8601 0.3511 0.1119 0.6574
##
## P value adjustment method: none
```

```
# Plot Evenness
ggplot(meta_alpha, aes(x = Group, y = Evenness,fill=Group)) +
  stat_boxplot(geom = 'errorbar', width =0.5) +
  geom_boxplot(fill=clr,outlier.alpha = 0.6, outlier.size = 0.5) +
  scale_y_continuous(limits=c(0.2,0.60))+
  ylab("Peilou's Evenness Index") +
  xlab(NULL) +
  theme_bw()+
   theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8),
        text = element_text(size = 9, family = 'Arial'))
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
## Removed 2 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



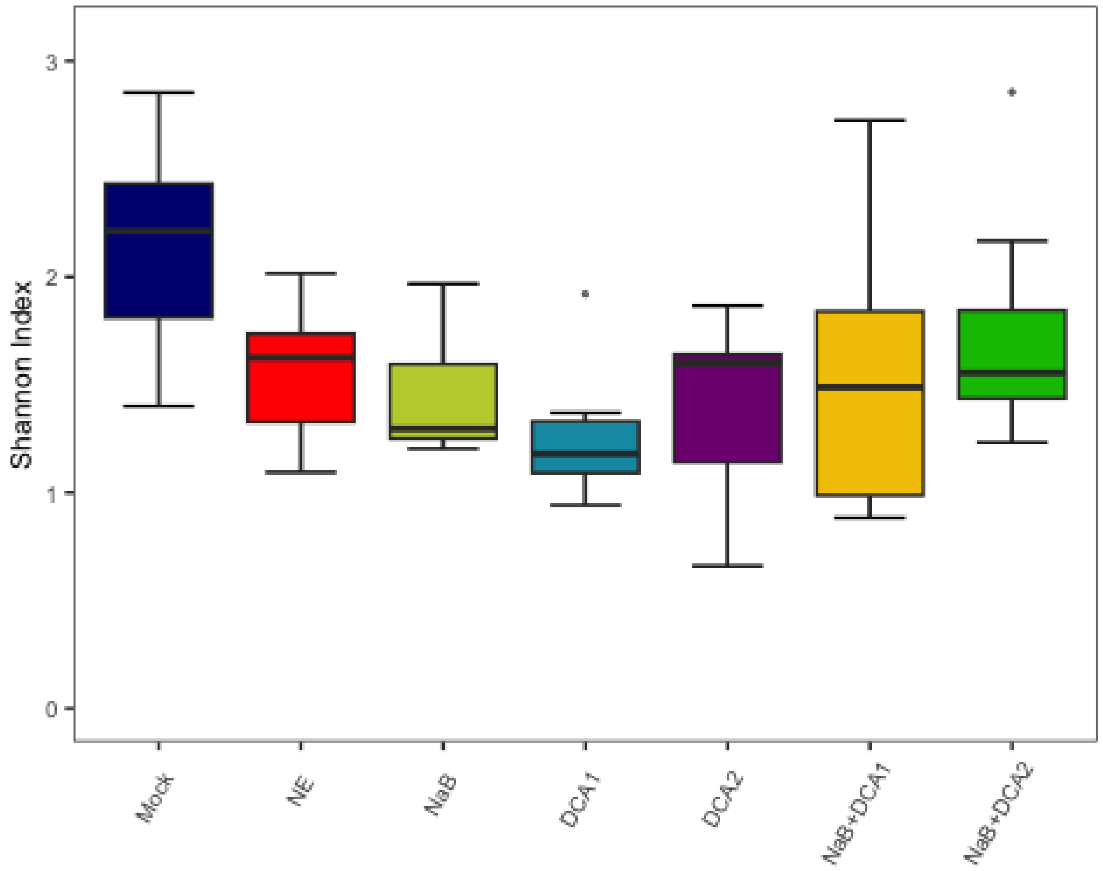
```
ggsave('Figures/Ileum/Diversity/Even.png',
       height = 2.4,
       width = 1.9,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Shannon Features

```
#Statistical analyzing
kruskal.test(Shannon ~ Group, meta_alpha)

##
##  Kruskal-Wallis rank sum test
##
## data:  Shannon by Group
## Kruskal-Wallis chi-squared = 16.789, df = 6, p-value = 0.01009

pairwise.wilcox.test(meta_alpha$Shannon, meta_alpha$Group, p.adjust.method =
'none')

##
##  Pairwise comparisons using Wilcoxon rank sum exact test
```

```
##
## data:  meta_alpha$Shannon and meta_alpha$Group
##
##          Mock    NE      NaB     DCA1    DCA2    NaB+DCA1
## NE       0.01111 -       -       -       -       -
## NaB      0.00524 0.46340 -       -       -       -
## DCA1     0.00058 0.08298 0.18928 -       -       -
## DCA2     0.00276 0.42345 0.91818 0.42345 -       -
## NaB+DCA1 0.04640 0.79845 0.95509 0.64538 0.67297 -
## NaB+DCA2 0.03103 0.90389 0.24629 0.00910 0.45610 0.54476
##
## P value adjustment method: none
```

```
# Plot Shannon OTUs
ggplot(meta_alpha, aes(x = Group, y = Shannon,fill=Group)) +
  stat_boxplot(geom = 'errorbar', width =0.5) +
  geom_boxplot(fill=clr,outlier.alpha = 0.6, outlier.size = 0.5) +
  ylab('Shannon Index') +
  scale_y_continuous(limits=c(0,3.1))+
  xlab(NULL) +
  theme_bw()+
   theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8),
        text = element_text(size = 9, family = 'Arial'))
```

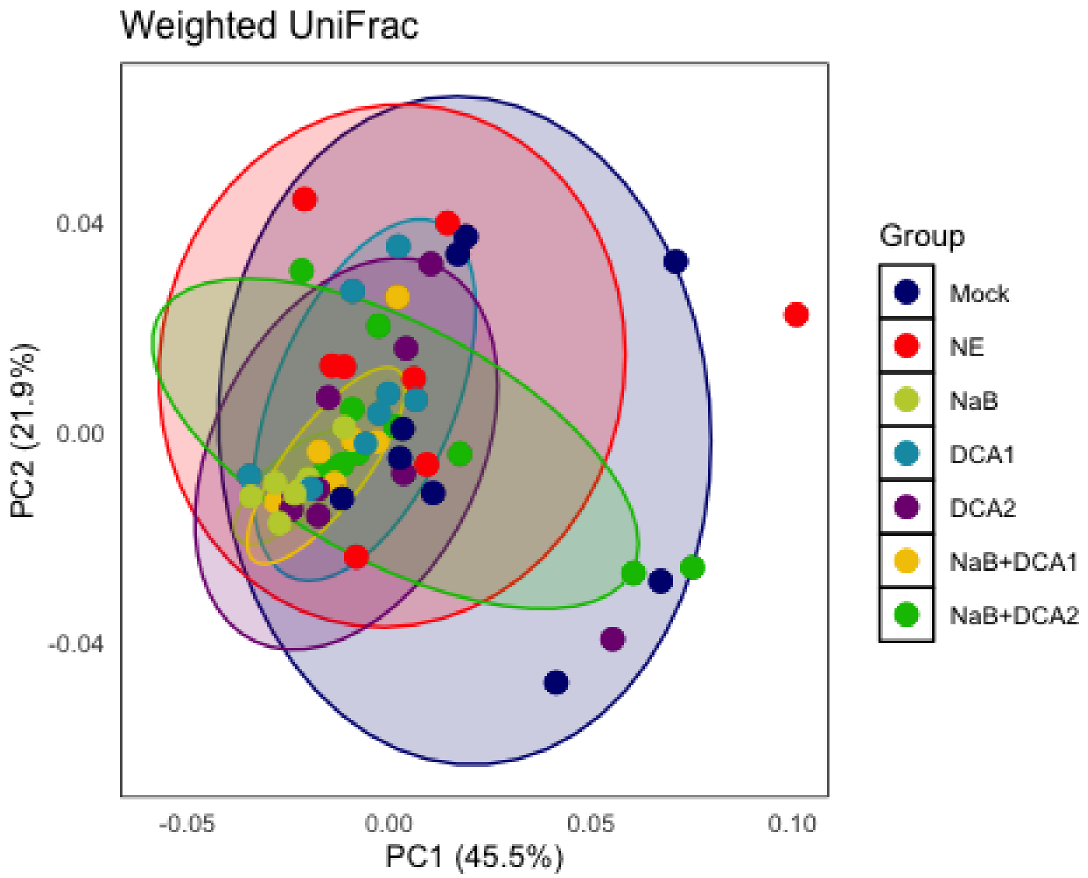


```
ggsave('Figures/Ileum/Diversity/Shannon.png',
       height = 2.4,
       width = 1.8,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Weighted UniFrac

```
clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")
set.seed(42)

weighted <- ordinate(
  physeq = bacteria,
  method = "PCoA",
  distance = "unifrac",weight=TRUE
)

#plot_scree(unweighted, "Scree plot, unifrac/PCoA")


weighted_table <- phyloseq::distance(bacteria, 'wunifrac')
```

```
sampledf <- data.frame(sample_data(bacteria))

adonis2(weighted_table ~ Group, data = sampledf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = weighted_table ~ Group, data = sampledf)
##          Df SumOfSqs      R2     F Pr(>F)
## Group     6 0.024282 0.23037 2.644  0.001 ***
## Residual 53 0.081122 0.76963
## Total    59 0.105404 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

perm <- pairwise.perm.manova(weighted_table, sampledf$Group, nperm =
999,p.method = 'none', R2=TRUE)


# Plot
update_geom_defaults("point", list(size = 3))

plot_ordination(
  physeq = bacteria,
  ordination = weighted,
  color = "Group",
  #label = "SampleID",
  title = "Weighted UniFrac"
) +
  stat_ellipse(geom = "polygon",level = 0.9, alpha=0.2, show.legend = FALSE,
aes(fill=Group)) +
  geom_point() +
  scale_color_manual(values = clr) +
  scale_fill_manual(values = clr) +
  labs(x= 'PC1 (45.5%)', y = 'PC2 (21.9%)') +
  theme(panel.grid.major = element_blank(), panel.grid.minor =
element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        strip.background = element_rect(color = "black",
                                        linewidth = 0.5),
        axis.ticks = element_blank(), text = element_text(size = 10, family =
'Arial'))
```

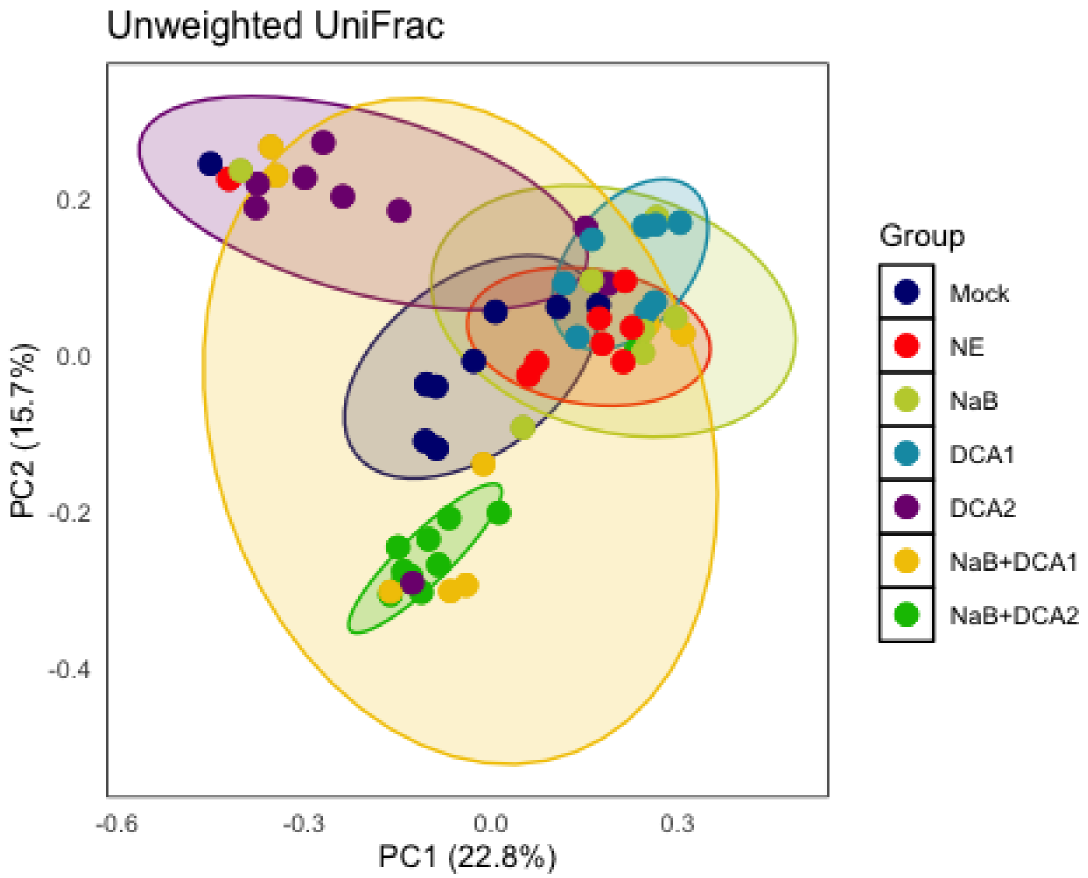


```
ggsave('Figures/Ileum/Diversity/Weighted.png',
       height = 3.0,
       width = 3.5,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

UnWeighted UniFrac

```
set.seed(42)
unweighted <- ordinate(
  physeq = bacteria,
  method = "PCoA",
  distance = "unifrac",weight=FALSE
)

#plot_scree(unweighted, "Scree plot, unifrac/PCoA")


unweighted_table <- phyloseq::distance(bacteria, 'uunifrac')
sampledf <- data.frame(sample_data(bacteria))
```

```
adonis2(unweighted_table ~ Group, data = sampledf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unweighted_table ~ Group, data = sampledf)
##          Df SumOfSqs      R2      F Pr(>F)
## Group     6   3.5404 0.29084 3.6228  0.001 ***
## Residual 53   8.6324 0.70916
## Total    59  12.1728 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

perm <- pairwise.perm.manova(unweighted_table, sampledf$Group, nperm =
999,p.method = 'none', R2=TRUE)


