Temporal 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)
```

```r
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

```r
load('temporal_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.ord`: DEICODE ordination generated by QIIME2 of the entire root microbiota data set
- `RA.ord.LE`: DEICODE ordination generated by QIIME2 of the loosely associated (L) root microbiota sub data set sampled at the early timepoint
- `RA.ord.LL`: DEICODE ordination generated by QIIME2 of the loosely associated (L) root microbiota sub data set sampled at the late timepoint
- `RA.ord.TE`: DEICODE ordination generated by QIIME2 of the loosely associated (T) root microbiota sub data set sampled at the early timepoint
- `RA.ord.TL`: DEICODE ordination generated by QIIME2 of the tightly associated (T) root microbiota sub data set sampled at the late timepoint
- `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 tightly associated (T) root microbiota sub data set
- `RA.dist.LE`: DEICODE distance matrix generated by QIIME2 of the loosely associated (L) root microbiota sub data set sampled at the early timepoint
- `RA.dist.LL`: DEICODE distance matrix generated by QIIME2 of the tightly associated (L) root microbiota sub data set sampled at the late timepoint
- `RA.dist.TE`: DEICODE distance matrix generated by QIIME2 of the loosely associated (T) root microbiota sub data set sampled at the early timepoint
- `RA.dist.TL`: DEICODE distance matrix generated by QIIME2 of the tightly associated (T) root microbiota sub data set sampled at the late timepoint
- `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

---

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)

# Linear Model
Model <- lm(Shannon ~ PHPP * timepoint, data = psL.meta, na.action = na.omit)
summary(Model)
anova(Model)
```

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)

# Linear Model
Model <- lm(Shannon ~ PHPP * timepoint, data = psT.meta, na.action = na.omit)
summary(Model)
anova(Model)
```

Timepoints “F” and “S” are one and two weeks after final application, respectively.

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 #using a temporary variable
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$PHPP * de$Timepoint , permutations = 999)
```

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

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

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

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

```
RA.dist <- RA.dist.T #using a temporary variable
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$PHPP * de$Timepoint , permutations = 999)
```

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

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

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

```


RA.ord <- RA.ord
pall <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL-F","CONTROL-S",
                              "ALIETTE-F","ALIETTE-S",
                              "LUNA-F","LUNA-S",
                              "MOVENTO-F","MOVENTO-S",
                              "SERENADE-F","SERENADE-S")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped, shape=Compartment)) +
  geom_point(alpha=0.9, size=12) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=12))) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  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(16,17,15), name="Compartment") +
  theme(legend.text.align = 0, legend.spacing.y = unit(1, 'cm'), legend.key.height = unit(1, 'cm')) +
  scale_color_manual(values=c("black", "blue4","red1", "red4", "purple1", "purple4","orange1", "orange3
                              "paleturquoise1", "paleturquoise4"), name="PHPP",
                     labels = c("Control - First TP","Control - Second TP",
                                "Aliette - First TP","Aliette - Second TP",
                                "Luna - First TP","Luna - Second TP",
                                "Movento - First TP","Movento - Second TP",
                                "Serenade - First TP","Serenade - Second TP"))
print(pall)
```

