Package ‘MetaPhyloTools’

October 9, 2025

Type Package

Title Tools for Analysing Phylogeographic Patterns from Metabarcoding Data

Version 0.1.0

Author Adrià Antich

Maintainer Adrià Antich <a.antich@ceab.csic.es>

Description A collection of tools to process and analyse phylogeographic patterns from metabarcoding data, including functions for data cleaning, phylogenetic analysis, and visualization of results.

License GPL-3

Imports dplyr,
ggplot2,
tidyr,
phytools,
ape,
vegan,
stringr,
readr,
purrr,
igraph,
scatterpie

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

R topics documented:

edenetwork_percolation 2
haplodata4ggplot 3
haploNet_data 4

haploNet_plot 6
haplo_ggplot 7
pairwise_djost 8
print_network 10
print_network_ggplot 12
rarefy_within_motu 15

Index **16**

`edenetwork_percolation` *Calculate Percolation Threshold for Ecological Networks*

Description

Calculates the percolation threshold for ecological networks by identifying the critical distance at which the network fragments into small components. This function implements the percolation analysis described in EDENetworks.

Usage

```
edenetwork_percolation(diss)
```

Arguments

`diss`	A distance matrix representing dissimilarities between nodes

Details

Percolation threshold is defined as follows in the original reference: from

Arnaud-Haond et al. EDENetworks: Ecological and Evolutionary Networks.

Percolation threshold: When links are removed from a connected network, it eventually fragments into small components. The point where this happens is called the percolation threshold. More accurately, this is the point where the so-called giant component (whose size is of the order of the network size) disappears and there is no long-range connectivity; even before the percolation threshold small disconnected fragments will appear, yet a substantial fraction of nodes belongs to the giant component. The precise location of this percolation point is made using the definition classically proposed for finite systems (Stauffer and Aharony, 1994) by calculating the average proposed for finite systems (Stauffer and Aharony, 1994) by calculating the average size of the clusters excluding the largest one:

S = sums<Smax(s^2ns)

as a function of the last distance value removed, thr, and identifying the critical distance with the one at which <S>* has a maximum. N is the total number of nodes not included in the largest cluster and ns is the number of clusters containing s nodes.

Value

A list containing:

graph The original igraph object

S_list Vector of S values for each threshold

critical_distance The critical distance at percolation threshold

References

Arnaud-Haond et al. EDENetworks: Ecological and Evolutionary Networks. https://www.researchgate.net/profile/Sophie-Arnaud-Haond/publication/268433406_EDENetworks_Ecological_and_Evolutionary_Networks/links/54f705a30cf28d6dec9c7ad8/EDENetworks-Ecological-and-Evolutionary-Networ pdf

Examples

```
# Create example distance matrix
dist_mat <- as.matrix(dist(iris[1:10, 1:4]))
result <- edenetwork_percolation(dist_mat)
print(result$critical_distance)
```

haplodata4ggplot *haplodata4ggplot*

Description

This function creates a list containing nodes and connections for a haplotype network plot, suitable for use with ggplot2.

Usage

```
haplodata4ggplot(haploNet_data_object, method = "pegas")
```

Arguments

`haploNet_data_object`

A list containing the haplotype network data named network, and a dataframe named pieseas. The latter contains the proportions of each sample or group of samples for each haplotype. This object can be generated using the function haploNet_data.

@param method A string indicating the method to use for plotting the network. Options are "pegas" for the pegas package or "fruchtermanreingold" for the Fruchterman-Reingold algorithm.

Details

This function prepares the data for plotting haplotype networks, using either the pegas package or the Fruchterman-Reingold algorithm to determine node positions. The nodes are represented as pie charts, with sizes proportional to haplotype frequencies. The aim of this function is to facilitate the creation of publication-ready haplotype network plots using ggplot2, addressing common issues such as legend overlap with the network.

Value

Returns a list with two data frames: nodes and connections. The nodes data frame contains the x and y coordinates, size, labels, and proportions of each haplotype. The connections data frame contains the coordinates of the connections between nodes.