# Plot
update_geom_defaults("point", list(size = 3))

plot_ordination(
  physeq = bacteria,
  ordination = unweighted,
  color = "Group",
  #label = "SampleID",
  title = "Unweighted UniFrac"
) +
  stat_ellipse(geom = "polygon",level = 0.8, alpha=0.2, show.legend = FALSE,
aes(fill=Group)) +
  geom_point() +
  scale_color_manual(values = clr) +
  scale_fill_manual(values = clr) +
  labs(x= 'PC1 (22.8%)', y = 'PC2 (15.7%)') +
  #scale_y_continuous(limits = c(-0.16,0.16)) +
  #scale_x_continuous(limits = c(-0.18, 0.05))+
  theme(panel.grid.major = element_blank(), panel.grid.minor =
element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        strip.background = element_rect(color = "black",
                                        linewidth = 0.5),
        axis.ticks = element_blank(), text = element_text(size = 10, family =
'Arial'))
```



```
ggsave('Figures/Ileum/Diversity/Unweighted.png',
       height = 3.0,
       width = 3.5,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Data Import-Composition analysis

The taxonomy, feature, and metadata tables were loaded into R and placed in a phyloseq object.

```
# remove history commands
rm(list=ls())

#Feature Table
otumat <- read.csv("QIIME2/feature_table/feature_table_tax.txt", sep="")
rownames(otumat) <- otumat[,1]
otumat <- otumat[,-c(1)]
#Sample Data
sampledata <- read.csv("Raw_data/sample_info.txt", sep="", header=T)
sampledata$Group<-as.factor(sampledata$Group)
```

```
sampledata$Site<-as.factor(sampledata$Site)

#Taxonomy table
taxmat <- read.csv(file = 'Processed_data/taxonomy.csv', header = T, sep =
",", stringsAsFactors = F) # delete percent coloums in excel

rownames(taxmat) <- taxmat[,1]
taxmat <- taxmat[,-1]
stopifnot(rownames(otumat) == rownames(taxmat))
#Convert taxonomy and OTU table to matrix
taxmat <- as.matrix(taxmat)
TAX <- tax_table(taxmat)
otumat <- data.matrix(otumat)
OTU <- otu_table(otumat, taxa_are_rows = T)
sample.data <- sample_data(sampledata)
rownames(sample.data) <- sample.data$SampleID
#colnames(otumat)
physeq <- phyloseq(TAX, OTU, sample.data)

colnames(tax_table(physeq)) <- c("Kingdom", "Phylum", "Class",
                                     "Order", "Family", "Genus","Species")
sample_data(physeq)$Group <- factor(sample_data(physeq)$Group, levels =
c('Mock', 'NE',"NaB","DCA1",'DCA2','NaB+DCA1','NaB+DCA2'))
sample_data(physeq)$Site <- factor(sample_data(physeq)$Site, levels =
c('Ileum', 'Cecum'))

# Subset sample based on site and trial
physeq_Ile <- subset_samples(physeq, Site == "Ileum")
physeq_Ile<-subset_taxa(physeq_Ile, Kingdom != "Eukaryota")
```

Remove OTUs that sum to 0 and a relative abundance less than 0.01%

```
# Remove ASVs present less than 5% of the sample
physeq <- physeq_Ile %>%
  prune_taxa(rowSums(otu_table(physeq_Ile) != 0) >= 4, .) # 64*0.05
```

#CSS normalization

```
#Normalize data using CSS method
# Convert phyloseq object to metageomeSeq object
metaSeq <- phyloseq_to_metagenomeSeq(physeq)
# Calculate pth quantile
p <- cumNormStatFast(metaSeq)
```

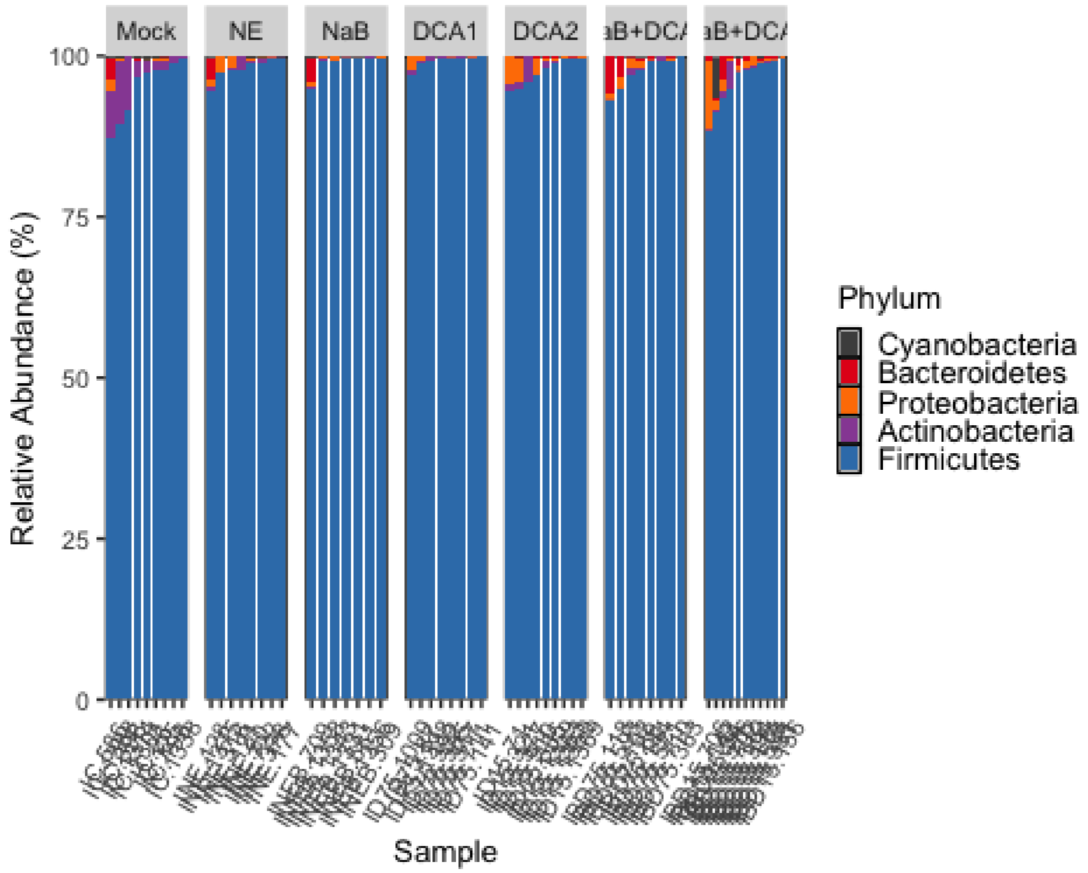
```
## Default value being used.
```

```
# CSS normalization
meta_sub <- cumNorm(metaSeq, p = p)
# Pull OTU table out of metagenomeSeq
otu_data <- as.data.frame(MRcounts(meta_sub, norm = T, log = F))
```

```
bacteria <- physeq
otu_table(bacteria) <- otu_table(otu_data, taxa_are_rows = T)
bacteria <- subset_samples(bacteria, SampleID != "IC.90" & SampleID !=
"ID15.701" & SampleID != "ID15.585" & SampleID != "ID75.574")
bacteria <- bacteria %>%
prune_taxa(taxa_sums(.) > 0, .)
```

Individual_phylum_barplot

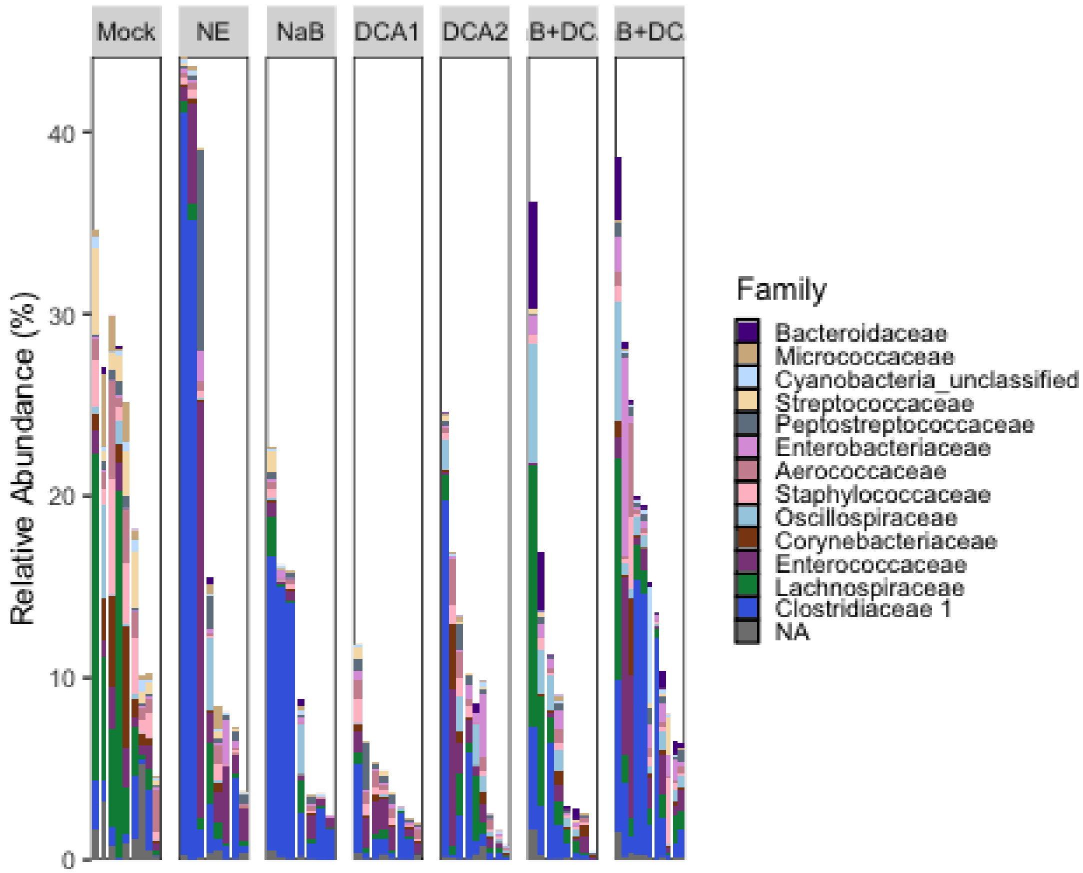
```
phylum_relabund <- bacteria %>%
  tax_glom(taxrank = "Phylum") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

phylum_relabund$Phylum <- gsub(phylum_relabund$Phylum, pattern =
'Bacteria_unclassified', replacement = 'Others')
phylum_relabund$Phylum <- reorder(phylum_relabund$Phylum,
phylum_relabund$Abundance)
phylum_relabund$Percent <- phylum_relabund$Abundance * 100
data <- subset(phylum_relabund, Phylum == 'Firmicutes')
data$Sample <- reorder(data$Sample, data$Abundance)
phylum_list <- levels(phylum_relabund$Phylum)
phylum_top5 <-phylum_list[5:9]

#phylum_top5 <- phylum_list[-3]
phylum_new <- phylum_relabund[which(phylum_relabund$Phylum %in%
phylum_top5),]
phylum_new$Phylum <- as.factor(phylum_new$Phylum)

phylum_new$Sample <- factor(phylum_new$Sample, levels = rev(data$Sample))
phylum_new$Phylum <- factor(phylum_new$Phylum, order = TRUE,levels =
c("Cyanobacteria", "Bacteroidetes", "Proteobacteria", "Actinobacteria",
"Firmicutes"))

my_colors <- brewer.pal(5, 'Set1')
my_colors <- c("#377EB8","#984EA3", "#FF7F00","#E41A1C", "grey30")

ggplot(phylum_new, aes(x = Sample, y = Percent, fill = Phylum)) +
  geom_bar(stat = 'identity') +
  facet_grid(.~Group,scales = "free") +
  scale_fill_manual(values = rev(my_colors)) +
  scale_y_continuous(expand=c(0,0))+  #(y axis start from 0)
  theme(panel.grid = element_blank(), panel.background = element_rect(color =
'black', fill = 'transparent')) +
  theme(legend.text = element_text(size=10,family = 'Arial'))+
  #Remove x axis title
  #theme(axis.title.x = element_blank()) +
  #theme(axis.text.x = element_blank(),axis.ticks.x = element_blank()) +
  theme(text = element_text(size = 10, family = 'Arial'),axis.text.x =
element_text(size = 8, hjust = 0.8,vjust = 0.8, angle = 60)) +
  #theme(legend.position = "bottome", legend.box = "vertical") +
  guides(fill = guide_legend(reverse = FALSE, keywidth = 0.5, keyheight =
0.5, ncol = 1)) +ylab("Relative Abundance (%)")
```



