Package: phylobar (via r-universe)

November 25, 2026

Title Interactive construction of stacked barplots using hierarchies

Version 1.0.0

Description The phylobar package supports interactive visualization of microbiome data by allowing a stacked barplot to be constructed using a guiding taxonomic or phylogenetic hierarchy. The package provides a strategy for collapsing and expanding the hierarchy to different levels of resolution and then for interactively ``painting" the stacked barplot by placing the mouse over different subtrees. This makes it possible to interactively test different color palettes at different resolution and identify taxonomic groups with interesting variation before settling on a final stacked barplot. One advantage of the approach is that multiple levels of taxonomic resolution can be compared at once within the same view.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/mkdiro-O/phylobar, http://mkdiro-o.github.io/phylobar

BugReports https://github.com/mkdiro-O/phylobar/issues

Roxygen list(markdown = TRUE)

Suggests BiocStyle, DESeq2, dplyr, ggplot2, mia, miaViz, microbiome, phyloseq, readr, rmarkdown, scater, seriation, stringr, testthat (>= 3.0.0), tibble, tidyr, zen4R

VignetteBuilder knitr

Imports ape, cluster, htmlwidgets, knitr, phangorn, purrr

Depends R (>= 4.5.0)

biocViews Software, Microbiome, Phylogenetics, Visualization

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake libglpk-dev make libuv1-dev libxml2-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:02 UTC

RemoteUrl https://github.com/bioc/phylobar

RemoteRef RELEASE_3_24

RemoteSha 0f73597557f604a1c0f9da33624d6a9103c55720

Contents

add_prefix 2
bray_curtis_dist 3
deseq_normalize 3
edgelist_to_phylo 4
node_hierarchy 5
node_totals 5
phylobar 6
phylobar_data 8
subset_cluster 9
taxonomy_to_tree 9

Index **11**

`add_prefix` *Add rank prefixes to taxonomic matrix values.*

Description

For each value in the taxonomic matrix (except the last column), adds a prefix based on the first character of the column (rank) name. For example, if the column is "Genus", the prefix will be "G_".

Usage

```
add_prefix(taxa)
```

Arguments

`taxa`	A character matrix with columns representing taxonomic ranks and rows representing taxa.

Value

A character matrix with prefixes added to each value (except the last column).

Examples

```
taxa <- matrix(
  c("Firmicutes", "Bacilli", "Lactobacillales",
    "Proteobacteria", "Gammaproteobacteria", "Enterobacterales"),
  ncol = 3, byrow = TRUE,
  dimnames = list(NULL, c("Phylum", "Class", "Order"))
)
add_prefix(taxa)
```

---

bray_curtis_dist *Bray-Curtis dissimilarity matrix*

Description

Pairwise Bray-Curtis dissimilarity between matrix rows. $\sum |x_i - y_i| / (\sum x_i + \sum y_i)$.

Usage

```
bray_curtis_dist(x)
```

Arguments

x	A matrix with samples as rows.

Value

A `dist` object of pairwise Bray-Curtis dissimilarities.

---

deseq_normalize *DESeq2 Size Factor Normalization*

Description

This normalizes a count matrix using the same approach as mStat_normalize_data from the MicrobiomeStat package. This can provide a useful alternative to purely compositional transformations when constructing the stacked bar plots. The only reason we don't use MicrobiomeStat directly is that the current CRAN version does not have this function (only the development GitHub version does).

Usage

```
deseq_normalize(otu)
```

Arguments

otu	A numeric matrix with taxa as rows and samples as columns. Zero-sum rows are dropped before normalization.

Value

A numeric matrix with taxa as rows and samples as columns containing the normalized counts. Any NaN or Inf entries (which can arise when a sample has all-zero counts) are replaced with 0.

Examples

```
otu <- matrix(c(10L, 0L, 5L, 20L, 3L, 0L), nrow = 3,
              dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
deseq_normalize(otu)
```

---

edgelist_to_phylo *Convert an edge list to an ape phylo object.*

Description

Takes a two column matrix (parents -> descendants) and constructs an ape phylo object from it.

Usage

```
edgelist_to_phylo(edgelist)
```

Arguments

edgelist	A two-column matrix where each row represents a parent-descendant relationship.

