Strawberry Trial

Maximilian Fernando Becker

2023-03-30

Bacterial community analysis

Becker *et al.* "Plant health protecting product application"

This document contains all statistical analyses conducted for the manuscript. Note that due to the random iterative nature of some analyses (such as PERMANOVA, CAP & PCA) some of the figure parameters will change slightly during reanalysis, though core results will remain essentially unchanged.

All data to reproduce analysis can be found here: `https://github.com/mfbeuq/becker_etal_PHPPapplication`

Load necessary ecological analysis libraries.

```
library(phyloseq)
library(microbiome)
library(ggord)
library(metagMisc)
library(ggpubr)
library(FSA)
library(knitr)
library(rmarkdown)
library(ape)
library(vegan)
library(philr)
library(compositions)
library(qiime2R)
library(plyr)
library(dplyr)
library(tidyr)
library(PMCMR)
library(tibble)
library(viridis)
library(gridExtra)
library(AcidPlots)
library(grid)
library(colorRamps)
```

```
library(rstatix)
library(dunn.test)
library(pairwiseAdonis)
library(dplyr)
library(ANCOMBC)
library(nlme)
library(tidyverse)
library(compositions)
library(readr)
library(DT)
library(matrixStats)
library(pheatmap)
library(RColorBrewer)
library(dendsort)
library(ComplexHeatmap)
library(circlize)
library(round)
library(lme4)
library(emmeans)
set.seed(225)
```

Contents of this workspace

```
load('concentration_trial.RData')
```

- `ps`: phyloseq object including the metadata, and the ASV table, the taxonomy table and the phylogenetic tree, all three created by QIIME2
- `psL.meta`: Shannon diversity index data of the loosely associated (L) root microbiota sub data set
- `psT.meta`: Shannon diversity index data of the tightly associated (T) root microbiota sub data set
- `RA.dist.L`: DEICODE distance matrix generated by QIIME2 of the loosely associated (L) root microbiota sub data set
- `RA.dist.T`: DEICODE distance matrix generated by QIIME2 of the loosely associated (T) root microbiota sub data set
- `RA.dist.L_FR`: DEICODE distance matrix generated by QIIME2 of the loosely associated (L) root microbiota of the fine roots (FR)
- `RA.dist.L_TR`: DEICODE distance matrix generated by QIIME2 of the loosely associated (L) root microbiota of the thick roots (TR)
- `RA.dist.T_FR`: DEICODE distance matrix generated by QIIME2 of the loosely associated (T) root microbiota of the fine roots (FR)
- `RA.dist.T_TR`: DEICODE distance matrix generated by QIIME2 of the tightly associated (T) root microbiota of the thick roots (TR)
- `RA.ord.all`: DEICODE ordination generated by QIIME2 of the loosely associated (L) root microbiota of the fine roots (FR)
- `RA.ord.L_FR`: DEICODE ordination generated by QIIME2 of the loosely associated (L) root microbiota of the fine roots (FR)
- `RA.ord.L_TR`: DEICODE ordination generated by QIIME2 of the loosely associated (L) root microbiota of the thick roots (TR)
- `RA.ord.T_FR`: DEICODE ordination generated by QIIME2 of the loosely associated (T) root microbiota of the fine roots (FR)
- `RA.ord.T_TR`: DEICODE ordination generated by QIIME2 of the tightly associated (T) root microbiota of the thick roots (TR)

- `q_meta`: the metadata table
- `deicode_theme`: a theme for generating the plots
- `plot_theme`: a theme for generating the plots
- `p.colors`: a color list for phyla coloring
- `read_distance`: function for importing a DEICODE distance matrix generated by QIIME2
- 'rename.ancombc.output": a function for renaming taxa in an ANCOM-BC table

and various lists for renaming factors

Alpha diversity statistics:

The following does the statistical analysis for the Shannon index values of the L-compartment

```
# Calculate Mean and SD
mean(psL.meta$Shannon)
sd(psL.meta$Shannon)

# GLM
Model <- aov(Shannon ~ Treatment * Rootpart + Treatment:Application + Treatment:Rootpart + Treatment:Ro

summary(Model)
anova(Model)

#Post Hoc
emmeans(Model, specs = pairwise ~ Rootpart)
emmeans(Model, specs = pairwise ~ Treatment:Application)
```

