Package ‘treeducken’

October 14, 2022

Type Package

Title Nested Phylogenetic Tree Simulator

Version 1.1.0

Description Simulates nested phylogenetic trees (gene trees in species tree, symbiont trees in host trees) using birth-death processes and transfers between lineages. Simulations of gene trees within species trees are performed using a three-tree model with species trees, locus trees, and gene trees. The cophylogenetic birth-death process is used to simulate sets of host and symbiont trees with extant associations between tips. For more information about the three-tree model see: Mallo et al. (2015) <doi:10.1093/sysbio/syv082>, Rasmussen and Kellis (2012) <doi:10.1101/gr.123901.111>.

License GPL-3

SystemRequirements C++11

Imports Rcpp (>= 1.0.2), apTreeshape, graphics, methods

Depends ape

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

LazyData true

Encoding UTF-8

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Author Wade Dismukes [aut, cre],
Tracy A. Heath [aut],
Josh Justison [ctb],
Damien de Vienne [ctb],
Liam Revell [ctb],
Emmanuel Paradis [ctb],
Klaus Schliep [ctb],
Ben Bolker [ctb],
Luke J. Harmon [ctb],
Joseph W. Brown [ctb]

Maintainer Wade Dismukes <`wade.dismukes@gmail.com`>

Repository CRAN

Date/Publication 2021-03-02 05:20:03 UTC

R topics documented:

`treeducken-package` *treeducken: simulates cophylogenetic systems & nested phylogenies*

Description

treeducken simulates cophylogenetic systems such as host and symbiont pairs. This is done using the `sim_cophylo_bdp` function. This function simulates a host tree and a symbiont tree simultaneously using a matrix describing the associations between contemporaneous hosts and symbionts. These simulations allow for varying rates of host-shift speciation and cospeciation in addition to independent birth and death rates of the host and the symbiont trees. treeducken is also able to simulate nested phylogenies such as might be expected in the case of gene tree and species tree scenarios.

Details

Package:	treeducken
Type:	Package
Version:	1.0
Date:	2020-03-25
License:	GPL-2
LazyLoad:	yes

Author(s)

Wade Dismukes <wade.dismukes@gmail.com>

References

Dismukes W. and Tracy A. Heath, Something something (work-in-progress)

See Also

`ape`, `geiger`, `TreeSim`, `PhyTools`

`add_events` *Add events from sim_cophyBD to plot.cophy*

Description

This will plot the classical cophylogenetic events onto the plot from plot.cophy if event_history is non-empty. At present this only works with phylograms.

Usage

```
add_events(
  cophy_obj,
  legend = TRUE,
  pch = NULL,
  col = NULL,
  gap = 1,
  fsize = 1,
  type = "phylogram",
  show_tip_label = TRUE
)
```

Arguments

`cophy_obj`	Cophylogenetic object
`legend`	Boolean to turn on or off the legend
`pch`	Length 8 vector of plotting symbols to be used
`col`	Length 8 vector of colors to be used in plot
`gap`	the gap between the two trees
`fsize`	the font size of tips (this must be set the same as used in plot.cophy)
`type`	the type of graph ("phylogram" or "cladogram")
`show_tip_label`	Boolean indicating whether the plot has tip labels or not

Details

The pch and color vectors place symbols or colors for the different events. The order this vector is input determines which symbol corresponds to which event. * Position 1 = cospeciation * Position 2 = host speciation * Position 3 = host extinction * Position 4 = symbiont speciation * Position 5 = symbiont extinction * Position 6 = host spread or host-switch symbiont speciation * Position 7 = anagenetic symbiont dispersal * Position 8 = anadenetic symbiont extirpation

By default a color vector is used in this order: purple, red, blue, darkorange, cyan, yellow, brown, seagreen

Examples

```
host_mu <- 1.0 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 10
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophylo_pair <- sim_cophyBD(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
```

```
                                host_exp_rate = host_shift_rate,
                                sdr = symb_mu,
                                sbr = symb_lambda,
                                numbsim = numb_replicates,

                                time_to_sim = time)
plot.cophy(cophylo_pair[[1]])
add_events(cophylo_pair[[1]], legend = FALSE)
```

`add_scalebar` *Add scale bar to cophylo plot*

Description

This function plots a host and symbiont tree given the object returned by ‘sim_cophyBD‘.

Usage

```
add_scalebar(host_coords, symb_coords, fsize)
```

Arguments

`host_coords`	Host x,y coordinates
`symb_coords`	Symb x,y coordinates
`fsize`	Font size of scale bar

`build_historical_association_matrix` *Reconstruct historical association matrix*

Description

Reconstruct historical association matrix

Usage

```
build_historical_association_matrix(t, tr_pair_obj)

get_assoc(t, tr_pair_obj)
```

Arguments

`t`	The time of interest
`tr_pair_obj`	The tree pair object from ‘sim_cophyBD‘

Details

Given a time and a tree pair object produced by the 'sim_cophyBD' object will produce the association matrix at that time point for the tree object. USER WARNING: this is still in development, and likely will not work all the time.

Value

Matrix of the associations at given time

Examples

```
host_mu <- 1.0 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 1
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophylo_pair <- sim_cophyBD(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
                           host_exp_rate = host_shift_rate,
                           sdr = symb_mu,
                           sbr = symb_lambda,
                           numbsim = numb_replicates,
                           time_to_sim = time)
time <- 1.0
assoc_mat_at_t <- get_assoc(t=time, tr_pair_obj = cophylo_pair[[1]])
```

`c.cophy` *Combine cophylogenetic sets into a multiCophy object*

Description

Combines cophylogenetic sets into a multiCophy object.

