

R version 4.5.0 (2025-04-11) -- "How About a Twenty-Six"

Copyright (C) 2025 The R Foundation for Statistical Computing

Platform: aarch64-apple-darwin20

#####Permanova Analysis#####

#load packages

> library(vegan)

This is vegan 2.7-2

> library(dplyr)

> library(permute)

#import distance matrix into R from Qiime as .tsv

> bray <- read.table("bray-distance-matrix.tsv",

+ header = TRUE,

+ row.names = 1,

+ sep = "\t",

+ check.names = FALSE)

> jaccard <- read.table("jaccard-distance-matrix.tsv",

+ header = TRUE,

+ row.names = 1,

+ sep = "\t",

+ check.names = FALSE)

> unweighted<- read.table("unweighted-distance-matrix.tsv",

+ header = TRUE,

```
+      row.names = 1,  
+      sep = "\t",  
+      check.names = FALSE)
```

```
#import metadata
```

```
> meta <- read.table("InvasMeta.tsv",  
+      header = TRUE,  
+      row.names = 1,  
+      sep = "\t",  
+      check.names = FALSE)
```

```
#make sure all rows are in both files
```

```
> all(rownames(meta) %in% rownames(bray))
```

```
[1] TRUE
```

```
> all(rownames(bray) %in% rownames(meta))
```

```
[1] TRUE
```

```
# restrict permutations to site to remove effect of site
```

```
> ctrl <- how(nperm = 9999)
```

```
> setBlocks(ctrl) <- meta$site
```

```
# run permanova
```

```
> set.seed(42)
```

```
> permanova_fixed <- adonis2(  
+   bray ~ Inoculant,  
+   data = meta,  
+   permutations = ctrl,
```

```
+ by      = "margin"
```

```
+ )
```

```
> permanova_fixed
```

```
Permutation test for adonis under reduced model
```

```
Marginal effects of terms
```

```
Blocks: meta$site
```

```
Permutation: free
```

```
Number of permutations: 9999
```

```
adonis2(formula = bray ~ Inoculant, data = meta, permutations = ctrl, by = "margin")
```

```
      Df SumOfSqs    R2    F Pr(>F)
```

```
Inoculant  1 0.018528 0.12497 1.4282 0.3356
```

```
Residual 10 0.129731 0.87503
```

```
Total    11 0.148259 1.00000
```

```
#####Partial Mantel Test#####
```

```
#load packages
```

```
library(vegan)
```

```
library(dplyr)
```

```
library(permute)
```

```
#read metadata, with site as numeric
```

```
> meta <- read.table("InvasMetaMantel.txt",
```

```
+      header = TRUE,
```

```
+      row.names = 1,
```

```

+         sep = "\t",
+         check.names = FALSE)

# Convert the column to numeric (in case it's a character or factor)
> meta$CopiesPerGSoil <- as.numeric(meta$CopiesPerGSoil)

# Create Euclidean distance from that numeric variable
> copies_dist <- dist(meta$CopiesPerGSoil, method = "euclidean")

# Verify its a distance matrix
> inherits(copies_dist, "dist")

```

```
[1] TRUE
```

```

#import bc distance matrix

> bray <- read.table("bray-distance-matrix.tsv",
+         header = TRUE,
+         row.names = 1,
+         sep = "\t",
+         check.names = FALSE)

#control for site with partial mantel,

#make sure "site" is numeric
> meta$site <- as.numeric(meta$site)

#make a "site" distance matrix
> site_dist <- dist(meta$site, method = "euclidean")

#make sure it's a distance matrix
> inherits(site_dist, "dist")

```

```
[1] TRUE
```

```
#run partial mantel
```

```
set.seed(123)
```

```
> mantel.partial(bray, copies_dist, site_dist, permutations = 9999)
```

Partial Mantel statistic based on Pearson's product-moment correlation

Call:

```
mantel.partial(xdis = bray, ydis = copies_dist, zdis = site_dist, permutations = 9999)
```

Mantel statistic r: -0.1775

Significance: 0.8196

Upper quantiles of permutations (null model):

90% 95% 97.5% 99%

0.378 0.459 0.578 0.805

Permutation: free

#####Indicator analysis#####

```
> library(indicspecies)
```

Loading required package: permute

```
> library(permute)
```

```
> library(tidyverse)
```

— Attaching core tidyverse packages

tidyverse 2.0.0 —

✓ dplyr 1.1.4 ✓ readr 2.1.5

✓ forcats 1.0.1 ✓ stringr 1.5.2

✓ ggplot2 4.0.0 ✓ tibble 3.3.0

✓ lubridate 1.9.4 ✓ tidyr 1.3.1


✓ purrr 1.1.0

— Conflicts —

tidyverse_conflicts() —

✗ dplyr::filter() masks stats::filter()

✗ dplyr::lag() masks stats::lag()

 Use the conflicted package to force all conflicts to become errors

```
> library(dplyr)
```

