

Package: plotly.microbiome (via r-universe)

November 18, 2024

Version 0.0.9003

Date 2022-05-03

Title Helper Functions for Plotly Graphs in Microbiome Work

Maintainer Lindsay V. Clark <lvclark@illinois.edu>

Imports plotly, dittoSeq, viridis, rlang

Suggests phyloseq, dplyr, tidyr, limma

Description Helper functions for nice interactive plots from phyloseq and other workflows.

License GPL (>= 2)

URL https://github.com/HPCBio/plotly_microbiome

Config/pak/sysreqs make libicu-dev libssl-dev

Repository <https://lvclark.r-universe.dev>

RemoteUrl https://github.com/HPCBio/plotly_microbiome

RemoteRef HEAD

RemoteSha 4c9fd921576a0c47fc908faedc06eb6764d42cfa

Contents

beta_diversity_3d	2
composition_df	4
findnonmissing	5
make_taxa_labels	6

Index	8
--------------	----------

beta_diversity_3d *Draw 3D Scatterplot of Beta Diversity*

Description

This is a wrapper function for `plot_ly` with the "scatterplot3d" option. It can be used on a data frame, or directly on the output of `phyloseq::ordinate` with the NMDS or PCoA method (or `vegan::metaMDS` or `ape::pcoa`, respectively), or on the output of `limma::plotMDS`. Default categorical colors are from **dittoSeq**.

Usage

```
beta_diversity_3d(x, ...)

## S3 method for class 'data.frame'
beta_diversity_3d(x, axes = colnames(x)[1:3],
                  color.column = colnames(x)[4],
                  label.column = colnames(x)[5],
                  color.key = NULL, ...)

## S3 method for class 'monoMDS'
beta_diversity_3d(x, metadata,
                  color.column,
                  label.column = NULL,
                  color.key = NULL, ...)

## S3 method for class 'pcoa'
beta_diversity_3d(x, metadata,
                  color.column,
                  label.column = NULL,
                  color.key = NULL, ...)

## S3 method for class 'MDS'
beta_diversity_3d(x, metadata,
                  color.column,
                  label.column = NULL,
                  color.key = NULL, ...)
```

Arguments

<code>x</code>	A data frame, "monoMDS", "pcoa", or "MDS" object containing ordination results.
<code>axes</code>	If <code>x</code> is a data frame, the column names for three axes to plot.
<code>color.column</code>	The name of a numeric or categorical column to be used for coloring points. This column should be found in <code>x</code> if <code>x</code> is a data frame, or in <code>metadata</code> otherwise.

label.column	The name of a character column to be used for labeling points. This column should be found in <code>x</code> if <code>x</code> is a data frame, or in metadata otherwise. It defaults to sample names if <code>x</code> is a "monoMDS" or "MDS".
color.key	If <code>color.column</code> refers to a character or factor column, a named vector of colors, with names corresponding to values in the column. The default is to use <code>dittoSeq::dittoColors</code> . If <code>color.column</code> refers to a numeric column, a long vector of colors to be used as the color scale. The default is to use <code>viridis</code> . Passed to the <code>colors</code> argument of <code>plot_ly</code> .
metadata	A data frame of sample metadata, in the same order as <code>x</code> .
...	Optional arguments passed to <code>plot_ly</code> .

Value

A "plotly" object.

Author(s)

Lindsay Clark

Examples

```
## Not run:
# Perform NMDS with Bray distance

ord_mfiber_prop1a <- ordinate(ps_mfiber_prop, "NMDS", "bray", k = 3)

beta_diversity_3d(ord_mfiber_prop1a,
                 sample_data(ps_mfiber_prop), "TRT", "Label")

# NMDS with UniFrac distance (ordinate won't let you adjust k)

dist_mfiber_4 <- phyloseq::distance(ps_mfiber_prop, "unifrac")
ord_mfiber_prop4a <- vegan::metaMDS(dist_mfiber_4, k = 3)

beta_diversity_3d(ord_mfiber_prop4a,
                 sample_data(ps_mfiber_prop), "TRT", "Label")

# MDS of gene expression with limma
mds1 <- plotMDS(logCPM.filt, top = 5000)
beta_diversity_3d(mds1, metadata = d.filt$samples,
                 color.column = "Group")

# Changing title and axis labels, manually setting colors, and saving

mfiber_colors <- c(CO = "magenta", BP = "black", MF = "turquoise",
                 FOS = "orange", RS = "skyblue", TP = "green")

p1 <- beta_diversity_3d(ord_mfiber_prop1a,
                     sample_data(ps_mfiber_prop), "TRT", "Label",
                     color.key = mfiber_colors) %>%
  plotly::layout(title = "NMDS with Bray distance",
```

```

    scene = list(xaxis = list(title = "Axis 1"),
                yaxis = list(title = "Axis 2"),
                zaxis = list(title = "Axis 3"))
htmlwidgets::saveWidget(partial_bundle(p1),
                        file = "results/NMDS_Bray_MFiber_plotly.html")

## End(Not run)

```

composition_df *Prepare a Data Frame for Composition Plots*

Description

This function takes a phyloseq object and prepares a data frame that can be used to generate composition plots with **ggplot2**. Works similarly to the **ggformat** function in **phyloseq.extended**.