`haploNet_data` *Create haplotype network data for visualization*

Description

This function processes MOTU (Molecular Operational Taxonomic Unit) data and metadata to create a haplotype network and associated pie chart data for phylogenetic analysis. It calculates mean abundances across samples grouped by a specified variable and generates network data suitable for haplotype network visualization.

Usage

```
haploNet_data(
  motu_tab,
  metadata,
  grouping_col,
  sample_col,
  seq_col,
  id_col,
  size_correction = 1
)
```

Arguments

`motu_tab` A data.frame containing MOTU abundance data. Rows represent MOTUs/haplotypes, columns represent samples. Must include columns specified by `id_col` and `seq_col`.

`metadata` A data.frame containing sample metadata with grouping information. Must include columns specified by `grouping_col` and `sample_col`.

`grouping_col` Character string. Name of the column in `metadata` that contains the grouping variable (e.g., "treatment", "location").

`sample_col` Character string. Name of the column in `metadata` that contains sample identifiers matching column names in `motu_tab`.

`seq_col` Character string. Name of the column in `motu_tab` that contains DNA sequences for haplotype analysis.

`id_col` Character string. Name of the column in `motu_tab` that contains unique identifiers for each MOTU/haplotype.

`size_correction`
Numeric. Factor to correct haplotype frequencies for visualization. Default is 1 (no correction).

Details

The function performs the following steps:

1. Groups samples by the specified grouping variable
2. Calculates mean abundances for each MOTU across samples within each group
3. Filters out MOTUs with zero total abundance
4. Converts DNA sequences to DNAbin format for phylogenetic analysis
5. Creates haplotypes and constructs a haplotype network
6. Attaches frequency information to the network for visualization

Value

A list containing two elements:

`network` A haplotype network object created by `pegas::haploNet()`, or NULL if only one haplotype is found. Contains network structure with frequency attributes.

`pieseas` A matrix with haplotypes as rows and groups as columns, containing mean abundances for pie chart visualization.

Note

- Requires packages ape and pegas
- DNA sequences should be aligned and of equal length
- Sample identifiers in metadata must match column names in motu_tab
- If only one haplotype is found, network will be NULL with a warning

Examples

```
## Not run:
# Example usage
result <- haploNet_data(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "treatment",
  sample_col = "sample_id",
  seq_col = "sequence",
  id_col = "motu_id",
  size_correction = 100
)
```

```
# Access results
network <- result$network
pie_data <- result$pieseas

## End(Not run)
```

haploNet_plot *Plot haplotype network with pie charts*

Description

This function creates a publication-ready haplotype network plot with pie charts showing the relative abundance or frequency of haplotypes across different groups. The plot includes a legend and title, and is saved as a PDF file.

Usage

```
haploNet_plot(haploNet_data_object, output_file, bg, plot_title)
```

Arguments

`haploNet_data_object`
: A list object returned by `haploNet_data()` containing network and pie chart data. Must include elements:
  - `network`A haplotype network object from `pegas::haploNet()`
  - `pieseas`A matrix with pie chart data for each haplotype

`output_file`
: Character string. File path for the output PDF plot (e.g., "network_plot.pdf").

`bg`
: A vector of colors for the pie chart segments. Length should match the number of groups in the pie chart data. Used for both pie segments and legend.

`plot_title`
: Character string. Main title to be displayed at the top of the plot.

Details

The function creates a multi-panel plot with:

- Main haplotype network with pie charts at each node
- Node sizes proportional to haplotype frequencies
- Title at the top of the plot
- Legend at the bottom showing group categories (currently hardcoded as "Inside"/"Outside")

Plot specifications:

- PDF dimensions: 15 x 15 inches
- High-resolution output suitable for publication
- Customizable colors and title
- Error handling for plot generation issues

Value

No return value. The function creates a PDF file at the specified location.