Individual_family_barplot

```
family_relabund <- bacteria %>%
  tax_glom(taxrank = "Family") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

family_relabund$Family <- gsub(family_relabund$Family, pattern =
'Bacteria_unclassified', replacement = 'Others')
family_relabund$Family <- gsub(family_relabund$Family, pattern =
'Firmicutes_unclassified', replacement = 'Others')
family_relabund$Family <- reorder(family_relabund$Family,
family_relabund$Abundance)
family_relabund$Percent <- family_relabund$Abundance * 100
data <- subset(family_relabund, Family == 'Lactobacillaceae')
```

```r
data$Sample <- reorder(data$Sample, data$Abundance)
family_list <- levels(family_relabund$Family)

family_top15 <-family_list[56:70]

family_new <- family_relabund[which(family_relabund$Family %in%
family_top15),]
family_new$Family <- as.factor(family_new$Family)

family_new$Sample <- factor(family_new$Sample, levels = rev(data$Sample))

family_new$Family <- factor(family_new$Family, order = TRUE,levels =
c("Bacteroidaceae","Micrococcaceae","Erysipelotrichaceae","Cyanobacteria_uncl
assified","Streptococcaceae","Peptostreptococcaceae","Enterobacteriaceae","Ae
rococcaceae","Staphylococcaceae","Oscillospiraceae","Corynebacteriaceae",
"Enterococcaceae", "Lachnospiraceae", "Clostridiaceae 1",
"Lactobacillaceae"))

family_color <-
c("royalblue","chocolate","springgreen4","orchid4","chocolate4","#beaed4","#a
6cee3", "pink", "pink3", "plum","slategray", "wheat", "slategray1", "tan",
"purple4")

family_color <-
c("chocolate","#beaed4","royalblue","springgreen4","orchid4","chocolate4","#a
6cee3", "pink", "pink3", "plum","slategray", "wheat", "slategray1", "tan",
"purple4")


ggplot(family_new, aes(x = Sample, y = Percent, fill = Family)) +
  geom_bar(stat = 'identity') +
  facet_grid(.~Group, scales = 'free') +
  scale_fill_manual(values = rev(family_color)) +
  scale_y_continuous(expand=c(0,0))+  #(y axis start from 0)
  theme(panel.grid = element_blank(), panel.background = element_rect(color =
'black', fill = 'transparent')) +
  theme(legend.text = element_text(size=8,family = 'Arial'))+
  #Remove x axis title
  theme(axis.title.x = element_blank(),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 90,
vjust = 0.8, hjust= 0.8)) +
  theme(axis.text.x = element_blank(),axis.ticks.x = element_blank()) +
  theme(text = element_text(size = 10, family = 'Arial')) +
  #theme(legend.position = "bottome", legend.box = "vertical") +
  guides(fill = guide_legend(reverse = FALSE, keywidth = 0.5, keyheight =
0.5, ncol = 1)) +ylab("Relative Abundance (%)")
```

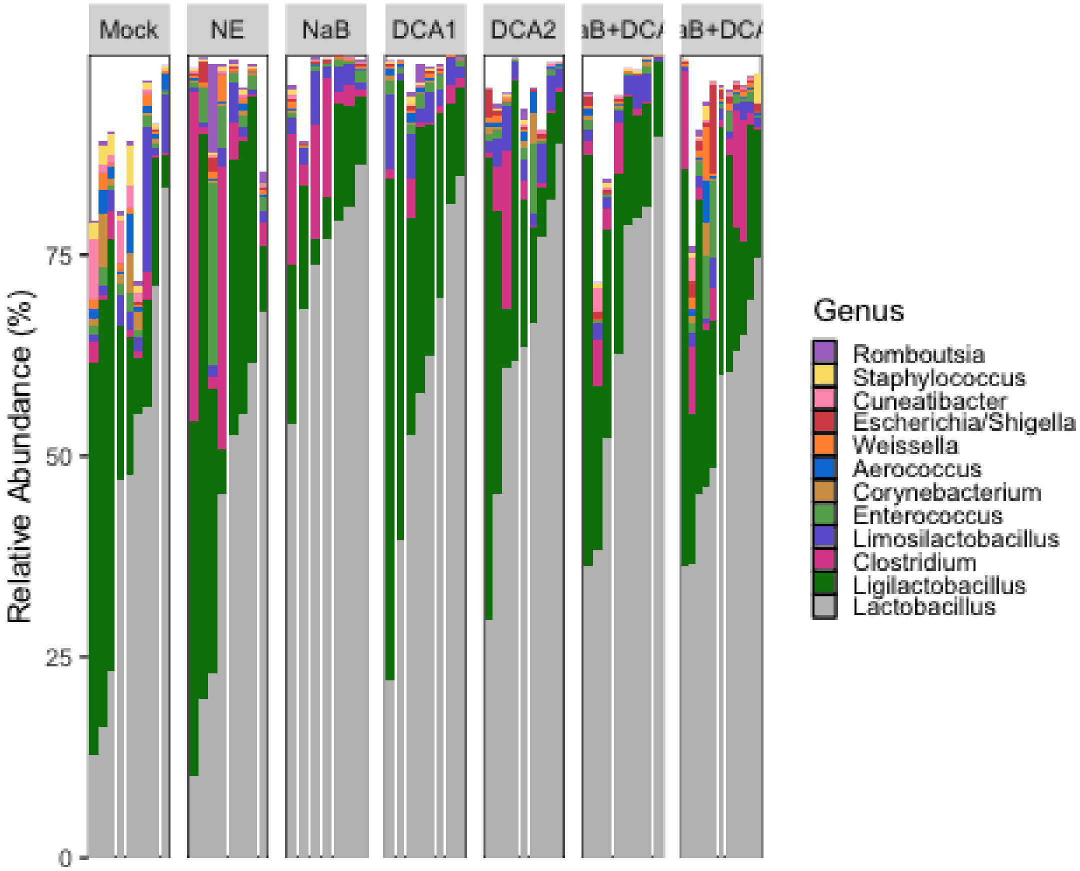


```
ggsave('Figures/Ileum/Abundance/Family_individual.png',
       height = 2.0,
       width = 7.0,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Genus_Barplot_Individual

```
genus_relabund <- bacteria %>%
  tax_glom(taxrank = "Genus") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

genus_relabund$Genus <- gsub(genus_relabund$Genus, pattern =
'Firmicutes_unclassified', replacement = 'Others')
genus_relabund$Genus <- reorder(genus_relabund$Genus,
genus_relabund$Abundance)
genus_relabund$Sample<-reorder(genus_relabund$Sample,
genus_relabund$Abundance)
```

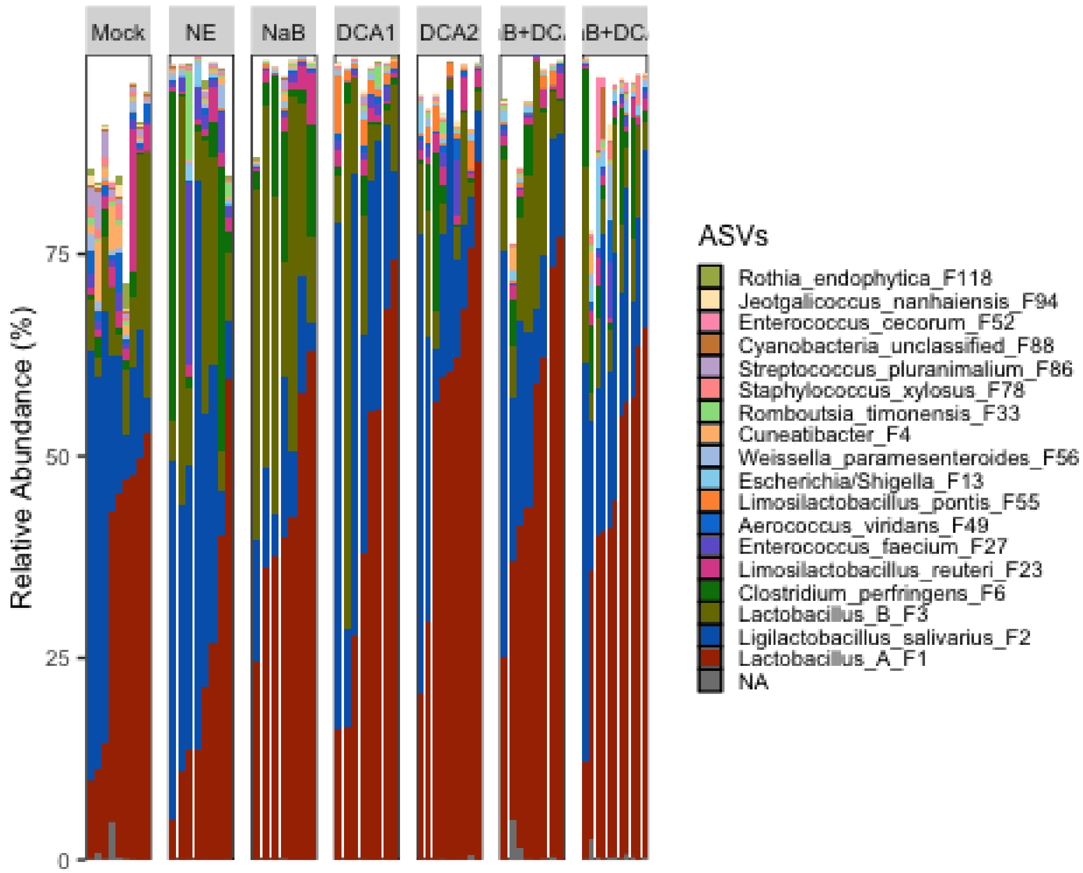
```
data <- subset(genus_relabund, Genus =='Lactobacillus')
data$Sample <- reorder(data$Sample, data$Abundance)
genus_relabund$Sample <-factor(genus_relabund$Sample, levels =
rev(data$Sample))

genus_relabund$Percent <- genus_relabund$Abundance * 100
genus_relabund$Genus <- as.factor(genus_relabund$Genus)
genus_relabund$Genus <- reorder(genus_relabund$Genus,
genus_relabund$Abundance)
genus_list <- levels(genus_relabund$Genus)

# Pull out top 15 Genus
genus_top15 <-genus_list[148:162]
genus_new <- genus_relabund[which(genus_relabund$Genus %in% genus_top15),]

genus_new$Genus <- factor(genus_new$Genus, order = TRUE,levels =
c("Cyanobacteria_unclassified",
"Streptococcus","Mediterraneibacter","Romboutsia", "Staphylococcus",
"Cuneatibacter","Escherichia/Shigella",
"Weissella","Aerococcus","Corynebacterium",
"Enterococcus","Limosilactobacillus", "Clostridium", "Ligilactobacillus",
"Lactobacillus"))

genus_colors <- c("#17BECFFF","#1F77B4FF", "#a6b352","grey","#007e07",
"#dc5097", "#6d63d2", "#64aa5d", "#d49d52", "#037cd7", "#ff933e",  "#d75154",
"#ff9bba", "#fadf77",  "#ab74c8")

ggplot(genus_new, aes(x = Sample, y = Percent, fill = Genus)) +
      geom_bar(stat = "identity") +
      facet_grid(.~Group, scales = 'free') +
  scale_fill_manual(values = rev(genus_colors)) +
  scale_y_continuous(expand=c(0,0))+  #(y axis start from 0)
  theme(panel.grid = element_blank(), panel.background = element_rect(color =
'black', fill = 'transparent')) +
  #theme(legend.title =element_blank())+
  theme(legend.text = element_text(size = 8,family = 'Arial'),
        axis.text.x = element_text(angle = 90, vjust = 0.8, hjust= 0.8))+
  #Remove x axis title
  theme(axis.title.x = element_blank()) +
  theme(axis.text.x = element_blank(),axis.ticks.x = element_blank()) +
  theme(text = element_text(size = 10, family = 'Arial')) +
  #theme(legend.position = "bottome", legend.box = "vertical") +
  guides(fill = guide_legend(reverse = FALSE, keywidth = 0.5, keyheight =
0.5, ncol = 1)) +ylab("Relative Abundance (%)")
```



Feature_barplot_individual

```
physeq_asv1 <- bacteria %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()

physeq_asv1$OTU <- reorder(physeq_asv1$OTU, physeq_asv1$Abundance)
physeq_asv1$ASVs <- paste0(physeq_asv1$Species,"_",physeq_asv1$OTU)
physeq_asv1$ASVs <- reorder(physeq_asv1$ASVs, physeq_asv1$Abundance)

physeq_asv1$Sample<-reorder(physeq_asv1$Sample, physeq_asv1$Abundance)
data <- subset(physeq_asv1, OTU == 'F1')

data$Sample <- reorder(data$Sample, data$Abundance)
physeq_asv1$Sample <-factor(physeq_asv1$Sample, levels = rev(data$Sample))

physeq_asv1$Percent <- physeq_asv1$Abundance * 100
physeq_asv1$ASVs <- as.factor(physeq_asv1$ASVs)
physeq_asv1$ASVs <- reorder(physeq_asv1$ASVs, physeq_asv1$Abundance)

# Pull out top 20 features
```

```
asv_list <- levels(physeq_asv1$ASVs)
bio_list <- asv_list[542:561]
bio_new <- physeq_asv1[which(physeq_asv1$ASVs %in% bio_list),]


bio_new$ASVs <- factor(bio_new$ASVs, levels = c("Rothia_endophytica_F118"
,"Corynebacterium_casei_F116","Corynebacterium_stationis_F105","Jeotgalicoccu
s_nanhaiensis_F94","Enterococcus_cecorum_F52","Cyanobacteria_unclassified_F88
","Streptococcus_pluranimalium_F86","Staphylococcus_xylosus_F78","Romboutsia_
timonensis_F33","Cuneatibacter_F4","Weissella_paramesenteroides_F56","Escheri
chia/Shigella_F13","Limosilactobacillus_pontis_F55","Aerococcus_viridans_F49"
,"Enterococcus_faecium_F27","Limosilactobacillus_reuteri_F23","Clostridium_pe
rfringens_F6","Lactobacillus_B_F3","Ligilactobacillus_salivarius_F2","Lactoba
cillus_A_F1"))


feat_colors <- c( "#64aa5d","#d49d52",'#a83000', "#0063b9", "#777700",
"#007e07", "#dc5097", "#6d63d2",  "#037cd7", "#ff933e", "#91d4f0",
"#AEC7E8FF", "#FFBB78FF", "#98DF8AFF","#FF9896FF", "#C5B0D5FF",
"peru","#ff9bba", "wheat1","#a6b352")

ggplot(bio_new, aes(x = Sample, y = Percent, fill = ASVs)) +
      geom_bar(stat = "identity") +
      facet_grid(.~Group, scales = 'free') +
  scale_fill_manual(values = rev(feat_colors)) +
  scale_y_continuous(expand=c(0,0))+  #(y axis start from 0)
  theme(panel.grid = element_blank(), panel.background = element_rect(color =
'black', fill = 'transparent')) +
  #theme(legend.title =element_blank())+
  theme(legend.text = element_text(size=7,family = 'Arial'))+
  #Remove x axis title
  theme(axis.title.x = element_blank(),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 90,
vjust = 0.8, hjust= 0.8)) +
  theme(axis.text.x = element_blank(),axis.ticks.x = element_blank()) +
  theme(text = element_text(size = 9, family = 'Arial')) +
  #theme(legend.position = "bottome", legend.box = "vertical") +
  guides(fill = guide_legend(reverse = FALSE, keywidth = 0.5, keyheight =
0.5, ncol = 1)) +ylab("Relative Abundance (%)")
```



Phylum-Mean