The following does the statistical analysis for the Shannon index values of the T-compartment

```
# Calculate Mean and SD
mean(psT.meta$Shannon)
sd(psT.meta$Shannon)

# GLM
Model <- aov(Shannon ~ Treatment * Rootpart + Treatment:Application + Treatment:Rootpart + Treatment:Ro

summary(Model)
anova(Model)

#Post Hoc
emmeans(Model, specs = pairwise ~ Rootpart)
emmeans(Model, specs = pairwise ~ Treatment:Application)
```

```
ps.meta$comp_rp <- paste0(ps.meta$Compartment, "-", ps.meta$Rootpart)
ps.meta$comp_rp <- factor(ps.meta$comp_rp, levels = c("Soil-Soil", "L-F", "L-T", "T-F", "T-T"))
p1 <- ggplot(ps.meta, aes(x=comp_rp, y=`shannon_entropy`, fill=Grouped)) +
```

```
  geom_boxplot() + deicode_theme + ylab("Shannon diversity index") + xlab("Compartment") +
  scale_fill_manual(values=c("gray30", group_col), name="Treatment", labels=c("Bulk soil", group_name))
  scale_x_discrete(labels=c("Soil-Soil" = "Bulk soil\n\nMean: 9.0\nSD: 0.1",
                            "L-F" = "Loosely - Fine roots\n\nMean: 7.7\nSD: 1.0",
                            "L-T" = "Loosely - Thick roots\n\nMean: 8.5\nSD: 0.3",
                            "T-F" = "Tightly - Fine roots\n\nMean: 7.4\nSD: 0.7",
                            "T-T" = "Tightly - Thick roots\n\nMean: 7.6\nSD: 0.3")) +
  theme(axis.title.x=element_blank(), axis.title.y=element_text(size=30), axis.text.y=element_text(size=
p1
```

Figure 1:

Beta Diversity:

Loosely associated root microbiome: First, the distance matrix calculated by QIIME2 needs to be merged with the metadata file:

```
RA.dist <- RA.dist.L
tmp_order <- colnames(RA.dist)
row.names(q_meta) <- q_meta$SampleID
de <- merge(q_meta, RA.dist, by=0, all=F, all.y=F)
row.names(de) <- de$SampleID
de <- de[ order(match(row.names(de), tmp_order)), ]
RA.dist <- RA.dist[ order(match(row.names(RA.dist), tmp_order)), ]
```

Then, PERMANOVA is calculated on the distance matrix

```
adonis2(formula = RA.dist ~ de$Rootpart * de$Grouped, permutations = 999)

# Pairwise adonis
pairwise.adonis(x=as.dist(RA.dist), factors=de[,"Grouped"], perm = 999, p.adjust.m = "fdr", reduce = "C
```

Furthermore, PERMDISP is calculated either for the PHPP treatment and the grouped variable

```
### Rootpart
mod <- betadisper(as.dist(RA.dist), de$Rootpart); anova(mod)
TukeyHSD(mod, which = "group", ordered = TRUE, conf.level = 0.95)

### Grouped
mod <- betadisper(as.dist(RA.dist), de$Grouped); anova(mod)
TukeyHSD(mod, which = "group", ordered = TRUE, conf.level = 0.95)
# extract only the Control comparisons and do BH correction by hand!
```

Tightly associated root microbiome: First, the distance matrix calculated by QIIME2 needs to be merged with the metadata file:

```
RA.dist <- RA.dist.T
tmp_order <- colnames(RA.dist)
row.names(q_meta) <- q_meta$SampleID
de <- merge(q_meta, RA.dist, by=0, all=F, all.y=F)
row.names(de) <- de$SampleID
de <- de[ order(match(row.names(de), tmp_order)), ]
RA.dist <- RA.dist[ order(match(row.names(RA.dist), tmp_order)), ]
```

Then, PERMANOVA is calculated on the distance matrix

```
adonis2(formula = RA.dist ~ de$Rootpart * de$Grouped, permutations = 999)

# Pairwise adonis
pairwise.adonis(x=as.dist(RA.dist), factors=de[,"Grouped"], perm = 999, p.adjust.m = "fdr", reduce = "C
```

