Chemo.04.ComunityAssembly

Angel Rain & Sara Beier

8/1/2022

Contents

1 Load packages and formatting

1.1 Loading packages

```
rm(list = ls())
library(phyloseq)
library(ggplot2)
library(vegan)
library(ecodist)
library(dplyr)
library(ape)  #
library(iCAMP)  #NULL model for microbial comm
library(cowplot)  #Multiple panel formatting
library(egg)  #Additional label for multiple panels
library(reshape)  #Dataframe formatting
library(stringr)  #Text manipulation
library(microViz)  #Microbiome (re_order function for phyloseq-objects)
library(rstatix)  #Versatile statistical package
```

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: chemostat number from 1 to 12; sampling time T: 1:9, DOM: DOM regime; Sal: disturbance regime) is created. Metabracoding sequence data (16s rRNA gene) were analyzed using the dada2-pipeline (Callahan et al. 2016, doi: 10.1038/nmeth.3869) using the code 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("oDOM",
    "eDOM"), each = 6), Sal = c("C", "D"))
schema$T = as.factor(schema$T)
schema$DOM = as.factor(schema$DOM)
schema$Sal = as.factor(schema$Sal)
tibble(schema)
```

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

```
# Loading phyloseq object from #dada2
ps <- readRDS("../data/dada2.output/chem.ps.rds")
# Phyloseq object contain abundance table, sample information, taxanomic
# information and the phylogenetic tree

# Loadgin 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 ]
```

```
# Removing initial inoculum samples for downstream analysis
ps = subset_samples(ps, sample.ID != "C10-0-LL" & sample.ID != "C10-0-HH")
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1447 taxa and 108 samples ]
## sample_data() Sample Data:       [ 108 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
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 108 samples ]
## sample_data() Sample Data:       [ 108 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 108 samples ]
## sample_data() Sample Data:       [ 108 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
head(sample_data(ps))
```

```
##           sample.ID Chem.ID T
## C10-1-09  C10-1-09      09 1
## C10-2-05  C10-2-05      05 2
## C10-3-01  C10-3-01      01 3
## C10-3-09  C10-3-09      09 3
## C10-4-05  C10-4-05      05 4
## C10-5-01  C10-5-01      01 5
```

```
# Some samples in phyloseq object are not included into this analyses
# (schema), so we proceed to reorder ps-data base in the schema$sample.ID

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

# Vizualize ordered ps-object
head(sample_data(ps))
```

```
##           sample.ID Chem.ID T
## C10-1-01  C10-1-01      01 1
## C10-1-02  C10-1-02      02 1
## C10-1-03  C10-1-03      03 1
## C10-1-04  C10-1-04      04 1
## C10-1-05  C10-1-05      05 1
## C10-1-06  C10-1-06      06 1
```

```
tail(sample_data(ps))
```

```
##           sample.ID Chem.ID T
## C10-9-07  C10-9-07      07 9
## C10-9-08  C10-9-08      08 9
## C10-9-09  C10-9-09      09 9
## C10-9-10  C10-9-10      10 9
## C10-9-11  C10-9-11      11 9
## C10-9-12  C10-9-12      12 9
```

2.2 Subset data by treatment

```
# Create empty list-objects
oD.ps = list()  # List to store oDOM results
oC.ps = list()  # List to store oDOM results

eD.ps = list()  # List to store eDOM results
eC.ps = list()  # List to store eDOM results

# Sample ID (Control and disturbance treatments)
oDOM_C = c("01", "03", "05")  #o-DOM Samples for control
oDOM_D = c("02", "04", "06")  #o-DOM Samples for disturbed treatments
eDOM_C = c("07", "09", "11")  #e-DOM Samples for control
eDOM_D = c("08", "10", "12")  #e-DOM Samples for disturbed treatments
```

