Chemo.01.ComunityComposition

Angel Rain & Sara Beier

7/26/2022

Contents

1 Load packages and formatting

1.1 Loading packages

```
rm(list = ls())
library(phyloseq)  # Phyloseq object
library(ggplot2)
library(vegan)
library(dplyr)
library(ape)
library(iCAMP)  # Null model
library(cowplot)  # Mutiple plot arrangement
```

```
library(egg)  # Tag edition
library(reshape)  # Dataframe formation
library(stringr)  # String manipulation
library(ggstatsplot)
library(microViz)
library(kableExtra)  # To nicely print tables
```

1.2 Setting colorblind palette

```
# Colorblind palette
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
    "#D55E00", "#CC79A7", "#808080", "#7570B3")
```

1.3 Loading and formating datasets

In this section the object schema with information about the experimental setup (Sample.ID; Chemostat.ID: chemostats number from 1 to 12; sampling time T: 1:9, DOM: DOM regime; Sal: disturbance regime) is created. Metabarcoding sequence data (16s rRNA gene) were analyzed using the dada2-pipeline (Callahan et al. 2016, doi: 10.1038/nmeth.3869) using the code detailed in script01_dada2.R and the resulting phyloseq project containing the ASV count table is uploaded.

```
# Create metadata for experimental setup
schema<-data.frame(sample.ID=paste0("C10-",rep(1:9,each=12),"-",sprintf("%02d",1:12)),
                   T=rep(1:9,each=12),
                   Chem.ID=sprintf("%02d",1:12),
                   DOM=rep(c("L","H"),each=6), #L: LDOM; H: HDOM
                   Sal=c("C","D")) #C: control; D: disturbed

tibble(schema)
```

```
## # A tibble: 108 x 5
##    sample.ID     T Chem.ID DOM   Sal
##    <chr>     <int> <chr>   <chr> <chr>
##  1 C10-1-01      1 01      L     C
##  2 C10-1-02      1 02      L     D
##  3 C10-1-03      1 03      L     C
##  4 C10-1-04      1 04      L     D
##  5 C10-1-05      1 05      L     C
##  6 C10-1-06      1 06      L     D
##  7 C10-1-07      1 07      H     C
##  8 C10-1-08      1 08      H     D
##  9 C10-1-09      1 09      H     C
## 10 C10-1-10      1 10      H     D
## # ... with 98 more rows
```

```
# Loading phyloseq object from #dada2
ps <- readRDS("../data/dada2.output/chem.ps.rds")

# Loading phylogenetic tree
chem.tree = read_tree("../data/dada2.output/dada-chem.GTR2")
```

```
phy_tree(ps) <- chem.tree  #Adding phylo-tree to the phyloseq object

# Phyloseq object contain abundance table, sample information, taxonomic
# information and the phylogenetic tree
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1447 taxa and 110 samples ]
## sample_data() Sample Data:       [ 110 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 1447 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1447 tips and 1445 internal nodes ]
```

2 Microbial community dynamic

2.1 Preprocess phyloseq-object

```
# Rarefy by minimum readnumber and transform to relative data set.seed(1)
# # Set seed for rarefaction
ps = rarefy_even_depth(ps, min(rowSums(otu_table(ps))), rngseed = 1, replace = F,
    trimOTUs = F)
```

```
## `set.seed(1)` was used to initialize repeatable random subsampling.

## Please record this for your records so others can reproduce.

## Try `set.seed(1); .Random.seed` for the full vector

## ...
```

```
# Estimating relative abundance
rOTUdf.rar <- prop.table(otu_table(ps), 1)

# New phyloseq-project with rarefied ASV table
otu_table(ps) <- otu_table(rOTUdf.rar, taxa_are_rows = FALSE)
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1447 taxa and 110 samples ]
## sample_data() Sample Data:       [ 110 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 1447 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1447 tips and 1445 internal nodes ]
```

```
# Keep ASVs with prevalence equivalent to more 0 reads
ps <- prune_taxa(taxa_sums(ps) > 0, ps)
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 973 taxa and 110 samples ]
## sample_data() Sample Data:       [ 110 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 973 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 973 tips and 971 internal nodes ]
```

```
# Setting up metadata Creating the additional identifier 'sample.names'
# for downstream analysis
schema$sample.names = str_replace_all(schema$sample.ID, "-", ".")
head(schema)
```

```
##   sample.ID T Chem.ID DOM Sal sample.names
## 1  C10-1-01 1      01   L   C     C10.1.01
## 2  C10-1-02 1      02   L   D     C10.1.02
## 3  C10-1-03 1      03   L   C     C10.1.03
## 4  C10-1-04 1      04   L   D     C10.1.04
## 5  C10-1-05 1      05   L   C     C10.1.05
## 6  C10-1-06 1      06   L   D     C10.1.06
```

```
# Re-Order ps object by sample ID from schema-object
new_order <- schema$sample.ID
ps = ps %>%
    ps_reorder(new_order)  #From microViz
```