Furthermore, PERMDISP is calculated either for the PHPP treatment and the grouped variable

```
### Rootpart
mod <- betadisper(as.dist(RA.dist), de$Rootpart); anova(mod)
TukeyHSD(mod, which = "group", ordered = TRUE, conf.level = 0.95)

### Grouped
mod <- betadisper(as.dist(RA.dist), de$Grouped); anova(mod)
TukeyHSD(mod, which = "group", ordered = TRUE, conf.level = 0.95)
# extract only the Control comparisons and do BH correction by hand!
```

```
# Panel A
RA.ord <- RA.ord.all
panelA <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, group_order)) %>%
  mutate(comp_rp = fct_relevel(comp_rp, c("B-B", "L-F", "L-T", "T-F", "T-T", "NC-NC", "Soil-Soil"))) %>%
  dplyr::filter(Grouped!='Soil') %>%
  ggplot(aes(x= PC1, y= PC2, color=Grouped, shape=comp_rp)) +
  geom_point(alpha=0.95, size=9, stroke=3) +
  deicode_theme +
  theme(legend.key.size = unit(1, "cm"), legend.text = element_text(size=28)) +
  guides(color = guide_legend(order = 1, override.aes = list(size=6)), #shape=c(rep(17,11),16
         shape = guide_legend(order = 2, override.aes = list(size=6))) +
  xlab(paste("PC1 - ",format(round(100*RA.ord$data$ProportionExplained[1,1],digits = 1), nsmall = 1),"%
  ylab(paste("PC2 - ",format(round(100*RA.ord$data$ProportionExplained[1,2],digits = 1), nsmall = 1),"%
  scale_shape_manual(values=c(1,16,15,17,18), name="Compartment",
                     labels = c("Bulk soil", "L - Fine roots", "L - Thick roots", "T - Fine roots", "T
  scale_color_manual(name="Treatment", values = c(group_col,"black"), aesthetics = c("colour", "fill"),
                     labels = group_name)
print(panelA)

##### Panel B ------------------------------------------------------------
group_col <- c("gray10","red1","purple1","orange1","paleturquoise3","green1")
```

```
group_name <- c("Control","Aliette","Luna","Movento", "Serenade","Bactiva")
app_name <- c("Control", "Recommended", "Double", "w felt", "w/o felt")
group_tmp <- c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE","BACTIVA")
app_tmp <- c("Control","Single","Double","Spross","Spross&Boden&noFelt")

### L: F ------------------------------------------------------
RA.ord <- RA.ord.L_FR
p1 <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Treatment = fct_relevel(Treatment, group_tmp)) %>%
  mutate(Application = fct_relevel(Application, app_tmp)) %>%
  ggplot(aes(x=PC1, y=PC2, color=Treatment, shape=Application)) +
  geom_point(alpha=0.9, size=10) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=6)),
         shape = guide_legend(order = 2, override.aes = list(size=6))) +
  xlab(paste("PC1 - ",format(round(100*RA.ord$data$ProportionExplained[1,1],digits = 1), nsmall = 1),"%
  ylab(paste("PC2 - ",format(round(100*RA.ord$data$ProportionExplained[1,2],digits = 1), nsmall = 1),"%
  scale_shape_manual(values=c(16,16,15,16,15), name="Application", labels=app_name) +
  scale_color_manual(name="Treatment", values =group_col, aesthetics = c("color"), labels=group_name) +
  theme(legend.key.height= unit(1.5, 'cm'))

### L: T ------------------------------------------------------
RA.ord <- RA.ord.L_TR
p2 <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Treatment = fct_relevel(Treatment, group_tmp)) %>%
  mutate(Application = fct_relevel(Application, app_tmp)) %>%
  ggplot(aes(x=PC1, y=PC2, color=Treatment, shape=Application)) +
  geom_point(alpha=0.9, size=10) + deicode_theme +