```
# oDOM level Filter by Sample
tmpL <- prune_samples(ps@sam_data[["Chem.ID"]] %in% oDOM_C, ps)
# Filter ASVs (taxa) to only those with abun equal to 0 in all the samples
oC.ps <- filter_taxa(tmpL, function(x) sum(x != 0) > 0, TRUE)
oC.ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 437 taxa and 27 samples ]
## sample_data() Sample Data:       [ 27 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 437 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 437 tips and 435 internal nodes ]
```

```
tmpL <- prune_samples(ps@sam_data[["Chem.ID"]] %in% oDOM_D, ps)
oD.ps <- filter_taxa(tmpL, function(x) sum(x != 0) > 0, TRUE)
oD.ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 394 taxa and 27 samples ]
## sample_data() Sample Data:       [ 27 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 394 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 394 tips and 392 internal nodes ]
```

```
# eDOM level Filter by Sample
tmpH <- prune_samples(ps@sam_data[["Chem.ID"]] %in% oDOM_C, ps)
# Filter ASVs (taxa) to only those with abun equal to 0 in all the samples
eC.ps <- filter_taxa(tmpH, function(x) sum(x != 0) > 0, TRUE)
eC.ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 437 taxa and 27 samples ]
## sample_data() Sample Data:       [ 27 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 437 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 437 tips and 435 internal nodes ]
```

```
tmpH <- prune_samples(ps@sam_data[["Chem.ID"]] %in% oDOM_D, ps)
# Filter ASVs (taxa) to only those with abun equal to 0 in all the samples
eD.ps <- filter_taxa(tmpH, function(x) sum(x != 0) > 0, TRUE)
eD.ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 394 taxa and 27 samples ]
## sample_data() Sample Data:       [ 27 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 394 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 394 tips and 392 internal nodes ]
```

2.3 Compute the beta-Nearest Taxon Index (bNTI)

The Influence of deterministic versus stochastic processes on microbial community dynamics was quantified during the course of the continuous culture experiment via null model analyses using the beta nearest taxon indices (bNTI) between sample pairs. For this purpose we applied bNTIs between two temporally succeeding samples separately for disturbance and DOM regimes by applying a sliding window setup in the continuous cultures.

```r
knitr::opts_chunk$set(cache = T)

NTI.out.oC = list()  #Loop for oDOM-Control for each sampling point
NTI.out.oD = list()  #Loop for oDOM-Disturbance for each sampling point

# o-DOM x Control
comm = otu_table(oC.ps)
dist = cophenetic(phy_tree(oC.ps))  #Standard distance calculation for tree used in the manual
set.seed(1)
NTI.out.oC = bNTIn.p(comm@.Data, dist, nworker = 2, memo.size.GB = 50, weighted = TRUE,
    exclude.consp = FALSE, rand = 1000, output.bMNTD = FALSE, sig.index = "SES",
    unit.sum = NULL, correct.special = FALSE, detail.null = FALSE, special.method = "MNTD")
```

```
## Now calculating observed betaMNTD. Begin at Fri Nov 10 10:05:46 2023. Please wait...

## Now randomizing by parallel computing. Begin at Fri Nov 10 10:05:47 2023. Please wait...
```

```r
# o-DOM x Disturbance
comm = otu_table(oD.ps)
dist = cophenetic(phy_tree(oD.ps))  #Standard distance calculation for tree used in the manual
NTI.out.oD = bNTIn.p(comm@.Data, dist, nworker = 2, memo.size.GB = 50, weighted = TRUE,
    exclude.consp = FALSE, rand = 1000, output.bMNTD = FALSE, sig.index = "SES",
    unit.sum = NULL, correct.special = FALSE, detail.null = FALSE, special.method = "MNTD")
```

```
## Now calculating observed betaMNTD. Begin at Fri Nov 10 10:08:13 2023. Please wait...

## Now randomizing by parallel computing. Begin at Fri Nov 10 10:08:15 2023. Please wait...
```

```r
NTI.out.eC = list()  #Loop for eDOM-Control for each sampling point
NTI.out.eD = list()  #Loop for eDOM-Disturbance for each sampling point

# e-DOM x Control
comm = otu_table(eC.ps)
dist = cophenetic(phy_tree(eC.ps))  #Standard distance calculation for tree used in the manual
NTI.out.eC = bNTIn.p(comm@.Data, dist, nworker = 2, memo.size.GB = 50, weighted = TRUE,
    exclude.consp = FALSE, rand = 100, output.bMNTD = FALSE, sig.index = "SES",
    unit.sum = NULL, correct.special = FALSE, detail.null = FALSE, special.method = "MNTD")
```

```
## Now calculating observed betaMNTD. Begin at Fri Nov 10 10:10:23 2023. Please wait...

## Now randomizing by parallel computing. Begin at Fri Nov 10 10:10:24 2023. Please wait...
```

```
# e-DOM x Disturbance
comm = otu_table(eD.ps)
dist = cophenetic(phy_tree(eD.ps))  #Standard distance calculation for tree used in the manual
NTI.out.eD = bNTIn.p(comm@.Data, dist, nworker = 2, memo.size.GB = 50, weighted = TRUE,
    exclude.consp = FALSE, rand = 1000, output.bMNTD = FALSE, sig.index = "SES",
    unit.sum = NULL, correct.special = FALSE, detail.null = FALSE, special.method = "MNTD")
```

```
## Now calculating observed betaMNTD. Begin at Fri Nov 10 10:10:44 2023. Please wait...

## Now randomizing by parallel computing. Begin at Fri Nov 10 10:10:45 2023. Please wait...
```

2.4 Reshape bNTI index into a data.frame

```
# Function to transform distance matrix into dataframe
dist2df_AR <- function(m) {
    xy <- t(combn(colnames(m$index), 2))
    tmp = data.frame(xy, dist = m$index[xy])
    tmp$t1 = str_split_fixed(as.character(tmp$X1), "-", 3)[, 2]  # Extract the time point from sample '
    tmp$t2 = str_split_fixed(as.character(tmp$X2), "-", 3)[, 2]  # Extract time point from sample 'y'
    tmp$dif = as.numeric(tmp$t2) - as.numeric(tmp$t1)  # Calculate the difference in time units
    return(tmp)
}

# oDOM control regime
df.NTI.oC <- dist2df_AR(NTI.out.oC)
df.NTI.oC = df.NTI.oC[df.NTI.oC$dif == 1, ]  # Get data space by only 1 time unit
df.NTI.oC$DOM = "oDOM"
df.NTI.oC$Treatment = "Control"
tibble(df.NTI.oC)
```

```
## # A tibble: 72 x 8
##    X1       X2         dist t1    t2      dif DOM   Treatment
##    <chr>    <chr>     <dbl> <chr> <chr> <dbl> <chr> <chr>
##  1 C10-1-01 C10-2-01 -0.878 1     2         1 oDOM  Control
##  2 C10-1-01 C10-2-03  0.979 1     2         1 oDOM  Control
##  3 C10-1-01 C10-2-05  1.86  1     2         1 oDOM  Control
##  4 C10-1-03 C10-2-01  0.168 1     2         1 oDOM  Control
##  5 C10-1-03 C10-2-03  2.09  1     2         1 oDOM  Control
##  6 C10-1-03 C10-2-05  1.51  1     2         1 oDOM  Control
##  7 C10-1-05 C10-2-01  0.469 1     2         1 oDOM  Control
##  8 C10-1-05 C10-2-03  0.695 1     2         1 oDOM  Control
##  9 C10-1-05 C10-2-05  1.31  1     2         1 oDOM  Control
## 10 C10-2-01 C10-3-01  0.383 2     3         1 oDOM  Control
## # ... with 62 more rows
```

```
# oDOM disturbance regime
df.NTI.oD <- dist2df_AR(NTI.out.oD)
df.NTI.oD = df.NTI.oD[df.NTI.oD$dif == 1, ]  # Get data space by only 1 time unit
df.NTI.oD$DOM = "oDOM"
df.NTI.oD$Treatment = "Disturbance"
tibble(df.NTI.oD)
```

```
## # A tibble: 72 x 8
##    X1       X2           dist t1    t2      dif DOM   Treatment
##    <chr>    <chr>       <dbl> <chr> <chr> <dbl> <chr> <chr>
##  1 C10-1-02 C10-2-02 -1.07   1     2          1 oDOM  Disturbance
##  2 C10-1-02 C10-2-04  0.982  1     2          1 oDOM  Disturbance
##  3 C10-1-02 C10-2-06 -0.0847 1     2          1 oDOM  Disturbance
##  4 C10-1-04 C10-2-02  1.87   1     2          1 oDOM  Disturbance
##  5 C10-1-04 C10-2-04  2.68   1     2          1 oDOM  Disturbance
##  6 C10-1-04 C10-2-06  2.03   1     2          1 oDOM  Disturbance
##  7 C10-1-06 C10-2-02  2.98   1     2          1 oDOM  Disturbance
##  8 C10-1-06 C10-2-04  1.55   1     2          1 oDOM  Disturbance
##  9 C10-1-06 C10-2-06  1.85   1     2          1 oDOM  Disturbance
## 10 C10-2-02 C10-3-02 -0.285  2     3          1 oDOM  Disturbance
## # ... with 62 more rows
```

```
# eDOM control regime
df.NTI.eC <- dist2df_AR(NTI.out.eC)
df.NTI.eC = df.NTI.eC[df.NTI.eC$dif == 1, ]  # Get data space by only 1 time unit
df.NTI.eC$DOM = "eDOM"
df.NTI.eC$Treatment = "Control"
tibble(df.NTI.eC)
```

```
## # A tibble: 72 x 8
##    X1       X2          dist t1    t2      dif DOM   Treatment
##    <chr>    <chr>      <dbl> <chr> <chr> <dbl> <chr> <chr>
##  1 C10-1-01 C10-2-01 -0.881 1     2          1 eDOM  Control
##  2 C10-1-01 C10-2-03  0.778 1     2          1 eDOM  Control
##  3 C10-1-01 C10-2-05  1.78  1     2          1 eDOM  Control
##  4 C10-1-03 C10-2-01  0.323 1     2          1 eDOM  Control
##  5 C10-1-03 C10-2-03  2.17  1     2          1 eDOM  Control
##  6 C10-1-03 C10-2-05  1.35  1     2          1 eDOM  Control
##  7 C10-1-05 C10-2-01  0.636 1     2          1 eDOM  Control
##  8 C10-1-05 C10-2-03  0.522 1     2          1 eDOM  Control
##  9 C10-1-05 C10-2-05  1.16  1     2          1 eDOM  Control
## 10 C10-2-01 C10-3-01  0.346 2     3          1 eDOM  Control
## # ... with 62 more rows
```

```
# eDOM disturbance regime
df.NTI.eD <- dist2df_AR(NTI.out.eD)
df.NTI.eD = df.NTI.eD[df.NTI.eD$dif == 1, ]  # Get data space by only 1 time unit
df.NTI.eD$DOM = "eDOM"
df.NTI.eD$Treatment = "Disturbance"
tibble(df.NTI.eD)
```

```
## # A tibble: 72 x 8
##    X1       X2          dist t1    t2      dif DOM   Treatment
##    <chr>    <chr>      <dbl> <chr> <chr> <dbl> <chr> <chr>
##  1 C10-1-02 C10-2-02 -1.07   1     2         1 eDOM  Disturbance
##  2 C10-1-02 C10-2-04  0.933  1     2         1 eDOM  Disturbance
##  3 C10-1-02 C10-2-06 -0.0639 1     2         1 eDOM  Disturbance
##  4 C10-1-04 C10-2-02  1.88   1     2         1 eDOM  Disturbance
##  5 C10-1-04 C10-2-04  2.74   1     2         1 eDOM  Disturbance
##  6 C10-1-04 C10-2-06  2.03   1     2         1 eDOM  Disturbance
##  7 C10-1-06 C10-2-02  3.10   1     2         1 eDOM  Disturbance
##  8 C10-1-06 C10-2-04  1.47   1     2         1 eDOM  Disturbance
##  9 C10-1-06 C10-2-06  1.99   1     2         1 eDOM  Disturbance
## 10 C10-2-02 C10-3-02 -0.317  2     3         1 eDOM  Disturbance
## # ... with 62 more rows
```

```
# Pooling the dataframes together
df.NTI.all = rbind(df.NTI.oC, df.NTI.oD, df.NTI.eC, df.NTI.eD)
tibble(df.NTI.all)
```

```
## # A tibble: 288 x 8
##    X1       X2         dist t1    t2      dif DOM   Treatment
##    <chr>    <chr>     <dbl> <chr> <chr> <dbl> <chr> <chr>
##  1 C10-1-01 C10-2-01 -0.878 1     2         1 oDOM  Control
##  2 C10-1-01 C10-2-03  0.979 1     2         1 oDOM  Control
##  3 C10-1-01 C10-2-05  1.86  1     2         1 oDOM  Control
##  4 C10-1-03 C10-2-01  0.168 1     2         1 oDOM  Control
##  5 C10-1-03 C10-2-03  2.09  1     2         1 oDOM  Control
##  6 C10-1-03 C10-2-05  1.51  1     2         1 oDOM  Control
##  7 C10-1-05 C10-2-01  0.469 1     2         1 oDOM  Control
##  8 C10-1-05 C10-2-03  0.695 1     2         1 oDOM  Control
##  9 C10-1-05 C10-2-05  1.31  1     2         1 oDOM  Control
## 10 C10-2-01 C10-3-01  0.383 2     3         1 oDOM  Control
## # ... with 278 more rows
```

2.5 Plot boxplot bNTI per DOM regime

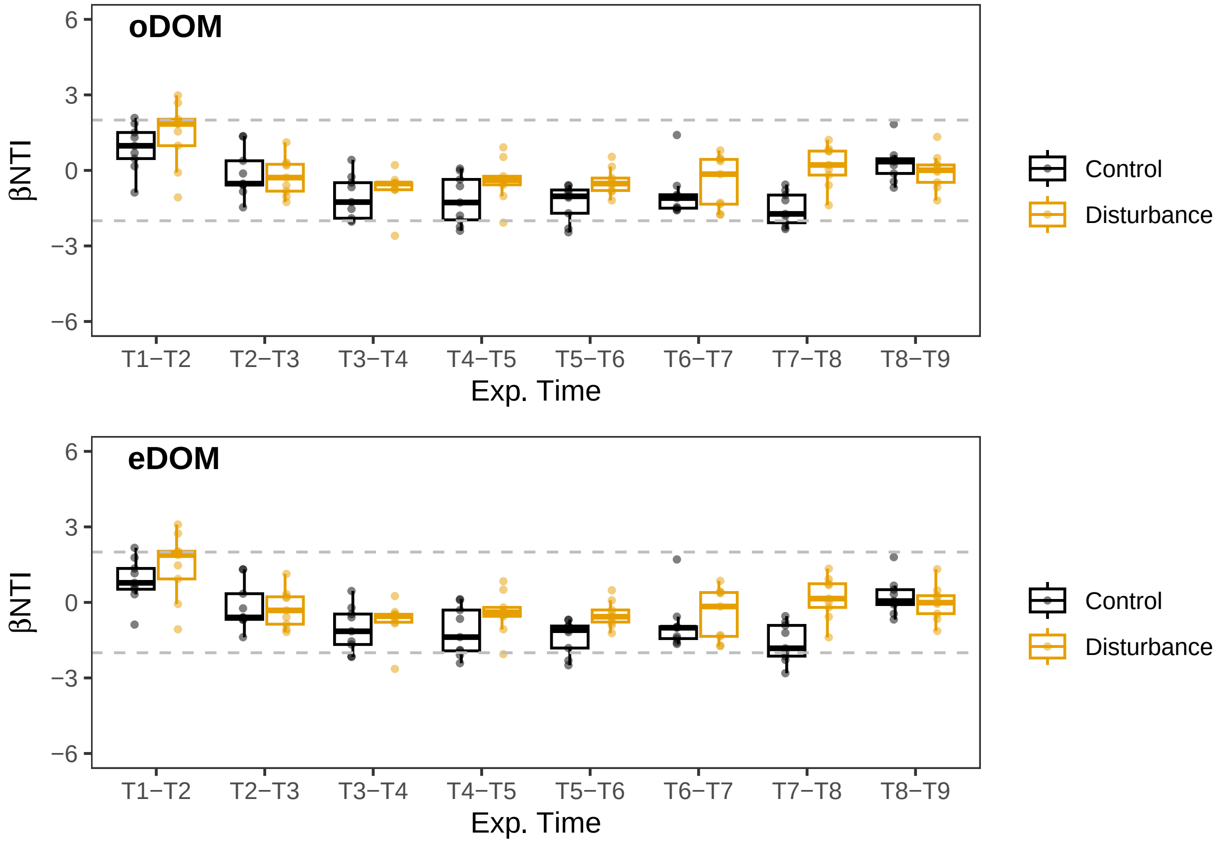
The bNTI evaluates whether the phylogenetic similarity between a pair of sample is significantly lower or higher than expected by chance relative to a reference species pool. Phylogenetic similarity surpassing the theoretical expectation (bNTI >2) indicates the prevalence of variable deterministic assembly processes during community succession. Phylogenetic similarity below the theoretical expectation (bNTI< -2), indicates the prevalence of homogeneous deterministic assembly processes during community succession. bNTIs between -2 and 2 indicate that community assembly is driven by stochastic rather than a deterministic processes.

```
# oDOM
plot.NTI.oDOM = df.NTI.all[df.NTI.all$DOM == "oDOM", ] %>%
    ggplot(aes((t2), dist, colour = (Treatment))) + geom_boxplot(outlier.size = -1,
    alpha = 0.5, size = 0.5) + geom_jitter(aes(group = interaction(t2, Treatment)),
    position = position_dodge(0.8), alpha = 0.5, size = 0.8) + ylab(expression(beta *
    NTI)) + xlab("Exp. Time") + scale_color_manual(values = cbbPalette, name = "") +
    theme_bw() + ylim(-6, 6) + geom_hline(aes(yintercept = -2), color = "grey",
    linetype = "dashed", size = 0.5) + geom_hline(aes(yintercept = 2), color = "grey",
    linetype = "dashed", size = 0.5) + theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()) + scale_x_discrete(labels = c(`2` = "T1-T2",
    `3` = "T2-T3", `4` = "T3-T4", `5` = "T4-T5", `6` = "T5-T6", `7` = "T6-T7",
    `8` = "T7-T8", `9` = "T8-T9"))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```

```
# plot.NTI.oDOM eDOM
plot.NTI.eDOM = df.NTI.all[df.NTI.all$DOM == "eDOM", ] %>%
    ggplot(aes((t2), dist, colour = (Treatment))) + geom_boxplot(outlier.size = -1,
    alpha = 0.5, size = 0.5) + geom_jitter(aes(group = interaction(t2, Treatment)),
    position = position_dodge(0.8), alpha = 0.5, size = 0.8) + ylab(expression(beta *
    NTI)) + xlab("Exp. Time") + scale_color_manual(values = cbbPalette, name = "") +
    theme_bw() + ylim(-6, 6) + geom_hline(aes(yintercept = -2), color = "grey",
    linetype = "dashed", size = 0.5) + geom_hline(aes(yintercept = 2), color = "grey",
    linetype = "dashed", size = 0.5) + theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()) + scale_x_discrete(labels = c(`2` = "T1-T2",
    `3` = "T2-T3", `4` = "T3-T4", `5` = "T4-T5", `6` = "T5-T6", `7` = "T6-T7",
    `8` = "T7-T8", `9` = "T8-T9"))

# Combined plot
plot_grid(plot.NTI.oDOM, plot.NTI.eDOM, ncol = 1, labels = c("oDOM", "eDOM"),
    label_x = 0.07, label_y = 0.97, label_size = 12)
```



2.6 Export figure

```
pdf("../figures/FigureS3_betaNTI.pdf", width = 7, height = 5)
plot_grid(plot.NTI.oDOM, plot.NTI.eDOM, ncol = 1, labels = c("oDOM", "eDOM"),
    label_x = 0.07, label_y = 0.97, label_size = 12)
dev.off()
```

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

2.7 Statistical analysis

```
df.NTI.all$Rep = rep(c(1, 2, 3), each = 3)
df.NTI.all$DOM = as.factor(df.NTI.all$DOM)
df.NTI.all$Treatment = as.factor(df.NTI.all$Treatment)
df.NTI.all$t2 = as.factor(df.NTI.all$t2)

# Testing assumptions Normality
df.NTI.all %>%
    group_by(DOM, Treatment, t2) %>%
    shapiro_test(dist)
```

```
## # A tibble: 32 x 6
##    t2    DOM   Treatment   variable statistic      p
##    <fct> <fct> <fct>       <chr>        <dbl>  <dbl>
##   1 2     HDOM  Control     dist         0.968 0.873
##   2 3     HDOM  Control     dist         0.914 0.346
##   3 4     HDOM  Control     dist         0.806 0.0236
##   4 5     HDOM  Control     dist         0.960 0.799
##   5 6     HDOM  Control     dist         0.920 0.389
##   6 7     HDOM  Control     dist         0.899 0.248
##   7 8     HDOM  Control     dist         0.957 0.763
##   8 9     HDOM  Control     dist         0.957 0.770
##   9 2     HDOM  Disturbance dist         0.955 0.742
## 10 3     HDOM  Disturbance dist         0.957 0.770
## # ... with 22 more rows
```

```
# Homogeneity of variances
df.NTI.all %>%
    group_by(t2) %>%
    levene_test(dist ~ DOM * Treatment)
```

```
## # A tibble: 8 x 5
##   t2      df1   df2 statistic     p
##   <fct> <int> <int>     <dbl> <dbl>
## 1 2         3    32     0.516 0.674
## 2 3         3    32     2.15  0.114
## 3 4         3    32     0.559 0.646
## 4 5         3    32     1.16  0.341
## 5 6          3    32      1.02  0.397
## 6 7          3    32      0.655 0.586
## 7 8          3    32      0.314 0.815
## 8 9          3    32      1.08  0.373
```

```
# Repeated measurement ANOVA
summary(aov(dist ~ DOM * Treatment * t2 + Error(Rep), data = df.NTI.all))
```

```
##
## Error: Rep
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  1 0.2117  0.2117
##
## Error: Within
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## DOM                1   3.67   3.672   3.849 0.050860 .
## Treatment          1   0.56   0.564   0.592 0.442491
## t2                 7  59.97   8.567   8.979 6.78e-10 ***
## DOM:Treatment      1  14.14  14.144  14.825 0.000149 ***
## DOM:t2             7  29.61   4.229   4.433 0.000114 ***
## Treatment:t2       7  17.98   2.568   2.692 0.010455 *
## DOM:Treatment:t2   7   5.67   0.811   0.850 0.547193
## Residuals        255 243.28   0.954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```