2.2 Setting-up matrix for barplot

```
# Combine relative count-table with taxonomic information in a new
# dataframe
df1 <- data.frame(tax_table(ps), t(otu_table(ps)))

# Aggregate relative counts by taxonomic Order
df2 <- aggregate(. ~ Order, data = df1[, c(4, 8:115)], sum, na.rm = TRUE)
tibble(df2)
```

```
## # A tibble: 29 x 109
##    Order C10.1~1 C10.1~2 C10.1~3 C10.1~4 C10.1~5 C10.1~6 C10.1~7 C10.1~8 C10.1~9
##    <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Acho~ 2.12e-4 3.19e-4 3.19e-4 2.12e-4 1.06e-4 1.06e-4 0       0       0
##  2 Acti~ 2.12e-3 1.70e-3 1.49e-3 1.49e-3 2.02e-3 2.34e-3 8.50e-4 6.37e-4 4.25e-4
##  3 Babe~ 0       0       0       0       0       0       0       0       0
##  4 Bact~ 0       0       0       0       0       0       0       0       0
##  5 Baln~ 0       0       0       0       0       0       0       0       0
##  6 Beta~ 7.54e-3 1.49e-3 2.01e-2 2.23e-3 2.23e-3 2.76e-3 6.69e-3 2.23e-3 1.59e-3
##  7 Caul~ 1.06e-4 0       2.12e-4 2.12e-4 0       0       0       1.06e-4 0
##  8 Chit~ 0       0       0       0       0       0       0       0       0
##  9 Cyto~ 0       3.19e-4 4.25e-4 4.25e-4 3.19e-4 4.25e-4 1.27e-3 1.49e-3 2.12e-4
## 10 Ente~ 2.90e-1 3.81e-1 2.76e-1 3.57e-1 2.46e-1 3.37e-1 2.03e-1 2.74e-1 1.40e-1
## # ... with 19 more rows, 99 more variables: C10.1.10 <dbl>, C10.1.11 <dbl>,
## #   C10.1.12 <dbl>, C10.2.01 <dbl>, C10.2.02 <dbl>, C10.2.03 <dbl>,
## #   C10.2.04 <dbl>, C10.2.05 <dbl>, C10.2.06 <dbl>, C10.2.07 <dbl>,
## #   C10.2.08 <dbl>, C10.2.09 <dbl>, C10.2.10 <dbl>, C10.2.11 <dbl>,
```

```
## #   C10.2.12 <dbl>, C10.3.01 <dbl>, C10.3.02 <dbl>, C10.3.03 <dbl>,
## #   C10.3.04 <dbl>, C10.3.05 <dbl>, C10.3.06 <dbl>, C10.3.07 <dbl>,
## #   C10.3.08 <dbl>, C10.3.09 <dbl>, C10.3.10 <dbl>, C10.3.11 <dbl>, ...
```

```
order <- df2[, 1]  #Create a vector with only taxonomic Orders
rownames(df2) <- df2[, 1]  # Set taxonomic orders as rownames
df2 <- df2[, -1]  #remove column with orders and keep only abundance data

# tcolSums to see representatively of the top orders ()values close to 1
# are reached)
summary(colSums(df2))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9686  0.9996  0.9998  0.9989  0.9999  1.0000
```

```
# Create a new object for the aggregated relative abundance
agg = df2
agg <- as.data.frame(agg)  #Formatting it as dataframe
rownames(agg) <- order  #add order information as rownames
agg$Sum.agg <- rowSums(agg)  # add an extra column with counts across treatments for all orders

# select the 10 most abundand orders
agg10 <- agg[with(agg, order(-Sum.agg)), ][1:10, 1:108]
agg10$order <- rownames(agg10)
tibble(agg10)
```

```
## # A tibble: 10 x 109
##    C10.1.01 C10.1.02 C10.1.03 C10.1.04 C10.1.05 C10.1.06 C10.1~1 C10.1~2 C10.1~3
##       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
##  1 0.0183   0.0150   0.00425  0.00627  0.0142   0.0159   1.89e-1 1.59e-1 1.11e-1
##  2 0.666    0.598    0.697    0.630    0.734    0.640    5.90e-1 5.46e-1 7.41e-1
##  3 0.290    0.381    0.276    0.357    0.246    0.337    2.03e-1 2.74e-1 1.40e-1
##  4 0        0.000106 0        0        0.000106 0.000106 3.19e-4 0       0
##  5 0.000106 0        0.000212 0.000212 0        0        0       1.06e-4 0
##  6 0.000637 0.000106 0        0        0        0.000106 1.06e-4 2.12e-4 2.12e-4
##  7 0.0134   0.000637 0.000319 0.000425 0.000212 0.000106 8.39e-3 1.48e-2 5.21e-3
##  8 0.000319 0        0        0.000212 0.000106 0        0       0       0
##  9 0.00754  0.00149  0.0201   0.00223  0.00223  0.00276  6.69e-3 2.23e-3 1.59e-3
## 10 0        0.000319 0.000425 0.000425 0.000319 0.000425 1.27e-3 1.49e-3 2.12e-4
## # ... with 100 more variables: C10.1.10 <dbl>, C10.1.11 <dbl>, C10.1.12 <dbl>,
## #   C10.2.01 <dbl>, C10.2.02 <dbl>, C10.2.03 <dbl>, C10.2.04 <dbl>,
## #   C10.2.05 <dbl>, C10.2.06 <dbl>, C10.2.07 <dbl>, C10.2.08 <dbl>,
## #   C10.2.09 <dbl>, C10.2.10 <dbl>, C10.2.11 <dbl>, C10.2.12 <dbl>,
## #   C10.3.01 <dbl>, C10.3.02 <dbl>, C10.3.03 <dbl>, C10.3.04 <dbl>,
## #   C10.3.05 <dbl>, C10.3.06 <dbl>, C10.3.07 <dbl>, C10.3.08 <dbl>,
## #   C10.3.09 <dbl>, C10.3.10 <dbl>, C10.3.11 <dbl>, C10.3.12 <dbl>, ...
```

```
# Check accumulated abundance after pooling by 10 most abundant orders
summary(colSums(agg10[, 1:108]))  # All above or around 0.95
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9674  0.9975  0.9985  0.9974  0.9993  1.0000
```

```r
# convert dataframe from wide to long format
agg10.long <- melt(agg10, id.vars = "order", variable.name = "treat")

# Combine aggregated data with metadata
agg10.long = merge(agg10.long, schema, by.x = "variable", by.y = "sample.names")

# Add experimental time (sampling day) as a new column
agg10.long$Time = as.data.frame(str_split_fixed(as.vector(agg10.long$variable),
    "", 7))[, 5]

# Formatting columns as factors
agg10.long$Time = as.factor(agg10.long$Time)
levels(agg10.long$Time) = c("4", "8", "15", "18", "22", "29", "36", "39", "41")
agg10.long$Sal = as.factor(agg10.long$Sal)

# Rename treatments
levels(agg10.long$Sal) = c("Control", "Disturbance")
agg10.long$DOM = as.factor(agg10.long$DOM)

# Rename DOM level
levels(agg10.long$DOM) = c("HDOM", "LDOM")

# Add replicate identifier
agg10.long$Rep = rep(c(1, 2, 3), each = 20)

tibble(agg10.long)
```

```
## # A tibble: 1,080 x 10
##    variable order            value sampl~1     T Chem.ID DOM   Sal   Time    Rep
##    <fct>    <chr>            <dbl> <chr>   <int> <chr>   <fct> <fct> <fct> <dbl>
##  1 C10.1.01 Rhodobacteral~ 1.83e-2 C10-1-~     1 01      LDOM  Cont~ 4         1
##  2 C10.1.01 Flavobacteria~ 6.66e-1 C10-1-~     1 01      LDOM  Cont~ 4         1
##  3 C10.1.01 Enterobactera~ 2.90e-1 C10-1-~     1 01      LDOM  Cont~ 4         1
##  4 C10.1.01 Sphingomonada~ 0       C10-1-~     1 01      LDOM  Cont~ 4         1
##  5 C10.1.01 Caulobacteral~ 1.06e-4 C10-1-~     1 01      LDOM  Cont~ 4         1
##  6 C10.1.01 Rhodospirilla~ 6.37e-4 C10-1-~     1 01      LDOM  Cont~ 4         1
##  7 C10.1.01 Pseudomonadal~ 1.34e-2 C10-1-~     1 01      LDOM  Cont~ 4         1
##  8 C10.1.01 Rhizobiales    3.19e-4 C10-1-~     1 01      LDOM  Cont~ 4         1
##  9 C10.1.01 Betaproteobac~ 7.54e-3 C10-1-~     1 01      LDOM  Cont~ 4         1
## 10 C10.1.01 Cytophagales   0       C10-1-~     1 01      LDOM  Cont~ 4         1
## # ... with 1,070 more rows, and abbreviated variable name 1: sample.ID
```

2.3 Microbial community relative abundance

Barplots indicate the change of the 10 most abundant order along the chemostat experiment by DOM level (HDOM and LDOM) by disturbance regime (Disturbed and undisturbed Controls). Early stages during the succession were dominated by members of Flavobacteriales and Enterobacterales orders at LDOM, as well as Rhodobacterales under the HDOM regime. At intermediate succession stages, Rhodobacterales and Caulobacterales increased in abundance. At the experiment end diverse taxonomic structures were observed and depending on the culture vessel for instance members of the Caulobacterales, Sphingomonadales, Rhodospirillales, or Enterobacterales dominated the community.

```
# Create an individual plot for LDOM treatment

plot0 <- agg10.long[agg10.long$DOM == "LDOM", ] %>%
    ggplot(aes(x = Time, y = value, fill = order)) + geom_bar(stat = "identity",
    width = 1) + scale_fill_manual(values = cbbPalette) + facet_grid(Sal ~ Rep) +
    xlab("Sampling day") + ylab("Relative abundance") + theme_bw() + theme(strip.placement.y = "outside
    axis.text.x = element_text(size = 7), strip.text.y = element_text(angle = 270),
    strip.background = element_blank()) + theme(plot.margin = margin(t = 2,
    r = -12, b = 0, l = 0, unit = "pt"))
plot0 <- plot0 + theme(legend.position = "none")

# Create an individual plot for HDOM treatment
plot1 <- agg10.long[agg10.long$DOM == "HDOM", ] %>%
    ggplot(aes(x = Time, y = value, fill = order)) + geom_bar(stat = "identity",
    width = 1) + scale_fill_manual(values = cbbPalette) + facet_grid(Sal ~ Rep) +
    xlab("Sampling day") + ylab("Relative abundance") + theme_bw() + theme(strip.placement.y = "outside
    axis.text.x = element_text(size = 7), strip.text.y = element_text(angle = 270),
    strip.background = element_blank()) + theme(plot.margin = margin(t = 2,
    r = 0, b = 0, l = 0, unit = "pt"))
```

2.3.1 Figure relative abundance

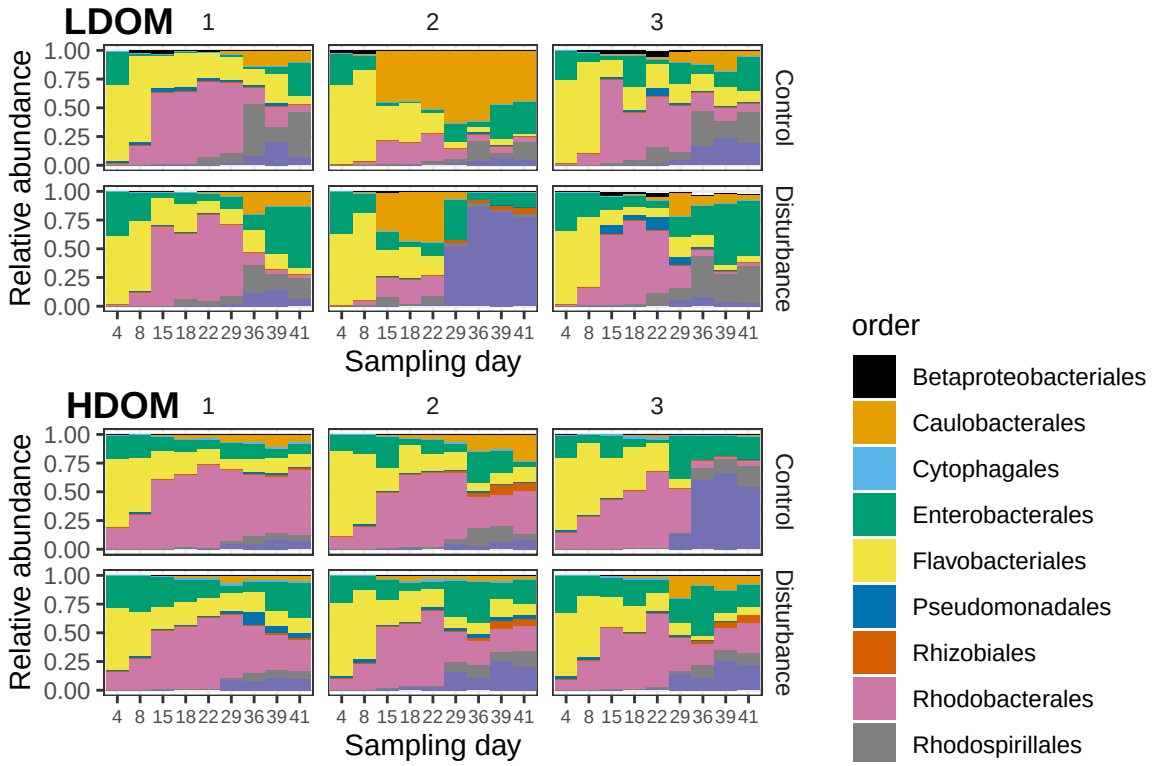

Figure 1: Community structure. Relative abundance of the 10 most abundant orders in the initial communities as well as in the resuscitated communities at LDOM and HDOM levels at each sampling day.

3 Betadiversity

3.1 Bray-Curtis and Unifrac ordination

```
# Bray-curtis (for relative abundance)
ord.pcoa.bray <- ordinate(ps, method = "PCoA", distance = "bray")
# Setting dataframes with ordination estimations
df.ord.pcoa.bray = data.frame(pco1 = ord.pcoa.bray$vectors[, 1], pco2 = ord.pcoa.bray$vectors[,
    2], Treatment = schema$Sal, DOM = schema$DOM, day = as.factor(schema$T),
    Chem.ID = schema$Chem.ID)
tibble(df.ord.pcoa.bray)
```

```
## # A tibble: 108 x 6
##      pco1    pco2 Treatment DOM   day   Chem.ID
##     <dbl>   <dbl> <chr>     <chr> <fct> <chr>
##  1 -0.484 -0.241  C         L     1     01
##  2 -0.429 -0.296  D         L     1     02
##  3 -0.533 -0.220  C         L     1     03
##  4 -0.471 -0.270  D         L     1     04
##  5 -0.472 -0.267  C         L     1     05
##  6 -0.448 -0.288  D         L     1     06
##  7 -0.440 -0.0473 C         H     1     07
##  8 -0.433 -0.0624 D         H     1     08
##  9 -0.459 -0.130  C         H     1     09
## 10 -0.457 -0.134  D         H     1     10
## # ... with 98 more rows
```

```
# Replace sampling ID by sampling day
levels(df.ord.pcoa.bray$day) = c("4", "8", "15", "18", "22", "29", "36", "39",
    "41")

# Visualize dataframe for Bray-curtis distance ordination
tibble(df.ord.pcoa.bray)
```

```
## # A tibble: 108 x 6
##      pco1    pco2 Treatment DOM   day   Chem.ID
##     <dbl>   <dbl> <chr>     <chr> <fct> <chr>
##  1 -0.484 -0.241  C         L     4     01
##  2 -0.429 -0.296  D         L     4     02
##  3 -0.533 -0.220  C         L     4     03
##  4 -0.471 -0.270  D         L     4     04
##  5 -0.472 -0.267  C         L     4     05
##  6 -0.448 -0.288  D         L     4     06
##  7 -0.440 -0.0473 C         H     4     07
##  8 -0.433 -0.0624 D         H     4     08
##  9 -0.459 -0.130  C         H     4     09
## 10 -0.457 -0.134  D         H     4     10
## # ... with 98 more rows
```

```
# Create a label vector for DNA-sampling days
labels = c("d4", "d8", "d15", "d18", "d22", "d29", "d36", "d39", "d41")
```

3.2 Bray-Curtis ordination visualization

```
# Figure all data points
plot.ord.pcoa.bray = ggplot(df.ord.pcoa.bray, aes(x = pco1, y = pco2)) + geom_point(aes(shape = interac
    DOM), colour = factor(day)), size = 5, alpha = 0.9) + scale_shape_manual(values = c(1,
    20, 2, 17), name = "Treatments", labels = c("C-HDOM", "D-HDOM", "C-LDOM",
    "D-LDOM")) + scale_color_manual(values = cbbPalette, name = "Sampling day") +
    theme_bw() + labs(title = "") + theme(text = element_text(size = 10, family = "ArialMT"),
    panel.grid.minor = element_blank()) + scale_x_continuous(breaks = seq(-0.5,
    0.5, 0.125), expand = c(0.02, 0.02)) + scale_y_continuous(breaks = seq(-0.5,
    0.5, 0.125), expand = c(0.02, 0.02)) + xlab(paste("PCOA1 [", round(ord.pcoa.bray$values[3][1,
    ] * 100, 0), "%]")) + ylab(paste("PCOA2 [", round(ord.pcoa.bray$values[3][2,
    ] * 100, 0), "%]")) + theme(legend.position = "none")

# Panels
plot.ord.pcoa.bray.b = ggplot(df.ord.pcoa.bray, aes(x = pco1, y = pco2)) + geom_point(aes(shape = intera
    DOM), colour = factor(day)), size = 3, alpha = 0.9) + scale_shape_manual(values = c(1,
    20, 2, 17), name = "Treatments", labels = c("C-HDOM", "D-HDOM", "C-LDOM",
    "D-LDOM")) + scale_color_manual(values = cbbPalette, name = "Sampling day") +
    theme_bw() + labs(title = "") + theme_bw() + labs(title = "", x = "", y = "") +
    theme(text = element_text(size = 10, family = "ArialMT"), panel.grid.minor = element_blank()) +
    scale_x_continuous(breaks = seq(-0.5, 0.5, 0.125), expand = c(0.02, 0.02)) +
    scale_y_continuous(breaks = seq(-0.5, 0.5, 0.125), expand = c(0.02, 0.02)) +
    facet_wrap(~day, ncol = 3, scales = "free") + theme(axis.text.x = element_blank(),
    axis.text.y = element_blank(), axis.ticks = element_blank(), legend.key.size = unit(0.8,
        "lines"))
#
plot.ord.pcoa.bray.b = tag_facet(plot.ord.pcoa.bray.b, open = "", close = ")",
    tag_pool = labels, hjust = -0.1, size = 3)
plot_grid(plot.ord.pcoa.bray, plot.ord.pcoa.bray.b, rel_widths = c(0.9, 1),
    ncol = 2, labels = c("A", "B"))
```

3.3 Unifrac ordination visualization

```
# Unifrac (Weighted=T for considering relative abundances)
ord.pcoa.unif <- ordinate(ps, "PCoA", "unifrac", weighted = T)
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as --
## SV_1192_Rhodospirillales -- in the phylogenetic tree in the data you provided.
```

```
# Setting dataframes with ordination estimations
df.ord.pcoa.unif = data.frame(pco1 = ord.pcoa.unif$vectors[, 1], pco2 = ord.pcoa.unif$vectors[,
    2], Treatment = schema$Sal, DOM = schema$DOM, day = as.factor(schema$T),
    Chem.ID = schema$Chem.ID)
tibble(df.ord.pcoa.unif)
```

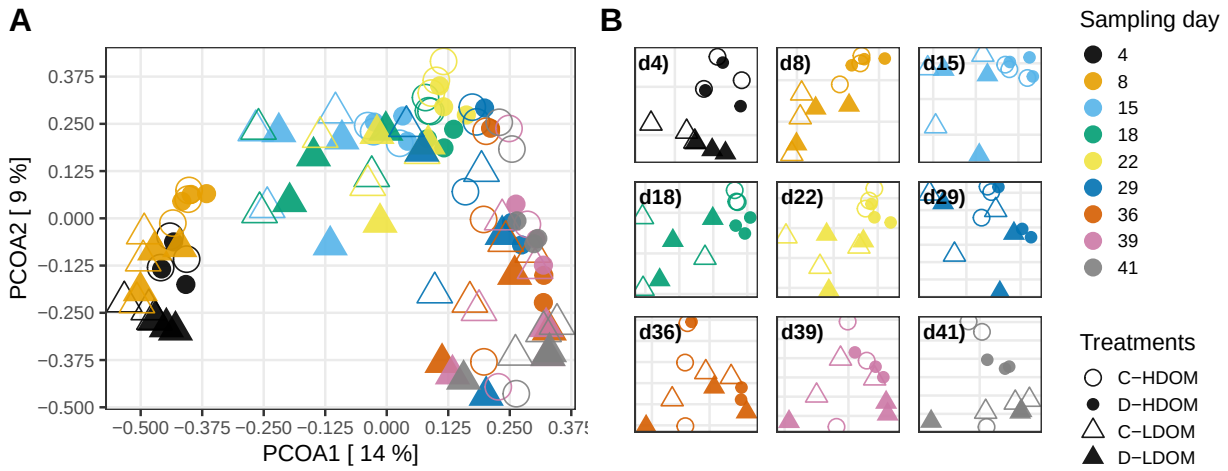

Figure 2: Community structure. A) Overview PCoA Bray-Curtis based biplot including all data points. B) PCoAs for individual sampling days; axes of the individual plots are differently scaled as indicated by the grid lines

```
## # A tibble: 108 x 6
##       pco1     pco2 Treatment DOM   day   Chem.ID
##      <dbl>    <dbl> <chr>     <chr> <fct> <chr>
##  1 -0.291 -0.0628  C         L     1     01
##  2 -0.284 -0.0834  D         L     1     02
##  3 -0.300 -0.0577  C         L     1     03
##  4 -0.291 -0.0779  D         L     1     04
##  5 -0.302 -0.0532  C         L     1     05
##  6 -0.290 -0.0764  D         L     1     06
##  7 -0.242 -0.0103  C         H     1     07
##  8 -0.240 -0.0353  D         H     1     08
##  9 -0.286  0.00409 C         H     1     09
## 10 -0.269 -0.0284  D         H     1     10
## # ... with 98 more rows
```

```
# Replace sampling ID by sampling day
levels(df.ord.pcoa.unif$day) = c("4", "8", "15", "18", "22", "29", "36", "39",
    "41")

# Visualize dataframe for Unifrac distance ordination
tibble(df.ord.pcoa.unif)
```

```
## # A tibble: 108 x 6
##       pco1     pco2 Treatment DOM   day   Chem.ID
##      <dbl>    <dbl> <chr>     <chr> <fct> <chr>
##  1 -0.291 -0.0628  C         L     4     01
##  2 -0.284 -0.0834  D         L     4     02
##  3 -0.300 -0.0577  C         L     4     03
##  4 -0.291 -0.0779  D         L     4     04
##  5 -0.302 -0.0532  C         L     4     05
##  6 -0.290 -0.0764  D         L     4     06
##  7 -0.242 -0.0103  C         H     4     07
##  8 -0.240 -0.0353  D         H     4     08
```

```
##  9 -0.286  0.00409 C         H     4     09
## 10 -0.269 -0.0284  D         H     4     10
## # ... with 98 more rows
```

```
# Figure all data points
plot.ord.pcoa.unif = ggplot(df.ord.pcoa.unif, aes(x = pco1, y = pco2)) + geom_point(aes(shape = interac
    DOM), colour = factor(day)), size = 5, alpha = 0.9) + scale_shape_manual(values = c(1,
    20, 2, 17), name = "Treatments", labels = c("C-HDOM", "D-HDOM", "C-LDOM",
    "D-LDOM")) + scale_color_manual(values = cbbPalette, name = "Sampling day") +
    theme_bw() + labs(title = "") + theme(text = element_text(size = 10, family = "ArialMT"),
    panel.grid.minor = element_blank()) + scale_x_continuous(breaks = seq(-0.5,
    0.5, 0.125), expand = c(0.02, 0.02)) + scale_y_continuous(breaks = seq(-0.5,
    0.5, 0.125), expand = c(0.02, 0.02)) + xlab(paste("PCOA1 [", round(ord.pcoa.unif$values[3][1,
    ] * 100, 0), "%]")) + ylab(paste("PCOA2 [", round(ord.pcoa.unif$values[3][2,
    ] * 100, 0), "%]")) + theme(legend.position = "none")

# Panels
plot.ord.pcoa.unif.b = ggplot(df.ord.pcoa.unif, aes(x = pco1, y = pco2)) + geom_point(aes(shape = intera
    DOM), colour = factor(day)), size = 3, alpha = 0.9) + scale_shape_manual(values = c(1,
    20, 2, 17), name = "Treatments", labels = c("C-HDOM", "D-HDOM", "C-LDOM",
    "D-LDOM")) + scale_color_manual(values = cbbPalette, name = "Sampling day") +
    theme_bw() + labs(title = "") + theme_bw() + labs(title = "", x = "", y = "") +
    theme(text = element_text(size = 10, family = "ArialMT"), panel.grid.minor = element_blank()) +
    scale_x_continuous(breaks = seq(-0.5, 0.5, 0.125), expand = c(0.02, 0.02)) +
    scale_y_continuous(breaks = seq(-0.5, 0.5, 0.125), expand = c(0.02, 0.02)) +
    facet_wrap(~day, ncol = 3, scales = "free") + theme(axis.text.x = element_blank(),
    axis.text.y = element_blank(), axis.ticks = element_blank(), legend.key.size = unit(0.8,
        "lines"))

plot.ord.pcoa.unif.b = tag_facet(plot.ord.pcoa.unif.b, open = "", close = ")",
    tag_pool = labels, hjust = -0.1, size = 3)
plot_grid(plot.ord.pcoa.unif, plot.ord.pcoa.unif.b, rel_widths = c(0.9, 1),
    ncol = 2, labels = c("A", "B"))
```

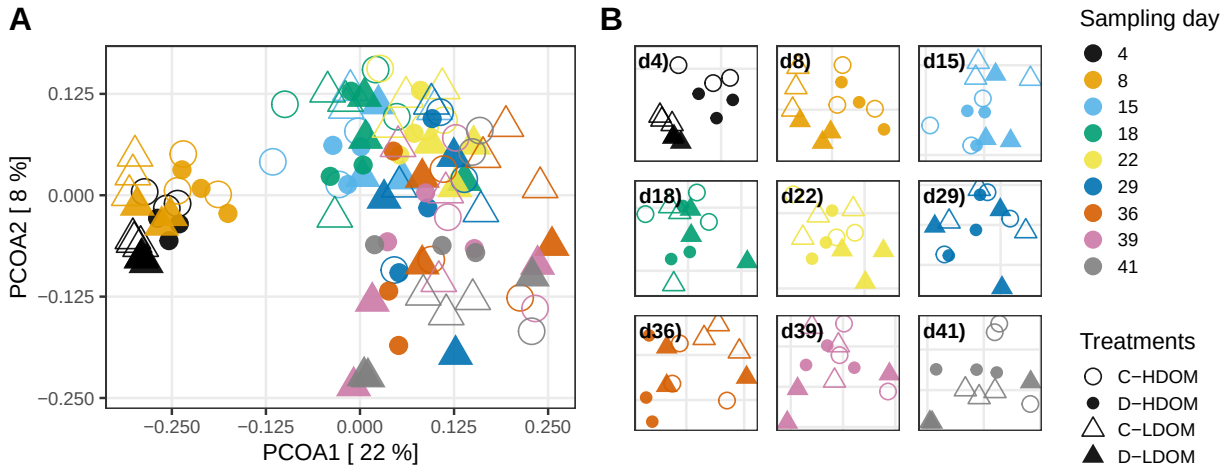

Figure 3: Community structure. A) Overview PCoA weighted Unifrac based biplot including all data points. B) PCoAs for individual sampling days; axes of the individual plots are differently scaled as indicated by the grid lines

4 Statistical analysis

4.1 Permanova

We performed a Permutational Multivariate Analysis of Variance (Permanova) usgin the difference distance matrix estimated (Bray-curtis and Unifrac distances). Factor considered for the analysis were: 1) DOM level availability, 2) Disturbance treatment (as "Sal" treatment) and 3) Time.

```
# Bray Curtis distance
dist.bray <- phyloseq::distance(ps, method = "bray")

# Unifrac distance
dist.wunif <- phyloseq::distance(ps, method = "wunifrac")
```

```
## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root as --
## SV_1236_Sphingomonadales -- in the phylogenetic tree in the data you provided.
```

```
# Permanova all factors: 1) DOM level availability, 2) Disturbance
# treatment (Sal) and 3) Time.

# Bray Curtis distance
set.seed(1)
permanova.bray <- adonis2(dist.bray ~ DOM * Sal * T, data = schema, permutations = 1000)
permanova.bray
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 1000
##
## adonis2(formula = dist.bray ~ DOM * Sal * T, data = schema, permutations = 1000)
##            Df SumOfSqs      R2       F   Pr(>F)
## DOM         1   3.1410 0.12065 22.4770 0.000999 ***
## Sal         1   0.3465 0.01331  2.4797 0.029970 *
## T           1   7.3069 0.28067 52.2876 0.000999 ***
## DOM:Sal     1   0.3915 0.01504  2.8014 0.010989 *
## DOM:T       1   0.3981 0.01529  2.8488 0.016983 *
## Sal:T       1   0.1034 0.00397  0.7399 0.597403
## DOM:Sal:T   1   0.3714 0.01427  2.6580 0.017982 *
## Residual  100  13.9744 0.53679
## Total     107  26.0332 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Unifrac distance
set.seed(1)
permanova.wunifrac <- adonis2(dist.wunif ~ DOM * Sal * T, data = schema, permutations = 1000)
permanova.wunifrac
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
```

```
## Permutation: free
## Number of permutations: 1000
##
## adonis2(formula = dist.wunif ~ DOM * Sal * T, data = schema, permutations = 1000)
##            Df SumOfSqs      R2       F   Pr(>F)
## DOM         1   0.3000 0.03767  7.3773 0.002997 **
## Sal         1   0.1150 0.01444  2.8282 0.047952 *
## T           1   3.2881 0.41288 80.8638 0.000999 ***
## DOM:Sal     1   0.0857 0.01076  2.1075 0.081918 .
## DOM:T       1   0.0284 0.00357  0.6988 0.514486
## Sal:T       1   0.0391 0.00491  0.9618 0.363636
## DOM:Sal:T   1   0.0413 0.00519  1.0163 0.345654
## Residual  100   4.0663 0.51058
## Total     107   7.9639 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```