  guides(color = guide_legend(order = 2, override.aes = list(size=6)),
         shape = guide_legend(order = 1, override.aes = list(size=6))) +
  xlab(paste("PC1 - ",format(round(100*RA.ord$data$ProportionExplained[1,1],digits = 1), nsmall = 1),"%
  ylab(paste("PC2 - ",format(round(100*RA.ord$data$ProportionExplained[1,2],digits = 1), nsmall = 1),"%
  scale_shape_manual(values=c(16,16,15,16,15), name="Application", labels=app_name) +
  scale_color_manual(name="Treatment", values =group_col, aesthetics = c("color"), labels=group_name) +
  theme(legend.key.height= unit(1.5, 'cm'))

### T: F ------------------------------------------------------
RA.ord <- RA.ord.T_FR
p3 <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Treatment = fct_relevel(Treatment, group_tmp)) %>%
  mutate(Application = fct_relevel(Application, app_tmp)) %>%
  ggplot(aes(x=PC1, y=PC2, color=Treatment, shape=Application)) +
  geom_point(alpha=0.9, size=10) + deicode_theme +
  guides(color = guide_legend(order = 2, override.aes = list(size=6)),
         shape = guide_legend(order = 1, override.aes = list(size=6))) +
  xlab(paste("PC1 - ",format(round(100*RA.ord$data$ProportionExplained[1,1],digits = 1), nsmall = 1),"%
  ylab(paste("PC2 - ",format(round(100*RA.ord$data$ProportionExplained[1,2],digits = 1), nsmall = 1),"%
  scale_shape_manual(values=c(16,16,15,16,15), name="Application", labels=app_name) +
```

```
  scale_color_manual(name="Treatment", values =group_col, aesthetics = c("color"), labels=group_name) +
  theme(legend.key.height= unit(1.5, 'cm'))

### T: T ------------------------------------------------------
RA.ord <- RA.ord.T_TR
p4 <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Treatment = fct_relevel(Treatment, group_tmp)) %>%
  mutate(Application = fct_relevel(Application, app_tmp)) %>%
  ggplot(aes(x=PC1, y=PC2, color=Treatment, shape=Application)) +
  geom_point(alpha=0.9, size=10) + deicode_theme +
  guides(color = guide_legend(order = 2, override.aes = list(size=6)),
         shape = guide_legend(order = 1, override.aes = list(size=6))) +
  xlab(paste("PC1 - ",format(round(100*RA.ord$data$ProportionExplained[1,1],digits = 1), nsmall = 1),"%
  ylab(paste("PC2 - ",format(round(100*RA.ord$data$ProportionExplained[1,2],digits = 1), nsmall = 1),"%
  scale_shape_manual(values=c(16,16,15,16,15), name="Application", labels=app_name) +
  scale_color_manual(name="Treatment", values =group_col, aesthetics = c("color"), labels=group_name) +
  theme(legend.key.height= unit(1.5, 'cm'))

### Merge subplots ----------------------------------------------------------

legend2 <- cowplot::get_legend(p1)
d1 <- ggpar(p1, legend = "", legend.title = "", font.x = c(22, "bold"), font.y = c(22, "bold"), font.ti
d2 <- ggpar(p2, legend = "", legend.title = "", font.x = c(22, "bold"), font.y = c(22, "bold"), font.ti
d3 <- ggpar(p3, legend = "", legend.title = "", font.x = c(22, "bold"), font.y = c(22, "bold"), font.ti
d4 <- ggpar(p4, legend = "", legend.title = "", font.x = c(22, "bold"), font.y = c(22, "bold"), font.ti
figure <- ggarrange(d1, d2, NULL, NULL, NULL, legend2, d3, d4, NULL, widths = c(2.5,2.5,1), heights = c
panelB <- figure + theme(axis.text.x = element_text(face = "bold", size = 24, colour = "black"),
                axis.text.y = element_text(face = "bold", size = 24, colour = "black"),
                axis.title = element_text(face = "bold", size = 24, colour = "black"))
print(panelB)
```

