Package ‘mixtree’

May 8, 2026

Title A Statistical Framework for Comparing Sets of Trees

Version 0.0.1

Description Statistical framework for comparing sets of trees using hypothesis testing methods. Designed for transmission trees, phylogenetic trees, and directed acyclic graphs (DAGs), the package implements chi-squared tests to compare edge frequencies between sets and PERMANOVA to analyse topological dissimilarities with customisable distance metrics, following Anderson (2001) <doi:10.1111/j.1442-9993.2001.01070.pp.x>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports igraph, treespace, vegan

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL `https://cygei.github.io/mixtree/`

NeedsCompilation no

Author Cyril Geismar [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0002-8486-5890`>)

Maintainer Cyril Geismar <`c.geismar21@imperial.ac.uk`>

Repository CRAN

Date/Publication 2025-03-05 13:00:06 UTC

Contents

Index **11**

`abouheif` *Compute the Abouheif distance matrix*

Description

The Abouheif distance is the product of the number of direct descendants of each node in the path between two nodes. It is a measure of the number of transmission events between two nodes.

Usage

```
abouheif(tree)
```

Arguments

`tree`	A data frame representing a transmission tree, with the first column containing the infector IDs and the second the infectee IDs.

Value

A square, symmetric matrix of Abouheif distances between nodes.

Examples

```
tree <- data.frame(from = c(1, 1, 2, 2, 3, 3), to = c(2, 3, 4, 5, 6, 7))
abouheif(tree)
```

`chisq_test` *Perform Chi-Square Test on Sets of Transmission Trees*

Description

Tests whether the distribution of infector-infectee pairs differs between sets of transmission trees.

Usage

```
chisq_test(..., method = c("chisq", "fisher"), test_args = list())
```

Arguments

`...`	Two or more sets of transmission trees. Each set is a list of data frames with columns `from` and `to`.
`method`	Test to use: `"chisq"` for Chi-Square or `"fisher"` for Fisher's Exact Test. Default is `"chisq"`.
`test_args`	A list of additional arguments for `stats::chisq.test` or `stats::fisher.test`. Default is an empty list.

Value

An `htest` object with the test results.

Examples

```
set.seed(1)
# No difference in the sets
setA <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
),
simplify = FALSE
)
setB <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
),
simplify = FALSE
)
chisq_test(setA, setB)

# Difference in the sets
setC <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 4, stochastic = TRUE)
),
simplify = FALSE
)
chisq_test(setA, setB, setC)
```

`euclidean` *Calculate the Euclidean distance between two distance matrices.*

Description

This function computes the Euclidean distance between the lower triangular parts of two given matrices.

Usage

```
euclidean(mat1, mat2)
```

Arguments

`mat1`	A numeric matrix.
`mat2`	A numeric matrix.

Value

A numeric value representing the Euclidean distance between the lower triangular parts of `mat1` and `mat2`.

Examples

```
mat1 <- matrix(c(1, 2, 3, 4), 2, 2)
mat2 <- matrix(c(4, 3, 2, 1), 2, 2)
euclidean(mat1, mat2)
```

`kendall` *Compute the Kendall distance matrix*

Description

Kendall's distance measures the depth of the most recent common infector (MRCI) for each pair of nodes with respect to the source (patient 0).

Usage

```
kendall(tree)
```

Arguments

`tree`	A data frame representing a transmission tree, with the first column containing the infector IDs and the second the infectee IDs.

Value

A square, symmetric matrix of Kendall's distances between nodes.

References

A Metric to Compare Transmission Trees - M Kendall · 2018

See Also

`findMRCIs`

Examples

```
tree <- data.frame(from = c(1, 1, 2, 2, 3, 3), to = c(2, 3, 4, 5, 6, 7))
kendall(tree)
```

`make_tree` *Generate a Transmission Tree*

Description

Creates a transmission tree with a specified number of cases and branches per case. The tree can be generated with fixed or Poisson-distributed branching factors.

Usage

```
make_tree(n_cases, R = 2, stochastic = FALSE, plot = FALSE)
```

Arguments

`n_cases`	Integer. The total number of cases (nodes) in the tree.
`R`	Integer. The fixed number of branches per case when `stochastic` is `FALSE`, or the mean of the Poisson distribution when `stochastic` is `TRUE`.
`stochastic`	Logical. If `TRUE`, the number of branches per case is sampled from a Poisson distribution with mean `R`. Default is `FALSE`.
`plot`	Logical. If `TRUE`, the function will plot the generated tree. Default is `FALSE`.

Value

An igraph object representing the transmission tree.

Examples

```
# Generate a deterministic transmission tree
deterministic_tree <- make_tree(n_cases = 15, R = 2, stochastic = FALSE, plot = TRUE)

# Generate a stochastic transmission tree
random_tree <- make_tree(n_cases = 15, R = 2, stochastic = TRUE, plot = TRUE)
```

`patristic` *Compute the Patristic distance matrix*

Description

The patristic distance is the number of generations separating any two nodes in a transmission tree.

Usage

```
patristic(tree)
```

Arguments

`tree`	A data frame representing a transmission tree, with the first column containing the infector IDs and the second the infectee IDs.

Value

A square, symmetric matrix of patristic distances between nodes.