Value

An object of class 'phylo' representing the tree structure.

Examples

```
# Example edge list: parent -> child
edgelist <- matrix(
  c("A", "B",
    "A", "C",
    "B", "D",
    "B", "E"),
  ncol = 2, byrow = TRUE,
  dimnames = list(NULL, c("parent", "child"))
)
tree <- edgelist_to_phylo(edgelist)
str(tree)
```

node_hierarchy *Construct Initial Hierarchy*

Description

This reshapes the list output from node_totals into the hierarchical format needed for the d3 tree visualization.

Usage

```
node_hierarchy(tree, totals, node = NULL)
```

Arguments

tree	An object of class phylo, representing the tree structure.
totals	A named list of node totals, as returned by node_totals.
node	Name of the node from which to start a recursion. Defaults to the root node.

Value

A nested list representing the hierarchy, with each node containing ' its name, value, summary, and children (if any).

Examples

```
library(ape)
tree <- rtree(5)
x_mat <- matrix(runif(15), ncol = 5)
colnames(x_mat) <- tree$tip.label

tree$node.label <- as.character(seq_len(tree$Nnode))
totals <- c(
  node_totals(tree, x_mat),
  as.list(data.frame(x_mat))
)
node_hierarchy(tree, totals)
```

node_totals *Compute Node Totals*

Description

This loops over all internal nodes in the tree and takes the sum over all descendant taxa, for each sample.

Usage

```
node_totals(tree, x_mat)
```

Arguments

tree	An object of class phylo, representing the tree structure. Must have tip labels matching the columns of x_mat.
x_mat	A numeric matrix of abundances, with samples in rows and features (tips) in columns. Column names should correspond to tree tip

Value

A named list where each element corresponds to an internal node (by node label) and contains a vector of totals for each sample, computed by summing abundances over all descendant tips.

Examples

```
library(ape)
tree <- rtree(5)
x_mat <- matrix(runif(15), ncol = 5)
colnames(x_mat) <- tree$tip.label
tree$node.label <- as.character(seq_len(tree$Nnode))
node_totals(tree, x_mat)
```

---

phylobar *Tree + Stacked Bars*

Description

phylobar is a visualization package that makes it possible to construct a stacked barplot by interactively "painting" an associated tree. This is an alternative to defining a color palette using a fixed taxonomic resolution. It also helps avoid the issue of grouping all rare taxa into a color for "other" since species can be chosen selectively, we can paint a few rare taxa but not the rest.

Usage

```
phylobar(
  x,
  tree,
  hclust_order = TRUE,
  palette = NULL,
  width = NULL,
  height = NULL,
  sample_font_size = 8,
  sample_label_margin = 10,
  sample_label_space = 50,
  sample_magnify = 1.5,
```

```
  sample_show_all = TRUE,
  element_id = NULL,
  rel_width = 0.4,
  rel_height = 0.85,
  rel_space = 10,
  legend_mode = TRUE,
  legend_x_start = 5,
  legend_spacing = 16
)
```

Arguments

x	A matrix of abundances. Samples along rows, features along columns.
tree	An object of class phylo, representing the tree structure.
hclust_order	Logical; if TRUE, reorder rows/columns by hierarchical clustering.
palette	Character vector of colors for stacked bars. If NULL, uses default palette: c("#9c7bbaff", "#6eb8acff", "#ce7b7bff", "#7b9cc4ff", "#c47ba0ff", "#e1d07eff").
width	Width of the widget in pixels. If NULL, uses window default.
height	Height of the widget in pixels. If NULL, uses window default.
sample_font_size	Font size for sample labels (integer).
sample_label_margin	Margin between sample labels and bars in pixels.
sample_label_space	Space allocated for sample labels in pixels.
sample_magnify	Magnification factor for hovered sample labels.
sample_show_all	Logical; if TRUE, show all sample labels.
element_id	Optional HTML element ID to attach the widget to.
rel_width	Width of the tree panel relative to the overall visualization. Defaults to 0.4.
rel_height	Relative height of the tree in the overall visualization. Defaults to 0.85. Adjust this if you need more/less space for the legend.
rel_space	Space between tree and barplot panels in pixels.
legend_mode	Logical; if TRUE (default), display labels for the painted subtrees in a legend near the bottom of the tree. If FALSE, include the labels within the tree itself.
legend_x_start	Horizontal starting position (in pixels) for the legend. Defaults to 4.
legend_spacing	Vertical spacing (in pixels) between legend entries.