Figure 4:

Phylogenetic Beta Diversity:

Figures S5 and S6: Load necessary packages and create theme for plots. Then, betaNTI and betaNRI, as well as tNST, are calculated for the compartments and timepoints individually.

```
library(microeco)
library(file2meco)
library(scales)
library(phyloseq)
library(magrittr)
library(ggplot2)
library(ggpubr)
library(dplyr)

dir.create("stat_results")
dir.create("stat_results/PD")
```

```r
# run analysis
for (comp in c("L","T")) {
  for (rs in c("F","T")) {

    # generate trans_nullmodel object
    pseq <- subset_samples(ps, Compartment==comp)
    pseq <- subset_samples(pseq, Rootpart==rs)

    dataset <- phyloseq2meco(pseq)
    dataset$tidy_dataset(); dataset

    # generate trans_nullmodel object
    t1 <- trans_nullmodel$new(dataset)


    #### beta NRI ------------------------------------------------------------
    method <- "bNRI"
    # see null.model parameter for other null models
    # null model run 500 times for the example
    t1$cal_ses_betampd(runs=999, abundance.weighted = TRUE)
    # return t1$res_ses_betampd

    # add betaNRI matrix to beta_diversity list
    dataset$beta_diversity[["betaNRI"]] <- t1$res_ses_betampd
    # create trans_beta class, use measure "betaNRI"
    t2 <- trans_beta$new(dataset = dataset, group = "Factor", measure = "betaNRI")
    # transform the distance for each group
    t2$cal_group_distance(); t2$res_group_distance

    # export data
    t2$res_group_distance %>%
      write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_rawdata_ssp.tsv"

    #calculate Mean and SD
    t2$res_group_distance %>% group_by(Factor) %>% summarise(Average=mean(Value), SD=sd(Value)) %>%
      write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_mean_ssp.tsv"), 

    # export ANOVA and post-hoc tests
    sink(file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_ANOVA_ssp.txt"))
    one.way <- aov(Value ~ Factor, data = t2$res_group_distance)
    agricolae::SNK.test(one.way, "Factor", alpha = 0.05, group=TRUE, main = NULL,console=TRUE)
    sink(file = NULL)


    # prepare the plot
    tmp <- t2$res_group_distance
    anova <- aov(Value ~ Factor, data = tmp)
    tukey <- TukeyHSD(anova)

    cld <- multcompView::multcompLetters4(anova, tukey)

    # table with factors and 3rd quantile
    dt <- group_by(tmp, Factor) %>%
```

```
  summarise(w=mean(Value), sd = sd(Value)) %>%
  arrange(desc(w))

# extracting the compact letter display and adding to the Tk table
cld <- as.data.frame.list(cld$Factor)
dt$cld <- cld$Letters
dt$Factor <- factor(dt$Factor, levels = c("Control", "Aliette_Single", "Aliette_Double", "Bactiva_S
tmp$Factor <- factor(tmp$Factor, levels = c("Control", "Aliette_Single", "Aliette_Double", "Bactiva_

# plot the results
plot = ggplot(tmp, aes(x=Factor, y=Value, fill=Factor)) +
  geom_boxplot() +
  geom_hline(yintercept = -2, linetype = 2) + geom_hline(yintercept = 2, linetype = 2) +
  scale_fill_manual(values=c("gray10", "red1", "red4","green1", "green4", "purple1", "purple4", "ora
  xlab("") + ylab("betaNRI") + theme(legend.position = "") +
  coord_cartesian(ylim = c(-6, 4)) +
  scale_y_continuous(breaks = seq(-6, 4, by = 2)) +
  plot_theme +
  geom_text(data = dt, aes(x = Factor, y = w, label = cld), size = 12, color = "black", hjust = 0.5
tiff(paste0("../Plots/PD/",method,"_",comp,"_",rs,"_factor_ssp.tiff"), units="in", width=8, height=
plot$layers[[2]]$aes_params$textsize <- 16
print(plot)
dev.off()


### beta NTI ----------------------------------------------------------
method <- "bNTI"
# null model run 500 times
t1$cal_ses_betamntd(runs=999, abundance.weighted = TRUE, nworker = 10, use_iCAMP_force = FALSE)
# add betaNRI matrix to beta_diversity list
dataset$beta_diversity[["betaNTI"]] <- t1$res_ses_betamntd
# create trans_beta class, use measure "betaNRI"
t2 <- trans_beta$new(dataset = dataset, group = "Factor", measure = "betaNTI")
# transform the distance for each group
t2$cal_group_distance(); t2$res_group_distance

# export data
t2$res_group_distance %>%
  write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_rawdata_ssp.tsv"

#calculate Mean and SD
t2$res_group_distance %>% group_by(Factor) %>% summarise(Average=mean(Value), SD=sd(Value)) %>%
  write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_mean_ssp.tsv"), 

# export ANOVA and post-hoc tests
sink(file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_factor_ANOVA_ssp.txt"))
one.way <- aov(Value ~ Factor, data = t2$res_group_distance); summary(one.way)
agricolae::SNK.test(one.way, "Factor", alpha = 0.05, group=TRUE, main = NULL,console=TRUE)
sink(file = NULL)

# prepare the plot
tmp <- t2$res_group_distance
anova <- aov(Value ~ Factor, data = tmp)
```

```
    tukey <- TukeyHSD(anova)

    cld <- multcompView::multcompLetters4(anova, tukey)

    # table with factors and 3rd quantile
    dt <- group_by(tmp, Factor) %>%
      summarise(w=mean(Value), sd = sd(Value)) %>%
      arrange(desc(w))

    # extracting the compact letter display and adding to the Tk table
    cld <- as.data.frame.list(cld$Factor)
    dt$cld <- cld$Letters
    dt$Factor <- factor(dt$Factor, levels = c("Control", "Aliette_Single", "Aliette_Double", "Bactiva_S
    tmp$Factor <- factor(tmp$Factor, levels = c("Control", "Aliette_Single", "Aliette_Double", "Bactiva_

    # plot the results
    plot = ggplot(tmp, aes(x=Factor, y=Value, fill=Factor)) +
      geom_boxplot() +
      geom_hline(yintercept = -2, linetype = 2) + geom_hline(yintercept = 2, linetype = 2) +
      scale_fill_manual(values=c("gray10", "red1", "red4","green1", "green4", "purple1", "purple4", "ora
      xlab("") + ylab("betaNTI") + theme(legend.position = "") +
      coord_cartesian(ylim = c(-6, 4)) +
      scale_y_continuous(breaks = seq(-6, 4, by = 2)) +
      plot_theme +
      geom_text(data = dt, aes(x = Factor, y = w, label = cld), size = 12, color = "black", hjust = 0.5
    plot$layers[[2]]$aes_params$textsize <- 16
    print(plot)
  }
}
```