Examples

```
tree <- data.frame(from = c(1, 1, 2, 2, 3, 3), to = c(2, 3, 4, 5, 6, 7))
patristic(tree)
```

`permanova_test` *Perform PERMANOVA on Sets of Transmission Trees*

Description

Tests for topological differences between sets of transmission trees using PERMANOVA (via `vegan::adonis2`).

Usage

```
permanova_test(
  ...,
  within_dist = patristic,
  between_dist = euclidean,
  test_args = list()
)
```

Arguments

`...`	Two or more sets of transmission trees. Each set is a list of dataframes with columns `from` (infector) and `to` (infectee).
`within_dist`	A function to compute pairwise distances within a tree. Takes a dataframe, returns a square matrix. Default is `patristic`.
`between_dist`	A function to compute distance between two trees. Takes two matrices, returns a numeric value. Default is `euclidean`.
`test_args`	A list of additional arguments to pass to `vegan::adonis2`. Default is an empty list.

Value

A `vegan::adonis2` object containing the test results.

Examples

```
set.seed(1)
# No difference in the sets
setA <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
),
simplify = FALSE
)
setB <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
),
simplify = FALSE
)
permanova_test(setA, setB)

# Difference in the sets
setC <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 4, stochastic = TRUE)
),
simplify = FALSE
)
permanova_test(setA, setB, setC)
```

`shuffle_graph_ids` *Shuffle Node IDs in a Graph*

Description

Randomly shuffles the IDs of the nodes in a given graph and optionally plots the shuffled graph.

Usage

```
shuffle_graph_ids(g, plot = FALSE)
```

Arguments

`g`	An igraph object representing the graph.
`plot`	Logical. If `TRUE`, the function will plot the shuffled graph. Default is `FALSE`.

Value

An igraph object with shuffled node IDs.

Examples

```
# Create an example graph
g <- make_tree(n_cases = 10, R = 2)

# Shuffle the node IDs
shuffled_graph <- shuffle_graph_ids(g, plot = TRUE)
```

`tree_test` *Test Differences Between Sets of Transmission Trees*

Description

Performs a statistical test to assess whether there are significant differences between sets of transmission trees. Supports PERMANOVA (via `"vegan::adonis2"`), Chi-Square, or Fisher's Exact Test.

Usage

```
tree_test(
  ...,
  method = c("permanova", "chisq", "fisher"),
  within_dist = patristic,
  between_dist = euclidean,
  test_args = list()
)
```

Arguments

`...`	Two or more sets of transmission trees. Each set must be a list of data frames with columns `from` (infector) and `to` (infectee).
`method`	A character string specifying the test method. Options are `"permanova"`, #' `"chisq"`, or `"fisher"`. Default is `"permanova"`.
`within_dist`	A function to compute pairwise distances within a tree for PERMANOVA. Takes a data frame, returns a square matrix. Default is `patristic`.
`between_dist`	A function to compute distance between two trees for PERMANOVA. Takes two matrices, returns a numeric value. Default is `euclidean`.
`test_args`	A list of additional arguments to pass to the underlying test function (`vegan::adonis2`, `stats::chisq.test`, or `stats::fisher.test`). Default is an empty list.

Details

This function compares sets of transmission trees using one of three statistical tests.

PERMANOVA: Evaluates whether the topology of transmission trees differs between sets.

- **Null Hypothesis (H0)**: There is no difference in tree topologies between sets.
- **Alternative Hypothesis (H1)**: At least one set of transmission trees has a different topological structure.

Chi-Square or Fisher's Exact Test: Evaluates whether the distribution of infector-infectee pairs differs between sets.

- **Null Hypothesis (H0)**: The frequency of infector-infectee pairs is consistent across all sets.
- **Alternative Hypothesis (H1)**: The frequency of infector-infectee pairs differs between at least two sets.

Value

- For `"permanova"`: A `"vegan::adonis2"` object containing the test results.
- For `"chisq"` or `"fisher"`: An `"htest"` object with the test results.

See Also

`permanova_test`, `chisq_test`

Examples

```
set.seed(1)
# Generate example sets
setA <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
), simplify = FALSE)
setB <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 2, stochastic = TRUE)
), simplify = FALSE)
setC <- replicate(10, igraph::as_long_data_frame(
  make_tree(n_cases = 10, R = 4, stochastic = TRUE)
), simplify = FALSE)

# PERMANOVA test
tree_test(setA, setB, setC,  method = "permanova")

# Chi-Square test
tree_test(setA, setB, setC, method = "chisq")
```

`validate_tree` *Validate a Transmission Tree*

Description

Checks if a transmission tree meets specific topology criteria for our test. The tree must be a directed acyclic graph (DAG), weakly connected, and have at most one infector per node.

Usage

```
validate_tree(tree)
```

Arguments

`tree`	A data frame with columns `from` and `to` representing the transmission tree.

Value

Invisible `TRUE` if the tree is valid. Throws an error if invalid.

Examples

```
good_tree <- data.frame(from = c(1, 2, 3), to = c(2, 3, 4))
validate_tree(good_tree)
bad_tree <- data.frame(from = c(1, 2, 3), to = c(2, 3, 2))
try(validate_tree(bad_tree))
```

Index