Figure 3 A:

```
## PCoA L-Early samples ---------------------------------------
RA.ord <- RA.ord.LE
p1E <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped)) +
  geom_point(alpha=0.9, size=12) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=12))) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  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),"%
  theme(legend.text.align = 0, legend.spacing.y = unit(1, 'cm'), legend.key.height = unit(1, 'cm')) +
  scale_color_manual(values=c("black", "red1", "purple1","orange1", "paleturquoise3"), name="PHPP",
                     labels = c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE"))


## PCoA L-Late samples ---------------------------------------
RA.ord <- RA.ord.LL
p1L <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped)) +
  geom_point(alpha=0.9, size=12) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=12))) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  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),"%
  theme(legend.text.align = 0, legend.spacing.y = unit(1, 'cm'), legend.key.height = unit(1, 'cm')) +
  scale_color_manual(values=c("black", "red1", "purple1","orange1", "paleturquoise3"), name="PHPP",
                     labels = c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE"))
```

```r
## PCoA T-Early samples ---------------------------------------
RA.ord <- RA.ord.TE
p2E <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped)) +
  geom_point(alpha=0.9, size=12) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=12))) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  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),"%
  theme(legend.text.align = 0, legend.spacing.y = unit(1, 'cm'), legend.key.height = unit(1, 'cm')) +
  scale_color_manual(values=c("black", "red1", "purple1","orange1", "paleturquoise3"), name="PHPP",
                     labels = c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE"))


## PCoA T-Late samples ---------------------------------------
RA.ord <- RA.ord.TL
p2L <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped)) +
  geom_point(alpha=0.9, size=12) + deicode_theme +
  guides(color = guide_legend(order = 1, override.aes = list(size=12))) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  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),"%
  theme(legend.text.align = 0, legend.spacing.y = unit(1, 'cm'), legend.key.height = unit(1, 'cm')) +
  scale_color_manual(values=c("black", "red1", "purple1","orange1", "paleturquoise3"), name="PHPP",
                     labels = c("CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE"))


# Create legend
RA.ord <- RA.ord.TL
legend <- RA.ord$data$Vectors %>%
  dplyr::select(SampleID, PC1, PC2) %>%
  left_join(q_meta) %>%
  mutate(Grouped = fct_relevel(Grouped, "CONTROL","ALIETTE","LUNA","MOVENTO","SERENADE")) %>%
  ggplot(aes(x=PC1, y=PC2, color=Grouped)) +
  scale_color_manual(values=c("black", "red1", "purple1","orange1", "paleturquoise3"), name="PHPP",
                     labels = c("Control","Aliette","Luna","Movento","Serenade"))  +
  deicode_theme + theme(legend.key.size = unit(1.5, "cm"), legend.text = element_text(size=28),
)

legend2 <- cowplot::get_legend(legend)
c1 <- ggpar(p1E, legend = "", legend.title = "", font.x = c(24, "bold"), font.y = c(24, "bold"), font.t
  annotate(geom="text", y = max(p1E$data$PC2), x = max(p1E$data$PC1), label="(1)", color="black", size=
  scale_y_continuous(labels = scales::number_format(accuracy = 0.1)) + scale_x_continuous(labels = scal
c2 <- ggpar(p1L, legend = "", legend.title = "", font.x = c(24, "bold"), font.y = c(24, "bold"), font.t
  annotate(geom="text", y = max(p1L$data$PC2), x = max(p1L$data$PC1), label="(2)", color="black", size=
  scale_y_continuous(labels = scales::number_format(accuracy = 0.1)) + scale_x_continuous(labels = scal
c3 <- ggpar(p2E, legend = "", legend.title = "", font.x = c(24, "bold"), font.y = c(24, "bold"), font.t
  annotate(geom="text", y = max(p2E$data$PC2), x = max(p2E$data$PC1), label="(3)", color="black", size=
  scale_y_continuous(labels = scales::number_format(accuracy = 0.1)) + scale_x_continuous(labels = scal
c4 <- ggpar(p2L, legend = "", legend.title = "", font.x = c(24, "bold"), font.y = c(24, "bold"), font.t
  annotate(geom="text", y = max(p2L$data$PC2), x = max(p2L$data$PC1), label="(4)", color="black", size=
  scale_y_continuous(labels = scales::number_format(accuracy = 0.1)) + scale_x_continuous(labels = scal
figure <- ggarrange(c1, c2, NULL,
                    NULL, NULL, legend2,
                    c3, c4, NULL, widths = c(2.5,2.5,1.5), heights = c(2.5,0.1,2.5),
                    ncol = 3, nrow = 3)
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"))
```

Figure 3 B:

Phylogenetic Beta Diversity:

Figures S3 and S4: 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")


### ind. timepoints - ssp-----------------------------------------------------

# same species pool
comp <- "T"
tp <- "F"

for (comp in c("L","T")) {
  for (tp in c("F","S")) {


    pseq <- subset_samples(ps, compartment==comp)
    pseq <- subset_samples(pseq, timepoint==tp)