Figure S9: tNST is calculated for each compartment and timepoint individually with both Bray's and Jaccard's distance.

```
for (comp in c("L","T")) {
  for (rs in c("F","T")) {
    for (dist in c("jaccard". "bray")) {

      method <- "NST"

      pseq <- subset_samples(ps, Compartment==comp)
      pseq <- subset_samples(pseq, Rootpart==rs)

      meta.df = as(sample_data(pseq), "matrix")
      meta.df = as.data.frame(meta.df)
      meta.df = meta.df[,"Factor", drop=FALSE]

      OTU1 = as(otu_table(pseq), "matrix")
      if(taxa_are_rows(pseq)){OTU1 <- t(OTU1)}
      comm.df = as.data.frame(OTU1)

      # calculate NST
      cal_nst <- NST::tNST(comm = comm.df, group= meta.df, rand = 999, nworker = 10, between.group = FA
```

```
      # bootstrap NST
      cal_nst.boot <- NST::nst.boot(nst.result = cal_nst, group = meta.df, rand = 999, trace = TRUE, two

      cal_nst.boot$compare[,-c(7,9:10)] %>% write.table( file = paste0("../stat_results/PD/",method,"_"

      # PERMANOVA
      cal_nst.permanova <- NST::nst.panova(nst.result = cal_nst, group = meta.df, rand = 999, trace=TRU
      cal_nst.permanova %>% write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_"

      tmp <- cal_nst.boot$detail$NST.boot
      df <- plyr::ldply (tmp,data.frame)

      # export data
      df %>% write.table( file = paste0("../stat_results/PD/",method,"_",comp,"_",rs,"_",dist,"_factor_

      df$.id <- factor(df$.id, levels = c("Control", "Aliette_Single", "Aliette_Double", "Bactiva_Single

      # plot
      p_tNSTI <- ggplot(df, aes(x=.id, y=X..i.., fill=.id)) +
        geom_boxplot() +
        xlab("") + ylab("tNST") +
        coord_cartesian(ylim = c(0, 1), expand = T) +
        plot_theme + theme(legend.position = "") +
        scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
        geom_hline(yintercept = 0, linetype = 1) + geom_hline(yintercept = 1, linetype = 2) +
        scale_fill_manual(values=c("gray10", "red1", "red4","green1", "green4", "purple1", "purple4", "
      p_tNSTI$layers[[2]]$aes_params$textsize <- 20
      print(p_tNSTI)
    }
  }
}
```