Usage

```

composition_df(psobj, rank = "Family",
               keepcols = c("Sample", "Group", "Label", "ID"),
               minprop = 0.05, mean_across_samples = NULL)

```

Arguments

psobj	A phyloseq object containing raw, potentially agglomerated, counts.
rank	The taxonomic rank across which taxa counts should be summed.
keepcols	Names of columns from <code>sample_data(psobj)</code>
minprop	Threshold for showing a taxon vs. lumping it into "Other". At least one sample must have at least this proportion of the OTU counts assigned to a given taxon for that taxon to be displayed.
mean_across_samples	An optional grouping variable, indicated as a character string matching one of the column names in <code>keepcols</code> . If provided, samples are lumped within groups. Proportions are averaged across samples, and counts are summed across samples.

Value

A tibble with the following columns:

- A column labeled ID listing sample names, or if provided, a column named the same as `mean_across_samples`.
- A column named the same as `rank` listing taxa names.
- A column named Proportion indicating the proportion of OTU counts assigned to a given taxon within a sample or group.

- A column named Counts indicating the total OTU counts assigned to a given taxon within a sample or group.
- Any other columns in keepcols. If mean_across_samples is provided, columns are dropped if they have more than one value for a given group.

Author(s)

Lindsay Clark

See Also

[findnonmissing](#) is used to determine which taxa to label as “Unclassified”.

Examples

```
## Not run:
# Columns to keep
kc <- c("Dog", "Trt", "Day", "Breed", "Group", "Label", "ID")

# Composition plot on individuals, grouped by experimental group
p1 <- composition_df(ps_glom, "Family", minprop = 0.1,
                    keepcols = kc)
ggplot(aes(x = ID, y = Proportion, fill = Family)) +
  geom_col() +
  facet_wrap(~ Group, scales = "free_x") +
  scale_fill_manual(values = dittoSeq::dittoColors(1)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

ggplotly(p1)

# Composition plot on treatments
p2 <- composition_df(ps_glom, "Family", minprop = 0.1,
                    keepcols = kc, mean_across_samples = "Trt")
ggplot(aes(x = Trt, y = Proportion, fill = Family)) +
  geom_col() +
  scale_fill_manual(values = dittoSeq::dittoColors(1)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

ggplotly(p2)

## End(Not run)
```

findnonmissing

Identify Non-Missing Taxonomic Labels

Description

In various taxonomic databases and pipelines, unknown or missing taxonomic labels may be indicated in a variety of ways, such as missing data, "unclassified", "uncultured", etc. This function identifies all of these that I have encountered so far.

Usage

```
findnonmissing(x)
```

Arguments

x A character vector of taxonomic labels, for example a single column of the tax_table slot of a phyloseq object.

Details

The following values will result in output of FALSE.

- NA
- An empty string.
- The words “unclassified”, “unidentified”, “uncultured”, “unknown”, or “metagenome” anywhere in the string, in any case.
- Values equal to “human_gut.”

Value

A logical vector, with TRUE if the taxonomic label reflects a taxonomic identity, and FALSE if it should be considered missing.

Author(s)

Lindsay V. Clark

Examples

```
findnonmissing(c("Streptococcus", "Blautia", "Horse metagenome", NA))
```

make_taxa_labels *Create a Label for Each Taxon*

Description

This function takes a table of taxonomic ranks, such as that stored in the tax_table slot of a phyloseq object, and creates a label for each taxon for use in plots and tables.

Usage

```
make_taxa_labels(taxtab)
```

Arguments

taxtab A matrix or data frame, with taxa in rows and taxonomic ranks in columns. The last column should be “Species”, the first column should be kingdom or domain, and columns in between should progress in order of rank.

Details

Species labels that pass `findnonmissing` are used, and otherwise species are labeled ""sp."" . The lowest rank that passes `findnonmissing` is pasted before the species label.

Value

A character vector containing the labels. If `taxtab` has row names, these are used to name the vector.

Author(s)

Lindsay V. Clark

Examples

```
tt <- matrix(c("Bacteria", "Firmicutes", "Clostridia",
              "Peptostreptococcales-Tissierellales", "Anaerovoracaceae",
              "Mogibacterium", "Unclassified",
              "Bacteria", "Firmicutes", "Clostridia", "Oscillospirales",
              "Ruminococcaceae", "Faecalibacterium", "prausnitzii",
              "Bacteria", "Firmicutes", "Clostridia",
              "Clostridia vadinBB60 group", "Unclassified", "Unclassified",
              "Unclassified"),
            nrow = 3, ncol = 7, byrow = TRUE,
            dimnames = list(c("db3b201de3a824d7356ce4d8360e5bc3",
                              "615ce01e68e098adc445d06e072ba255",
                              "6f643e7faeab3e737b38f5304b841d97"),
                           c("Kingdom", "Phylum", "Class", "Order",
                              "Family", "Genus", "Species"))))

make_taxa_labels(tt)
```

Index

beta_diversity_3d, 2

composition_df, 4

findnonmissing, 5, 5, 7

make_taxa_labels, 6