    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$new(dataset = dataset, group = "PHPP", measure = "betaNRI")
# transform the distance for each group
t2$cal_group_distance(); t2$res_group_distance

# export data
t2$res_group_distance %>%
  write.table( file = paste0("stat_results/PD/",method,"_",comp,"_",tp,"_rawdata_ssp.tsv"), row.name

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

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

##### **Figure S4:**
p_bNRI <- ggplot(t2$res_group_distance, aes(x=PHPP, y=Value, fill=PHPP)) +
  geom_boxplot() +
  geom_hline(yintercept = -2, linetype = 2) + geom_hline(yintercept = 2, linetype = 2) +
  scale_fill_manual(values=c("red1", "black", "purple1","orange1", "paleturquoise3"), name="Treatmen
                     labels = c("ALIETTE","CONTROL","LUNA","MOVENTO","SERENADE")) +
  xlab("") + ylab("betaNRI") +
  coord_cartesian(ylim = c(-8, 4)) +
  scale_y_continuous(breaks = seq(-8, 4, by = 2)) +
  plot_theme
p_bNRI$layers[[2]]$aes_params$textsize <- 20
print(p_bNRI)

### beta NTI ----------------------------------------------------------
method <- "bNTI"
# null model run 500 times
t1$cal_ses_betamntd(runs=999, abundance.weighted = TRUE, nworker = 10, use_iCAMP = FALSE)
# return t1$res_ses_betamntd
# 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$new(dataset = dataset, group = "PHPP", measure = "betaNTI")
    # transform the distance for each group
    t2$cal_group_distance(); t2$res_group_distance

    # export data
    t2$res_group_distance %>%
      write.table( file = paste0("stat_results/PD/",method,"_",comp,"_",tp,"_rawdata_ssp.tsv"), row.nam

