Package ‘treestructure’

May 8, 2026

Type Package

Title Detect Population Structure Within Phylogenetic Trees

Version 0.7.0

Date 2025-09-24

Description Algorithms for detecting population structure from the history of coalescent events recorded in phylogenetic trees. This method classifies each tip and internal node of a tree into disjoint sets characterized by similar coalescent patterns.

License GPL (>= 2)

Suggests ggtree, ggplot2, knitr, rmarkdown, getopt, bookdown, phangorn, treeio

Depends R (>= 4.1.0)

Imports ape (>= 5.0), rlang

LinkingTo Rcpp

VignetteBuilder knitr

RoxygenNote 7.3.3

Encoding UTF-8

URL `https://emvolz-phylodynamics.github.io/treestructure/`, `https://github.com/emvolz-phylodynamics/treestructure`

BugReports `https://github.com/emvolz-phylodynamics/treestructure/issues`

NeedsCompilation yes

Author Erik Volz [aut, cre] (ORCID: <`https://orcid.org/0000-0001-6268-8937`>), Fabricia F. Nascimento [ctb] (ORCID: <`https://orcid.org/0000-0001-9426-6835`>), Vinicius B. Franceschi [ctb] (ORCID: <`https://orcid.org/0000-0002-0006-9337`>)

Maintainer Erik Volz <`erik.volz@gmail.com`>

Repository CRAN

Date/Publication 2025-09-27 13:00:02 UTC

Contents

`addtips` *Compare and add tips into new treestructure object*

Description

Compares a new input tree to an old treestructure fit and merges tips into a new treestructure object. Tips in the new tree that are not in the new treestructure will be merged. Merging is carried out based on a phylogenetic criterion. The new tips are added to the cluster which shares its MRCA (most recent common ancestor).

Usage

```
addtips(trst, tre)
```

Arguments

`trst`	Original treestructure fit that that will be updated.
`tre`	A new tree (ape::phylo) which may contain samples not in trst. This tree must be rooted, but does not need to be time-scaled or binary.

Value

A new treestructure fit.

Author(s)

Erik Volz

Examples

```
set.seed(072023)
# simulate two trees and bind them to simulate structure
tr1 <- ape::rcoal( 50 )
tr2 <- ape::rcoal( 100 )
tr1$tip.label <- gsub(tr1$tip.label, patt = 't', rep = 's')
tr1$edge.length <- tr1$edge.length*.5
tr1$root.edge <- 1
tr2$root.edge <- 1
tr <- ape::bind.tree(tr1, tr2, position = .5 ) |> ape::multi2di()
```

```
# subsample the tree to simulating missing tips and estimate structure
ex <- sample( tr$tip.label, size = 30, replace = FALSE)
tr0 <- ape::drop.tip( tr, ex )
(s0 <- treestructure::trestruct( tr0 ))

# assign structure to the previously missing tips
(s <- treestructure::addtips( s0, tr ))
```

`plot.TreeStructure` *Plot TreeStructure tree with cluster and partition variables*

Description

Plot TreeStructure tree with cluster and partition variables

Usage

```
## S3 method for class 'TreeStructure'
plot(x, use_ggtree = TRUE, ...)
```

Arguments

`x`	A TreeStructure object
`use_ggtree`	Toggle ggtree or ape plotting behavior
`...`	Additional arguments passed to ggtree or ape::plot.phylo

Examples

```
#tree <- ape::read.tree( system.file('sim.nwk', package = 'treestructure') )
# you can run the example below before plotting
#struc <- trestruct( tree )

#because it can take a minute or so to run treestructure, we will load it here
struc <- readRDS( system.file('struc_plot_example.rds', package='treestructure') )
#plot treestructure object

suppressWarnings(plot(struc))
```

`treestructure.test` *Test treestructure hypothesis*

Description

Test the hypothesis that two clades within a tree were generated by the same coalescent process.

Usage

```
treestructure.test(tre, x, y, nsim = 10000)
```

Arguments

`tre`	An ape::phylo tree, must be binary and rooted
`x`	A character vector of tip labels or numeric node numbers. If numeric, can include internal node numbers.
`y`	as x, but must be disjoint with x
`nsim`	Number of simulations (larger = slower and more accurate)

Examples

```
tree <- ape::read.tree( system.file('sim.nwk', package = 'treestructure') )

# you can run the example below before running test
#struc <- trestruct( tree )

#because it can take a minute or so to run treestructure, we will load it here
struc <- readRDS( system.file('struc_plot_example.rds', package='treestructure') )

#run the test

results <- treestructure.test(tree, x = struc$clusterSets[[1]],
                              y = struc$clusterSets[[2]])

print(results)
```

`trestruct` *Detect cryptic population structure in time trees*

Description

Estimates a partition of a time-scaled tree by contrasting coalescent patterns.

Usage

```
trestruct(
  tre,
  minCladeSize = 25,
  minOverlap = -Inf,
  nodeSupportValues = FALSE,
  nodeSupportThreshold = 95,
  nsim = 10000,
  level = 0.01,
  ncpu = 1,
  verbosity = 1,
  debugLevel = 0,
  levellb = 0.001,
  levelub = 0.1,
  res = 11
)
```

Arguments

`tre`	A tree of type ape::phylo. Must be rooted. If the tree has multifurcations, it will be converted to a binary tree before processing.
`minCladeSize`	All clusters within partition must have at least this many tips.
`minOverlap`	Threshold time overlap required to find splits in a clade.
`nodeSupportValues`	Node support values such as produced by bootstrap or Bayesian credibility scores. Must be logical or vector with length equal to number of internal nodes in the tree. If nodeSupportValues = TRUE, then the function will get the information on node support from the tree. If numeric vector, these values should be between 0 and 100.
`nodeSupportThreshold`	Threshold node support value between 0 and 100. Nodes with support lower than this threshold will not be tested.
`nsim`	Number of simulations for computing null distribution of test statistics.
`level`	Significance level for finding new split within a set of tips. Can also be NULL, in which case the optimal level is found according to the CH index (see details).
`ncpu`	If > 1 will compute statistics in parallel using multiple CPUs.
`verbosity`	If > 0 will print information about progress of the algorithm.
`debugLevel`	If > 0 will produce additional data in return value.
`levellb`	If optimizing the ‘level‘ parameter, this is the lower bound for the search.
`levelub`	If optimizing the ‘level‘ parameter, this is the upper bound for the search.
`res`	If optimizing the ‘level‘ parameter, this is the number of values to test.

Details

Estimates a partition of a time-scaled tree by contrasting coalescent patterns. The algorithm is premised on a Kingman coalescent null hypothesis for the ordering of node heights when contrasting two clades, and a test statistic is formulated based on the rank sum of node times in the tree. If node support values are available (as computed by bootstrap procedures), the method can optionally exclude designation of structure on poorly supported nodes. The method will not designate structure on nodes with zero branch length relative to their immediate ancestor. The significance level for detecting significant partitions of the tree can be provided, or a range of values can be examined. The CH index based on within- and between-cluster variance in node heights can be used to select a significance level if none is provided.

Value

A TreeStructure object which includes cluster and partition assignment for each tip of the tree.

References

Volz EM, Carsten W, Grad YH, Frost SDW, Dennis AM, Didelot X. Identification of hidden population structure in time-scaled phylogenies. Systematic Biology 2020; 69(5):884-896.

Author(s)

Erik M Volz

Examples

```
tree <- ape::rcoal(50)
struct <-  trestruct( tree )
print(struct)
```

Index