Usage

```
## S3 method for class 'cophy'
c(...)

## S3 method for class 'multiCophy'
c(...)
```

Arguments

`...`	Values of class ‘cophy‘

Value

An object of class ‘multiCophy‘

Functions

- `c.multiCophy`: Combines two multiCophy objects into one multiCophy object

Author(s)

Wade Dismukes and Emmanuel Paradis

See Also

‘c‘ generic function

Examples

```
h_lambda <- 1.0
h_mu <- 0.3
c_lambda <- 0.0
s_lambda <- 1.0
s_mu <- 0.3
s_her <- 0.0
host_symb_sets <-  sim_cophylo_bdp(hbr = h_lambda,
                                  hdr = h_mu,
                                  sbr = s_lambda,
                                  cosp_rate = c_lambda,
                                  sdr = s_mu,
                                  host_exp_rate = s_her,
                                  time_to_sim = 1.0,
                                  numbsim = 2)
host_symb_sets2 <- sim_cophylo_bdp(hbr = h_lambda,
                                  hdr = h_mu,
                                  sbr = s_lambda,
                                  cosp_rate = c_lambda,
                                  sdr = s_mu,
                                  host_exp_rate = s_her,
                                  time_to_sim = 1.0,
                                  numbsim = 2)
multi_host_symb <- c(host_symb_sets[[1]], host_symb_sets2[[2]])
multi_host_symb_alt <- c(host_symb_sets, host_symb_sets2)
```

`calculate_expected_leaves_locustree`

Calculate expected leaves of a locus tree

Description

Calculate expected leaves of a locus tree

Usage

```
calculate_expected_leaves_locustree(t, dup_rate, loss_rate, num_species)

ave_tips_lt(t, dup_rate, loss_rate, num_species)
```

Arguments

`t`	time to simulate until (the length of the species tree)
`dup_rate`	gene birth rate
`loss_rate`	gene death rate
`num_species`	number of leaves on the species tree

Details

Calculates the expected number of leaves for a birth-death simulation given a gene birth and death rate, a time, and the number of leaves on the species tree that the locus tree is to be simulated upon.

Value

The expected number of leaves

References

Mallo, D., de Oliveira Martins, L., & Posada, D. (2016). SimPhy: phylogenomic simulation of gene, locus, and species trees. Systematic biology, 65(2), 334-344.

Examples

```
gene_birth_rate <- 1.0
gene_death_rate <- 0.5
time <- 2
num_species <- 10
ave_tips_lt(time,
                                gene_birth_rate,
                                gene_death_rate,
                                num_species)
```

`calculate_expected_leaves_sptree`

Calculate expected leaves of a species tree

Description

Calculate expected leaves of a species tree

Usage

```
calculate_expected_leaves_sptree(lambda, mu, t)

ave_tips_st(lambda, mu, t)
```

Arguments

`lambda`	speciation rate
`mu`	extinction rate
`t`	time to simulate until

Details

Calculates the expected number of leaves for a birth-death simulation given a speciation and extinction rate and a time.

Value

The expected number of leaves

References

Mooers, A., Gascuel, O., Stadler, T., Li, H., & Steel, M. (2012). Branch lengths on birth–death trees and the expected loss of phylogenetic diversity. Systematic biology, 61(2), 195-203.

Examples

```
spec_rate <- 1.0
ext_rate <- 0.5
time <- 2
ave_tips_st(spec_rate, ext_rate, time)
```

`collapse_locus_subtree` *Collapse a clade into a single tip*

Description

Collapse a clade into a single tip

Usage

```
collapse_locus_subtree(list_of_subtrees, locus_to_collapse)

collapse_clade(list_of_subtrees, locus_to_collapse)
```

Arguments

`list_of_subtrees`	a list of type ‘multiPhylo‘
`locus_to_collapse`	a subtree found within a subst of ‘list_of_subtrees‘

Details

Takes a clade as input and collapses that clade to one tip in all trees in ‘list_of_subtrees‘.

Value

multiPhy (list of trees) with the subtree in question collapse

Examples

```
lambda <- 1.0
mu <- 0.2
nt <- 10
trees <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
subtrees_of_trees <- ape::subtrees(trees[[1]])
st_of_interest <- subtrees_of_trees[[1]]
collapse_st_of_interest <- collapse_locus_subtree(trees, st_of_interest)
```

`convert_assoc_table_to_matrix` *Convert a table with host and symbiont associations to a matrix*

Description

Converts a table of associations to an association matrix with rows as symbionts and columns as host.

Usage

```
convert_assoc_table_to_matrix(assoc_table)

make_mat(assoc_table)
```

Arguments

`assoc_table`	A dataframe with two columns

Details

Converts a dataframe with first column listing the host individually and the second column as the symbionts. If hosts have more than one symbiont list these with commas. For example, if the table is a tab-delimited file then a row should read: "Hostus_mostus Symbiont_1, Symbiont_2".

Value

A matrix with rows as symbionts and columns as hosts with 1's representing an association.

Examples

```
file_path <- system.file("extdata",
                         "gopher_lice_mapping.txt",
                          package = "treeducken")
gopher_lice_map <- read.table(file_path,
                              stringsAsFactors = FALSE,
                              header = TRUE)
gopher_lice_assoc_matrix <- convert_assoc_table_to_matrix(gopher_lice_map)
```

`convert_to_cophy` *Converts an object into an object of type cophy*

Description

Functions for converting either a list of three components (host_tree, symb_tree, and association_mat) into an object of class cophy Otherwise turns arguments into the cophy object if inputting a hostTree of type ‘phylo‘, a symbiont tree of type ‘phylo‘, and a matrix of type eventHistory.

Usage

```
convert_to_cophy(hostTree, symbTree, assocMat, eventHistory = NULL)

to_cophy(hostTree, symbTree, assocMat, eventHistory = NULL)
```

Arguments

`hostTree`	An object of type ‘phylo‘
`symbTree`	An object of type ‘phylo‘
`assocMat`	A matrix with rows being extant symbionts and columns being extant hosts
`eventHistory`	An optional data frame of four columns: Symbiont Index, Host Index, Event Type (see details), and Event Time

Details

The association matrix must be with rows equal to the number of extant symbionts and columns equal to the number of extant hosts. Non-zero values in this matrix indicate associations (typically this will be a matrix of just zeros and ones).

The eventHistory parameter has four columns: Symbiont Index, Host Index, Event Type (see details), and Event Time. The indexing of the first two columns should follow the indexing of the ‘phylo‘ objects ‘hostTree‘ and ‘symbTree‘. The types of events are as follows: * HG - a host speciation event * HL - a host extinction event * C - a cospeciation event * SG - a symbiont speciation event * SL - a symbiont extinction event * AG - an association gain between symbiont x and host y * AL - an association loss between symbiont x and host y

Value

An object of type cophy

See Also

is.cophy

Examples

```
gopher_lice_map_path <- system.file("extdata",
                                  "gopher_lice_mapping.txt",
                                   package = "treeducken")
gopher_lice_map <- read.table(gopher_lice_map_path,
                              stringsAsFactors = FALSE,
                              header = TRUE)
gopher_tree_path <- system.file("extdata",
                                "gophers_bd.tre",
                                 package = "treeducken")
gopher_lice_assoc_matrix <- convert_assoc_table_to_matrix(gopher_lice_map)
gopher_tree <- ape::read.nexus(gopher_tree_path)
lice_tree_path <- system.file("extdata",
                              "lice_bd.tre",
                               package = "treeducken")
lice_tree <- ape::read.nexus(lice_tree_path)
gopher_lice_cophy <- to_cophy(hostTree = gopher_tree,
                                         symbTree = lice_tree,
                                         assocMat = gopher_lice_assoc_matrix)
```

---

cophy_summary_stat_by_indx

Calculates summary statistics for cophylogenetic objects

Description

For cophylogenetic objects produced in treeducken via ‘sim_cophyBD‘, calculates the numbers of different events of interest. In addition, calculates and tests the ParaFit test.

Usage

```
cophy_summary_stat_by_indx(cophy_obj, cophy_obj_indx)

summarize_1cophy(cophy_obj, cophy_obj_indx)

cophy_summary_stat(cophy_obj)

summarize_cophy(cophy_obj)
```

Arguments

`cophy_obj`	The cophylogenetic object produced via ‘sim_cophyBD‘
`cophy_obj_indx`	The index with ‘cophy_obj‘ for ‘summarize_1cophy‘

Value

A vector consisting of (in order) cospeciations, host speciations, host extinctions, symbiont speciations, symbiont extinctions, host spread/switch speciations, symbiont dispersals, symbiont extirpations, parafit statistic, and parafit p-value

A dataframe containing statistics relevant to cophylogenetic analysis

Functions

- `cophy_summary_stat_by_indx`: Calculates the summary statistics for one index of the list of cophylogenetic objects

Examples

```
host_mu <- 0.5 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 1
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophy_pair <- sim_cophyBD(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
                           host_exp_rate = host_shift_rate,
                           sdr = symb_mu,
                           sbr = symb_lambda,
                           numbsim = numb_replicates,
                           time_to_sim = time)
summary_stats <- summarize_cophy(cophy_pair)
```

`count_cherries` *Calculate cherry statistic for gene-trees*

Description

Calculate cherry statistic according to the definition given in McKenzie and Steel 2000 (see below for reference)

Usage

```
count_cherries(tree)
```

Arguments

`tree`	an object of class "phylo"

Details

This calculates the value for the cherry test statistic on a rooted tree. Note that this does not perform the actual hypothesis test against Yule or uniform tree models.

Value

The value fo cherries on a tree

Author(s)

Emmanuel Paradis

References

McKenzie, A. and Steel, M. (2000) Distributions of cherries for two models of trees. Mathematical Biosciences, 164, 81–92.

Examples

```
# first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
treeducken::count_cherries(tr[[1]])
# to do the hypothesis test you can use the ape version of this function
ape::cherry(tr[[1]])
```

`draw_cophy` *Internal tree plot function*

Description

internal plot function from ape::plotCophylo2 under GPL v. 2

Usage

```
draw_cophy(
  x,
  y,
  assoc = assoc,
  use_edge_length = use_edge_length,
  length_line = length_line,
  type = type,
  return = return,
  col = col,
  lwd = lwd,
  lty = lty,
```

```
  show_tip_label = show_tip_label,
  font = font,
  fsize = fsize,
  gap = gap,
  show_scalebar = show_scalebar,
  scalebar_fsize = scalebar_fsize,
  ...
)
```

Arguments

`x`	Host tree as phylo object
`y`	Symb tree as phylo object
`assoc`	Association matrix
`use_edge_length`	Boolean to draw trees with edge length or not
`length_line`	Length of interactions lines
`type`	string "phylogram" or "cladogram"
`return`	Return an object or no (default = FALSE)
`col`	What color to draw links between trees
`lwd`	Width of links between trees
`lty`	Type of line to draw between trees
`show_tip_label`	Boolean for showing labels
`font`	What font to use (bold, italic (default), etc.)
`fsize`	What size font as a character expansion factor (same as cex)
`gap`	Gap between tips and tip names
`show_scalebar`	Boolean for turning on and off the scalebar
`scalebar_fsize`	Font size of scalebar
`...`	Other plotting parameters

References

Paradis E. & Schliep K. 2019. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. Bioinformatics 35: 526-528.

`draw_curve` *Curve draw function*

Description

internal function to draw curved links between tips modified from Liam Revell phytools package under GPL v. 2

Usage

```
draw_curve(x, y, scale = 0.01, ...)
```

Arguments

`x`	x positions on graph
`y`	y positions on graph
`scale`	Scale of the logistic (which is where the curve comes from)
`...`	Other plotting parameters

Author(s)

Wade Dismukes and Liam J Revell

References

Revell, L.J. (2012), phytools: an R package for phylogenetic comparative biology (and other things). Methods in Ecology and Evolution, 3: 217-223. doi:10.1111/j.2041-210X.2011.00169.x

---

`drop_extinct` *Drops extinct tips from tree*

Description

Drops extinct tips from tree

Usage

```
drop_extinct(phy, tol = NULL)
```

Arguments

`phy`	a ’phylo’ class object
`tol`	tolerance in decimal values for branch lengths

Value

A 'phylo' class object with extinct tips removed

Author(s)

LJ Harmon, and JW Brown This is a direct port of the geiger function, I import it here for convenience. This code is copied under GPL 3 license.

References

Pennell M, Eastman J, Slater G, Brown J, Uyeda J, Fitzjohn R, Alfaro M, Harmon L (2014). "geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees." Bioinformatics, 30, 2216-2218

Examples

```
mu <- 0.5 # death rate
lambda <- 2.0 # birth rate
numb_replicates <- 10
numb_extant_tips <- 4
# simulate trees under the GSA so first simulates a tree with
# numb_extant_tips * 100 tips counting each time we have a tree with 10 tips
# then randomly picks one of those trees

tree_list <- sim_sptree_bdp(sbr = lambda,
                sdr = mu,
                numbsim = numb_replicates,
                n_tips = numb_extant_tips)
pruned <- drop_extinct(tree_list[[1]])
```

`estimate_node_heights` *Calculate expected time to branching point of a species tree*

Description

Calculates the expected time to branching point of a species tree for a birth-death simulation given a speciation and extinction rate and a number of leaves, and a branching point.

Usage

```
estimate_node_heights(lambda, mu, n, k = 1)

node_heights(lambda, mu, n, k = 1)
```

Arguments

`lambda`	speciation rate
`mu`	extinction rate
`n`	number of tips on tree
`k`	branching point (k = 1 is the root and is the default)

Details

By default this branching point is 1 which corresponds to the root, k = 2 corresponds to the first branching point after the root, k = 3 the second, and so on. For more details see Gernhard 2008.

Value

The expected branching time

References

Gernhard, T. (2008). The conditioned reconstructed process. Journal of theoretical biology, 253(4), 769-778.

Examples

```
spec_rate <- 1.0
ext_rate <- 0.5
nt <- 10
node_heights(lambda = spec_rate, mu = ext_rate, n = nt)

node_heights(lambda = spec_rate, mu = ext_rate, n = nt, k = 2)
```

`event_history` *Summarize a cophylogenetic set*

Description

Several utility functions for cophylogenetic set summarization. Including functions for printing an entire summary, and a summary of each part: host_tree, symb_tree, association_mat, and event_history.

Usage

```
event_history(cophy)

## S3 method for class 'cophy'
event_history(cophy)

## S3 method for class 'cophy'
summary(object, ...)
```

Arguments

`cophy`	Cophylogenetic set
`object`	An object of class ‘cophy‘
`...`	Further arguments used in generic classes

Details

The summary for a cophylogenetic set outputs a summary of the host tree and the symbiont tree. The number of rows and columns of the association matrix, and a summary of the event_history.

Value

Summary returns NULL.

Functions

- `event_history`: Returns event history of a cophylogenetic set
- `event_history.cophy`: Returns event history of a cophylogenetic set

Author(s)

Wade Dismukes and Emmanuel Paradis

See Also

sim_cophylo_bdp, summary for the generic, multiCophy, c.cophy

Examples

```
h_lambda <- 1.0
h_mu <- 0.3
c_lambda <- 0.0
s_lambda <- 1.0
s_mu <- 0.3
s_her <- 0.0
host_symb_sets <- sim_cophylo_bdp(hbr = h_lambda,
                                  hdr = h_mu,
                                  sbr = s_lambda,
                                  cosp_rate = c_lambda,
                                  sdr = s_mu,
                                  host_exp_rate = s_her,
                                  time_to_sim = 1.0,
                                  numbsim = 1)
summary(host_symb_sets[[1]])
```

`genetree_summary_stat` *Calculate summary statistics for gene trees*

Description

Calculates summary statistics including Colless' statistic, gamma statistic of the locus tree input as an index as part of a list, gamma statistic of gene tree, sackin statistic, cherry statistic, and time to most recent common ancestor

Usage

```
genetree_summary_stat(lt_obj, lt_indx)

summarize_gt(lt_obj, lt_indx)
```

Arguments

`lt_obj`	Locus tree object obtain from 'sim_lt_gt_mlc'
`lt_indx`	Index of locus tree object of interest

Value

Dataframe with summary statistics for each gene tree

Examples

```
# first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
# for a locus tree with 100 genes sampled per locus tree
gentrees <- sim_msc(tr[[1]],
                    ne = 10000,
                    num_sampled_individuals = 1,
                    num_genes = 100)

gt_df <- summarize_gt(gentrees, lt_indx = 1)
```

`get_loci` *Separate a locus tree into loci*

Description

Separate a locus tree into loci

Usage

```
get_loci(locus_tree)
```

Arguments

`locus_tree`	tree of type ‘phy‘

Details

This separates loci based on node labels "D[A-Z]". This is intended to be used internally, but should work with other trees where duplications are marked similarly.

Value

list of subtrees (with ‘locus_tree‘ at the end‘)

Examples

```
# first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
# for a locus tree with 100 genes sampled per locus tree
gene_br <- 0.1
gene_dr <- 0.02
transfer_rate <- 0.0
locus_tree <- sim_ltBD(species_tree = tr[[1]],
                  gbr = gene_br,
                  gdr = gene_dr,
                  lgtr = transfer_rate,
                  num_loci = 1)
locus_tree_subtrees <- get_loci(locus_tree[[1]])
```

`get_tip_labels_tree_list` *Get all the tip labels of a ‘multiPhylo‘ object*

Description

Get all the tip labels of a ‘multiPhylo‘ object

Usage

```
get_tip_labels_tree_list(multi_tree)

get_tipnames(multi_tree)
```

Arguments

`multi_tree`	an object of class ‘multiPhylo‘

Details

Retrieves the member "tip.label" from each tree in multi_tree

Value

a list of the same length as ‘multi_tree‘ with only the tip labels

Examples

```
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 5, n_tips = nt)
tips_of_tr <- get_tip_labels_tree_list(tr)
```

`host_tree` *Print a cophylogenetic set*

Description

Prints a cophylogenetic set or a list of cophylogenetic sets.

Usage

```
host_tree(cophy)

## S3 method for class 'cophy'
host_tree(cophy)

## S3 method for class 'multiCophy'
host_tree(cophy)

symb_tree(cophy)

## S3 method for class 'cophy'
symb_tree(cophy)

## S3 method for class 'multiCophy'
symb_tree(cophy)

association_mat(cophy)

## S3 method for class 'cophy'
association_mat(cophy)

## S3 method for class 'multiCophy'
association_mat(cophy)

## S3 method for class 'multiCophy'
event_history(cophy)

## S3 method for class 'cophy'
print(x, ...)

## S3 method for class 'multiCophy'
print(x, details = FALSE, ...)
```

Arguments

`cophy`	An object of class ‘cophy‘
`x`	An object of class ‘cophy‘ or class ‘multiCophy‘
`...`	Further arguments used in generic classes
`details`	A logical value, outputs brief summary of each set in the list.

Details

The association matrix is output with symbionts in columns and hosts in rows. The event history data frame has codes for the following events: "CSP" = cospeciation/codivergence, "HSP" = host speciation, "HX" = host extinction, "SSP" = symbiont speciation, "SX" = Symbiont extinction, "DISP" = Symbiont dispersal, and "EXTP" = symbiont extirpation, "SHE" symbiont speciation with host spread or host switch.

Value

Print returns NULL. host_tree returns NULL, symb_tree returns NULL, association_mat returns the dimensions of the matrix, event_history returns NULL.

Functions

- `host_tree`: Returns host tree of a cophylogenetic set
- `host_tree.cophy`: Returns host tree of a cophylogenetic set
- `host_tree.multiCophy`: Returns host tree of each member of a list of cophylogenetic sets
- `symb_tree`: Returns symb tree of a cophylogenetic set
- `symb_tree.cophy`: Returns symb tree of a cophylogenetic set
- `symb_tree.multiCophy`: Returns symb tree of each member of a list of cophylogenetic sets
- `association_mat`: Returns association matrix of a cophylogenetic set
- `association_mat.cophy`: Returns association matrix of a cophylogenetic set
- `association_mat.multiCophy`: Returns association matrix for each member of a list of cophylogenetic sets
- `event_history.multiCophy`: Returns event_history for each member of a list of cophylogenetic sets
- `print.multiCophy`: Prints a list of cophylogenetic sets

Author(s)

Wade Dismukes, Ben Bolker, and Emmanuel Paradis

See Also

sim_cophylo_bdp, print for the generic, multiCophy, c.cophy

Examples

```
h_lambda <- 1.0
h_mu <- 0.3
c_lambda <- 0.0
s_lambda <- 1.0
s_mu <- 0.3
s_her <- 0.0
host_symb_sets <- sim_cophylo_bdp(hbr = h_lambda,
                                  hdr = h_mu,
                                  sbr = s_lambda,
                                  cosp_rate = c_lambda,
                                  sdr = s_mu,
                                  host_exp_rate = s_her,
                                  time_to_sim = 1.0,
                                  numbsim = 4)
print(host_symb_sets[[1]])
host_tree(host_symb_sets[[1]])
symb_tree(host_symb_sets[[1]])
```

```
association_mat(host_symb_sets[[1]])
event_history(host_symb_sets[[1]])
print(host_symb_sets)
```

`is.cophy` *Test for the cophygenetic set object*

Description

Tests if an object is of class ‘cophy‘

Usage

```
is.cophy(cophy)

is.multiCophylo(multiCophy)
```

Arguments

`cophy`	an object to test to see if it is of class ‘cophy‘
`multiCophy`	an object to test for multiCophy

Details

Checks that an object is of class ‘cophy‘. For multicophy checks that the class is ‘multiCophylo‘ and that each element is of class ‘cophy‘.

Value

A logical vector

Functions

- `is.multiCophylo`: Tests for ‘multiCophylo‘ composed of ‘cophy‘ objects

See Also

as.cophy

Examples

```
h_lambda <- 1.0
h_mu <- 0.3
c_lambda <- 0.0
s_lambda <- 1.0
s_mu <- 0.3
s_her <- 0.0
host_symb_sets <- sim_cophyBD(hbr = h_lambda,
                                  hdr = h_mu,
```

```
                                  sbr = s_lambda,
                                  cosp_rate = c_lambda,
                                  sdr = s_mu,
                                  host_exp_rate = s_her,
                                  time_to_sim = 2.0,
                                  numbsim = 1)
is.cophy(host_symb_sets[[1]])
is.multiCophylo(host_symb_sets)
```

`is_extinct` *Identify extinct tips from tree*

Description

This is a direct port of the geiger function, I import it here for convenience. This code is copied under GPL 3 license.

Usage

```
is_extinct(phy, tol = NULL)
```

Arguments

`phy`	a 'phylo' class object
`tol`	tolerance in decimal values for branch lengths

Value

A list of the tips that are extinct

References

Pennell M, Eastman J, Slater G, Brown J, Uyeda J, Fitzjohn R, Alfaro M, Harmon L (2014). "geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees." Bioinformatics, 30, 2216-2218

Examples

```
mu <- 0.5 # death rate
lambda <- 2.0 # birth rate
numb_replicates <- 10
numb_extant_tips <- 4
# simulate trees under the GSA so first simulates a tree with
# numb_extant_tips * 100 tips counting each time we have a tree with 10 tips
# then randomly picks one of those trees

tree_list <- sim_sptree_bdp(sbr = lambda,
                sdr = mu,
                numbsim = numb_replicates,
```

```
              n_tips = numb_extant_tips)
is_extinct(tree_list[[1]])
```

`make_textbox` *Internal tree plot function*

Description

internal function to make textbox for tip labels modified from phytools::TEXTBOX package under GPL v. 2

Usage

```
make_textbox(x, y, label, pos, offset, cex, font)
```

Arguments

`x`	x coordinates
`y`	y coordinates
`label`	Labels as vector of strings
`pos`	Position in plot environment
`offset`	How offset from tips
`cex`	a numerical vector giving the amount by which characters should be scaled relative to the default
`font`	font choice

Author(s)

Wade Dismukes and Liam J Revell

References

Revell, L.J. (2012), phytools: an R package for phylogenetic comparative biology (and other things). Methods in Ecology and Evolution, 3: 217-223. doi:10.1111/j.2041-210X.2011.00169.x

---

`parafit_stat` *Calculate the ParafitGlobal statistic on 2 trees and their association matrix*

Description

Calculate the ParafitGlobal Statistic to be used in the hypothesis test described in Legendre et al. (2002). The null hypothesis of this test being that the evolution of the two trees together with their associations at the present have been independent.

Usage

```
parafit_stat(host_tr, symb_tr, assoc_mat)

parafit_test(host_tr, symb_tr, assoc_mat, D, reps = 99)
```

Arguments

`host_tr`	The host tree of class "phy"
`symb_tr`	The symbiont tree of class "phy
`assoc_mat`	Association matrix between the extant tips of ‘host_tr‘ and ‘symb_tr‘
`D`	the statistic calculated using ‘parafit_stat‘
`reps`	Number of permutations to perform on the association matrix for the hypothesis test

Details

‘parafit_stat‘ drops any non-extant tips from the tree. Then the phylogenetic distance matrix is obtained for both host and symbiont tree. Next the principal coordinates are found for the host and symbiont distance matrices before these PCoA vectors are used in the following matrix multiplication following Legendre et al. (2002): D = H t(A) A. The trace is then found of this to get our ParaFitGlobal Statistic.

The test function ‘parafit_test‘ performs a row-wise permutation of the association matrix as described in Legendre et al. 2002. This is performed a number of times set by the user (default is 999) and a p-value is output.

The value from this is input into the test function. Note that this gives only the raw statistic unlike ‘ape::parafit‘. That is the only reason it is implemented here in treeducken (similar to ‘treeducken::cherries‘).

Value

A p-value for the hypothesis test described above

Functions

- `parafit_test`: Perform ParaFit Hypothesis Test

References

Legendre, P., Y. Desdevises and E. Bazin. 2002. A statistical test for host-parasite coevolution. Systematic Biology, 51(2), 217–234.

See Also

parafit_test

Examples

```
tr_pair <- sim_cophyBD(hbr=0.1,
                          hdr=0.05,
                          sdr=0.1,
                          host_exp_rate=0.4,
                          sbr = 0.05,
                          cosp_rate = 1.0,
                          numbsim = 1,
                          time_to_sim = 1)
# maybe we are interested in only cophylogenetic object 1
ht <- tr_pair[[1]]$host_tree
st <- tr_pair[[1]]$symb_tree
A <- tr_pair[[1]]$association_mat
pfs <- parafit_stat(host_tr = ht, symb_tr = st, assoc_mat = A)

parafit_test(ht, st, A, pfs, reps = 19)
```

`plot.cophy` *Plot host and symbiont pair with current associations*

Description

This function plots a host and symbiont tree given the object returned by ‘sim_cophyBD‘.

Usage

```
## S3 method for class 'cophy'
plot(
  x,
  use_edge_length = TRUE,
  type = "phylogram",
  col = par("fg"),
  lwd = par("lwd"),
  lty = par("lty"),
  show_tip_label = TRUE,
  gap = 1,
  font = 3,
  fsize = 1,
  show_div_bar = FALSE,
```

```
  ...
)

## S3 method for class 'multiCophy'
plot(x, ...)
```

Arguments

`x`	object of class multiCophy
`use_edge_length`	Boolean to draw trees with edge length or not
`type`	string "phylogram" or "cladogram"
`col`	What color to draw links between trees
`lwd`	Width of links between trees
`lty`	Type of line to draw between trees
`show_tip_label`	Boolean for showing labels
`gap`	Size of the gap between the tips and tip names
`font`	What font to use (bold, italic (default), etc.)
`fsize`	What size font as a character expansion factor (same as cex)
`show_div_bar`	Shows a bar under both trees with ticks where the divergences are (default: F)
`...`	other plotting parameters

Details

This function is mostly an altered version of the cophyloplot function written by Damien de Vienne Copyright 2008 - 2010 under GPL.

Value

a plot of the host and symbiont tree with extant interactions

Functions

- `plot.multiCophy`: Plots multiple cophy plots

Author(s)

Wade Dismukes and Damien de Vienne

Examples

```
host_mu <- 1.0 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 10
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
```

```
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophylo_pair <- sim_cophyBD(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
                           host_exp_rate = host_shift_rate,
                           sdr = symb_mu,
                           sbr = symb_lambda,
                           numbsim = numb_replicates,

                           time_to_sim = time)
plot.cophy(cophylo_pair[[1]])
```

`retrieve_parent_genetrees` *Retrieve all gene trees of the parent tree from a list generated from sim_mlc*

Description

Retrieve all gene trees of the parent tree from a list generated from sim_mlc

Retrieves the gene trees of the child subtrees

Usage

```
retrieve_parent_genetrees(gene_tree_list)

get_parent_gts(gene_tree_list)

retrieve_child_genetrees(gene_tree_list)

get_child_gts(gene_tree_list)
```

Arguments

`gene_tree_list`	A list of length 2: "parent_tree" and "child_trees" both of which are of class "multiPhylo"

Value

A ‘multiPhylo‘ object of only the gene trees generated on the parent subtree

Functions

- `retrieve_child_genetrees`: Returns a list of objects of class ‘multiPhylo‘

Examples

```
#' # first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
# for a locus tree with 100 genes sampled per locus tree
gene_br <- 0.1
gene_dr <- 0.02
transfer_rate <- 0.2
locus_tree <- sim_ltBD(species_tree = tr[[1]],
                  gbr = gene_br,
                  gdr = gene_dr,
                  lgtr = transfer_rate,
                  num_loci = 1)
effect_popsize <- 1e6
gene_tree_obj <- sim_mlc(locus_tree[[1]],
                                      effect_popsize,
                                      num_reps = 2)
parent_trees <- retrieve_parent_genetrees(gene_tree_obj)
```

`sim_cophyBD` *Simulates a host-symbiont system using a cophylogenetic birth-death process*

Description

Simulates a host-symbiont system using a cophylogenetic birth-death process

Usage

```
sim_cophyBD(
  hbr,
  hdr,
  sbr,
  sdr,
  host_exp_rate,
  cosp_rate,
  time_to_sim,
  numbsim,
  host_limit = 0L,
  hs_mode = FALSE
)

sim_cophylo_bdp(
  hbr,
  hdr,
  sbr,
```

```
  sdr,
  host_exp_rate,
  cosp_rate,
  time_to_sim,
  numbsim,
  host_limit = 0
)
```

Arguments

`hbr`	host tree birth rate
`hdr`	host tree death rate
`sbr`	symbiont tree birth rate
`sdr`	symbiont tree death rate
`host_exp_rate`	host shift speciation rate
`cosp_rate`	cospeciation rate
`time_to_sim`	time units to simulate until
`numbsim`	number of replicates
`host_limit`	Maximum number of hosts for symbionts (0 implies no limit)
`hs_mode`	Boolean turning host expansion into host switching (explained above) (default = FALSE)

Details

Simulates a cophylogenetic system using birth-death processes. The host tree is simulated following a constant rate birth-death process with an additional parameter - the cospeciation rate. This rate works as the speciation rate with the additional effect that if cospeciation occurs the symbiont tree also speciates. The symbiont tree is related to the host tree via an association matrix that describes which lineages are associated with which. The symbiont tree has an independent birth-death process with the addition of a host shift speciation rate that allows for the addition of more associated hosts upon symbiont speciation.

Host expansions are similar to the more commonly found host switching. In this model, host-expansion speciation describes events where a symbiont speciates and at that time, both descendants retain the ancestral host associations. Randomly one of these descendant symbionts then randomly acquires a new host. When the option ‘host_switch_mode = TRUE‘, the behavior of this changes to a more traditional host switching where one descendant retains the ancestral range and the other gains a novel host association.

Value

A list containing the ‘host_tree‘, the ‘symbiont_tree‘, the association matrix in the present, with hosts as rows and smybionts as columns, and the history of events that have occurred.

Examples

```
host_mu <- 0.5 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 10
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophylo_pair <- sim_cophyBD(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
                           host_exp_rate = host_shift_rate,
                           sdr = symb_mu,
                           sbr = symb_lambda,
                           numbsim = numb_replicates,
                           time_to_sim = time)
```

`sim_cophyBD_ana` *Simulates a host-symbiont system using a cophylogenetic birth-death process*

Description

Simulates a host-symbiont system using a cophylogenetic birth-death process

Usage

```
sim_cophyBD_ana(
  hbr,
  hdr,
  sbr,
  sdr,
  s_disp_r,
  s_extp_r,
  host_exp_rate,
  cosp_rate,
  time_to_sim,
  numbsim,
  host_limit = 0L,
  hs_mode = FALSE
)

sim_cophylo_bdp_ana(
  hbr,
```

```
  hdr,
  sbr,
  sdr,
  s_disp_r,
  s_extp_r,
  host_exp_rate,
  cosp_rate,
  time_to_sim,
  numbsim,
  host_limit = 0
)
```

Arguments

`hbr`	host tree birth rate
`hdr`	host tree death rate
`sbr`	symbiont tree birth rate
`sdr`	symbiont tree death rate
`s_disp_r`	symbiont dispersal rate to new hosts
`s_extp_r`	symbiont exirpation rate on h
`host_exp_rate`	host shift speciation rate
`cosp_rate`	cospeciation rate
`time_to_sim`	time units to simulate until
`numbsim`	number of replicates
`host_limit`	Maximum number of hosts for symbionts (0 implies no limit)
`hs_mode`	Boolean turning host expansion into host switching (explained above) (default = FALSE)

Details

Simulates a cophylogenetic system using birth-death processes with anagenetic processes allowing symbiont to gain or loss associations with hosts. The host tree is simulated following a constant rate birth-death process with an additional parameter - the cospeciation rate. This rate works as the speciation rate with the additional effect that if cospeciation occurs the symbiont tree also speciates. The symbiont tree is related to the host tree via an association matrix that describes which lineages are associated with which. The symbiont tree has an independent birth-death process with the addition of a host shift speciation rate that allows for the addition of more associated hosts upon symbiont speciation. The anagenetic processes are modeled using a poisson process occurring along the tree. The dispersal to hosts is at present random; there is no preferential host expansion.

Host expansions are similar to the more commonly found host switching. In this model, host-expansion speciation describes events where a symbiont speciates and at that time, both descendants retain the ancestral host associations. Randomly one of these descendant symbionts then randomly acquires a new host. When the option ‘host_switch_mode = TRUE‘, the behavior of this changes to a more traditional host switching where one descendant retains the ancestral range and the other gains a novel host association.

Value

A list containing the ‘host_tree‘, the ‘symbiont_tree‘, the association matrix in the present, with hosts as rows and smybionts as columns, and the history of events that have occurred.

Examples

```
host_mu <- 0.5 # death rate
host_lambda <- 2.0 # birth rate
numb_replicates <- 10
time <- 1.0
symb_mu <- 0.2
symb_lambda <- 0.4
host_shift_rate <- 0.0
cosp_rate <- 2.0

cophylo_pair <- sim_cophyBD_ana(hbr = host_lambda,
                           hdr = host_mu,
                           cosp_rate = cosp_rate,
                           s_disp_r = 1,
                           s_extp_r = 0.4,
                           host_exp_rate = host_shift_rate,
                           sdr = symb_mu,
                           sbr = symb_lambda,
                           numbsim = numb_replicates,
                           time_to_sim = time)
```

`sim_ltBD` *Simulates locus tree using constant rate birth-death-transfer process*

Description

Given a species tree simulates a locus or gene family tree along the species tree. Short for simulates a locus tree under a birth-death-transfer process.