Differential abundance analysis:

```
library(dplyr)
library(ANCOMBC)
library(nlme)
library(tidyverse)
library(compositions)
library(readr)
library(DT)
library(matrixStats)
library(pheatmap)
library(RColorBrewer)
library(viridis)
library(dendsort)
library(qiime2R)
library(phyloseq)
library(microbiome)
library(ComplexHeatmap) #Heatmap function
library(circlize)
```

```
library(round)
library(tidyr)
library(readr)
library(tibble)

dir.create(paste0("stat_results/ANCOMBC/"))
dir.create(paste0("stat_results/ANCOMBC/pairwise"))
dir.create(paste0("stat_results/ANCOMBC/pairwise/",tax))
tax="genus"

# Treatment: calculate ------------------------------------------------------------------
for (comp in c("L","T")){

  for (rs in c("F","T")){

    pseq <- subset_samples(ps, Compartment==comp)
    dir.create(paste0("stat_results/ANCOMBC/pairwise/",tax,"/",rs))
    pseq <- subset_samples(pseq, Rootpart==rs)
    genus_data <- tax_glom(pseq, taxrank = rank_names(pseq)[6], NArm = FALSE)

    var1 <- "Factor"

    ## Control ------------------------------------------------------------------
    var2 <- "Control"

    sample_data(genus_data)$Treatment <- factor(sample_data(genus_data)$Treatment, levels =
                                                  c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE","BA

    sample_data(genus_data)$Factor <- factor(sample_data(genus_data)$Factor, levels =
                                                  c("Control","Aliette_Single","Aliette_Double","Luna_S
                                                    ))

    sample_data(genus_data)$Application <- as.factor(sample_data(genus_data)$Application)

    # Run ancombc function
    out = ANCOMBC::ancombc(phyloseq = genus_data, formula = "Factor",
                           p_adj_method = "holm", zero_cut = 0.90, lib_cut = 10000,
                           group = var1, struc_zero = FALSE, neg_lb = FALSE,
                           tol = 1e-5, max_iter = 100, conserve = TRUE,
                           alpha = 0.1, global = FALSE)
    res = out$res

    #Coefficients
    tab_coef = res$W
    colnames(tab_coef)
    col_name = c("Control - Aliette Single","Control - Aliette Double","Control - Luna Single","Control
    colnames(tab_coef) = col_name

    source("ancom_bc_rename_variables.R", echo = T, spaced = T)

    tmp <- tab_diff
    tmp2 <- merge(tmp, tab_w, by=0)
    write_csv(tmp2, file = paste0("stat_results/ANCOMBC/pairwise/",tax,"/",rs,"/",comp,"-",var2,".csv"))
```

```
  }
}

## Treatment Control : combine datasets ------------------------------------------------------

for (comp in c("L","T")){

  for (rs in c("F","T")){

    pseq <- subset_samples(ps, Compartment==comp)
    pseq <- subset_samples(pseq, Rootpart==rs)
    genus_data <- tax_glom(pseq, taxrank = rank_names(pseq)[6], NArm = FALSE)

    var1 <- "Factor"

    var2="Control";R1 <- read.csv(file = paste0("stat_results/ANCOMBC/pairwise/",tax,"/",rs,"/",comp,"-

    tmp <- Reduce(function(x, y) merge(x, y, all=TRUE, by="Row.names"), list(R1))
    rownames(tmp) <- tmp$Row.names
    tmp <- tmp[,-1]

    PGroup <- transform_sample_counts(genus_data, function(x)100* x / sum(x))
    OTUg <- otu_table(PGroup)
    AverageD <- as.data.frame(rowMeans(OTUg))
    names(AverageD) <- c("Mean")
    SD <- as.data.frame(rowSds(OTUg),na.rm = T)
    names(SD) <- c("SD")
    tmp_stat <- cbind(AverageD,SD)
    tmp5 <- merge(tmp, tmp_stat, by=0)

    write_csv(tmp5, file = paste0("stat_results/ANCOMBC/pairwise/",tax,"/",rs,"/",comp,"-",var1,"-",var

  }
}


# Make heatmap ---------------------------------

for (comp in c("L","T")){

  for (rs in c("F","T")){
    pseq <- subset_samples(ps, Compartment==comp)
    pseq <- subset_samples(pseq, Rootpart==rs)
    genus_data <- tax_glom(pseq, taxrank = rank_names(pseq)[6], NArm = FALSE)

    GTable <- read.csv(file = paste0("stat_results/ANCOMBC/pairwise/",tax,"/",rs,"/",comp,"-",var1,"-",
    ancom <- GTable[(GTable$Mean>=0.1),]

    tmp <- as.data.frame(tax_table(pseq)); tmp <- tmp[,-c(1,7:8)]
    tmp <- merge(tmp, ancom, by=0)
    rownames(tmp) <- tmp$Row.names; tmp$Row.names = NULL