Note

- Requires the pegas package for network plotting
- Legend labels are currently hardcoded as "Inside" and "Outside"
- The function assumes a two-group comparison (modify legend for more groups)
- If plotting fails, an error message is displayed with details

Examples

```
## Not run:
# Create network data first
network_data <- haploNet_data(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "location",
  sample_col = "sample_id",
  seq_col = "sequence",
  id_col = "motu_id"
)

# Create the plot
haploNet_plot(
  haploNet_data_object = network_data,
  output_file = "haplotype_network.pdf",
  bg = c("red", "blue"),
  plot_title = "Haplotype Network Analysis"
)

## End(Not run)
```

haplo_ggplot *Create Haplotype Network Plots with ggplot2*

Description

Creates haplotype network visualizations using ggplot2 with customizable legend positioning and scatter pie charts for nodes.

Usage

```
haplo_ggplot(data, legend_pos = "left_bottom")
```

Arguments

data A list containing 'nodes' and 'connections' data frames for the network. The 'nodes' data frame should contain x, y coordinates and size information. The 'connections' data frame should specify network edges.

legend_pos Position for the legend. Options: "left_bottom", "right_bottom", "top_left", "top_right", "left", "right", "top", "bottom", "none"

Details

This function creates publication-ready haplotype network plots with nodes represented as pie charts (using scatterpie) and customizable legend positioning to avoid overlap with the network. The aim of this function is to solve a problem often encountered when plotting haplotype networks with ggplot2, where the legend can overlap with the network itself.

Value

A ggplot object representing the haplotype network

Examples

```
# Example with mock network data
nodes <- data.frame(
  x = runif(5),
  y = runif(5),
  size = runif(5, 1, 3),
  pop1 = runif(5),
  pop2 = runif(5)
)
connections <- data.frame(from = c(1,2,3), to = c(2,3,4))
data <- list(nodes = nodes, connections = connections)
plot <- haplo_ggplot(data)
```

---

pairwise_djost *Calculate Pairwise Jost's D Differentiation*

Description

Calculates pairwise Jost's D differentiation indices between samples, optionally grouped by sample groups and per MOTU (Molecular Operational Taxonomic Unit).

Usage

```
pairwise_djost(
  df,
  sample_names,
  sample_groups = NULL,
  motu_col = "MOTU",
  rarefy = TRUE,
  rarefy_to = min
)
```

Arguments

`df` Data frame containing MOTU abundance data

`sample_names` Vector of column names representing samples

`sample_groups` Optional vector of group assignments for samples. Must be the same length as sample_names and in the same order. If NULL, each sample is treated as its own group

`motu_col` Name of the column containing MOTU identifiers (default: "MOTU")

`rarefy` Logical, whether to rarefy data within MOTU (default: TRUE)

`rarefy_to` Function to determine rarefaction depth (default: min)

Details

Jost's D is a measure of differentiation that is independent of within-population diversity. This function calculates it for each MOTU separately and can group samples for analysis. The function rarefies data within each MOTU to standardize sampling effort before calculating Jost's D this rarefaction uses the funcion vegan::rrarefy and the specified rarefaction depth defined by the user with the parameter rarefy_to (default is the minimum sample sum across samples with non-zero counts within each MOTU).

Value

A list containing Jost's D matrices for each MOTU, and optionally grouped results if sample_groups is provided

Examples

```
# Example with mock data
df <- data.frame(
  MOTU = rep(c("MOTU1", "MOTU2"), each = 5),
  sample1 = rpois(10, 5),
  sample2 = rpois(10, 3),
  sample3 = rpois(10, 4)
)
result <- pairwise_djost(df, c("sample1", "sample2", "sample3"))
```

print_network *Create and save a haplotype network plot to PDF*

Description

This function is a convenient wrapper that combines haplotype network data generation and plotting into a single step. It processes MOTU data and metadata to create a haplotype network visualization and saves it directly to a PDF file.

Usage

```
print_network(
  motu_tab,
  output_file = "haplotype_network.pdf",
  metadata,
  grouping_col,
  sample_col,
  seq_col = "sequence",
  id_col = "id",
  size_correction = 1,
  plot_title = "Haplotype Network",
  bg = NULL,
  ...
)
```

Arguments

`motu_tab`	A data.frame containing MOTU abundance data. Rows represent MOTUs/haplotypes, columns represent samples. Must include columns specified by `id_col` and `seq_col`.
`output_file`	Character string. File path for the output PDF plot. Default is "haplotype_network.pdf".
`metadata`	A data.frame containing sample metadata with grouping information. Must include columns specified by `grouping_col` and `sample_col`.
`grouping_col`	Character string. Name of the column in `metadata` that contains the grouping variable (e.g., "treatment", "location").
`sample_col`	Character string. Name of the column in `metadata` that contains sample identifiers matching column names in `motu_tab`.
`seq_col`	Character string. Name of the column in `motu_tab` that contains DNA sequences for haplotype analysis. Default is "sequence".
`id_col`	Character string. Name of the column in `motu_tab` that contains unique identifiers for each MOTU/haplotype. Default is "id".
`size_correction`	Numeric. Factor to correct haplotype frequencies for visualization. Default is 1 (no correction).

`plot_title` Character string. Main title to be displayed on the plot. Default is "Haplotype Network".

`bg` A vector of colors for the pie chart segments. If NULL, default colors will be used.

`...` Additional arguments passed to underlying functions.

Details

This function is a streamlined workflow that:

1. Calls `haploNet_data()` to process the input data and create network structure
2. Calls `haploNet_plot()` to generate and save the plot to PDF
3. Displays a confirmation message with the output file path

The resulting PDF contains:

- Haplotype network with nodes representing haplotypes
- Pie charts at each node showing group composition
- Node sizes proportional to haplotype frequencies
- Legend and title
- High-resolution output (15x15 inches) suitable for publication

Value

Invisible NULL. The function is called for its side effect of creating a PDF file.

Note

- This is a convenience function that combines `haploNet_data()` and `haploNet_plot()`
- Requires packages `ape` and `pegas`
- For more control over the plotting process, use `haploNet_data()` and `haploNet_plot()` separately
- DNA sequences should be aligned and of equal length

See Also

`haploNet_data`, `haploNet_plot`, `print_network_ggplot`

Examples

```
## Not run:
# Basic usage
print_network(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "treatment",
  sample_col = "sample_id",
  output_file = "treatment_network.pdf"
)

# With custom parameters
print_network(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "location",
  sample_col = "sample_id",
  seq_col = "dna_sequence",
  id_col = "motu_id",
  size_correction = 1000,
  plot_title = "Geographic Distribution",
  bg = c("red", "blue", "green"),
  output_file = "geographic_network.pdf"
)

## End(Not run)
```

print_network_ggplot *Create haplotype network visualization using ggplot2*

Description

This function creates a modern, customizable haplotype network plot using ggplot2. It processes MOTU data and metadata to generate a network visualization with nodes representing haplotypes and edges showing relationships between them.

Usage

```
print_network_ggplot(
  motu_tab,
  output_file = "haplotype_network.pdf",
  metadata,
  grouping_col,
  sample_col,
  seq_col = "sequence",
  id_col = "id",
  size_correction = 1,
  plot_title = "Haplotype Network",
  bg = NULL,
  method = "pegas",
  legend_pos = "left_bottom",
  ...
)
```

Arguments

`motu_tab` A data.frame containing MOTU abundance data. Rows represent MOTUs/haplotypes, columns represent samples. Must include columns specified by `id_col` and `seq_col`.

`output_file` Character string. File path for the output PDF plot. Default is "haplotype_network.pdf". Currently not used in this function.

`metadata` A data.frame containing sample metadata with grouping information. Must include columns specified by `grouping_col` and `sample_col`.

`grouping_col` Character string. Name of the column in `metadata` that contains the grouping variable (e.g., "treatment", "location").

`sample_col` Character string. Name of the column in `metadata` that contains sample identifiers matching column names in `motu_tab`.