#read metadata

```
> meta <- read.table("InvasMeta.txt",
```

```
+       header = TRUE,
```

```
+       row.names = 1,
```

```
+       sep = "\t",
```

```
+       check.names = FALSE,
```

```
+       comment.char = "",
```

```
+ stringsAsFactors = FALSE)
```

```
#read feature table
```

```
> ft_raw <- read.table("exported-collapsed-table-L7-m2.txt",
```

```
+ header = TRUE,
```

```
+ row.names = 1,
```

```
+ sep = "\t",
```

```
+ check.names = FALSE,
```

```
+ comment.char = "")
```

```
#ensure samples x taxa
```

```
> clean_ids <- function(x) {
```

```
+ x <- trimws(x)
```

```
+ x <- gsub("\u00A0", " ", x) # non-breaking space
```

```
+ x <- gsub("\\s+", " ", x)
```

```
+ x
```

```
+ }
```

```
> rownames(meta) <- clean_ids(rownames(meta))
```

```
> rownames(ft_raw) <- clean_ids(rownames(ft_raw))
```

```
> colnames(ft_raw) <- clean_ids(colnames(ft_raw))
```

```
#If samples are columns (QIIME export), transpose to samples x taxa
```

```
> match_rows <- sum(rownames(ft_raw) %in% rownames(meta))
```

```
> match_cols <- sum(colnames(ft_raw) %in% rownames(meta))
```

```
> FT <- if (match_cols > match_rows) t(as.matrix(ft_raw)) else as.matrix(ft_raw)
```

```
> mode(FT) <- "numeric"
```

```
#align samples
```

```
> keep_ids <- intersect(rownames(FT), rownames(meta))
```

```
> if (length(keep_ids) < 2L) stop("No overlapping sample IDs between FT and metadata.")
```

```
> FT <- FT[keep_ids, , drop = FALSE]
```

```
> meta <- meta[keep_ids, , drop = FALSE]
```

```
#build groups from CopiesPerGSoil (Low/High)
```

```
# make numeric safely (strip commas if present)
```

```
> meta$CopiesPerGSoil <- as.numeric(gsub(",", "", as.character(meta$CopiesPerGSoil)))
```

```
> if (all(is.na(meta$CopiesPerGSoil))) stop("CopiesPerGSoil is all NA after numeric conversion.")
```

```
> group <- cut(meta$CopiesPerGSoil,
```

```
+     breaks = c(-Inf, 25000, Inf),
```

```
+     labels = c("Low", "High"),
```

```
+     include.lowest = TRUE)
```

```
#drop samples with NA group (if any)
```

```
> has_group <- !is.na(group)
```

```
> FT <- FT[has_group, , drop = FALSE]
```

```
> meta <- meta[has_group, , drop = FALSE]
```

```
> group <- droplevels(group[has_group])
```

```
> if (nlevels(group) < 2L || any(table(group) < 2L)) {
```

```
+   print(table(group))
```

```
+   stop("Indicator analysis needs at least 2 groups with ≥2 samples each.")
```

```
+ }
```

```
> nlevels(group)
```



```
[1] 2
> table(group)
group
Low High
6 6
```

```
# run IndVal.g
> set.seed(123)
> ctrl <- how(nperm = 9999)
> indval <- multipatt(FT, cluster = group, func = "IndVal.g", control = ctrl)
> summary(indval, indvalcomp = TRUE, alpha = 1)
```

Multilevel pattern analysis

Association function: IndVal.g

Significance level (alpha): 1

Total number of species: 21

Selected number of species: 5

Number of species associated to 1 group: 5

List of species associated to each combination:

Group Low #sps. 1

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomeromycetes;f__Glomeromycetes;g__Glomeromycetes;s__Paraglomus_brasilianum 0.8571

B

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomeromycetes;f__Glomeromycetes;g__Glomeromycetes;s__Paraglomus_brasilianum 0.5000

stat

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomeromycetes;f__Glomeromycetes;g__Glomeromycetes;s__Paraglomus_brasilianum 0.655

p.value

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomeromycetes;f__Glomeromycetes;g__Glomeromycetes;s__Paraglomus_brasilianum 0.302

Group High #sps. 4

A

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__uncultured;s__uncultured_fungus 1.0000

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Diversisporales;f__Gigasporaceae;g__Dentiscutata;s__Dentiscutata_reticulata 0.6263

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Claroideoglomeraceae;g__Claroideoglomus;s__Claroideoglomus_etunicatum 1.0000

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__Glo mus;s__uncultured_Glomeromycotina 1.0000

B

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__uncultured;s__uncultured_fungus 0.6667

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Diversisporales;f__Gigasporaceae;g__Dentiscutata;s__Dentiscutata_reticulata 0.8333

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Claroideoglomeraceae;g__Claroideoglomus;s__Claroideoglomus_etunicatum 0.3333

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__Glo
mus;s__uncultured_Glomeromycotina 0.3333

stat

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__unc
ultured;s__uncultured_fungus 0.816

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Diversisporales;f__Gigasporaceae;g__
_Dentiscutata;s__Dentiscutata_reticulata 0.722

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Claroideoglomeracea
e;g__Claroideoglomus;s__Claroideoglomus_etunicatum 0.577

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__Glo
mus;s__uncultured_Glomeromycotina 0.577

p.value

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__unc
ultured;s__uncultured_fungus 0.0633

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Diversisporales;f__Gigasporaceae;g__
_Dentiscutata;s__Dentiscutata_reticulata 0.1766

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Claroideoglomeracea
e;g__Claroideoglomus;s__Claroideoglomus_etunicatum 0.4599

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__Glo
mus;s__uncultured_Glomeromycotina 0.4494

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__unc
ultured;s__uncultured_fungus .

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Diversisporales;f__Gigasporaceae;g__
_Dentiscutata;s__Dentiscutata_reticulata

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Claroideoglomeracea
e;g__Claroideoglomus;s__Claroideoglomus_etunicatum

d__Eukaryota;p__Mucoromycota;c__Glomeromycetes;o__Glomerales;f__Glomeraceae;g__Glo
mus;s__uncultured_Glomeromycotina

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1