Figure1B

2025-08-06

Intro

Non-Metric Multidimensional Scaling (NMDS) performed on the Bray-Curtis compositional dissimilarity matrices of Operational Protein Units (OPUs), to reproduce the analyses and the Figure 1A from the original publication Seasonal dynamics of the coastal microbiome and its association with environmental factors.

1. Set the environment

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-8
```

```
library(ggpubr)
library(corrgram)
```

```
##
## Attaching package: 'corrgram'
##
## The following object is masked from 'package:lattice':
##
##     panel.fill
```

```
library(lubridate)
library(grid)
```

2. Load data

```
ABUND <- readr::read_tsv("../data/opus_workable.tsv", show_col_types = FALSE)
METADATA <- readr::read_tsv("../data/metadata_workable.tsv", show_col_types = FALSE)
```

3. Format abundance table

Because the OPU abundance table is in long format (to speed up import), we need to convert it back to wide format.

```
ABUND <- ABUND %>%
  tidyr::pivot_wider(names_from = opu_id, values_from = abundance) %>%
  tibble::column_to_rownames("Date") %>%
  as.matrix()
```

3. Run NMDS

NMDS is performed on Bray–Curtis dissimilarities computed from Hellinger-transformed OPU abundance profiles.

```
ABUND_norm <- vegan::decostand(ABUND, method = "hellinger")
set.seed(123)
ABUND_norm_nmds <- vegan::metaMDS(ABUND_norm, k = 2, try = 50, trymax = 50, threshold = 0.9)
```

```
## Run 0 stress 0.1759656
## Run 1 stress 0.1725844
## ... New best solution
## ... Procrustes: rmse 0.06349945  max resid 0.2089412
## Run 2 stress 0.1687237
## ... New best solution
## ... Procrustes: rmse 0.1273724  max resid 0.5343585
## Run 3 stress 0.1682768
## ... New best solution
## ... Procrustes: rmse 0.01184472  max resid 0.0462124
## Run 4 stress 0.1725844
## Run 5 stress 0.1750414
## Run 6 stress 0.1682768
## ... New best solution
## ... Procrustes: rmse 5.858705e-06  max resid 1.428457e-05
## ... Similar to previous best
## Run 7 stress 0.1682768
## ... Procrustes: rmse 4.732532e-05  max resid 0.0001709377
## ... Similar to previous best
## Run 8 stress 0.1682768
## ... Procrustes: rmse 1.560506e-05  max resid 4.755194e-05
## ... Similar to previous best
## Run 9 stress 0.1753588
## Run 10 stress 0.1688158
## Run 11 stress 0.1750414
## Run 12 stress 0.1682769
## ... Procrustes: rmse 2.740805e-05  max resid 7.00654e-05
## ... Similar to previous best
## Run 13 stress 0.1734958
## Run 14 stress 0.1725844
## Run 15 stress 0.1688158
## Run 16 stress 0.1685561
## ... Procrustes: rmse 0.02410741  max resid 0.07368704
## Run 17 stress 0.1725844
## Run 18 stress 0.1685561
## ... Procrustes: rmse 0.02410765  max resid 0.07367428
## Run 19 stress 0.1750414
```

```
## Run 20 stress 0.1759655
## Run 21 stress 0.1773317
## Run 22 stress 0.1734958
## Run 23 stress 0.1753587
## Run 24 stress 0.1688158
## Run 25 stress 0.1759655
## Run 26 stress 0.1759655
## Run 27 stress 0.1753587
## Run 28 stress 0.1725844
## Run 29 stress 0.178165
## Run 30 stress 0.2098772
## Run 31 stress 0.1767414
## Run 32 stress 0.173496
## Run 33 stress 0.1682768
## ... Procrustes: rmse 4.19164e-06  max resid 1.576108e-05
## ... Similar to previous best
## Run 34 stress 0.1734958
## Run 35 stress 0.178165
## Run 36 stress 0.1734958
## Run 37 stress 0.1750414
## Run 38 stress 0.1682768
## ... Procrustes: rmse 6.700011e-06  max resid 2.522765e-05
## ... Similar to previous best
## Run 39 stress 0.1682768
## ... Procrustes: rmse 3.524255e-05  max resid 0.0001320547
## ... Similar to previous best
## Run 40 stress 0.1753587
## Run 41 stress 0.1682768
## ... Procrustes: rmse 1.399582e-05  max resid 3.780497e-05
## ... Similar to previous best
## Run 42 stress 0.1688158
## Run 43 stress 0.1725844
## Run 44 stress 0.1759655
## Run 45 stress 0.192892
## Run 46 stress 0.1682768
## ... Procrustes: rmse 3.165445e-05  max resid 0.000110926
## ... Similar to previous best
## Run 47 stress 0.178165
## Run 48 stress 0.178165
## Run 49 stress 0.178165
## Run 50 stress 0.1682768
## ... Procrustes: rmse 8.605362e-06  max resid 2.957725e-05
## ... Similar to previous best
## *** Best solution repeated 10 times
```

4. Merge tables to plot

```
ABUND_norm_nmds_ext <- ABUND_norm_nmds$points %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Date") %>%
  dplyr::mutate(Date = as.Date(Date)) %>%
  dplyr::left_join(METADATA, by = "Date")

# Factor order for seasons
```

```r
ABUND_norm_nmds_ext$Season <- factor(ABUND_norm_nmds_ext$Season,
                                     levels = c("Summer", "Autumn", "Winter", "Spring"))
```

5. Add centroid coordinates

The centroid coordinates for each season are calculated to draw segments connecting the centroids.

```r
ctd_coords <- ABUND_norm_nmds_ext %>%
  dplyr::mutate(Year = lubridate::year(Date)) %>%
  dplyr::group_by(Season) %>%
  dplyr::summarize(mean_nmds1 = mean(MDS1), mean_nmds2 = mean(MDS2), .groups = "drop") %>%
  dplyr::arrange(Season)

ctd_coords$segment_x    <- NA
ctd_coords$segment_y    <- NA
ctd_coords$segment_xend <- NA
ctd_coords$segment_yend <- NA

for (i in 1:(nrow(ctd_coords) - 1)) {
  ctd_coords$segment_x[i]    <- ctd_coords$mean_nmds1[i]
  ctd_coords$segment_y[i]    <- ctd_coords$mean_nmds2[i]
  ctd_coords$segment_xend[i] <- ctd_coords$mean_nmds1[i + 1]
  ctd_coords$segment_yend[i] <- ctd_coords$mean_nmds2[i + 1]
}

i_last <- nrow(ctd_coords)
ctd_coords$segment_x[i_last]    <- ctd_coords$mean_nmds1[i_last]
ctd_coords$segment_y[i_last]    <- ctd_coords$mean_nmds2[i_last]
ctd_coords$segment_xend[i_last] <- ABUND_norm_nmds_ext %>%
  dplyr::filter(Season == "Summer") %>% dplyr::summarize(mean_nmds1 = mean(MDS1)) %>% dplyr::pull(mean_
ctd_coords$segment_yend[i_last] <- ABUND_norm_nmds_ext %>%
  dplyr::filter(Season == "Summer") %>% dplyr::summarize(mean_nmds2 = mean(MDS2)) %>% dplyr::pull(mean_
```

7. Plot: MDS1 vs MDS2

```r
x_title <- "MDS1"; y_title <- "MDS2"
x_max <- max(ABUND_norm_nmds_ext$MDS1) + max(ABUND_norm_nmds_ext$MDS1)/10
x_min <- min(ABUND_norm_nmds_ext$MDS1) - abs(min(ABUND_norm_nmds_ext$MDS1)/10)
y_max <- max(ABUND_norm_nmds_ext$MDS2) + max(ABUND_norm_nmds_ext$MDS2)/10
y_min <- min(ABUND_norm_nmds_ext$MDS2) - abs(min(ABUND_norm_nmds_ext$MDS2)/10)

text_size <- 13
season_colors <- c("#c93f1b", "#98482b", "#154360", "#3c7810")
nmds_plot_margin <- c(0.2, 1, 0.2, 0.2)

nmds_plot_mds1_vs_mds2 <- ggplot(ABUND_norm_nmds_ext, aes(MDS1, MDS2, color = Season), alpha = 0.8) +
  geom_point(alpha = 0.95, size = 3) +
  scale_color_manual(values = season_colors) +
  geom_vline(xintercept = 0, color = "gray40", linetype = "dotted") +
  geom_hline(yintercept = 0, color = "gray40", linetype = "dotted") +
  xlab(x_title) + ylab(y_title) +
  theme_bw() +
  theme(
    axis.title = element_text(size = text_size + 4),
```

```
    axis.text  = element_text(size = text_size),
    legend.text  = element_text(size = text_size),
    legend.title = element_text(size = text_size),
    plot.margin  = grid::unit(nmds_plot_margin, "lines")
  ) +
  guides(shape = guide_legend(override.aes = list(size = 3)),
         color = guide_legend(override.aes = list(shape = 19, size = 3))) +
  xlim(c(x_min, x_max)) + ylim(c(y_min, y_max)) +
  annotate("text", label = paste("Stress:", round(ABUND_norm_nmds$stress, 3)),
           x = -1.4, y = 0.85, size = 3) +
  geom_segment(data = ctd_coords,
               aes(x = segment_x, y = segment_y, xend = segment_xend, yend = segment_yend),
               arrow = arrow(length = grid::unit(0.1, "inches")),
               color = "gray10", alpha = 0.7, linewidth = 1) +
  geom_text(data = ctd_coords, aes(x = mean_nmds1, y = mean_nmds2),
            label = c("Summer", "Autumn", "Winter", "Spring"),
            vjust = c(-0.5, 0, 1.4, 0), hjust = c(0.5, 1.2, 0, -0.3),
            fontface = "bold", size = 2, show.legend = FALSE) +
  geom_text(aes(label = Date), vjust = 2, size = 1.5, check_overlap = TRUE, show.legend = FALSE)
```

8. Create box plot for MDS1

```
ABUND_norm_nmds_ext$Season_rev <- factor(ABUND_norm_nmds_ext$Season,
                                         levels = c("Spring", "Winter", "Autumn", "Summer"))

x_title_boxplot_mds1 <- x_title
y_title_boxplot_mds1 <- "Season"
boxplot_margin_pc1 <- c(0.2, 7.5, 0.2, 4.85)

boxplot_mds1_vs_season <- ggplot(ABUND_norm_nmds_ext, aes(x = MDS1, y = Season_rev, fill = Season_rev))
  geom_boxplot() +
  scale_fill_manual(values = rev(season_colors), name = "Season") +
  xlab(x_title_boxplot_mds1) + ylab(y_title_boxplot_mds1) +
  theme_bw() +
  theme(
    axis.title.x = element_text(size = text_size + 4, margin = grid::unit(c(4,0,0,0), "mm")),
    axis.title.y = element_text(size = text_size + 1, margin = grid::unit(c(0,2,0,0), "mm")),
    axis.text.x  = element_text(size = text_size),
    axis.text.y  = element_text(size = text_size - 3),
    legend.text  = element_text(size = text_size),
    legend.title = element_text(size = text_size),
    plot.margin  = grid::unit(boxplot_margin_pc1, "lines")
  ) +
  xlim(c(x_min, x_max)) +
  geom_segment(aes(x = 0.7, y = 1.5, xend = 0.7, yend = 3.5),
               color = "gray10", alpha = 0.7, linewidth = 0.1) +
  annotate("text", x = 0.75, y = 2.5, label = "***", size = 2, angle = 90)
```

9. Create boxplot for MDS2

```
x_title_boxplot_mds2 <- "Season"
y_title_boxplot_mds2 <- y_title
boxplot_margin_pc2 <- c(0.2, 0.2, 0.2, 0.2)
```

```
boxplot_mds2_vs_season <- ggplot(ABUND_norm_nmds_ext, aes(x = Season_rev, y = MDS2, fill = Season_rev))
  geom_boxplot() +
  scale_fill_manual(values = rev(season_colors), name = "Season") +
  xlab(x_title_boxplot_mds2) + ylab(y_title_boxplot_mds2) +
  theme_bw() +
  theme(
    axis.title.x = element_text(size = text_size + 4, margin = grid::unit(c(4,0,0,0), "mm")),
    axis.title.y = element_text(size = text_size + 4, margin = grid::unit(c(0,2,0,0), "mm")),
    axis.text    = element_text(size = text_size),
    legend.text  = element_text(size = text_size),
    legend.title = element_text(size = text_size),
    plot.margin  = grid::unit(boxplot_margin_pc2, "lines")
  ) +
  ylim(c(y_min, y_max))
```

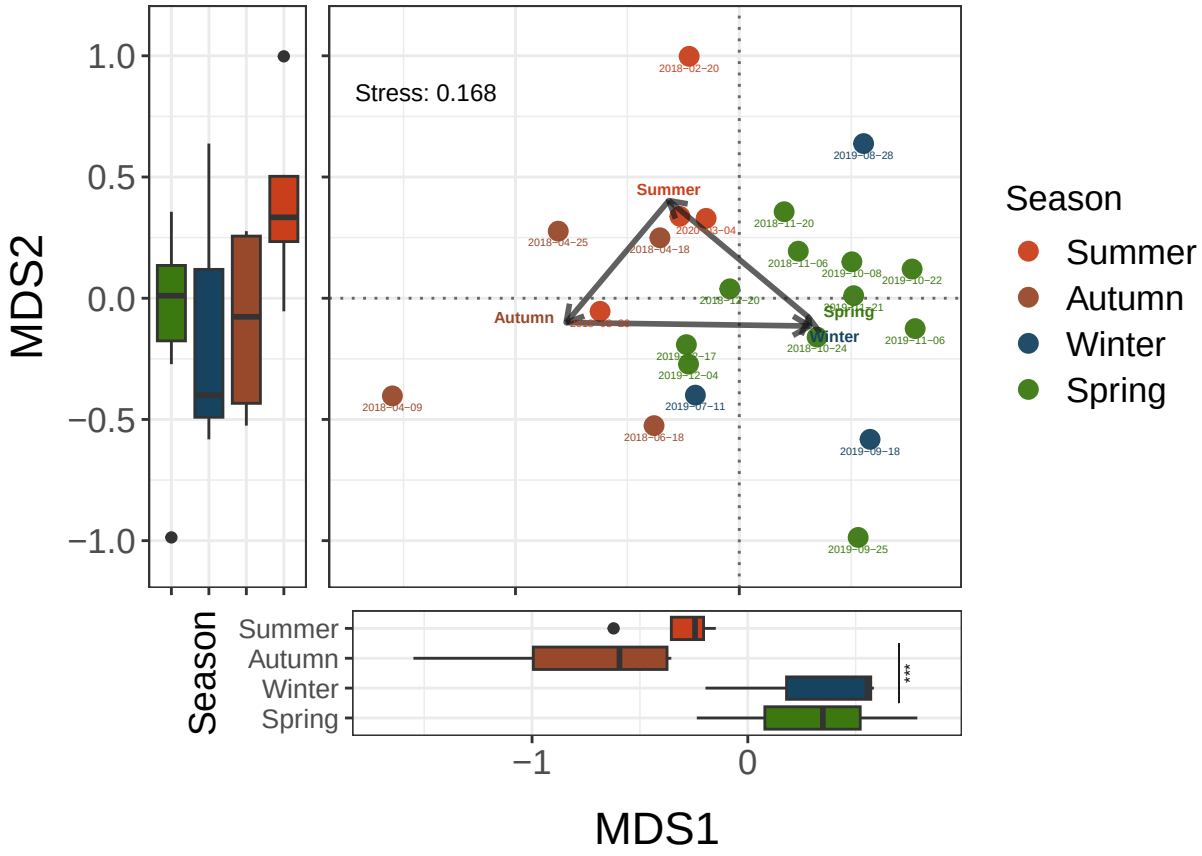
10. Merge plots

```
nmds_plot_and_mds1_boxplot <- ggpubr::ggarrange(
  boxplot_mds2_vs_season + ggpubr::rremove("legend") + ggpubr::rremove("x.title") + ggpubr::rremove("x.
  nmds_plot_mds1_vs_mds2 + ggpubr::rremove("x.title") + ggpubr::rremove("x.text") + ggpubr::rremove("y.
  ncol = 2, nrow = 1, widths = c(0.25, 0.75)
)

nmds_plot_and_mds1_and_mds2_boxplot <- ggpubr::ggarrange(
  nmds_plot_and_mds1_boxplot,
  boxplot_mds1_vs_season + ggpubr::rremove("legend"),
  nrow = 2, ncol = 1, heights = c(0.70, 0.3)
)
```

11. Visualize plot

```
nmds_plot_and_mds1_and_mds2_boxplot
```



12. Perform ANOVA Welch test: MDS1

```
ABUND_norm_nmds_ext$Semester <- ifelse(ABUND_norm_nmds_ext$Season %in% c("Winter","Spring"),
                                       "Winter-Spring", "Autumn-Summer")
anova_welch_mds1 <- oneway.test(MDS1 ~ Semester, data = ABUND_norm_nmds_ext, var.equal = FALSE)
anova_welch_mds1
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  MDS1 and Semester
## F = 20.42, num df = 1.00, denom df = 11.95, p-value = 0.0007106
```

13. Perform ANOVA Welch test: MDS2

```
ABUND_norm_nmds_ext$Semester2 <- ifelse(ABUND_norm_nmds_ext$Season %in% c("Summer","Spring"),
                                        "Summer-Spring", "Autumn-Winter")
anova_welch_mds2 <- oneway.test(MDS2 ~ Semester2, data = ABUND_norm_nmds_ext, var.equal = FALSE)
anova_welch_mds2
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  MDS2 and Semester2
## F = 0.53877, num df = 1.000, denom df = 10.491, p-value = 0.479
```