`seq_col` Character string. Name of the column in `motu_tab` that contains DNA sequences for haplotype analysis. Default is "sequence".

`id_col` Character string. Name of the column in `motu_tab` that contains unique identifiers for each MOTU/haplotype. Default is "id".

`size_correction`
Numeric. Factor to correct haplotype frequencies for visualization. Default is 1 (no correction).

`plot_title` Character string. Main title to be displayed on the plot. Default is "Haplotype Network".

`bg` A vector of colors for the groups. If NULL, default colors will be used.

`method` Character string. Layout method for the network. Must be either "pegas" (uses pegas package layout) or "fruchtermanreingold" (uses Fruchterman-Reingold layout). Default is "pegas".

`legend_pos` Character string. Position of the legend. Default is "left_bottom".

`...` Additional arguments passed to underlying functions.

Details

The function performs the following workflow:

1. Calls `haploNet_data()` to process input data and create network structure
2. Uses `haplodata4ggplot()` to convert network data to ggplot-compatible format
3. Creates the final plot using `haplo_ggplot()`

The function supports two layout methods:

- `"pegas"`: Uses the default layout from the pegas package
- `"fruchtermanreingold"`: Uses the Fruchterman-Reingold force-directed layout

Value

A ggplot object containing the haplotype network visualization.

Note

- Requires packages ape, pegas, and ggplot2
- This function depends on helper functions: haploNet_data(), haplodata4ggplot(), and haplo_ggplot()
- The output_file parameter is included for consistency but not currently used
- DNA sequences should be aligned and of equal length

See Also

haploNet_data, haploNet_plot

Examples

```
## Not run:
# Basic usage
plot <- print_network_ggplot(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "treatment",
  sample_col = "sample_id",
  plot_title = "Treatment Comparison"
)

# With custom parameters
plot <- print_network_ggplot(
  motu_tab = my_motu_data,
  metadata = my_metadata,
  grouping_col = "location",
  sample_col = "sample_id",
  seq_col = "dna_sequence",
  id_col = "motu_id",
  size_correction = 100,
  method = "fruchtermanreingold",
  bg = c("#FF6B6B", "#4ECDC4", "#45B7D1"),
  legend_pos = "right"
)

# Display the plot
print(plot)

# Save the plot
ggsave("my_network.pdf", plot, width = 12, height = 10)

## End(Not run)
```

rarefy_within_motu *Rarefy Data Within MOTUs*

Description

Performs rarefaction of sample data within each MOTU (Molecular Operational Taxonomic Unit) to standardize sampling effort.

Usage

```
rarefy_within_motu(
  df,
  sample_names,
  motu_col = "MOTU",
  rarefy = TRUE,
  rarefy_to = min,
  rel_abund = TRUE
)
```

Arguments

df	Data frame containing MOTU abundance data
sample_names	Vector of column names representing samples
motu_col	Name of the column containing MOTU identifiers (default: "MOTU")
rarefy	Logical, whether to perform rarefaction (default: TRUE)
rarefy_to	Function or numeric value to determine rarefaction depth (default: min function)
rel_abund	Logical, whether to convert to relative abundance after rarefaction (default: TRUE)

Details

This function rarefies data within each MOTU separately, which is important for metabarcoding data where different MOTUs may have very different abundance ranges.

Value

A data frame with rarefied abundance data

Examples

```
# Example with mock data
df <- data.frame(
  MOTU = rep(c("MOTU1", "MOTU2"), each = 3),
  sample1 = rpois(6, 10),
  sample2 = rpois(6, 15)
)
result <- rarefy_within_motu(df, c("sample1", "sample2"))
```

Index

edenetwork_percolation, 2

haplo_ggplot, 7
haplodata4ggplot, 3
haploNet_data, 4, *11*, *14*
haploNet_plot, 6, *11*, *14*

pairwise_djost, 8
print_network, 10
print_network_ggplot, *11*, 12

rarefy_within_motu, 15