```
phylum_relabund <- bacteria %>%
  tax_glom(taxrank = "Phylum") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

phylum_relabund$Phylum <- gsub(phylum_relabund$Phylum, pattern =
'Bacteria_unclassified', replacement = 'Others')
phylum_relabund$Percent <- phylum_relabund$Abundance * 100
phylum_relabund$Phylum <- reorder(phylum_relabund$Phylum,
phylum_relabund$Percent)
phylum_relabund <- phylum_relabund[,c(2,5,12,13)]#Sample, Group,Phylum,
Percent
#phylum_relabund <- spread(phylum_relabund, Genus, Percent)
phylum_relabund<-phylum_relabund %>%
  distinct(Sample, Phylum, .keep_all = TRUE) %>%
  spread(Phylum, Percent)

phylum_relabund <- phylum_relabund[,-1]
phylum_relabund[is.na(phylum_relabund)] <- 0
```

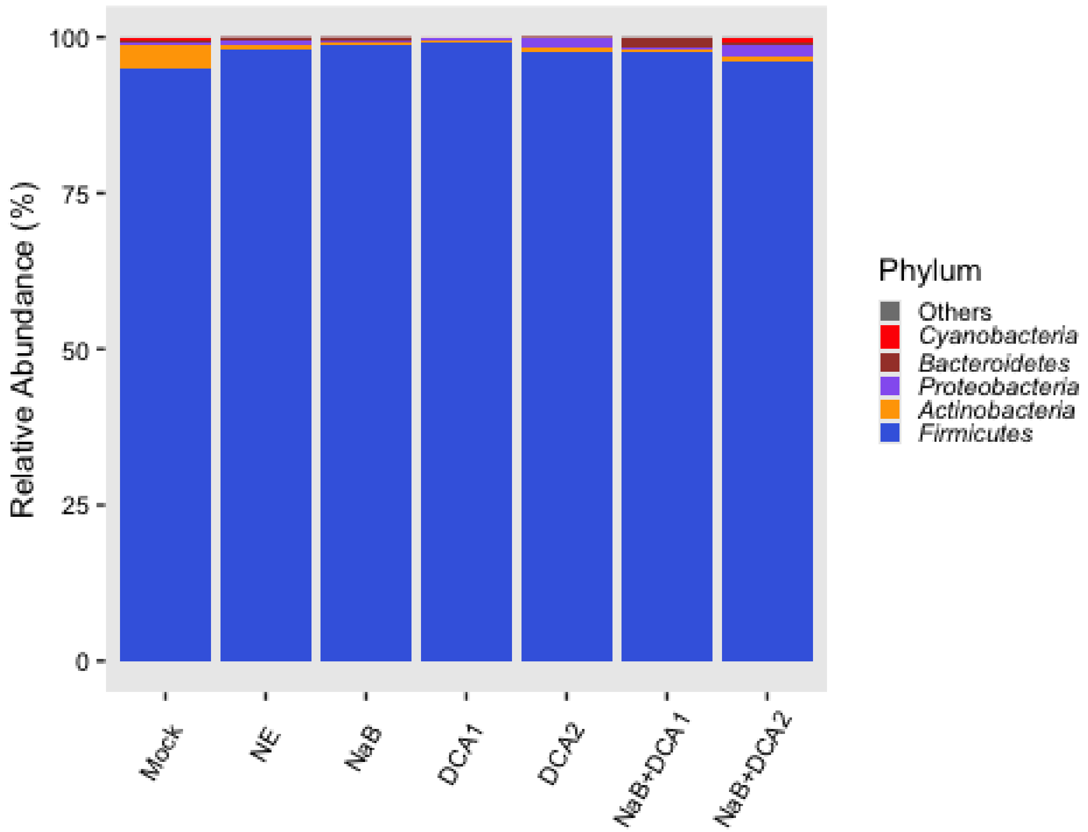
```r
phylum_relabund_2 <- phylum_relabund %>%
  group_by(Group) %>%
  summarise_each(funs(mean))
```

```
## Warning: `summarise_each()` was deprecated in dplyr 0.7.0.
## ℹ Please use `across()` instead.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## ℹ Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
#write.csv(median_genus_2, 'Genus_Relabund.csv')

pie <- phylum_relabund_2[,c(1:10)]

count <- pie[,-c(1:5)]
#Create the column 'Sums', which lists the sum of the top 15 genus.
count$Sums <- apply(count, 1, sum)
#Calculate the sum of the now-removed lowly abundant features by subtraction.
count$Others <- (100 - count$Sums)
# Remove the sums column.
count <- count[,-6]
#Add the 'Gender' and 'BW' columns back in.
count$Group <- pie$Group


long_mean <- gather(count, Phylum, Abundance, 1:6)
long_mean$Phylum <- as.factor(long_mean$Phylum)

long_mean$Phylum <- reorder(long_mean$Phylum, long_mean$Abundance)
long_mean$Phylum <- factor(long_mean$Phylum, levels =
c("Others","Cyanobacteria", "Bacteroidetes", "Proteobacteria",
"Actinobacteria", "Firmicutes"))

p_colors <- c('royalblue', 'orange', '#9666f1',"#A64036","red","grey50")

colsPhylum <-c("Firmicutes"="royalblue","Actinobacteria"=
"orange","Proteobacteria"= "#9666f1",Bacteroidetes = "#A64036",Cyanobacteria
= "red",Others = "grey50")
```

```r
labsPhylum <-c(expression(paste(italic("Firmicutes"))),
               expression(paste(italic("Actinobacteria"))),
               expression(paste(italic("Proteobacteria"))),
               expression(paste(italic("Bacteroidetes"))),
               expression(paste(italic("Cyanobacteria"))),"Others")

breaksPhylum <-c("Firmicutes", "Actinobacteria","Proteobacteria",
"Bacteroidetes","Cyanobacteria","Others")

ggplot(long_mean, aes(x = Group, y = Abundance, fill = Phylum)) +
  geom_bar(stat = 'identity') +
  #facet_grid(.~Gender, scales = 'free' ) +
  scale_fill_manual(values = colsPhylum,breaks = breaksPhylum, labels =
labsPhylum ) +
  #theme(legend.title=element_blank())+
  theme(text = element_text(size = 10, family = 'Arial'),legend.text.align =
0,
        axis.text = element_text(size = 8,family = 'Arial', color = "black"),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8)) +
  guides(fill = guide_legend(reverse = TRUE, keyheight = 0.5, keywidth = 0.5,
ncol = 1)) +
  ylab("Relative Abundance (%)") +
  xlab("")

## Warning: The `legend.text.align` argument of `theme()` is deprecated as of
ggplot2
## 3.5.0.
## ℹ Please use theme(legend.text = element_text(hjust)) instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
ggsave('Figures/Ileum/Abundance/phylum.png',
       height = 3.0,
       width = 2.9,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Phylum statistical analysis

```
sem <- function(x) sd(x)/sqrt(length(x))

phylum_relabund <- bacteria %>%
  tax_glom(taxrank = "Phylum") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

phylum_relabund$Phylum <- gsub(phylum_relabund$Phylum, pattern =
'Bacteria_unclassified', replacement = 'Others')
phylum_relabund$Phylum <- reorder(phylum_relabund$Phylum,
phylum_relabund$Abundance)
phylum_relabund$Percent <- phylum_relabund$Abundance * 100
```

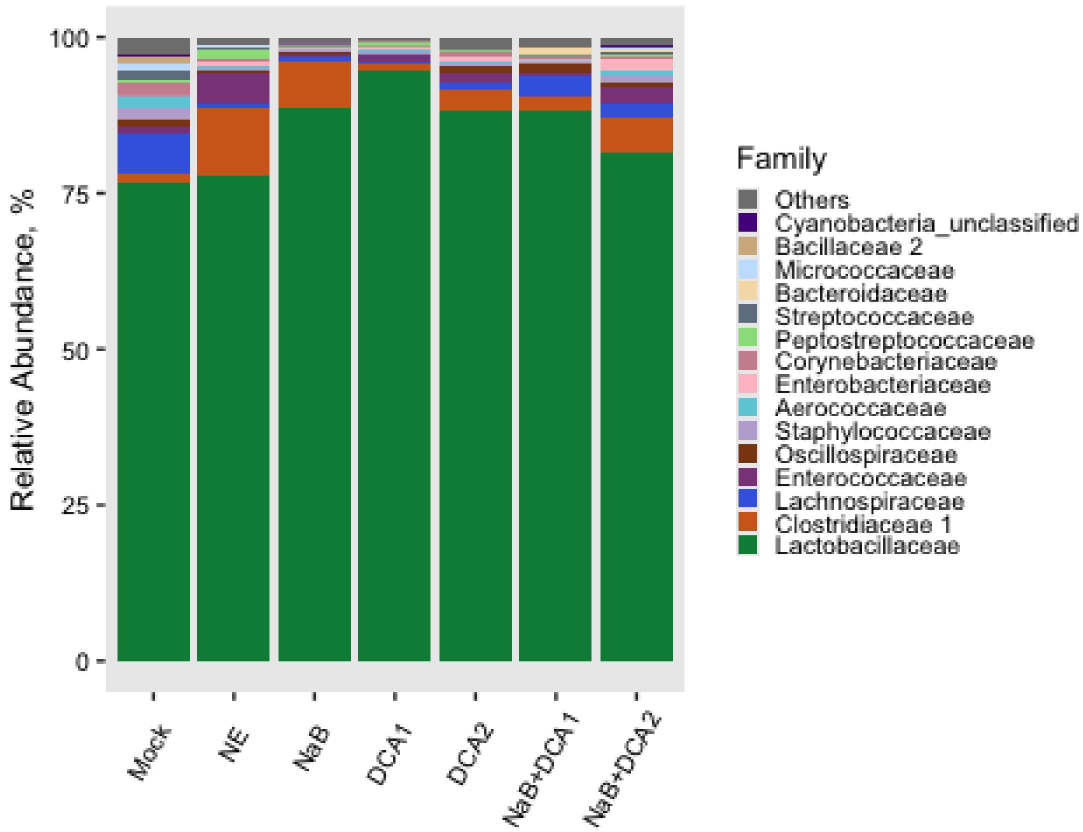
```
data <- subset(phylum_relabund, Phylum == 'Firmicutes')
data$Sample <- reorder(data$Sample, data$Abundance)
phylum_relabund$Sample <- factor(phylum_relabund$Sample, levels =
rev(data$Sample))

# Pull out the Sample, Percent, Group, and Phylum columns
# Collapse Relabund
mean_phylum <-  phylum_relabund[,c(2,5,12,13)] #Sample, Group, Phylum,
Percent

# Spread into a data frame where each phyla has its own column
mean_phylum <- spread(mean_phylum, Phylum, Percent)
#mean_phylum<- mean_phylum[,-c(1,2,3)]
# Remove Sample and Group columns
mean <- mean_phylum[,-c(1,2)]
 # Convert NAs to 0s
mean[is.na(mean)] <- 0
mean_phylum[is.na(mean_phylum)] <- 0

# Calculate the mean of each phylum in each Trt
mean_phylum_2 <- aggregate(mean, by=list(mean_phylum$Group), mean)

# Transform so that rows are phyla
mean_phylum_3 <- t(mean_phylum_2)
mean_phylum_3 <- as.data.frame(mean_phylum_3)
colnames(mean_phylum_3) <-mean_phylum_3[1,]
mean_phylum_3 <-mean_phylum_3[-1,]

# Calcualte the standard deviation of each phylum
sem_phylum <- aggregate(mean, by=list(mean_phylum$Group), sem)
sem_phylum_2 <- t(sem_phylum)
sem_phylum_2 <- as.data.frame(sem_phylum_2)
colnames(sem_phylum_2) <-sem_phylum_2[1,]
sem_phylum_2 <-sem_phylum_2[-1,]

# Run Statistics
# Pull out the column names that include the list of phyla
phyla <- colnames(mean_phylum)
# Remova the Sample and lesion score from th elist
phyla <- phyla[-c(1:2)]

# Create empty data frame for loop
Phyla_P <- data.frame(matrix(nrow = length(phyla), ncol = 1))
 # Label the rows and columns of the empty data frame
rownames(Phyla_P) <- phyla
colnames(Phyla_P) <- "P"

Phyla_P <- NULL
```

```
for (i in 1:length(phyla)){
  k <- kruskal.test(mean_phylum[,2+i] ~ mean_phylum[,2], mean_phylum)
  data <- data.frame(Phylum = phyla[i], P = k[3])
  Phyla_P <- rbind(Phyla_P, data)
}


Phyla_P$BH <- p.adjust(Phyla_P$p.value, method = 'BH')
rownames(Phyla_P) <-Phyla_P$Phylum
Phyla_P <- cbind(Phyla_P, mean_phylum_3,sem_phylum_2)

write.csv(Phyla_P, 'Processed_data/Abundance/Ileum/phylum_stats.csv')
```