Value

An htmlwidget visualization attached to the element element_id on the output HTML page.

Examples

```
library(ape)
tree <- rtree(5)
x <- matrix(rpois(15, 1), ncol = 5)
phylobar(x, tree)
```

`phylobar_data` *Prepare tree data for phylobar visualization*

Description

Prepare tree data for phylobar visualization

Usage

```
phylobar_data(x, tree, hclust_order = TRUE)
```

Arguments

`x`	A matrix of abundances. Samples along rows, features along columns.
`tree`	A n object of class phylo, representing the tree structure.
`hclust_order`	Logical; if TRUE, reorder rows/columns by hierarchical clustering.

Value

A list with tree_data and labels.

Examples

```
library(ape)
tree <- rtree(5)
tree$node.label <- paste0("node", seq_len(4))
x <- matrix(runif(15), nrow = 3)
colnames(x) <- tree$tip.label
rownames(x) <- paste0("sample", seq_len(3))
phylobar_data(x, tree)
```

subset_cluster *Subsample rows of a matrix using clustering*

Description

This function takes a matrix `x` and subsamples its rows by clustering them. One representative row is selected from each cluster, which helps in visualizing a large dataset in a semi-representative way.

Usage

```
subset_cluster(x, k = 100, method = c("hclust", "medoid"))
```

Arguments

`x`	A matrix whose rows will be clustered and subsampled.
`k`	The number of clusters to form (default is 100).
`method`	Clustering method to use. `"hclust"` (the default) uses Euclidean distances and hierarchical clustering, selecting an arbitrary member from each cluster. `"medoid"` uses Bray-Curtis dissimilarity and K-medoids (PAM) clustering, selecting the medoid of each cluster.

Value

A matrix containing one representative row from each cluster.

Examples

```
mat <- matrix(rnorm(1000), nrow = 100)
rownames(mat) <- seq_len(100)
result <- subset_cluster(mat, k = 10)
dim(result) # only 10 representatives

# Using Bray-Curtis + K-medoids (better for compositional data)
counts <- matrix(rpois(1000, 5), nrow = 100)
rownames(counts) <- seq_len(100)
result2 <- subset_cluster(counts, k = 10, method = "medoid")
```

taxonomy_to_tree *Convert a taxonomic table to an ape phylo object.*

Description

Creates a phylo from a taxonomic tree, skipping over any NA assignments. Assumes that the columns are sorted from coarsest to finest taxonomic resolution.

Usage

```
taxonomy_to_tree(taxa)
```

Arguments

taxa	A data.frame or matrix with columns representing taxonomic ranks (sorted coarsest to finest) and rows representing taxa. NA or empty values are skipped.

Value

An object of class 'phylo' representing the taxonomic tree.

Examples

```
taxa <- matrix(
  c("Firmicutes", "Bacilli", "Lactobacillales",
    "Proteobacteria", "Gammaproteobacteria", "Enterobacterales"),
  ncol = 3, byrow = TRUE,
  dimnames = list(NULL, c("Phylum", "Class", "Order"))
)
tree <- taxonomy_to_tree(taxa)
str(tree)

# A more involved example with missing values
taxa <- matrix(
  c("Firmicutes", "Bacilli", "Lactobacillales", "ASV1",
    "Proteobacteria", "Gammaproteobacteria", "Enterobacterales", "ASV2",
    "Firmicutes", "Bacilli", NA, "ASV3"),
  ncol = 4, byrow = TRUE,
  dimnames = list(NULL, c("Phylum", "Class", "Order", "ASV"))
)
taxmat <- add_prefix(taxa)
tree <- taxonomy_to_tree(taxmat)
str(tree)
```

Index

add_prefix, 2

bray_curtis_dist, 3

deseq_normalize, 3

edgelist_to_phylo, 4

node_hierarchy, 5
node_totals, 5

phylobar, 6
phylobar_data, 8

subset_cluster, 9

taxonomy_to_tree, 9