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

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

    ##### **Figure S3:**
    p_bNTI <- ggplot(t2$res_group_distance, aes(x=PHPP, y=Value, fill=PHPP)) +
      geom_boxplot() +
      geom_hline(yintercept = -2, linetype = 2) + geom_hline(yintercept = 2, linetype = 2) +
      scale_fill_manual(values=c("red1", "black", "purple1","orange1", "paleturquoise3"), name="Treatmen
                         labels = c("ALIETTE","CONTROL","LUNA","MOVENTO","SERENADE")) +
      xlab("") + ylab("betaNTI") +
      coord_cartesian(ylim = c(-8, 4)) +
      scale_y_continuous(breaks = seq(-8, 4, by = 2)) +
      plot_theme
    p_bNTI$layers[[2]]$aes_params$textsize <- 16
    print(p_bNTI)
  }
}
```

Figure S8: tNST is calculated for each compartment and timepoint individually with both Bray's and Jaccard's distance.

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

      method <- "NST"

      pseq <- subset_samples(ps, Compartment==comp)
      pseq <- subset_samples(pseq, timepoint==tp)

      meta.df = as(sample_data(pseq), "matrix")
      meta.df = as.data.frame(meta.df)
      meta.df = meta.df[,"PHPP", 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 = TRUE, output.rand = TRUE, dist.method = dist)

      # bootstrap NST
      cal_nst.boot <- NST::nst.boot(nst.result = cal_nst, group = meta.df, rand = 999, trace = TRUE,
                               two.tail = TRUE, out.detail = TRUE, between.group = FALSE,
                               nworker = 10, SES = FALSE, RC = FALSE)

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

      # PERMANOVA
      cal_nst.permanova <- NST::nst.panova(nst.result = cal_nst, group = meta.df, rand = 999,
                                           trace=TRUE, SES=FALSE, RC=FALSE)
      cal_nst.permanova %>% write.table( file = paste0("stat_results/PD/",method,"_",comp,"_",tp,"_",di

      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,"_",tp,"_",dist,"_rawdata.ts

      # 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 +
        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("red1", "black", "purple1","orange1", "paleturquoise3"), name="Treat
                           labels = c("ALIETTE","CONTROL","LUNA","MOVENTO","SERENADE"))
      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)
library(circlize)
library(round)
library(tidyr)
library(tibble)

tax <- "genus"
dir.create("stat_results/ANCOMBC/")
dir.create("stat_results/ANCOMBC/genus")

# Timepoints individually: -----------------------------------------------------------------
for (comp in c("L","T")){

  pseq <- subset_samples(ps, Compartment==comp)
  genus_data <- tax_glom(pseq, taxrank <- rank_names(pseq)[6], NArm = FALSE)
  var1 <- "PSM_TP" #is equal to the grouped variable
  tax="genus"

  ## to control early --------------------------------------------------------------------
  var2="Early"
  sample_data(genus_data)$PSM_TP <- factor(sample_data(genus_data)$PSM_TP, levels = c("CONTROL-F","CONTI
                                                                                      "ALIETTE-F","ALIET
                                                                                      "LUNA-F","LUNA-S"
                                                                                      "MOVENTO-F","MOVEI
                                                                                      "SERENADE-F","SERI

  # Run ancombc function
  out = ancombc(phyloseq = genus_data, formula = "PSM_TP",
                p_adj_method = "holm", zero_cut = 0.90, lib_cut = 10000,
                group = var1, struc_zero = TRUE, 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_E - Control_L",
               "Control_E - Aliette_E", "Control_E - Aliette_L",
               "Control_E - Luna_E", "Control_E - Luna_L",
               "Control_E - Movento_E", "Control_E - Movento_L",
               "Control_E - Serenade_E", "Control_E - Serenade_L")
  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/",tax,"/",comp,"-",var2,".csv"))
  ## to control late --------------------------------------------------------------------
  var2="Late"
  sample_data(genus_data)$PSM_TP <- factor(sample_data(genus_data)$PSM_TP, levels = c("CONTROL-S","CONTR
                                                                                      "ALIETTE-F","ALIET
                                                                                      "LUNA-F","LUNA-S"
                                                                                      "MOVENTO-F","MOVEN
                                                                                      "SERENADE-F","SERE

  # Run ancombc function
  out = ancombc(phyloseq = genus_data, formula = "PSM_TP",
                p_adj_method = "holm", zero_cut = 0.90, lib_cut = 10000,
                group = var1, struc_zero = TRUE, 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_L - Control_E",
               "Control_L - Aliette_E", "Control_L - Aliette_L",
               "Control_L - Luna_E", "Control_L - Luna_L",
               "Control_L - Movento_E", "Control_L - Movento_L",
               "Control_L - Serenade_E", "Control_L - Serenade_L")
  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/",tax,"/",comp,"-",var2,".csv"))

}


# Timepoints individually: combine datasets ---------------------------------------------------

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

  var1 <- "PSM_TP"

  var2="Early";R1 <- read.csv(file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"-",var2,".csv"))[,c(1
  var2="Late";R2 <- read.csv(file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"-",var2,".csv"))[,c(1,4
  tmp <- Reduce(function(x, y) merge(x, y, all=TRUE, by="Row.names"), list(R1,R2))
  rownames(tmp) <- tmp$Row.names; tmp <- tmp[,-1]

  pseq <- subset_samples(ps, compartment==comp)
  pseq <- subset_samples(pseq, timepoint=="F")
  genus_data <- tax_glom(pseq, taxrank <- rank_names(pseq)[6], NArm = FALSE)
  PGroup <- transform_sample_counts(genus_data, function(x)100* x / sum(x))
  OTUg <- otu_table(PGroup)
  AverageD <- as.data.frame(rowMeans(OTUg))
  names(AverageD) <- "Early Mean"
  SD <- as.data.frame(rowSds(OTUg),na.rm = T)
  names(SD) <- "Early SD"
  tmp_early <- cbind(AverageD,SD)

  pseq <- subset_samples(ps, compartment==comp)
  pseq <- subset_samples(pseq, timepoint=="S")
  genus_data <- tax_glom(pseq, taxrank <- rank_names(pseq)[6], NArm = FALSE)
  PGroup <- transform_sample_counts(genus_data, function(x)100* x / sum(x))
  OTUg <- otu_table(PGroup)
  AverageD <- as.data.frame(rowMeans(OTUg))
  names(AverageD) <- "Late Mean"
  SD <- as.data.frame(rowSds(OTUg),na.rm = T)
  names(SD) <- "Late SD"
  tmp_late <- cbind(AverageD,SD)

  tmp4 <- merge(tmp_early,tmp_late, by=0, all=TRUE)
  rownames(tmp4) <- tmp4$Row.names; tmp4 <- tmp4[,-1]
  tmp5 <- merge(tmp,tmp4, by=0, all=TRUE)

  #export as .csv
  write_csv(tmp5, file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"-",var1,"_results.csv"))
}

# Treatment at individual timepoints --------------------------------
for (comp in c("L","T")){

  ###ANCOM results
  tmp <- read.csv(file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"-",var1,"_results.csv"))

  rownames(tmp) <- tmp$Row.names; tmp$Row.names = NULL
  pseq <- subset_samples(ps, compartment==comp)
  genus_data <- tax_glom(pseq, taxrank <- rank_names(pseq)[6], NArm = FALSE)

  ttable <- as.data.frame(tax_table(genus_data))
  GTable <- merge(ttable,tmp, by=0)
  rownames(GTable) <- GTable$Row.names; GTable$Row.names = NULL
  head(GTable)

  ancom <- GTable[(GTable$Late.Mean>=0.1 | GTable$Early.Mean>=0.1),] %>% drop_na(P)

  ancom <- ancom[,-c(1,7:8)]
  ancom$Phylum <- ancom$P
  ancom <- merge(ancom, 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(8:12,18:21)])!=0),] %>% drop_na(Phylum)
  head(ancom)
  ancom <- ancom[,-c(2)]
  colnames(ancom)[1:5] <- c("Phylum","Class","Order","Family","Genus")

  tax.clean <- ancom
  colnames(tax.clean)
  tax.clean <- rename.ancombc.output(ancom)
  write_csv(tax.clean, file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"_subplots_combined.csv"))
  file.copy(from = paste0("stat_results/ANCOMBC/",tax,"/",comp,"_subplots_combined.csv"), to = paste0("
  #redo in excel! -> add specific row names if necessary
}


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

  mat.df <- read.csv(file = paste0("stat_results/ANCOMBC/",tax,"/",comp,"_subplots_combined_mod.csv"),
  str(mat.df)

  colnames(mat.df) <- c("Phylum","Control.Early:Late.sig","E:Control:Aliette.sig","E:Control:Luna.sig",

  #IMPORTANT: specify ONLY the columns with the differentials
  col.order <- c("Control Early -> Late","Early: Control -> Aliette", "Early: Control -> Luna","Early: 
  mat.diff <- as.matrix(mat.df[,c(7:11,16:19)])
  sig_mat <- as.matrix(mat.df[,c(2:6,12:15)])
  #rownames(mat.diff) <- mat.df$Taxonomy
  min_lc <- min(mat.diff, na.rm = T)
  max_lc <- max(mat.diff, na.rm = T)
  colors <- colorRamp2(c(min_lc, 0, max_lc), c("blue", "white", "red"))

  hb = rowAnnotation('Mean Abundance [%]' = anno_barplot(as.vector(mat.df$`Early 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(mat.diff , name = "logfold change",
               column_gap = unit(5, "mm"),
               row_split = mat.df$Phylum,
               row_gap = unit(5, "mm"),
               row_names_gp = gpar(fontsize = 32, fontface = "bold",
                                   col = mat.df$`P-color`),
               column_split = c("A","B","B","B","B","C","C","C","C"),
               na_col = "grey",
               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", 
               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 = FALSE,
               column_order = col.order,
               show_parent_dend_line = FALSE,
               cell_fun = function(j, i, x, y, width, height, fill) {
                 if(sig_mat[i, j] == "TRUE")
                   grid.text("*", x, y, gp = gpar(fontsize = 28, fontface = "bold"))
               },
               right_annotation = c(hb))
  draw(ht, padding = unit(c(10, 1, 1, 26), "cm")) # add space for titles
  draw(lgd, x = unit(45, "cm"), y = unit(8, "cm"))
}
```

Figure S11: