Figure4B

Emiliano Pereira

2025-08-06

Intro

Functions differentially associated between semesters based on the IndVal analysis to reproduce the results of Figure 4B from the original publication Seasonal dynamics of the coastal microbiome and its association with environmental factors.

1. Set the environment

```
library(tidyverse)
library(vegan)
library(doParallel)
library(indicspecies)
source("../scripts/resources/rare_indval.R")
```

2. Load data

`opus2ko_non_rare_long.tsv.gz` contains the non-rarefied OPU abundance profiles, in a long format, having each OPU annotated against the KOfam database.
`metagenomic_sample_name2date.tsv` is table mapping the sample name of the metagenomic samples and the date. `date2season2community.tsv` is a table mapping the date, season, and community columns. `KO_class.csv` contains a custom coarse grained functional classification of KOs. `ko2desc.tsv` tables describing the functions of each KO, as provided by the KOfam database.

```
KO_TBL <- read_tsv("../data/opus2ko_non_rare_long.tsv.gz", show_col_types = FALSE)
SAMPLENAME2DATE <- read_tsv("../data/metagenomic_sample_name2date.tsv", show_col_types = FALSE)
DATE2SEASON2COMMUITY <- read_tsv("../data/date2season2community.tsv", show_col_types = FALSE)
KO_CLASS <- read_tsv("../data/KO_class.csv", show_col_types = FALSE)
KO2DESC <- read_tsv("../data/kofam/ko2desc.tsv", col_names = F, show_col_types = FALSE)
colnames(KO2DESC) <- c("KO_ID", "Code", "desc_ko")
```

3. Collapse keeping only necessary columns

```
KO_TBL_collapsed <- KO_TBL %>%
                    group_by(KO_ID, sample_name) %>%
                    summarize(abund = sum(abund))
```

```
## `summarise()` has grouped output by 'KO_ID'. You can override using the `.groups` argument.
```

4. Map dates and semester

```
KO_TBL_collapsed_ext <- left_join(x = KO_TBL_collapsed,
                                  y = SAMPLENAME2DATE,
                                  by = "sample_name") %>%
```

```
                        left_join(x = .,
                                  y = DATE2SEASON2COMMUITY,
                                  by = "Date")
```

5. Convert to wide format

```
KO_TBL_collapsed_ext_wide <- KO_TBL_collapsed_ext %>%
                             dplyr::select(KO_ID, Date, abund) %>%
                             pivot_wider(names_from = KO_ID,
                                         values_from = abund,
                                         values_fill = 0) %>%
                             column_to_rownames("Date")
dim(KO_TBL_collapsed_ext_wide)
```

```
## [1]   22 9214
```

7. Round values to integers

```
# Set the number of cores to use for parallel processing;
# this should be adjusted to the local computational capacitites.
n_cores <- 44
doParallel::registerDoParallel(n_cores)
n <- dim(KO_TBL_collapsed_ext_wide)[2]

abund_ko_int_list <- foreach(i = 1:n) %dopar% round(KO_TBL_collapsed_ext_wide[,i], 0)
KO_TBL_int <- do.call("cbind", abund_ko_int_list)
colnames(KO_TBL_int) <- colnames(KO_TBL_collapsed_ext_wide)
rownames(KO_TBL_int) <- rownames(KO_TBL_collapsed_ext_wide)
```

8. Define variables to run IndVal over iterated rarefactions

```
# Set the number of cores to use for parallel processing;
# this should be adjusted to the local computational capacitites.
n_cores <- 44
doParallel::registerDoParallel(n_cores)

abundance_table <- KO_TBL_int
sample_size <- rowSums(abundance_table) %>% min()
semester <- DATE2SEASON2COMMUITY %>%
            dplyr::select(Date, Community) %>%
            column_to_rownames("Date")

semester_ord <- semester[rownames(abundance_table),, drop = F]
#Sanity check
all(rownames(semester_ord) == rownames(abundance_table))
```

```
## [1] TRUE
```

```
vector_group <- semester_ord$Community
```

9. Run IndVal over iterated rarefactions

To accelerate the execution of the following function, we will use the paramters nperm = 99. However, to reproduce the results from the original publication, this parameter should be set to 9999.

```
rare_indval_output <- rare_indval(iterations = 100,
                                  abundance_table = abundance_table,
                                  sample_size = sample_size,
                                  vector_group = vector_group,
                                  nperm = 99,
                                  p_value = 1e-2)
```

```
## `summarise()` has grouped output by 'asv_id'. You can override using the `.groups` argument.
```

10. Select KOs with a 80% consistency or greater

```
rare_indval_output_s1_80 <- rare_indval_output %>%
                            filter(consistency > 80 & semester == "S1")

rare_indval_output_s2_80 <- rare_indval_output %>%
                            filter(consistency > 80 & semester == "S2")

# Check dimensions
dim(rare_indval_output_s1_80)
```

```
## [1] 1 5
```

```
dim(rare_indval_output_s2_80)
```

```
## [1] 7 5
```

11. KO classification

```
indval_output_df_sig_s1_80_wdesc <- rare_indval_output_s1_80 %>%
                                rename(KO_ID = asv_id) %>%
                                left_join(x = .,
                                          y = KO_CLASS,
                                          by = "KO_ID") %>%
                                left_join(x = .,
                                          y = KO2DESC,
                                          by = "KO_ID")


indval_output_df_sig_s2_80_wdesc <- rare_indval_output_s2_80 %>%
                                   rename(KO_ID = asv_id) %>%
                                   left_join(x = .,
                                             y = KO_CLASS,
                                             by = "KO_ID") %>%
                                   left_join(x = .,
                                             y = KO2DESC,
                                             by = "KO_ID")
```

12. Map abundance

```
sample_min <- rowSums(KO_TBL_int) %>% min()
ABUND <- rrarefy(KO_TBL_int, sample_min)

samples_s1 <- semester_ord %>% filter(Community == "S1") %>% rownames()
samples_s2 <- semester_ord %>% filter(Community == "S2") %>% rownames()
```

```
ABUND_s1 <- ABUND[samples_s1,]
ABUND_s2 <- ABUND[samples_s2,]

ABUND_s1 <- ABUND_s1[, colSums(ABUND_s1) > 0]
ABUND_s2 <- ABUND_s2[, colSums(ABUND_s2) > 0]

kos_abund_s1 <- data.frame(ko_abund_rel = colSums(ABUND_s1)/sum(colSums(ABUND_s1)),
                           KO_ID = colnames(ABUND_s1))
kos_abund_s2 <- data.frame(ko_abund_rel = colSums(ABUND_s2)/sum(colSums(ABUND_s2)),
                           KO_ID = colnames(ABUND_s2))

indval_output_df_sig_s1_80_wdesc_wabund <- left_join(x = indval_output_df_sig_s1_80_wdesc,
                                                     y = kos_abund_s1,
                                                     by = "KO_ID")

indval_output_df_sig_s2_80_wdesc_wabund <- left_join(x = indval_output_df_sig_s2_80_wdesc,
                                                     y = kos_abund_s2,
                                                     by = "KO_ID")
```

14. Define KO Class colors

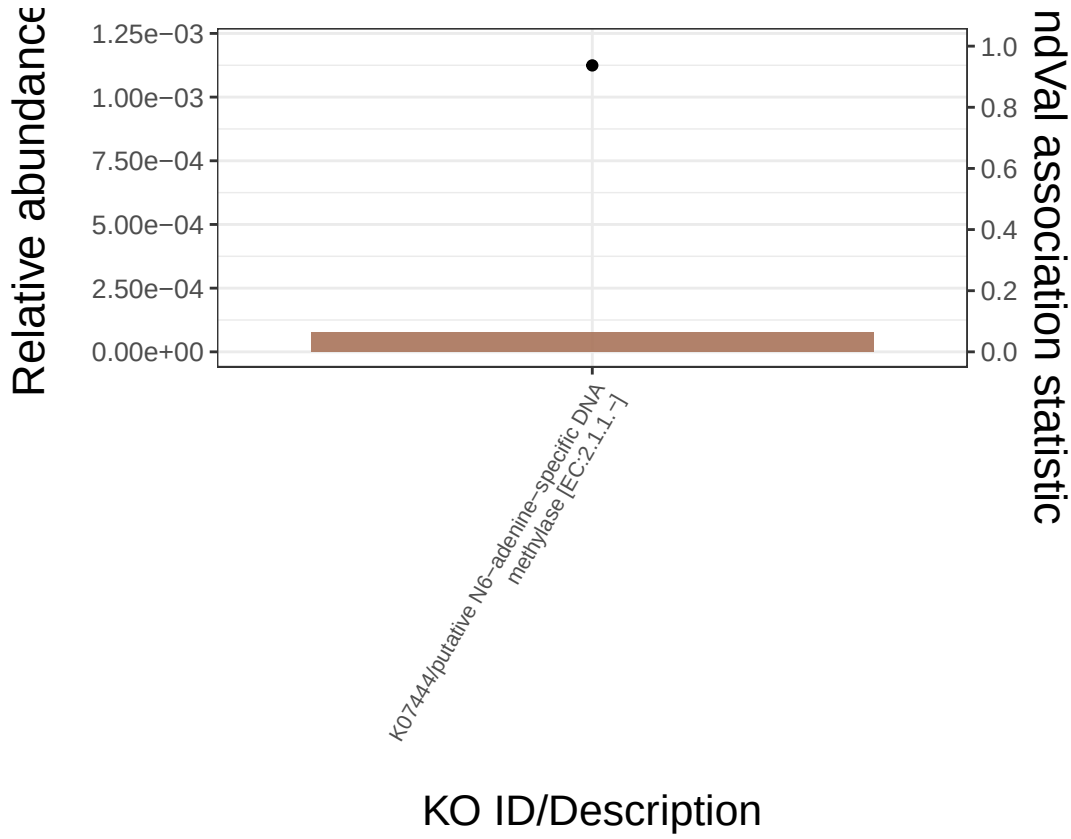
```
ko_class <- c(indval_output_df_sig_s1_80_wdesc$KO_class, indval_output_df_sig_s2_80_wdesc$KO_class) %>%

ko_class_palette <- c(
  "#8B3E2F",  # earthy red (burnt sienna)
  "#C97E2A",  # clay orange (ochre)
  "#A9745A",  # canyon brown (raw umber)
  "#4A6C59",  # moss green (pine)
  "#5F9EA0", # Cadet Blue
  "#6A8BAA",   # muted blue (mineral water)
  "#3E5F9E",  # slate blue (dusty indigo)
  "#6D597A",  # dusty purple (dried lavender)
  "#483D8B" # Dark Slate Blue

)

names(ko_class_palette) <- ko_class[order(ko_class)]
indval_output_df_sig_s1_80_wdesc_wabund$color <- ko_class_palette[indval_output_df_sig_s1_80_wdesc_wabu
indval_output_df_sig_s2_80_wdesc_wabund$color <- ko_class_palette[indval_output_df_sig_s2_80_wdesc_wabu
```

15. Create barplots: S1

```
scaling_factor_s1 <- 1.2e-3
text_size <- 10

# format axis labels
indval_output_df_sig_s1_80_wdesc_wabund$labels <- paste(indval_output_df_sig_s1_80_wdesc_wabund$KO_ID,
                                                        indval_output_df_sig_s1_80_wdesc_wabund$desc_k
                                                        sep = "/") %>%
                                                  sub("(([^ ]+ +){2}[^ ]+) +", "\\1\n", .)

barplot_kos_indval_s1 <- indval_output_df_sig_s1_80_wdesc_wabund %>%
                         ggplot(.,
```

```
                                aes(x = labels,
                                    y = ko_abund_rel,
                                    fill = KO_class)) +
                                geom_bar(stat = "identity", alpha = 0.9) +
                                scale_fill_manual(values = indval_output_df_sig_s1_80_wdesc_wabund$col
                                geom_line(aes(x = labels,
                                              y = stat_mean*scaling_factor_s1,
                                              group = 1), color = "black") +
                                geom_point(aes(x = labels,
                                               y = stat_mean*scaling_factor_s1,
                                               group = 1), color = "black") +
                                scale_y_continuous(
                                                   limits  = c(0, scaling_factor_s1 + 1e-5),
                                                   labels = function(x) format(x, scientific = TRUE),
                                                   name = "Relative abundance",
                                                   sec.axis = sec_axis(~ . / scaling_factor_s1,
                                                                       name = "Mean IndVal association
                                                                       breaks = seq(0, 1, 0.2))) +
                               xlab("KO ID/Description") +
                               theme_bw() +
                               theme(
                                 axis.text.x = element_text(size = text_size-2, angle = 60, hjust = 1)
                                 axis.text.y = element_text(size = text_size),
                                 axis.title.x = element_text(size = text_size +6, margin = unit(c(5,0,
                                 axis.title.y = element_text(size = text_size +7, margin = unit(c(0,5,
                                 plot.title = element_text(size = text_size+4, hjust = 0.5),
                                 strip.text = element_text(size = text_size+4),
                                 strip.background = element_blank(),
                                 plot.margin = margin(r = 1, l = 4, t= 0.5,b = 1, unit = "lines")
                               )   +  # remove legend
                               theme(legend.position = "none")

barplot_kos_indval_s1
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



16. Create barplots: S2

```
scaling_factor_s2 <- 9.5e-4
text_size <- 10

indval_output_df_sig_s2_80_wdesc_wabund %>% filter(KO_ID == "K02497") %>% pull(desc_ko)
```

```
## [1] "HemX protein"
```

```
# format axis labels
indval_output_df_sig_s2_80_wdesc_wabund$labels <- paste(indval_output_df_sig_s2_80_wdesc_wabund$KO_ID,
                                                        indval_output_df_sig_s2_80_wdesc_wabund$desc_ko
                                                        sep = "/") %>%
                                                  sub("(([^ ]+ +){2}[^ ]+) +", "\\1\n", .)

barplot_kos_indval_s2 <- indval_output_df_sig_s2_80_wdesc_wabund %>%
                         ggplot(.,
                                aes(x = labels,
                                    y = ko_abund_rel,
                                    fill = KO_class)) +
                         geom_bar(stat = "identity", alpha = 0.9) +
                         scale_fill_manual(values = indval_output_df_sig_s2_80_wdesc_wabund$color) +
                         geom_line(aes(x = labels,
                                       y = stat_mean*scaling_factor_s2,
                                       group = 1), color = "black") +
                         geom_point(aes(x = labels,
```

```
                      y = stat_mean*scaling_factor_s2,
                      group = 1), color = "black") +
  scale_y_continuous(
    limits  = c(0, scaling_factor_s2),
    labels = function(x) format(x, scientific = TRUE),
    name = "Relative abundance",
    sec.axis = sec_axis(~ . / scaling_factor_s2,
                        name = "Mean IndVal association statistic",
                        breaks = seq(0, 1, 0.2))) +
  xlab("KO ID/Description") +
  theme_bw() +
  theme(
    axis.text.x = element_text(size = text_size-2, angle = 60, hjust = 1),
    axis.text.y = element_text(size = text_size),
    axis.title.x = element_text(size = text_size +6, margin = unit(c(5,0,0,0),
    axis.title.y = element_text(size = text_size +7, margin = unit(c(0,5,0,0),
    plot.title = element_text(size = text_size+4, hjust = 0.5),
    strip.text = element_text(size = text_size+4),
    strip.background = element_blank(),
    plot.margin = margin(r = 1, l = 2.5, t= 0.5,b = 1, unit = "lines")
  )   # +
  # theme(legend.position = "none")
```

```
barplot_kos_indval_s2
```