    head(tmp)
```

```
    tmp$Phylum <- tmp$P
    ancom <- merge(tmp, p.colors, by="Phylum", all.y = F)
    ancom$G[ancom$G=="uncultured"] <- "Unclassified"
    ancom$G[is.na(ancom$G)] <- "Unclassified"
    ancom <- add_column(ancom, Name = ancom$G, .after = "G")
    ancom <- ancom[(rowSums(ancom[,c(8:17)])!=0),] %>% drop_na(P.y)
    head(ancom)
    ancom <- ancom[,-c(2,31)]
    colnames(ancom)[1:5] <- c("Phylum","Class","Order","Family","Genus")

    tax.clean <- ancom
    colnames(tax.clean)

    tax.clean <- rename.ancombc.output(ancom)


    #ancom <- ancom[order(ancom$Mean, decreasing = TRUE),]
    write_csv(tax.clean, file = paste0("stat_results/ANCOMBC/pairwise/subplots_combined_",tax,"_",var2,
    file.copy(from = paste0("stat_results/ANCOMBC/pairwise/subplots_combined_",tax,"_",var2,"_",comp,"-
              to = paste0("stat_results/ANCOMBC/pairwise/subplots_combined_",tax,"_",var2,"_",comp,"-",
    #redo in excel! -> add
  }
}


for (comp in c("L","T")){

  for (rs in c("F","T")){
mat.df <- read.csv(file = paste0("stat_results/ANCOMBC/pairwise/subplots_combined_",tax,"_",var2,"_",co
str(mat.df)

colnames(mat.df) <- c("Phylum", "Control -> Aliette Single SIG", "Control -> Aliette Double SIG", "Conti

#IMPORTANT: specify ONLY the columns with the differentials
col.order <- c("Control -> r.Aliette", "Control -> d.Aliette", "Control -> r.Luna", "Control -> d.Luna"
sig_mat <- mat.df[,c(2:11)] #TRUE/FALSE
mat.diff <- mat.df[,c(12:21)] #log fold changes

heatmap_function = function(num_matrix, omatrix , sig_matrix) {

  min_lc <- min(num_matrix, na.rm = T)
  max_lc <- max(num_matrix, na.rm = T)
  colors <- colorRamp2(c(min_lc, 0, max_lc), c("blue", "white", "red"))

  hb1 = rowAnnotation('L.F Mean\nAbundance [%]' = anno_barplot(as.vector(omatrix$Mean), gp = gpar(fill =

  lgd = Legend(at = c(round(min_lc+min_lc*0.05,digits = 1), 0, round(max_lc+max_lc*0.05, digits = 1)),

  ht = Heatmap(num_matrix , name = "logfold change",
               column_gap = unit(5, "mm"),
               row_split = omatrix$Phylum,
               column_split = c("A","A","B","B","C","C","D","D","E","E"),
               row_gap = unit(5, "mm"),
```

```
                row_names_gp = gpar(fontsize = 32, fontface = "bold",
                                    col = omatrix$`P.color`),
                na_col = "grey",
                column_order = col.order,
                col = colors,
                rect_gp = gpar(col = "white", lwd = 2), #white spaces in between
                border = TRUE,
                cluster_rows = FALSE, #remove cluster
                show_column_dend = FALSE, #remove cluster
                show_heatmap_legend = FALSE, #r emove legend
                row_names_side = "right",
                row_names_max_width = unit(2, "cm"),
                row_names_rot = 0,
                row_names_centered = FALSE,
                row_title_gp = gpar(fontsize = 28),
                row_title_rot = 0,
                column_title = paste0("ANCOM-BC W-value"),
                column_title_gp = gpar(fill = "black", col = "white", fontsize = 28, fontface = "bold", l
                column_title_side = "top",
                column_names_max_height = unit(6, "cm"),
                column_names_gp = gpar(fontsize = 30, fontface = "bold", col = "black"),
                column_names_rot = 90,
                column_names_centered = TRUE,
                show_parent_dend_line = FALSE,
                cell_fun = function(j, i, x, y, width, height, fill) {
                  if(sig_matrix[i, j] == "TRUE")
                    grid.text("*", x, y, gp = gpar(fontsize = 28, fontface = "bold"))
                },
                right_annotation = c(hb1))
  results <- c(fct = ht, lgd = lgd)
  return(results)
}

# all but proteos
j = 1; i = nrow(mat.diff)
full_map = heatmap_function(num_matrix = mat.diff[j:i,], omatrix = mat.df[j:i,], sig_matrix = sig_mat[j
draw(full_map$fct, padding = unit(c(9, 1, 1, 30), "cm")) # add space for titles
draw(full_map$lgd, x = unit(75, "cm"), y = unit(6, "cm"))
  }
}
```

Figure S12: