01 Community description patterns

Angel Rain-Franco

01/03/2024

Contents

1 Setting up the workspace

1.1 Loading Packages

```
rm(list = ls())
library(cowplot)
library(vegan)
library(egg)
library(readxl)
library(FactoMineR)
library(factoextra)
library(RColorBrewer)  #Expand color palette
library(dplyr)
library(stringr)  # For editing string
library(reshape2)  #For 'melt' function
library(ggsignif)  #Significance values for ggplot
library(viridis)  #Nice green and yellow color palette
library(rstatix)
library(olsrr)
library(nlme)
library(pheatmap)
library(NST)
library(tinytex)
library(kableExtra)
library(ggnewscale)
library(ggvenn)  # Venn diagram
```

1.2 Loading colorblind palette

```
# Setting the colorblind palette
cbp1 <- c("#D55E00", "#999999", "#0072B2", "#E69F00", "#56B4E9",
    "#FFDB6D", "#009E73", "#CC79A7", "#293352", "#F0E442", "#ff0000")

# Expanding the standard palette Set3 for downstream
# analysis
mycolors2 <- colorRampPalette(brewer.pal(12, "Set3"))(12)
```

1.3 Load OTU table and metadata

```
## Load relative reads number matrix
df1 = data.frame(read.csv("../data/OTU_table_merged200_SILVA_megablast.csv",
    row.names = 1))
## Load metadata
meta = data.frame(read.csv("../data/metadata_merged200_SILVA_megablast.csv",
    row.names = 1))

# Formatting metadata
meta$time = as.factor(meta$Cycle)
```

```
levels(meta$time) = c(0, 1, 4, 6, 7)  #Cycle number
meta$time = as.numeric(as.character(meta$time))
meta$Replicate = as.factor(sprintf("%02d", meta$Replicate))  # Add Microcosm ID

tibble(meta)
```

```
## # A tibble: 100 x 4
##    Cycle Replicate M.ID   time
##    <chr> <fct>     <chr> <dbl>
##  1 C0    01        C01       0
##  2 C0    10        C010      0
##  3 C0    11        C011      0
##  4 C0    12        C012      0
##  5 C0    13        C013      0
##  6 C0    14        C014      0
##  7 C0    15        C015      0
##  8 C0    16        C016      0
##  9 C0    17        C017      0
## 10 C0    18        C018      0
## # i 90 more rows
```

2 Community patterns

2.1 Description of microcosms community composition at Genus level

```
# Overview at Genus level
df2 <- aggregate(. ~ Genus, data = df1[, c(6, 8:107)], sum, na.rm = TRUE)

genus <- df2[, 1]  #Create a vector with Genus
rownames(df2) <- df2[, 1]  #Add it as rownames
df2 <- df2[, -1]  #remove column with orders and keep only abundance data
# New object with the aggregated reads
agg = df2

# Change format of agg matrix to dataframe
agg <- as.data.frame(agg)
rownames(agg) <- genus  #add genus information
agg$Sum.agg <- rowSums(agg)  # add column with counts across treatments for all genera

# Summarizing the most abundant genera Get the more
# abundant genera at Cycle 7
agg$Sum.agg10T <- rowSums(agg[, meta$M.ID[meta$Cycle == "C7"]])
## Get the more abundant genera from Cycle 0 to Cycle 6
agg$Sum.agg10C <- rowSums(agg[, meta$M.ID[meta$Cycle != "C7"]])

# select the most abundant genera from the full experiment
agg10T <- agg[with(agg, order(-Sum.agg10T)), ][1:11, 1:100]
agg10C <- agg[with(agg, order(-Sum.agg10C)), ][1:20, 1:100]

# Combine both dataframes avoiding duplicates
```

```
duprows <- which(!is.na(match(rownames(agg10C), rownames(agg10T))))
agg10combined <- rbind(agg10T, agg10C[-duprows, ])

agg10combined$genus <- rownames(agg10combined)

# convert from wide to long format
agg10.long <- melt(agg10combined, id.vars = "genus", variable.name = "treat")
agg10.long$M.ID = str_remove(agg10.long$treat, "V1.")

# Combine metadata with results
agg10.long = merge(agg10.long, meta, by = "M.ID")

# Reformat Replicate with 02 digits
agg10.long$Replicate = sprintf("%02d", as.numeric(agg10.long$Replicate))
agg10.long$Replicate = as.factor(agg10.long$Replicate)
```

```
## Set Genus as factor
agg10.long$order = factor(agg10.long$genus, row.names(agg10combined))

# Reorder factor levels according C6 microcosms
agg10.long$Replicate = factor(agg10.long$Replicate, c(c("04",
    "05", "08", "11", "12", "16", "17", "10", "18"), c("01",
    "03", "07"), c("02", "13", "14"), c("06", "09", "15", "19",
    "20")))

# Classify them
agg10.long$Class = "NA"
agg10.long$Class[which(agg10.long$order %in% rownames(agg10T))] = "Top10"
agg10.long$Class[(agg10.long$Class) == "NA"] = "Others"
# set column Class as factor
agg10.long$Class <- as.factor(agg10.long$Class)

tibble(agg10.long)
```

```
## # A tibble: 2,100 x 9
##    M.ID  genus            treat    value Cycle Replicate  time order       Class
##    <chr> <chr>            <fct>    <dbl> <chr> <fct>     <dbl> <fct>       <fct>
##  1 C01   Pseudomonas      C01   0        C0    01            0 Pseudomonas Top10
##  2 C01   Acidovorax       C01   0.0405   C0    01            0 Acidovorax  Top10
##  3 C01   Aeromonas        C01   0.695    C0    01            0 Aeromonas   Top10
##  4 C01   f_Comamonadaceae C01   0.00756  C0    01            0 f_Comamona~ Top10
##  5 C01   Pelomonas        C01   0        C0    01            0 Pelomonas   Top10
##  6 C01   Bosea            C01   0        C0    01            0 Bosea       Top10
##  7 C01   Cellvibrio       C01   0        C0    01            0 Cellvibrio  Top10
##  8 C01   Ottowia          C01   0.000344 C0    01            0 Ottowia     Top10
##  9 C01   Rhodoferax       C01   0.0113   C0    01            0 Rhodoferax  Top10
## 10 C01   Ramlibacter      C01   0.00550  C0    01            0 Ramlibacter Top10
## # i 2,090 more rows
```

```
# loop to plot microbial relative abundance per cycle

list.plots <- list()
cycles.index <- c("C0", "C1", "C4", "C6", "C7")
```

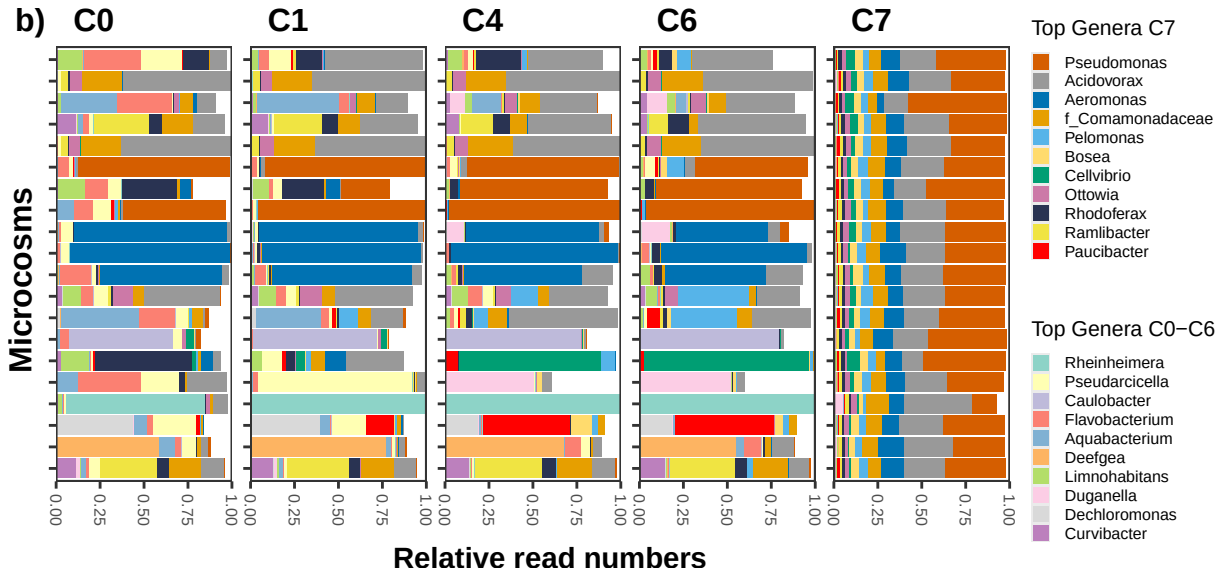
```
for (i in 1:length(cycles.index)) {
    list.plots[[i]] <- ggplot(agg10.long %>%
        subset(Cycle == cycles.index[i]), aes(x = Replicate,
        y = value, fill = order)) + geom_bar(stat = "identity") +
        scale_fill_manual(values = c(cbp1, mycolors2)) + scale_y_continuous(limits = c(0,
        1), expand = c(0, 0)) + guides(fill = guide_legend(override.aes = list(size = 1))) +
        theme(legend.text = element_text(size = 5)) + guides(fill = guide_legend(ncol = 1)) +
        labs(title = "", x = NULL, y = "") + my_theme + coord_flip() +
        theme(legend.position = "none")
}

# Temporal plot to create legends
plot_tmp <- ggplot() + geom_bar(data = agg10.long %>%
    subset(Cycle == "C7"), aes(x = Replicate, y = value, fill = order),
    stat = "identity") + scale_fill_manual(name = "Top Genera C7",
    values = c(cbp1, mycolors2), breaks = levels(agg10.long$order)[1:11]) +
    new_scale_fill() + geom_bar(data = agg10.long %>%
    subset(Cycle == "C7"), aes(x = Replicate, y = value, fill = order),
    stat = "identity") + scale_fill_manual(name = "Top Genera C0-C6",
    values = c(mycolors2, mycolors2), breaks = levels(agg10.long$order)[12:22]) +
    theme(legend.key.width = unit(0.25, "cm"), legend.key.height = unit(0.25,
        "cm"), legend.key = element_rect(fill = NULL, colour = NULL),
        legend.text = element_text(size = 6), legend.title = element_text(size = 8)) +
    guides(fill = guide_legend(ncol = 1))

# Extract legend
legend.plot <- get_legend(plot_tmp)
rm(plot_tmp)
```

2.2 Relative read number reads over the experiment (Figure 1b ms)

Proportions of reads of 16S rRNA genes affiliated with the most abundant genera in metagenomes from the experimental communities at C0, C1, C4, C6 and C7.

```
## pdf
##   2
```

```
plot_grid(plot_grid(NULL, list.plots[[1]], list.plots[[2]], list.plots[[3]],
    list.plots[[4]], list.plots[[5]], rel_widths = c(0.12, 0.45,
        0.45, 0.45, 0.45, 0.45), labels = c("b)", "C0", "C1",
        "C4", "C6", "C7"), vjust = 1.7, hjust = -0.6, label_size = 12,
    nrow = 1), legend.plot, nrow = 1, rel_widths = c(1, 0.2),
    align = "top") + draw_label("Relative read numbers", x = 0.45,
    y = 0, vjust = 0, angle = 0, fontface = "bold", size = 11) +
    draw_label("Microcosms", x = 0, y = 0.5, vjust = 1.5, angle = 90,
        fontface = "bold", size = 12)
```



2.3 Get communities with common and abundant genera

#Common dominant taxa are here operationally defined as the genus with highest relative abundance in a community and high co-occurrency (>3) following Avolio et al., 2019.

```
df.order <- df2[order(rowSums(df2[, 61:80]), decreasing = TRUE),
    61:80]
df.dominance <- data.frame(sample.ID = colnames(df.order), most_abundant = rownames(df.order)[max.col(t
tibble(df.dominance)
```

```
## # A tibble: 20 x 2
##    sample.ID most_abundant
##    <chr>     <chr>
##  1 C61       Aeromonas
##  2 C610      Pelomonas
##  3 C611      Rheinheimera
##  4 C612      Duganella
##  5 C613      Pseudomonas
##  6 C614      Pseudomonas
##  7 C615      Acidovorax
##  8 C616      Cellvibrio
##  9 C617      Caulobacter
## 10 C618      Pelomonas
## 11 C619      Acidovorax
## 12 C62       Pseudomonas
## 13 C620      Acidovorax
## 14 C63       Aeromonas
## 15 C64       Ramlibacter
## 16 C65       Deefgea
## 17 C66       Acidovorax
## 18 C67       Aeromonas
## 19 C68       Paucibacter
## 20 C69       Acidovorax
```

2.4 Patterns of diversity

2.4.1 Alpha and beta diversity over the incubation cycles

```
### Shannon diversity index
meta$shannon = vegan::diversity(t(df1[, 8:107]))
### Richness
meta$Richness = colSums(df1[, 8:107] != 0)

## Evenness (Pielou)
meta$Even = meta$shannon/log(meta$Richness)
tibble(meta)
```

```
## # A tibble: 100 x 7
##    Cycle Replicate M.ID   time shannon Richness  Even
##    <chr> <fct>     <chr> <dbl>   <dbl>    <dbl> <dbl>
##  1 C0    01        C01       0    2.97      116 0.626
##  2 C0    10        C010      0    3.57      176 0.690
##  3 C0    11        C011      0    2.71       72 0.633
##  4 C0    12        C012      0    3.47      125 0.719
##  5 C0    13        C013      0    3.43      107 0.735
##  6 C0    14        C014      0    3.06      113 0.648
##  7 C0    15        C015      0    3.82      149 0.763
##  8 C0    16        C016      0    3.02      122 0.629
##  9 C0    17        C017      0    3.27      123 0.680
## 10 C0    18        C018      0    3.18       85 0.716
## # i 90 more rows
```

```
### Richness across microcosms
df.rich <- data.frame(Cycle = c("C0", "C1", "C4", "C6", "C7"),
    nOTUS = c(sum(rowSums(df1[, 8:27] != 0) > 0), sum(rowSums(df1[,
        28:47] != 0) > 0), sum(rowSums(df1[, 48:67] != 0) > 0),
        sum(rowSums(df1[, 68:87] != 0) > 0), sum(rowSums(df1[,
            88:107] != 0) > 0)))

tibble(df.rich)
```

```
## # A tibble: 5 x 2
##   Cycle nOTUS
##   <chr> <int>
## 1 C0      722
## 2 C1      654
## 3 C4      709
## 4 C6      737
## 5 C7      469
```

2.4.2 Calculation Raup–Crick (bRC) index

```
dist.method = "bray"  # Select distance method
```

```
comm <- df1[, 8:107]
TaxonID <- row.names(comm)
comm <- comm[-1, ]
meta <- meta[order(meta$M.ID), ]
meta.groupi <- data.frame(cluster = meta[, c("cluster")])
row.names(meta.groupi) <- meta$M.ID

tnst = tNST(comm = t(comm), meta.com = NULL, group = meta.groupi,
    dist.method = "bray", abundance.weighted = TRUE, rand = 1000,
    output.rand = TRUE, nworker = 8, LB = FALSE, null.model = "PF",
    between.group = TRUE, SES = TRUE, RC = TRUE)
```

```
## All match very well.

## Now randomizing by parallel computing. Begin at Mon May 20 20:15:00 2024. Please wait...
```

Organize Raup-Crick (bRC) data

```
nst.temp <- tnst[["index.pair.grp"]]
nst.temp$Cycle1 <- str_sub(nst.temp$name1, 1, 2)
nst.temp$Cycle2 <- str_sub(nst.temp$name2, 1, 2)
nst.temp$Micro1 <- str_sub(nst.temp$name1, 3, 5)
nst.temp$Micro2 <- str_sub(nst.temp$name2, 3, 5)
# Rename groups
levels(nst.temp$group) <- c("Acidovorax-type", "Aeromonas-type",
    "Pseudomonas-type", "Others")
nst.temp$group <- factor(nst.temp$group, c("Acidovorax-type",
    "Pseudomonas-type", "Aeromonas-type", "Others"))
```

2.4.3 Prepare RC figures

```
set.seed(666)
plot_RC <- nst.temp[nst.temp$Cycle1 == nst.temp$Cycle2 & nst.temp$group !=
    "Others" & nst.temp$Cycle1 != "C7", ] %>%
    ggplot(aes(x = Cycle1, y = RC.ij.bray)) + geom_violin(aes(fill = group),
    alpha = 0.3, size = 0.5) + geom_jitter(aes(fill = group),
    width = 0.2, alpha = 0.7, size = 1.5, shape = 21) + geom_hline(yintercept = c(-0.95,
    0.95), linetype = "dashed") + theme_bw() + theme(panel.grid.minor = element_blank(),
    panel.grid.major = element_blank()) + labs(y = expression(beta *
    "RC"), x = "Cycle") + ylim(-1.05, 1.1) + scale_fill_manual(values = c("#999999",
    "#D55E00", "#0072B2")) + theme(plot.margin = unit(c(0.1,
    0.2, 0, 0), "cm")) + theme(plot.title = element_text(vjust = -1,
    size = 8), text = element_text(size = 8.5), axis.title.y = element_text(size = 8.5,
    vjust = -2)) + facet_wrap(~group, nrow = 1) + theme(legend.position = "none")
plot_RC <- tag_facet(plot_RC, tag_pool = c("Acidovorax-type",
    "Pseudomonas-type", "Aeromonas-type"), hjust = -0.1, open = "",
    close = "", fontface = 3, size = 2.8)
```

2.4.4 Calculate proportions of pairwise bRC

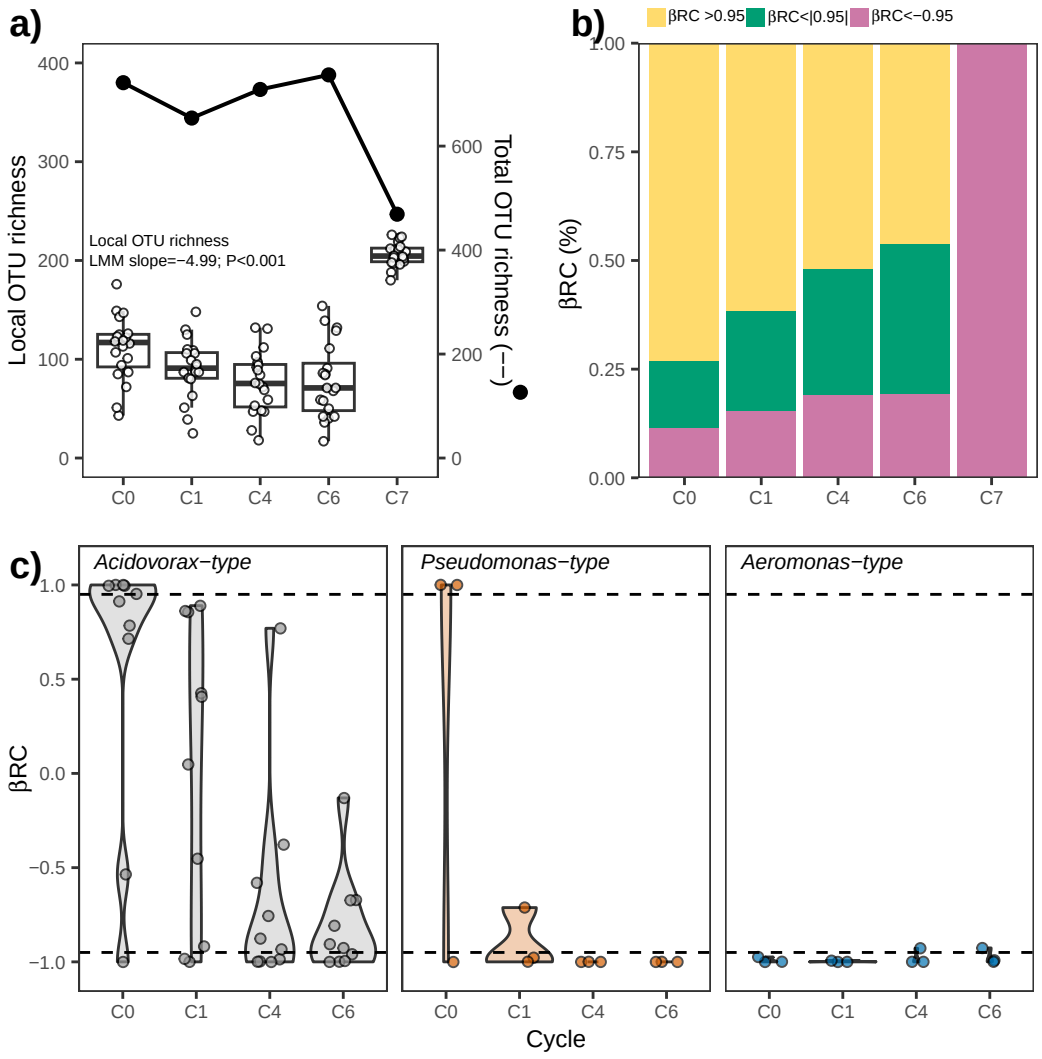
```
data.bRC = nst.temp[nst.temp$Cycle1 == nst.temp$Cycle2, ]
data.bRC$category <- "bRC<|0.95|"
data.bRC$category[data.bRC$RC.ij.bray > 0.95] <- "bRC>0.95"
data.bRC$category[data.bRC$RC.ij.bray < (-0.95)] <- "bRC<-0.95"

agg.bRC <- aggregate(RC.ij.bray ~ category + Cycle1, data = data.bRC,
    length)
agg.bRC$category <- as.factor(agg.bRC$category)  #Set it as factor
agg.bRC$category <- factor(agg.bRC$category, c("bRC>0.95", "bRC<|0.95|",
    "bRC<-0.95"))  #Reorder factor levels

agg.bRC$norm.count <- agg.bRC$RC.ij.bray/52

prop.bRC <- agg.bRC %>%
    ggplot(aes(x = Cycle1, y = norm.count, fill = category)) +
    geom_bar(stat = "identity") + scale_fill_manual(values = c(cbp1[6:9]),
    name = "", labels = c(expression(beta * "RC >0.95"), expression(beta *
        "RC<|0.95|"), expression(beta * "RC<-0.95"))) + scale_y_continuous(limits = c(0,
    1), expand = c(0, 0)) + labs(title = "", x = NULL, y = expression(beta *
    "RC (%) ")) + theme_bw() + theme(text = element_text(size = 8.5),
    axis.title.y = element_text(size = 8.5, vjust = -0.05)) +
    theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
        legend.position = c(0.35, 1.06), legend.direction = "horizontal",
        legend.title = element_blank(), legend.text = element_text(size = 5.5,
            margin = margin(0, -0.2, 0, 0.01, unit = "cm")),
        legend.margin = margin(-10, -20, -10, -10), legend.key.size = unit(0.3,
            "cm")) + theme(plot.margin = unit(c(0.1, 0.15, 0.1,
    0), "cm"))
```

```
## Warning: A numeric ‘legend.position‘ argument in ‘theme()‘ was deprecated in ggplot2
## 3.5.0.
## i Please use the ‘legend.position.inside‘ argument of ‘theme()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

2.4.5 Export Alpha and beta-diversity (Figure 2 ms)

OTU richness (a) per microcosms (left axis) and total OTU richness across microcosms (right axis) over cycles. b) Percentage of Raup–Crick (bRC) pairwise dissimilarity distances among all microcosms. c) Per identified community-common genera cluster during cycles.

```
## pdf
##   2
```



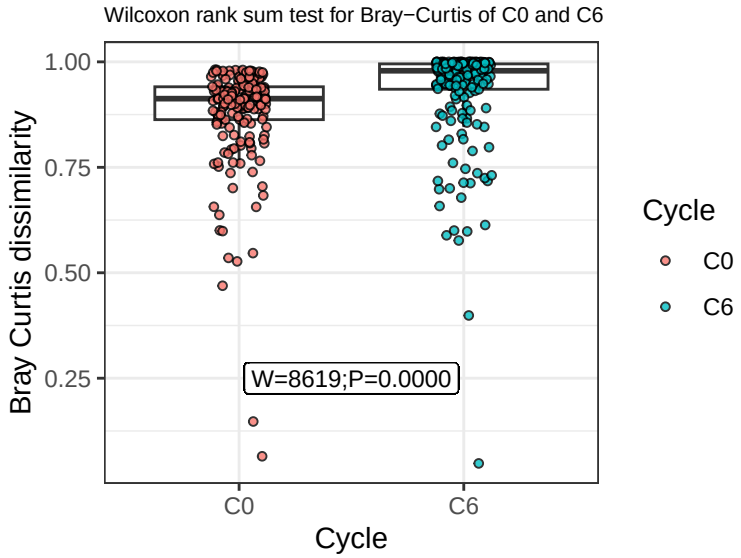
2.5 Compare pairwise compositional dissimilarity (Bray Curtis-based) between C0 and C6

```
# calculate the Bray-Curtis dissimilarity matrix on the C0:
ps.rel = df1[, meta$M.ID[meta$Cycle == "C0"]]
colnames(ps.rel) = meta$Replicate[meta$Cycle == "C0"]
data.dist0 <- vegdist(as.matrix(t(ps.rel)), method = "bray",
    binary = F)

# Calculate the Bray-Curtis dissimilarity matrix on the C6:
ps.rel = df1[, meta$M.ID[meta$Cycle == "C6"]]
colnames(ps.rel) = meta$Replicate[meta$Cycle == "C6"]
data.dist6 <- vegdist(as.matrix(t(ps.rel)), method = "bray",
    binary = F)

# Mantel test relationship between C0 and C6
# dissimilarities
```

```
set.seed(666)
mantel(data.dist0, data.dist6, permutations = 1000)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = data.dist0, ydis = data.dist6, permutations = 1000)
##
## Mantel statistic r: 0.5542
##       Significance: 0.000999
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.105 0.140 0.189 0.252
## Permutation: free
## Number of permutations: 1000
```

2.5.1 Comparison Bray Curtis dissimilarity at C0 and C6 cycles



3 Description of community types

3.1 Veen diagram

```
# Subset data from C6
df.C6 <- df2[, c(61:80)]

# Genus taxonomic classification
my_genus_col = data.frame(df1[, c(5, 6)])
my_genus_col <- unique(my_genus_col)
```

```
row.names(my_genus_col) <- my_genus_col$Genus

# Remove not present taxa
df.C6 <- df.C6[rowSums(df.C6) > 0, ]  # Reduced from 119 to 85 Genera

# Average by community type
acido = rowMeans(df.C6[, meta$M.ID[meta$Cycle == "C6" & meta$cluster ==
    "Acidovorax"]])
pseu = rowMeans(df.C6[, meta$M.ID[meta$Cycle == "C6" & meta$cluster ==
    "Pseudomonas"]])
aero = rowMeans(df.C6[, meta$M.ID[meta$Cycle == "C6" & meta$cluster ==
    "Aeromonas"]])
othe = rowMeans(df.C6[, meta$M.ID[meta$Cycle == "C6" & meta$cluster ==
    "Others"]])
df.simple <- data.frame(acido, pseu, aero, othe)

# Combining samples per cluster Get only data from Cycle 6
# for Venn diagram and Heatmap
df.simple.cluster <- df.simple[, c("acido", "pseu", "aero")]

# Remove the uniques uniques taxa from the. 'Others
# communities
df.simple.cluster <- df.simple.cluster[rowSums(df.simple.cluster) >
    0, ]
df.simple.cluster <- as.data.frame(df.simple.cluster > 0)
names(df.simple.cluster) <- c("Acidovorax-type", "Pseudomonas-type",
    "Aeromonas-type")

# veen diagram plot (save for later)
venn_plot <- ggvenn(df.simple.cluster, fill_color = c("#868686FF",
    "#fc6f00", "#0073C2FF", "#f0f2f2"), stroke_size = 0.25, set_name_size = 1.7,
    show_percentage = F, text_size = 2)
```

3.2 Heatmap Cycle 6 community types

```
# remove genera from others
df.simple.hm <- df.simple[, c("acido", "pseu", "aero")]
# Remove the unique taxa from the. 'Others communities
df.simple.hm <- df.simple.hm[rowSums(df.simple.hm) > 0, ]

# Add classification
col.for.labels <- data.frame(genus = row.names(df.simple.hm))
col.for.labels$classification = NA
col.for.labels$classification[(df.simple.hm[, "acido"] > 0)] = "Acidovorax"
col.for.labels$classification[(df.simple.hm[, "pseu"] > 0)] = "Pseudomonas"
col.for.labels$classification[(df.simple.hm[, "aero"] > 0)] = "Aeromonas"
col.for.labels$classification[rowSums(df.simple.hm[, 1:3] > 0) >
    1] = "Shared"
col.for.labels$classification[rowSums(df.simple.hm[, 1:3] > 0) ==
    3] = "Widespread"
df.simple.hm.log10 <- log10(df.simple.hm + 1e-04)
```

```
# Rename Allorhizobium-Neorhizobium-Pararhizobium-Rhizobium
# for ANPR
row.names(df.simple.log10)[27] <- "ANPR"
col.for.labels$genus[27] <- "ANPR"

names(df.simple.log10) <- c("Acidovorax-type", "Pseudomonas-type",
    "Aeromonas-type")
## create the breaks to state 0 as white
bk2 = seq(-4, -0.09119, length = 70)

# set different color vectors for each interval
col1 = "white"  #set the order of greys
col2 = viridis(69)
colors2 <- c(col1, col2)

my_heatmap <- pheatmap::pheatmap(t(df.simple.log10), color = colors2,
    breaks = bk2, annotation_colors = NA, angle_col = 45, fontsize = 4.7,
    legend = TRUE, cluster_rows = F, cluster_cols = F, cellwidth = 5.4,
    cellheight = 7, margin = c(0, 0, 0, 0), labels_col = paste0("f",
        c(1:70)))
```

```
# Add color by community type
my_heatmap[["gtable"]][["grobs"]][[2]][["label"]] <- row.names(df.simple.log10)
cols = col.for.labels[order(match(rownames(df.simple.log10),
    my_heatmap$gtable$grobs[[2]]$label)), ]$colors
my_heatmap$gtable$grobs[[2]]$gp = gpar(col = cols)
```

3.3 Compare dominant genus from the community clusters

```
df2 <- aggregate(. ~ Genus, data = df1[, c(6, 8:107)], sum, na.rm = TRUE)

df.long <- melt(df2, id.vars = "Genus", variable.name = "M.ID")

# Include cluster information to the long format taxa table
df.long <- merge(df.long, meta, by = "M.ID")

agg.cluster <- aggregate(value ~ Genus + cluster + Cycle, data = df.long,
    mean)

## Remove the 'Others' communities
agg.cluster <- agg.cluster[agg.cluster$cluster != "Others", ]
agg.cluster <- droplevels(agg.cluster)
agg.cluster$Genus <- as.factor(agg.cluster$Genus)

# plot figure
agg.plot <- agg.cluster[agg.cluster$Cycle %in% c("C0", "C1",
    "C4", "C6") & agg.cluster$Genus %in% c("Acidovorax", "Aeromonas",
    "Pseudomonas"), ] %>%
    ggplot(aes(x = Cycle, y = (value) * 100, fill = (Genus))) +
```

```
    geom_bar(position = "stack", stat = "identity") + scale_y_continuous(expand = c(0,
    0), limits = c(0, 100)) + theme_bw() + facet_grid(~cluster) +
    xlab("Cycle") + ylab("% reads number") + scale_fill_manual(values = c("#999999",
    "#0072B2", "#D55E00"), name = NULL) + theme(axis.title.y = element_text(size = 6,
    vjust = -1.55), text = element_text(size = 6), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), legend.position = "top",
    legend.text = element_text(size = 5), legend.key.size = unit(1,
        "lines"), legend.box = "horizontal", legend.margin = margin(0,
        -20, -10, -10), plot.margin = unit(c(0, 0, -0.05, 0),
        "cm"), legend.key = element_rect(color = NA, fill = NA),
    legend.spacing.x = unit(0.04, "cm"))
```

3.4 Alpha diversity patterns

```
df.rmANOVA <- data.frame(Variable = c("Richness", "Pielou evenness"),
    Normality = NA, HOV = NA, Comm.type.pvalue = NA, Cycle.pvalue = NA)
# ANOVAs
ref.anova.rich = aov(Richness ~ cluster * Cycle + Error(Replicate),
    data = meta[meta$Cycle != "C7", ])
ref.anova.even = aov(Even ~ cluster * Cycle + Error(Replicate),
    data = meta[meta$Cycle != "C7", ])
```

	Assumptions		rmANOVA	
Variable	Normality	HOV	Comm.type.pvalue	Cycle.pvalue
Richness	TRUE	TRUE	0.7938670	0.0000002
Pielou evenness	TRUE	TRUE	0.0070899	0.0446801

##a-diversity plots (C0-C6)

```
plot.Richness = meta[meta$Cycle != "C7", ] %>%
    ggplot(aes(cluster, Richness, fill = Cycle)) + geom_boxplot(outlier.shape = NA,
    alpha = 0.5, size = 0.2) + xlab("") + ylab("OTU Richness") +
    geom_point(position = position_jitterdodge(jitter.width = 0.15),
        shape = 21, size = 0.5, stroke = 0.1) + theme_bw() +
    scale_fill_manual(values = c(cbp1[c(6, 7, 8, 10)]), name = "") +
    theme(axis.title.y = element_text(size = 6), text = element_text(size = 6),
        axis.text.x = element_blank(), panel.grid.minor = element_blank(),
        panel.grid.major = element_blank()) + theme(legend.position = "top",
    legend.text = element_text(size = 5.5, margin = margin(0,
        -0.2, 0, 0.01, unit = "cm")), legend.key.size = unit(1,
        "lines"), legend.box = "horizontal") + theme(plot.margin = unit(c(-0.5,
    0.1, -0.25, 0.15), "cm"), axis.text.y = element_text(size = 5),
    legend.box = "horizontal", legend.margin = margin(0, 0, -10,
        -10))

# legend_plot<-get_legend(plot.Richness)
# plot.Richness<-plot.Richness+theme(legend.position =
# 'none')

plot.Evennes = meta[meta$Cycle != "C7", ] %>%
```

```
    ggplot(aes(cluster, Even, fill = Cycle)) + geom_boxplot(outlier.shape = NA,
    alpha = 0.5, size = 0.2) + xlab("") + ylab("Pielou Evennes") +
    geom_point(position = position_jitterdodge(jitter.width = 0.15),
        shape = 21, size = 0.5, stroke = 0.1) + ylim(0.25, 1) +
    theme_bw() + theme(legend.position = "none") + theme(axis.title.y = element_text(size = 6),
    text = element_text(size = 6), axis.text.x = element_text(angle = 13,
        hjust = 0.5), panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
    scale_fill_manual(values = c(cbp1[c(6, 7, 8, 10)]), name = "") +
    theme(legend.position = "none") + geom_signif(comparisons = list(c("Acidovorax",
    "Aeromonas"), c("Pseudomonas", "Aeromonas"), c("Others",
    "Aeromonas")), map_signif_level = TRUE, annotations = c("***",
    "***", "***"), y_position = c(0.78, 0.82, 0.87), textsize = 2.5,
    vjust = 0.5, col = "black", step_increase = 0.05, tip_length = 0.01,
    orientation = "x", size = 0.17) + theme(plot.margin = unit(c(-0.05,
    0.1, -0.45, 0.15), "cm"))
```

3.5 Export figure 3 manuscript

a) Community specific and shared genera in community types dominated by Acidovorax, Pseudomonas, or Aeromonas at cycle 6. b) Relative abundance of the three representative genera per community type (percentage of reads number). c) Shannon diversity (H) and Pielou Evenness indices per community type in the experimental cycles. d) Relative abundances of genera in the three community types.

```
## pdf
##   2
```