Family-mean

```
family_relabund <- bacteria %>%
  tax_glom(taxrank = "Family") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

family_relabund$Family <- gsub(family_relabund$Family, pattern =
'Bacteria_unclassified', replacement = 'Others')
family_relabund$Family <- reorder(family_relabund$Family,
family_relabund$Abundance)
family_relabund$Percent <- family_relabund$Abundance * 100

family_relabund <- family_relabund[,c(2,5,15,16)]#Sample, Group, Family,
Percent
#family_relabund <- spread(family_relabund, Genus, Percent)
family_relabund<-family_relabund %>%
  distinct(Sample, Family, .keep_all = TRUE) %>%
  spread(Family, Percent)

family_relabund <- family_relabund[,-1]
family_relabund[is.na(family_relabund)] <- 0

family_relabund_2 <- family_relabund %>%
  group_by(Group) %>%
  summarise_each(funs(mean))

#Copy all columns from 'genus_relabund_2' to new object 'pie'.
pie <- family_relabund_2[,c(1:73)]

count <- pie[,-c(1:58)]
#Create the column 'Sums', which lists the sum of the top 15 genus.
count$Sums <- apply(count, 1, sum)
#Calculate the sum of the now-removed lowly abundant features by subtraction.
count$Others <- (100 - count$Sums)
```

```r
#Remove the sums column.
count <- count[,-c(16)]
#Add the 'Group' and 'BW' columns back in.
count$Group <- pie$Group

#1:n, where n is the number of columns containing genus relabund info.
long_mean <- gather(count, Family, Abundance, 1:16)
long_mean$Family <- as.factor(long_mean$Family)
long_mean$Family <- reorder(long_mean$Family, long_mean$Abundance)

long_mean$Family <- factor(long_mean$Family, levels =
c("Others","Cyanobacteria_unclassified","Bacillaceae
2","Micrococcaceae","Bacteroidaceae","Streptococcaceae","Peptostreptococcacea
e","Corynebacteriaceae","Enterobacteriaceae","Aerococcaceae","Staphylococcace
ae","Oscillospiraceae", "Enterococcaceae", "Lachnospiraceae", "Clostridiaceae
1", "Lactobacillaceae"))

family_color <-
c("springgreen4","chocolate","royalblue","orchid4","chocolate4","#beaed4","#6
DCCDAFF", "pink", "pink3", "#98DF8AFF","slategray", "wheat", "slategray1",
"tan","purple4", "grey50")

ggplot(long_mean, aes(x = Group, y = Abundance, fill = Family)) +
  geom_bar(stat = 'identity') +
  scale_fill_manual(values = rev(family_color)) +
  #theme(legend.title=element_blank())+
  theme(text = element_text(size = 10, family = 'Arial'),
        axis.text = element_text(size = 8,family = 'Arial', color = "black"),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8)) +
  guides(fill = guide_legend(reverse = FALSE, keyheight = 0.5, keywidth =
0.5, ncol = 1)) +
  ylab("Relative Abundance, %") +
  xlab("")
```



```
ggsave('Figures/Ileum/Abundance/family.png',
       height = 2.8,
       width = 3.35,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

Family-Statitical analysis

```
family_relabund <- bacteria %>%
  tax_glom(taxrank = "Family") %>%                    # group at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt()  # Melt to long format

family_relabund$Family <- gsub(family_relabund$Family, pattern =
'Bacteria_unclassified', replacement = 'Others')
family_relabund$Family <- gsub(family_relabund$Family, pattern =
'Firmicutes_unclassified', replacement = 'Others')
family_relabund$Family <- reorder(family_relabund$Family,
family_relabund$Abundance)
family_relabund$Percent <- family_relabund$Abundance * 100
```

```r
family_relabund <- family_relabund[,c(2,5,15,16)]
#family_relabund <- spread(family_relabund, Family, Percent)
family_relabund<-family_relabund %>%
  distinct(Sample, Family, .keep_all = TRUE) %>%
  spread(Family, Percent)

family_mean <- family_relabund[,-c(1,2)]

family_mean[is.na(family_mean)] <- 0

family_relabund[is.na(family_relabund)] <- 0

family_mean_2 <- aggregate(family_mean, by=list(family_relabund$Group), mean)
family_mean_2 <- t(family_mean_2)
family_mean_2 <- as.data.frame(family_mean_2)
colnames(family_mean_2) <-family_mean_2[1,]
family_mean_2 <-family_mean_2[-1,]

# Calcualte the standard deviation of each family
sem_family <- aggregate(family_mean, by=list(family_relabund$Group), sem)
sem_family_2 <- t(sem_family)
sem_family_2 <- as.data.frame(sem_family_2)
colnames(sem_family_2) <-sem_family_2[1,]
sem_family_2 <-sem_family_2[-1,]

# Pull out the column names that include the list of families
families <- colnames(family_relabund)
# Remove the Sample and Group names from the list
families <- families[-c(1:2)]
# Create empty data frame for loop
Families_P <- data.frame(matrix(nrow = length(families), ncol = 1))
# Label the rows and columns of the empty data frame
rownames(Families_P) <- families
colnames(Families_P) <- "P"

family_relabund[is.na(family_relabund)] <- 0
Families_P <- NULL

for (family in 1:length(families)){
  k <- kruskal.test(family_relabund[,2+family] ~ family_relabund[,2],
family_relabund)
  data <- data.frame(Family= families[family], P = k[3])
  Families_P <- rbind(Families_P, data)
}

Families_P$BH <- p.adjust(Families_P$p.value, method = 'BH')
rownames(Families_P) <-Families_P$Family
```

```
Families_P <- cbind(Families_P, family_mean_2,sem_family_2)
Families_P

write.csv(Phyla_P, 'Processed_data/Abundance/Ileum/Family_stats.csv')


pairwise.wilcox.test(family_relabund$Bacteroidaceae, family_relabund$Group,
p.adjust.method = 'none')
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity
correction
##
## data:  family_relabund$Bacteroidaceae and family_relabund$Group
##
##           Mock    NE      NaB     DCA1    DCA2    NaB+DCA1
## NE        0.46426 -       -       -       -       -
## NaB       0.58414 0.94080 -       -       -       -
## DCA1      0.04507 0.17090 0.14235 -       -       -
## DCA2      0.31699 0.04758 0.07706 0.00305 -       -
## NaB+DCA1  0.08101 0.03796 0.05865 0.00457 0.16035 -
## NaB+DCA2  0.00495 0.00592 0.01119 0.00056 0.00968 0.48199
##
## P value adjustment method: none
```

```
##### add compact letter display to the results of pairwise Wilcoxon test
Table = suppressWarnings(pairwise.wilcox.test(family_relabund$Bacteroidaceae,
family_relabund$Group,p.adjust.method = 'none'))
Table2 = Table$p.value
Table2
```

```
##                 Mock          NE        NaB         DCA1        DCA2
NaB+DCA1
## NE       0.464258737          NA         NA           NA          NA
NA
## NaB      0.584143389 0.940802657         NA           NA          NA
NA
## DCA1     0.045071066 0.170903520 0.14234505           NA          NA
NA
## DCA2     0.316986801 0.047577056 0.07706378 0.0030503660          NA
NA
## NaB+DCA1 0.081006536 0.037959472 0.05865019 0.0045693046 0.160351028
NA
## NaB+DCA2 0.004950082 0.005922449 0.01118645 0.0005644807 0.009681572
0.4819941
```

```
Table3 = fullPTable(Table2)
Table3
```

```
##                 Mock          NE        NaB         DCA1        DCA2
## Mock     1.000000000 0.464258737 0.58414339 0.0450710662 0.316986801
## NE       0.464258737 1.000000000 0.94080266 0.1709035202 0.047577056
## NaB      0.584143389 0.940802657 1.00000000 0.1423450481 0.077063782
## DCA1     0.045071066 0.170903520 0.14234505 1.0000000000 0.003050366
## DCA2     0.316986801 0.047577056 0.07706378 0.0030503660 1.000000000
## NaB+DCA1 0.081006536 0.037959472 0.05865019 0.0045693046 0.160351028
## NaB+DCA2 0.004950082 0.005922449 0.01118645 0.0005644807 0.009681572
##             NaB+DCA1     NaB+DCA2
## Mock     0.081006536 0.0049500819
## NE       0.037959472 0.0059224492
## NaB      0.058650194 0.0111864546
## DCA1     0.004569305 0.0005644807
## DCA2     0.160351028 0.0096815717
## NaB+DCA1 1.000000000 0.4819940608
## NaB+DCA2 0.481994061 1.0000000000
```

```
multcompLetters(Table3)
```

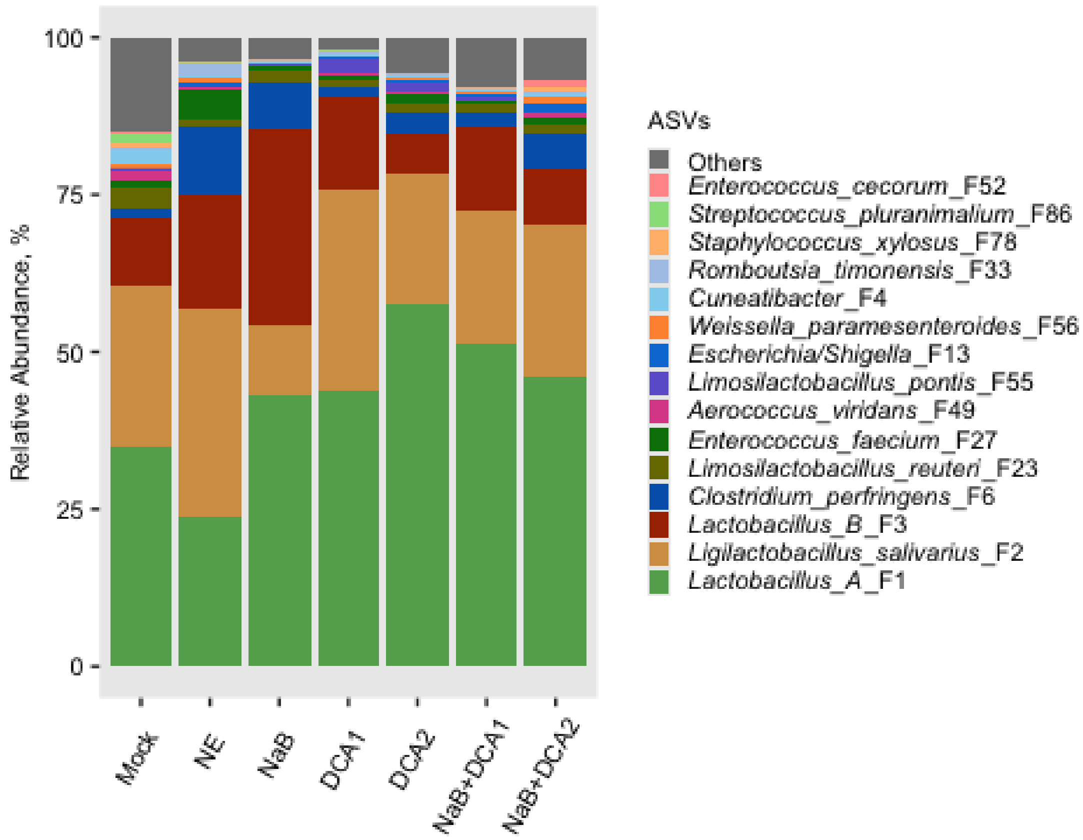
```
##     Mock       NE      NaB     DCA1     DCA2 NaB+DCA1 NaB+DCA2 
##     "ab"     "ac"    "abc"      "c"      "b"     "bd"      "d"
```

Feature

```
## Prepare to plot relative abundance data
    # melt to long format (for ggploting)
    # prune out phyla below 0 in each sample
feature_relanund <- bacteria %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel.
abundance
  psmelt() %>%
  arrange(OTU)

feature_relanund$ASVs <-
paste0(feature_relanund$Species,"_",feature_relanund$OTU)
feature_relanund$ASVs <- reorder(feature_relanund$ASVs,
feature_relanund$Abundance)
feature_relanund$Percent <- feature_relanund$Abundance * 100

feature_relanund <- feature_relanund[,c(2,5,18,19)]
#genus_relabund <- spread(genus_relabund, Genus, Percent)
feature_relanund<-feature_relanund %>%
  distinct(Sample, ASVs, .keep_all = TRUE) %>%
  spread(ASVs, Percent)

feature_relanund <- feature_relanund[,-1]
feature_relanund[is.na(feature_relanund)] <- 0

feature_relanund_2 <- feature_relanund %>%
  group_by(Group) %>%
  summarise_each(funs(mean))
```

```
# Copy all columns from 'genus_relabund_2' to new object 'pie'.
pie <- feature_relanund_2[,c(1:562)]
#Remove lowly abundant features, leaving only the top 30 most abundant.
count <- pie[,-c(1:547)]
#Create the column 'Sums', which lists the sum of the top 30 genus.
count$Sums <- apply(count, 1, sum)
#Calculate the sum of the now-removed lowly abundant features by subtraction.
count$Others <- (100 - count$Sums)
#Remove the sums column.
count <- count[,-16]
#Add the 'Gender' and 'BW' columns back in.
count$Group <- pie$Group

long_mean <- gather(count, ASVs, Abundance, 1:16)
long_mean$ASVs <- as.factor(long_mean$ASVs)
long_mean$ASVs <- reorder(long_mean$ASVs, long_mean$Abundance)

long_mean$ASVs <- factor(long_mean$ASVs, levels
=c("Others","Enterococcus_cecorum_F52","Streptococcus_pluranimalium_F86","Sta
phylococcus_xylosus_F78","Romboutsia_timonensis_F33","Cuneatibacter_F4","Weis
sella_paramesenteroides_F56","Escherichia/Shigella_F13","Limosilactobacillus_
pontis_F55","Aerococcus_viridans_F49","Enterococcus_faecium_F27","Limosilacto
bacillus_reuteri_F23","Clostridium_perfringens_F6","Lactobacillus_B_F3","Ligi
lactobacillus_salivarius_F2","Lactobacillus_A_F1"))

labsPhylum <-c(expression(paste(italic("Lactobacillus_A"),"_F1")),

expression(paste(italic("Ligilactobacillus_salivarius"),"_F2")),
               expression(paste(italic("Lactobacillus_B"),"_F3")),
               expression(paste(italic("Clostridium_perfringens"),"_F6")),

expression(paste(italic("Limosilactobacillus_reuteri"),"_F23")),
               expression(paste(italic("Enterococcus_faecium"),"_F27")),
               expression(paste(italic("Aerococcus_viridans"),"_F49")),

expression(paste(italic("Limosilactobacillus_pontis"),"_F55")),
               expression(paste(italic("Escherichia/Shigella"),"_F13")),

expression(paste(italic("Weissella_paramesenteroides"),"_F56")),
               expression(paste(italic("Cuneatibacter"),"_F4")),
               expression(paste(italic("Romboutsia_timonensis"),"_F33")),
               expression(paste(italic("Staphylococcus_xylosus"),"_F78")),

expression(paste(italic("Streptococcus_pluranimalium"),"_F86")),

expression(paste(italic("Enterococcus_cecorum"),"_F52")),"Others")

breaksPhylum <-
c("Others","Enterococcus_cecorum_F52","Streptococcus_pluranimalium_F86","Stap
```

```r
hylococcus_xylosus_F78","Romboutsia_timonensis_F33","Cuneatibacter_F4","Weiss
ella_paramesenteroides_F56","Escherichia/Shigella_F13","Limosilactobacillus_p
ontis_F55","Aerococcus_viridans_F49","Enterococcus_faecium_F27","Limosilactob
acillus_reuteri_F23","Clostridium_perfringens_F6","Lactobacillus_B_F3","Ligil
actobacillus_salivarius_F2","Lactobacillus_A_F1")


feat_colors <- c( "#64aa5d","#d49d52",'#a83000', "#0063b9", "#777700",
"#007e07", "#dc5097", "#6d63d2",  "#037cd7", "#ff933e", "#91d4f0",
"#AEC7E8FF", "#FFBB78FF", "#98DF8AFF","#FF9896FF", "#C5B0D5FF",
"peru","#ff9bba", "wheat1","#a6b352","grey50")

colsPhylum <- c("Lactobacillus_A_F1"
="#64aa5d","Ligilactobacillus_salivarius_F2" ="#d49d52","Lactobacillus_B_F3"
='#a83000', "Clostridium_perfringens_F6" ="#0063b9",
"Limosilactobacillus_reuteri_F23" ="#777700", "Enterococcus_faecium_F27"
="#007e07", "Aerococcus_viridans_F49" ="#dc5097",
"Limosilactobacillus_pontis_F55" ="#6d63d2", "Escherichia/Shigella_F13" =
"#037cd7","Weissella_paramesenteroides_F56" = "#ff933e", "Cuneatibacter_F4"
="#91d4f0",  "Romboutsia_timonensis_F33"
="#AEC7E8FF","Staphylococcus_xylosus_F78" =
"#FFBB78FF","Streptococcus_pluranimalium_F86" =
"#98DF8AFF","Enterococcus_cecorum_F52" ="#FF9896FF","Others" ="grey50")


ggplot(long_mean, aes(x = Group, y = Abundance, fill = ASVs)) +
  geom_bar(stat = 'identity') +
  scale_fill_manual(values = colsPhylum,breaks = rev(breaksPhylum), labels =
labsPhylum ) +
  #theme(legend.title=element_blank())+
 theme(legend.text = element_text(size=8,family = 'Arial'),legend.text.align
= 0,
       text = element_text(size = 8, family = 'Arial'),
        axis.text = element_text(size = 8,family = 'Arial', color = "black"),
        panel.grid = element_blank(),axis.text.x = element_text(angle = 60,
vjust = 0.8, hjust= 0.8)) +
  guides(fill = guide_legend(reverse = TRUE, keyheight = 0.5, keywidth = 0.5,
ncol = 1)) +
  ylab("Relative Abundance, %") +
  xlab("")
```



```
ggsave('Figures/Ileum/Abundance/ASV.png',
       height = 3.0,
       width = 3.85,
       unit = 'in',
       scale = 1,
       dpi = 300)
```

ASV Statistical analysis

```
## Run statistics

  # Collaps relabund
feature_relanund <- bacteria %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance
  psmelt() %>%
  arrange(OTU)

feature_relanund$Features <- paste0(feature_relanund$Species,"_",feature_relanund$OTU)
feature_relanund$Features <- reorder(feature_relanund$Features, feature_relanund$Abundance)
```

```r
feature_relanund$Percent <- feature_relanund$Abundance * 100

feature_relanund <- feature_relanund[,c(2,5,18,19)]
feature_relanund<-feature_relanund %>%
  distinct(Sample, Features, .keep_all = TRUE) %>%
  spread(Features, Percent)

feature_mean <- feature_relanund[,-c(1,2)]
feature_mean[is.na(feature_mean)] <- 0
feature_relanund[is.na(feature_relanund)] <- 0

mean_feature <- aggregate(feature_mean, by=list(feature_relanund$Group),
mean)
mean_feature <- as.data.frame(t(mean_feature))
colnames(mean_feature) <-mean_feature[1,]
mean_feature <-mean_feature[-1,]

# Calcualte the standard deviation of each genus
sem_feature <- aggregate(feature_mean, by=list(feature_relanund$Group), sem)
sem_feature_2 <- t(sem_feature)
sem_feature_2 <- as.data.frame(sem_feature_2)
colnames(sem_feature_2) <-sem_feature_2[1,]
sem_feature_2 <-sem_feature_2[-1,]

# Pull out the column names that include the list of feature
feat <- colnames(feature_relanund)
# Remove the Sample and Group names from the list
feat <- feat[-c(1,2)]
# Create empty data frame for loop
Feature_P <- data.frame(matrix(nrow = length(feat), ncol = 1))
# Label the rows and columns of the empty data frame
rownames(Feature_P) <- feat
colnames(Feature_P) <- "P"

feature_relanund[is.na(feature_relanund)] <- 0

Feature_P <- NULL

for (feature in 1:length(feat)){
  k <- kruskal.test(feature_relanund[,2+feature] ~ feature_relanund[,2],
feature_relanund)
  data <- data.frame(Feature = feat[feature], P = k[3])
  Feature_P <- rbind(Feature_P, data)
}

Feature_P$BH <- p.adjust(Feature_P$p.value, method = 'BH')
rownames(Feature_P) <-Feature_P$Family
Feature_P <- cbind(Feature_P, mean_feature,sem_feature_2)
```

Scatter plots

F1

```
set.seed(1000)

kruskal.test(feature_relanund$Lactobacillus_A_F1~Group, feature_relanund)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  feature_relanund$Lactobacillus_A_F1 by Group
## Kruskal-Wallis chi-squared = 14.152, df = 6, p-value = 0.02798
```

```
Table=pairwise.wilcox.test(feature_relanund$Lactobacillus_A_F1,feature_relanu
nd$Group,p.adjust.method = "none",paired = F)
Table2 = Table$p.value
Table2
```

```
##                Mock         NE       NaB      DCA1      DCA2  NaB+DCA1
## NE       0.32126697         NA        NA        NA        NA        NA
## NaB      0.75769231 0.05407925        NA        NA        NA        NA
## DCA1     0.27659399 0.06495726 1.0000000        NA        NA        NA
## DCA2     0.01061292 0.00370218 0.1737762 0.1387906        NA        NA
## NaB+DCA1 0.19958865 0.01476301 0.3356643 0.4418026 0.6058412        NA
## NaB+DCA2 0.22986425 0.02034876 0.4789467 0.8403853 0.1308050 0.7167844
```

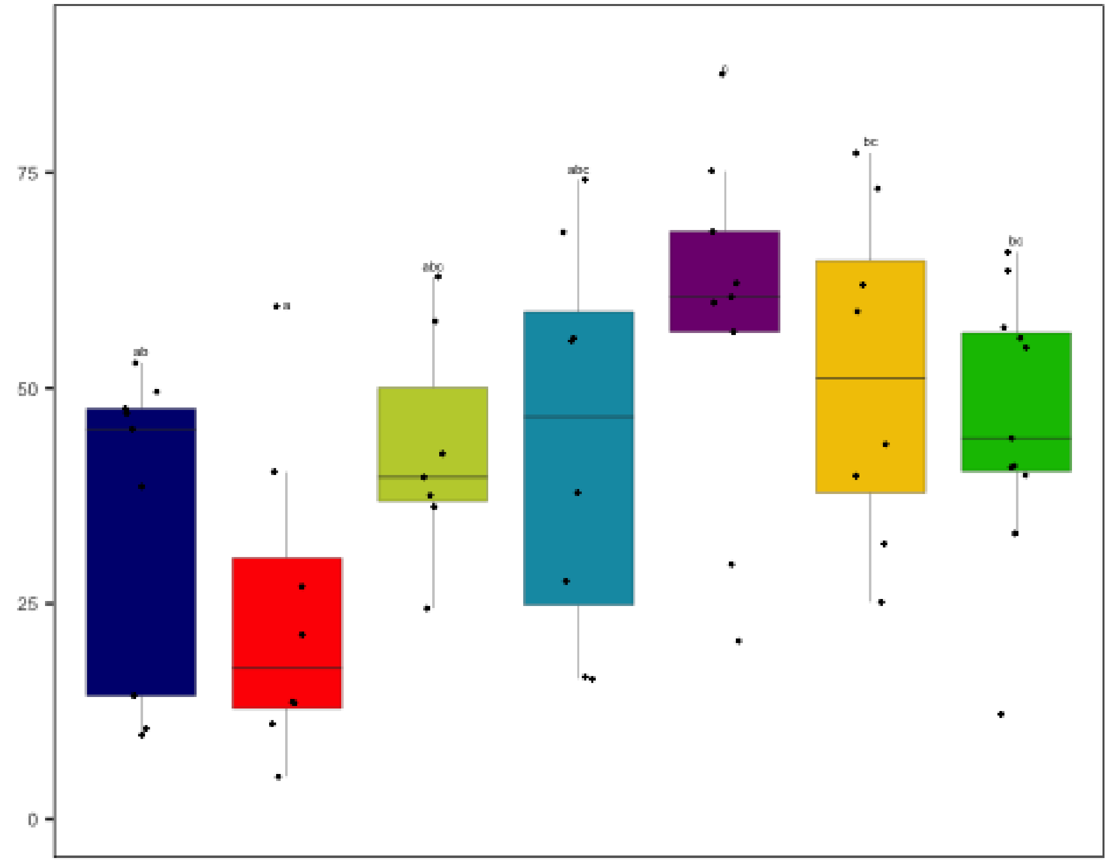
```
Table3 = fullPTable(Table2)
Table3
```

```
##                Mock         NE        NaB       DCA1       DCA2   NaB+DCA1
## Mock     1.00000000 0.32126697 0.75769231 0.27659399 0.01061292 0.19958865
## NE       0.32126697 1.00000000 0.05407925 0.06495726 0.00370218 0.01476301
## NaB      0.75769231 0.05407925 1.00000000 1.00000000 0.17377622 0.33566434
## DCA1     0.27659399 0.06495726 1.00000000 1.00000000 0.13879062 0.44180264
## DCA2     0.01061292 0.00370218 0.17377622 0.13879062 1.00000000 0.60584122
## NaB+DCA1 0.19958865 0.01476301 0.33566434 0.44180264 0.60584122 1.00000000
## NaB+DCA2 0.22986425 0.02034876 0.47894671 0.84038528 0.13080495 0.71678442
##            NaB+DCA2
## Mock     0.22986425
## NE       0.02034876
## NaB      0.47894671
## DCA1     0.84038528
## DCA2     0.13080495
## NaB+DCA1 0.71678442
## NaB+DCA2 1.00000000
```

```
cld <- multcompLetters(Table3)[["Letters"]]
cld
```

```
##      Mock        NE       NaB      DCA1      DCA2 NaB+DCA1 NaB+DCA2
##      "ab"       "a"     "abc"     "abc"       "c"      "bc"      "bc"
```

```
dt <- group_by(feature_relanund, Group) %>%
  summarise(w=mean(Lactobacillus_A_F1), quant = quantile(Lactobacillus_A_F1,
probs = 0.99))

# extracting the compact letter display and adding to the Tk table
dt$cld <- cld

clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")

ggplot(feature_relanund, aes(x=Group, y= Lactobacillus_A_F1,fill=Group)) +
geom_boxplot(position=position_dodge(0.1),size=0.1,outlier.colour =
NA)+geom_jitter(shape=19,size= 0.2,
position=position_jitterdodge(jitter.width=1))+
  scale_fill_manual(values = clr)+
  geom_text(data = dt, aes(y = quant, label = cld), size = 1.5, vjust=-1,
hjust =0.5)+
  scale_y_continuous(limits = c(0,90)) +
  labs(title=NULL,
       y=NULL,
       x=NULL) +
   theme(title =element_text(size=7),axis.text.y =element_text(size=6),
panel.border = element_rect(fill = NA),panel.background = element_rect(fill =
NA),legend.position = "none",axis.text.x = element_blank(),axis.ticks.x =
element_blank())
```



```
ggsave('Figures/Ileum/Dotplots/F1.png',
       height = 1.0,
       width =1.1,
       unit = 'in',
       dpi = 300)
```

F2

```
set.seed(1000)

kruskal.test(feature_relanund$Ligilactobacillus_salivarius_F2~Group,
feature_relanund)

##
##  Kruskal-Wallis rank sum test
##
## data:  feature_relanund$Ligilactobacillus_salivarius_F2 by Group
## Kruskal-Wallis chi-squared = 9.8437, df = 6, p-value = 0.1314

Table=pairwise.wilcox.test(feature_relanund$Ligilactobacillus_salivarius_F2,f
eature_relanund$Group,p.adjust.method = "none",paired = F)
Table2 = Table$p.value
Table2
```

```
##               Mock         NE         NaB      DCA1      DCA2  NaB+DCA1
## NE       0.6058412         NA          NA        NA        NA        NA
## NaB      0.1416084 0.04009324          NA        NA        NA        NA
## DCA1     0.3703826 0.64537685 0.020512821        NA        NA        NA
## DCA2     0.7304401 0.37038256 0.469755245 0.1671740        NA        NA
## NaB+DCA1 0.8883587 0.27863248 0.054079254 0.1948718 0.4807075        NA
## NaB+DCA2 0.7103001 0.31004737 0.005907491 0.3950411 0.2298643 0.4920219
```

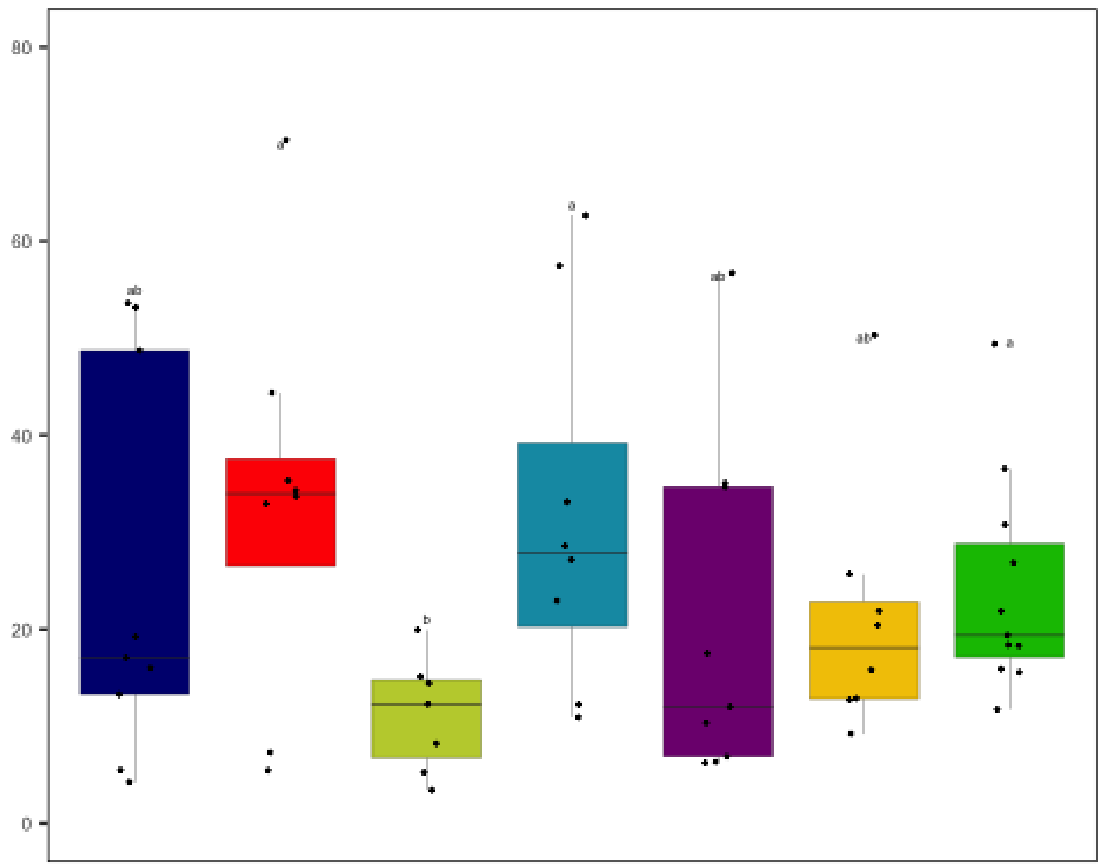
```
Table3 = fullPTable(Table2)
Table3
```

```
##               Mock         NE         NaB       DCA1      DCA2   NaB+DCA1
## Mock     1.0000000 0.60584122 0.141608392 0.37038256 0.7304401 0.88835870
## NE       0.6058412 1.00000000 0.040093240 0.64537685 0.3703826 0.27863248
## NaB      0.1416084 0.04009324 1.000000000 0.02051282 0.4697552 0.05407925
## DCA1     0.3703826 0.64537685 0.020512821 1.00000000 0.1671740 0.19487179
## DCA2     0.7304401 0.37038256 0.469755245 0.16717400 1.0000000 0.48070753
## NaB+DCA1 0.8883587 0.27863248 0.054079254 0.19487179 0.4807075 1.00000000
## NaB+DCA2 0.7103001 0.31004737 0.005907491 0.39504115 0.2298643 0.49202191
##             NaB+DCA2
## Mock     0.710300071
## NE       0.310047366
## NaB      0.005907491
## DCA1     0.395041147
## DCA2     0.229864253
## NaB+DCA1 0.492021910
## NaB+DCA2 1.000000000
```

```
cld <- multcompLetters(Table3)[["Letters"]]
cld
```

```
##     Mock       NE      NaB     DCA1     DCA2 NaB+DCA1 NaB+DCA2
##     "ab"      "a"      "b"      "a"     "ab"     "ab"      "a"
```

```
dt <- group_by(feature_relanund, Group) %>%
  summarise(w=mean(Ligilactobacillus_salivarius_F2), quant =
quantile(Ligilactobacillus_salivarius_F2, probs = 0.99))

# extracting the compact letter display and adding to the Tk table
dt$cld <- cld



clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")

ggplot(feature_relanund, aes(x=Group, y=
Ligilactobacillus_salivarius_F2,fill=Group)) +
geom_boxplot(position=position_dodge(0.1),size=0.1,outlier.colour =
NA)+geom_jitter(shape=19,size= 0.2,
position=position_jitterdodge(jitter.width=1))+
  scale_fill_manual(values = clr)+
  geom_text(data = dt, aes(y = quant, label = cld), size = 1.5, vjust=-1,
hjust =0.5)+
  scale_y_continuous(limits = c(0,80)) +
  labs(title=NULL,
       y=NULL,
       x=NULL) +
   theme(title =element_text(size=7),axis.text.y =element_text(size=6),
panel.border = element_rect(fill = NA),panel.background = element_rect(fill =
NA),legend.position = "none",axis.text.x = element_blank(),axis.ticks.x =
element_blank())
```



```
 ggsave('Figures/Ileum/Dotplots/F2.png',
        height = 1.0,
        width =1.1,
        unit = 'in',
        dpi = 300)
```

Lefse Analysis in R

Data Import-Lefse Analysis

The taxonomy, feature, and metadata tables were loaded into R and placed in a phyloseq object.

```r
# remove history commands
rm(list=ls())

#Feature Table
otumat <- read.csv("QIIME2/feature_table/feature_table_tax.txt", sep="")
rownames(otumat) <- otumat[,1]
otumat <- otumat[,-c(1)]
#Sample Data
sampledata <- read.csv("Raw_data/sample_info.txt", sep="", header=T)
sampledata$Group<-as.factor(sampledata$Group)
sampledata$Site<-as.factor(sampledata$Site)

#Taxonomy table
taxmat <- read.csv(file = 'Processed_data/taxonomy_lef.csv', header = T, sep
= ",", stringsAsFactors = F) # delete percent coloums in excel

rownames(taxmat) <- taxmat[,1]
taxmat <- taxmat[,-1]
stopifnot(rownames(otumat) == rownames(taxmat))
#Convert taxonomy and OTU table to matrix
taxmat <- as.matrix(taxmat)
TAX <- tax_table(taxmat)
otumat <- data.matrix(otumat)
OTU <- otu_table(otumat, taxa_are_rows = T)
sample.data <- sample_data(sampledata)
rownames(sample.data) <- sample.data$SampleID
#colnames(otumat)
physeq <- phyloseq(TAX, OTU, sample.data)

colnames(tax_table(physeq)) <- c("Kingdom", "Phylum", "Class",
                                 "Order", "Family", "Genus","Species")
sample_data(physeq)$Group <- factor(sample_data(physeq)$Group, levels =
c('Mock', 'NE','NaB',"DCA1",'DCA2','NaB+DCA1','NaB+DCA2'))
sample_data(physeq)$Site <- factor(sample_data(physeq)$Site, levels =
c('Ileum', 'Cecum'))

# Subset sample based on site and trial
physeq_Ile <- subset_samples(physeq, Site == "Ileum")
physeq_Ile<-subset_taxa(physeq_Ile, Kingdom != "Eukaryota" & Phylum !=
"Bacteria_unclassified")
```

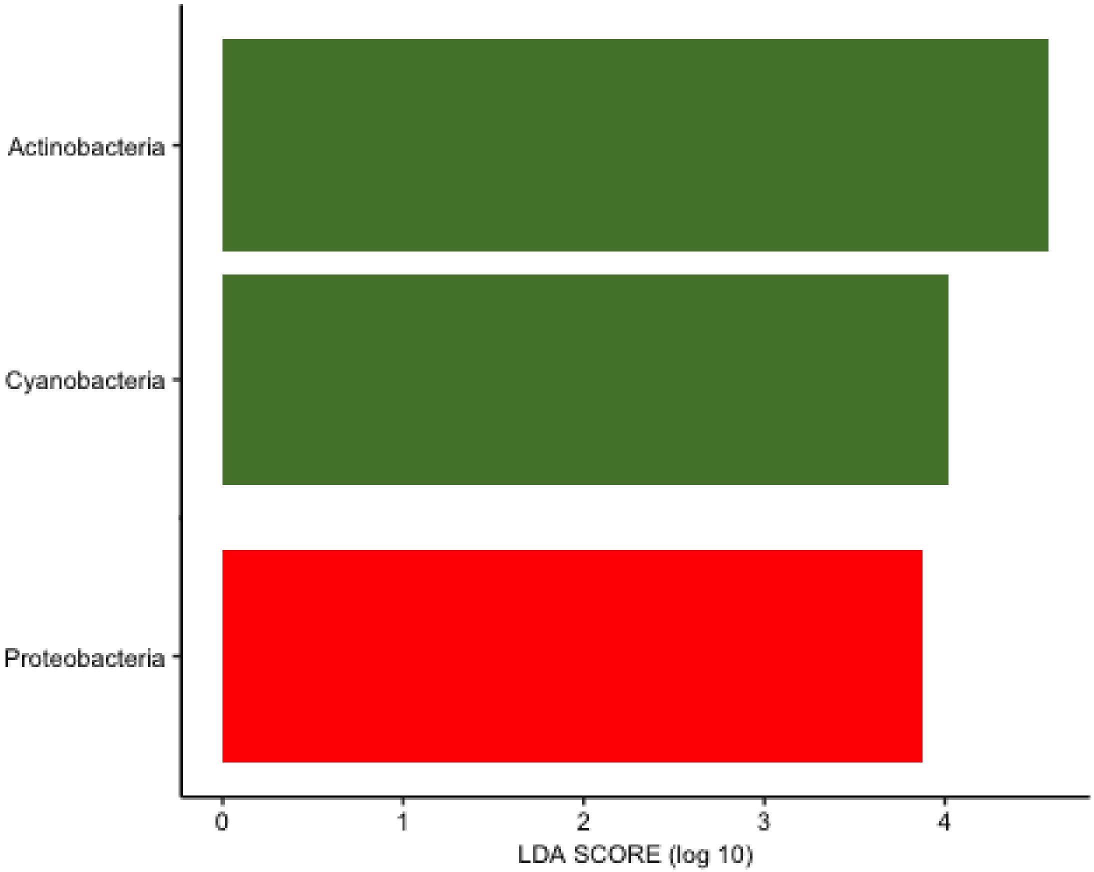
```
# Remove ASVs present less than 5% of the sample
physeq <- physeq_Ile %>%
  prune_taxa(rowSums(otu_table(physeq_Ile) != 0) >= 4, .) # 64*0.05
```

CSS normalization

```
#Normalize data using CSS method
# Convert phyloseq object to metageomeSeq object
metaSeq <- phyloseq_to_metagenomeSeq(physeq)
# Calculate pth quantile
p <- cumNormStatFast(metaSeq)

## Default value being used.

# CSS normalization
meta_sub <- cumNorm(metaSeq, p = p)
# Pull OTU table out of metagenomeSeq
otu_data <- as.data.frame(MRcounts(meta_sub, norm = T, log = F))
bacteria <- physeq
otu_table(bacteria) <- otu_table(otu_data, taxa_are_rows = T)
myTaxa = names(sort(taxa_sums(bacteria), decreasing = TRUE)[1:75])
bacteria = prune_taxa(myTaxa, bacteria)
#bacteria <- subset_samples(bacteria, SampleID != "IC.90" & SampleID !=
"ID15.701" & SampleID != "ID15.585" & SampleID != "ID75.574")
#bacteria <- bacteria %>%
#prune_taxa(taxa_sums(.) > 0, .)
```

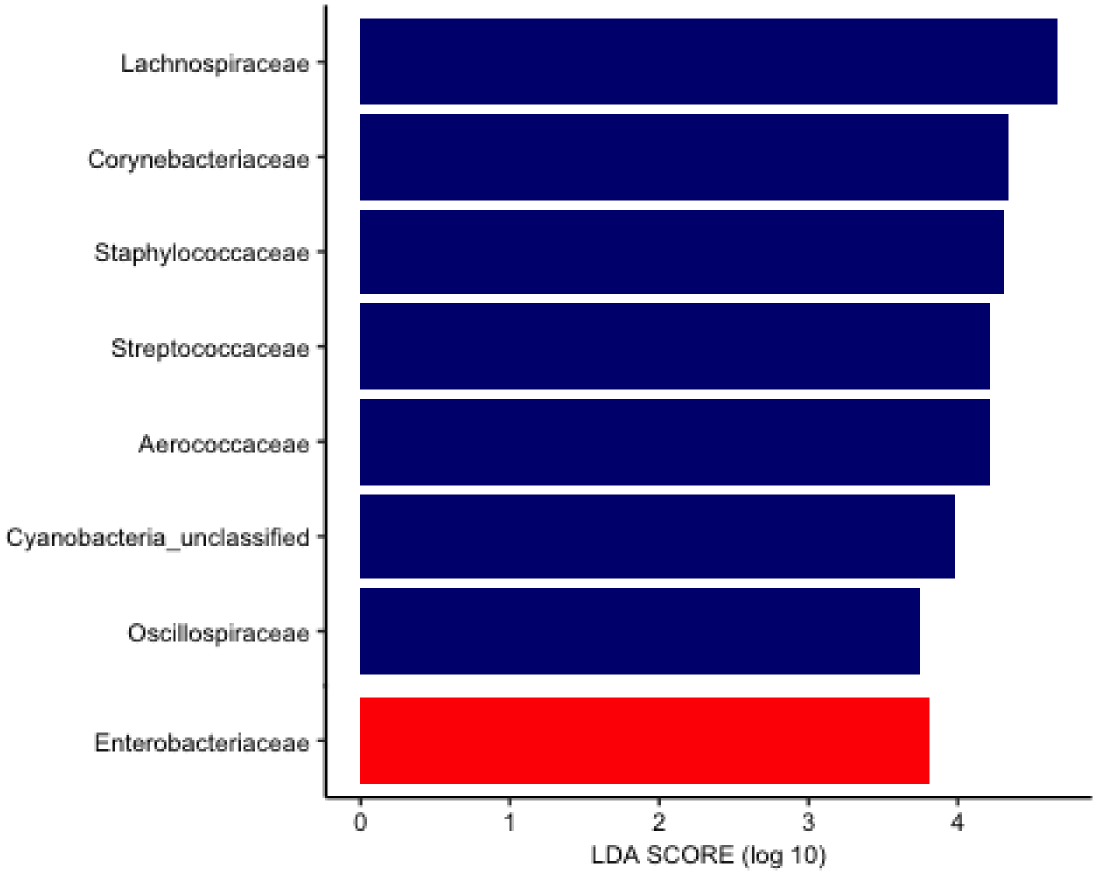
Lefse analysis-Mock/NE

```
#### Mock/NE #####
clr <-c("#00007E","red","#C0CF3AFF","#0D99B2","#7E007E","#F2C600","#00BF00")
bacteria_NC <-subset_samples(bacteria, Group %in% c("Mock","NE"))
bacteria_NC <- bacteria_NC %>%
prune_taxa(taxa_sums(.) > 0, .)

set.seed(1000)
lef_out<-run_lefse(bacteria_NC, group = "Group", norm = "CPM",
                   taxa_rank = "Phylum",
                   kw_cutoff = 0.05, lda_cutoff = 3)
dat <- marker_table(lef_out) %>% data.frame()

ggplot(dat,aes(ef_lda,reorder(feature, ef_lda))) +
  geom_col(aes(fill=enrich_group)) +
  facet_grid(rows = vars(enrich_group), scales = "free_y", space = "free_y")
+
   scale_fill_manual(name = "Day",values = c("#548236","red")) +
   #coord_flip() +
   labs(y=NULL, x="LDA SCORE (log 10)")+ theme_classic() +
theme(legend.position = "none", text = element_text(size = 8, family =
'Arial'),axis.text.x = element_text(size = 8,family = 'Arial', color =
"black"),axis.text.y = element_text(size = 8,family = 'Arial', color =
"black")) +
  theme(panel.spacing.y = unit(-0.1, "lines")) + theme(strip.text.y =
element_blank())
```



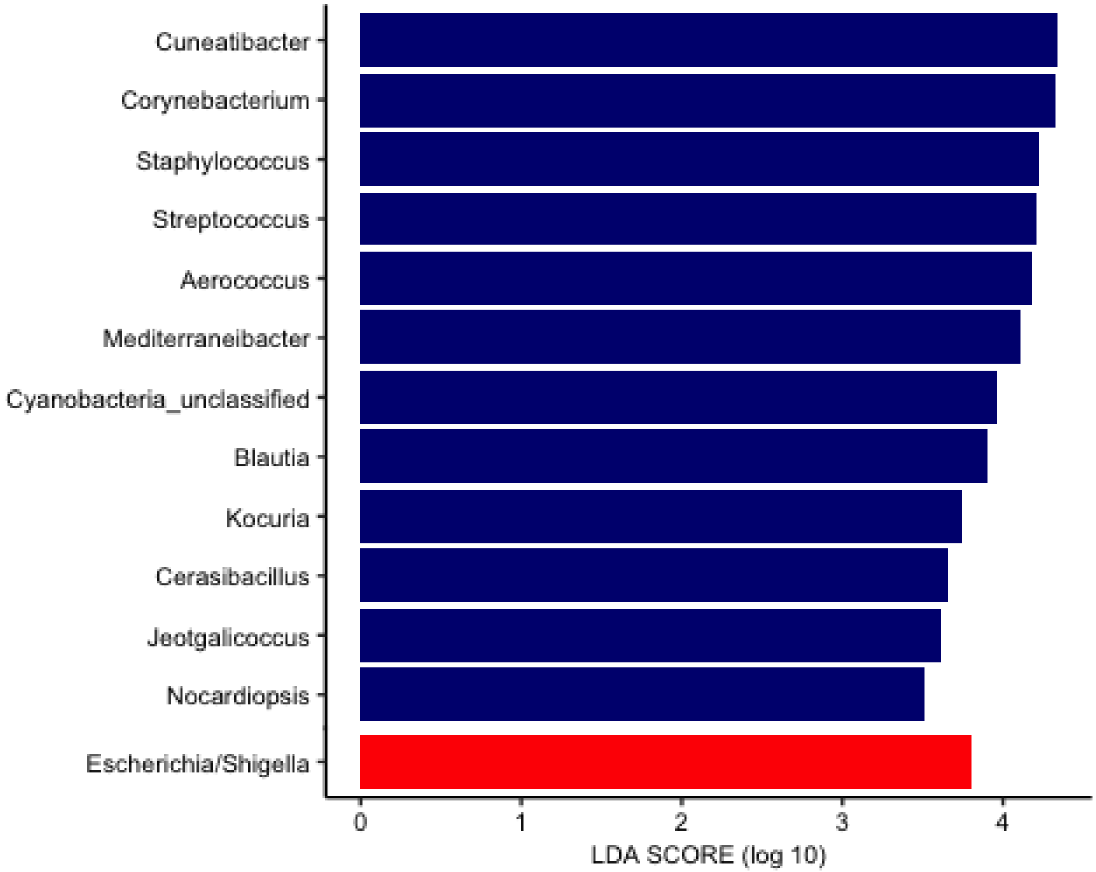
```
ggsave('Figures/Ileum/lefse/Mock_NE/phylum.png',
       height = 1.2,
       width = 2.5,
       unit = 'in',
       dpi = 300)

set.seed(1000)
gp = tax_glom(bacteria_NC, taxrank = "Family")
rb = transform_sample_counts(gp, function(x) {x/sum(x)} )
myTaxa = names(sort(taxa_sums(rb), decreasing = TRUE)[1:15])
bacteria_t = prune_taxa(myTaxa, gp)

lef_out<-run_lefse(bacteria_t, group = "Group", norm = "CPM",
                   taxa_rank = "Family",
                   kw_cutoff = 0.05, lda_cutoff = 3)
dat <- marker_table(lef_out) %>% data.frame()
```

```
ggplot(dat,aes(ef_lda,reorder(feature, ef_lda))) +
  geom_col(aes(fill=enrich_group)) +
  facet_grid(rows = vars(enrich_group), scales = "free_y", space = "free_y")
+
   scale_fill_manual(name = "Day",values = c("#00007E","red")) +
   #coord_flip() +
   labs(y=NULL, x="LDA SCORE (log 10)")+ theme_classic() +
theme(legend.position = "none", text = element_text(size = 8, family =
'Arial'),axis.text.x = element_text(size = 8,family = 'Arial', color =
"black"),axis.text.y = element_text(size = 8,family = 'Arial', color =
"black")) +
  theme(panel.spacing.y = unit(-0.1, "lines")) + theme(strip.text.y =
element_blank())
```

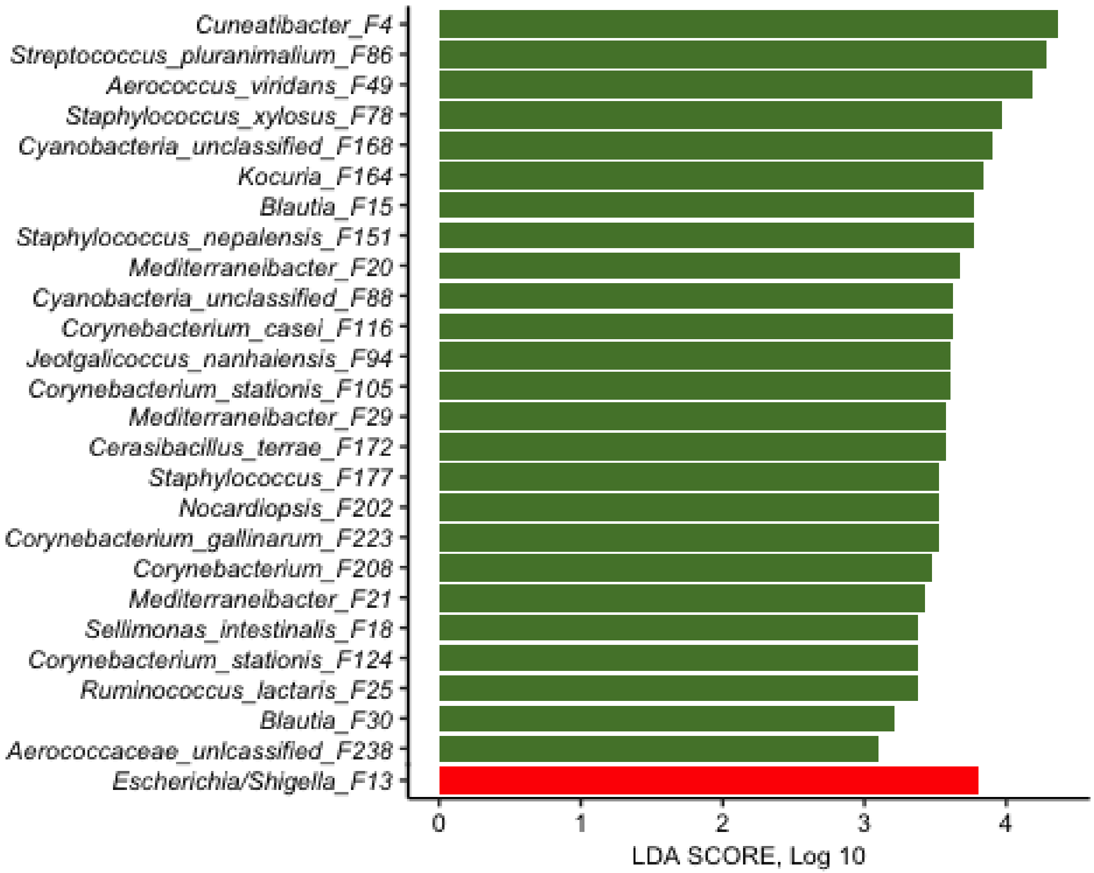


```
ggsave('Figures/Ileum/lefse/Mock_NE/Family.png',
       height = 1.65,
       width = 3.3,
       unit = 'in',
       dpi = 300)

set.seed(1000)
gp = tax_glom(bacteria_NC, taxrank = "Genus")
rb = transform_sample_counts(gp, function(x) {x/sum(x)} )
```

```r
myTaxa = names(sort(taxa_sums(rb), decreasing = TRUE)[1:25])
bacteria_t = prune_taxa(myTaxa, gp)

lef_out<-run_lefse(bacteria_t, group = "Group", norm = "CPM",
                   taxa_rank = "Genus",
                   kw_cutoff = 0.05, lda_cutoff = 3)
dat <- marker_table(lef_out) %>% data.frame()

ggplot(dat,aes(ef_lda,reorder(feature, ef_lda))) +
  geom_col(aes(fill=enrich_group)) +
  facet_grid(rows = vars(enrich_group), scales = "free_y", space = "free_y")
+
   scale_fill_manual(name = "Day",values = c("#00007E","red")) +
   #coord_flip() +
   labs(y= NULL, x="LDA SCORE (log 10)")+ theme_classic() +
theme(legend.position = "none", text = element_text(size = 8, family =
'Arial'),axis.text.x = element_text(size = 8,family = 'Arial', color =
"black"),axis.text.y = element_text(size = 8,family = 'Arial', color =
"black")) +
  theme(panel.spacing.y = unit(-0.1, "lines")) + theme(strip.text.y =
element_blank())
```



```
ggsave('Figures/Ileum/lefse/Mock_NE/Genus.png',
       height = 2.45,
       width = 3.35,
       unit = 'in',
       dpi = 300)

set.seed(1000)
gp = tax_glom(bacteria_NC, taxrank = "Species")
rb = transform_sample_counts(gp, function(x) {x/sum(x)} )
myTaxa = names(sort(taxa_sums(rb), decreasing = TRUE)[1:75])
bacteria_t = prune_taxa(myTaxa, gp)

lef_out<-run_lefse(bacteria_t, group = "Group", norm = "CPM",
                   taxa_rank = "Species",
                   kw_cutoff = 0.05, lda_cutoff = 3)

dat <- marker_table(lef_out) %>% data.frame()

ggplot(dat,aes(ef_lda,reorder(feature, ef_lda))) +
  geom_col(aes(fill=enrich_group)) +
  facet_grid(rows = vars(enrich_group), scales = "free_y", space = "free_y")
+
  scale_fill_manual(name = "Day",values = c("#548236","red")) +
   #coord_flip() +
   labs(y=NULL, x="LDA SCORE, Log 10")+ theme_classic() +
theme(legend.position = "none", text = element_text(size = 8, family =
'Arial'),axis.text.x = element_text(size = 8,family = 'Arial', color =
"black"),axis.text.y = element_text(size = 8,family = 'Arial', color =
"black",face ="italic")) +
  theme(panel.spacing.y = unit(-0.1, "lines")) + theme(strip.text.y =
element_blank())
```



```
ggsave('Figures/Ileum/lefse/Mock_NE/ASV.png',
       height = 4.5,
       width = 3.6,
       unit = 'in',
       dpi = 300)
```

Picrust2-Function Prediction

```
# Feature Table and rep-seqs
picrust2_pipeline.py -s QIIME2/data_filtered_rep_seqs/dna-sequences.fasta -i
QIIME2/feature_table/feature_table.txt -o picrust -p 16

# Add description
add_descriptions.py -i EC_metagenome_out/pred_metagenome_unstrat.tsv.gz -m EC
\
                    -o
EC_metagenome_out/pred_metagenome_unstrat_descrip.tsv.gz

add_descriptions.py -i KO_metagenome_out/pred_metagenome_unstrat.tsv.gz -m KO
\
                    -o
KO_metagenome_out/pred_metagenome_unstrat_descrip.tsv.gz
```

```
add_descriptions.py -i pathways_out/path_abun_unstrat.tsv.gz -m METACYC \
                    -o pathways_out/path_abun_unstrat_descrip.tsv.gz

# Add Pathway level 3
pathway_pipeline.py -i KO_metagenome_out/pred_metagenome_unstrat.tsv.gz -o
KEGG_pathways_out --no_regroup --map
KO_metagenome_out/KEGG_pathways_to_KO.txt

add_descriptions.py -i KEGG_pathways_out/path_abun_unstrat.tsv.gz \
                    -o KEGG_pathways_out/path_abun_unstrat_descrip.tsv.gz \
                    --custom_map_table
KO_metagenome_out/KEGG_pathways_info.tsv
```