Usage

```
sim_ltBD(species_tree, gbr, gdr, lgtr, num_loci, transfer_type = "random")

sim_locustree_bdp(
  species_tree,
  gbr,
  gdr,
  lgtr,
  num_loci,
  transfer_type = "random"
)
```

Arguments

`species_tree`	species tree to simulate along
`gbr`	gene birth rate
`gdr`	gene death rate
`lgtr`	gene transfer rate
`num_loci`	number of locus trees to simulate
`transfer_type`	The type of transfer input. Acceptable options: "cladewise" or "random"

Details

Given a species tree will perform a birth-death process coupled with transfer. The simulation runs along the species tree speciating and going extinct in addition to locus tree birth and deaths. Thus with parameters set to 0.0 a tree identical to the species tree is returned (it is relabel however).

Transfers are implemented as a birth-death process. One descendant lineage retains its species identity the other gains a new identity. At present, two types of transfers are implemented: "random" an "cladewise". The random transfer mode transfers one randomly chooses a contemporaneous lineage. Cladewise transfers choose lineages based on relatedness with more closely related lineages being more likely.

Value

List of objects of the tree class (as implemented in APE)

References

Rasmussen MD, Kellis M. Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Res. 2012;22(4):755–765. doi:10.1101/gr.123901.111

Examples

```
# first simulate a species tree
mu <- 0.5 # death rate
lambda <- 2.0 # birth rate
numb_replicates <- 10
numb_extant_tips <- 4
# simulate trees under the GSA so first simulates a tree with
# numb_extant_tips * 100 tips counting each time we have a tree with 10 tips
# then randomly picks one of those trees

sp_tree <- sim_stBD(sbr = lambda,
                sdr = mu,
                numbsim = numb_replicates,
                n_tips = numb_extant_tips)

gene_br <- 1.0
gene_dr <- 0.2
transfer_rate <- 0.2
sim_ltBD(species_tree = sp_tree[[1]],
                  gbr = gene_br,
```

```
                    gdr = gene_dr,
                    lgtr = transfer_rate,
                    num_loci = 10)
```

`sim_msc` *Simulate multispecies coalescent on a species tree*

Description

Simulates the multispecies coalescent on a species tree.

Usage

```
sim_msc(
  species_tree,
  ne,
  num_sampled_individuals,
  num_genes,
  rescale = TRUE,
  mutation_rate = 1L,
  generation_time = 1L
)

sim_multispecies_coal(
  species_tree,
  ne,
  num_sampled_individuals,
  num_genes,
  rescale = TRUE,
  mutation_rate = 1
)
```

Arguments

`species_tree`	input species tree of class "phylo"
`ne`	Effective population size
`num_sampled_individuals`	number of individuals sampled within each lineage
`num_genes`	number of genes to simulate within each locus
`rescale`	Rescale the tree into coalescent units (otherwise assumes it is in those units)
`mutation_rate`	The rate of mutation per generation
`generation_time`	The number of time units per generation

Details

This a multispecies coalescent simulator with two usage options. The function can rescale the given tree into coalescent units given the ‘mutation_rate‘, ‘ne‘, and the ‘generation_time‘. These result in a tree with coalescent times in units of expected number of mutations per site. The generation_time parameter default is 1 time unit per generation if the units of the tree are in millions of years The mutation_rate parameter is by default set to 1 mutations per site per generation (which is nonsensical). Rescale is set to true by default.

If rescale is set to false the tree is assumed to be in coalescent units and ‘ne‘ is used as the population genetic parameter theta.

Value

A list of coalescent trees

References

Bruce Rannala and Ziheng Yang (2003) Bayes Estimation of Species Divergence Times and Ancestral Population Sizes Using DNA Sequences From Multiple Loci Genetics August 1, 2003 vol. 164 no. 4 1645-1656 Mallo D, de Oliveira Martins L, Posada D (2015) SimPhy: Phylogenomic Simulation of Gene, Locus and Species Trees. Syst. Biol. doi: http://dx.doi.org/10.1093/sysbio/syv082

See Also

sim_ltBD, sim_stBD, sim_stBD_t

Examples

```
# first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
# for a locus tree with 100 genes sampled per locus tree
gentrees <- sim_msc(tr[[1]],
                    ne = 10000,
                    mutation_rate = 1e-9,
                    generation_time = 1e-6,
                    num_sampled_individuals = 1,
                    num_genes = 100)
```

`sim_multilocus_coal` *Simulates multi-locus coalescent on a given locus tree*

Description

separates a locus tree into loci broken up by duplications and simulates the coalescent on each loci.

Usage

```
sim_multilocus_coal(
  locus_tree,
  effective_pop_size,
  generation_time = 1,
  mutation_rate = 1e-06,
  num_reps
)

sim_mlc(
  locus_tree,
  effective_pop_size,
  generation_time = 1,
  mutation_rate = 1e-06,
  num_reps
)
```

Arguments

`locus_tree`	a locus tree from ‘sim_ltBD‘ of class ‘phy‘
`effective_pop_size`	the effective population size
`generation_time`	unit time per generation (default 1 year per generation)
`mutation_rate`	number of mutations per unit time
`num_reps`	number of coalescent simulations per locus

Details

This simulation follows the algorithm given in Rasmussen and Kellis 2012. The locus tree is scaled into coalescent units prior to being used. The generation_time parameter default assumes 1 generation per year if the units of the tree are in millions of years. The mutation_rate parameter is by default set to 1e-6 mutations per year (this is totally arbitrary). Also note that the return type is a list of many trees so for sufficiently complicated locus trees with ‘num_reps‘ set to a larger value may slow things considerably so use with caution.

Value

A list of list of gene trees of length ‘num_reps‘ simulated along each locus. The first member of the list is the parent tree, all others are child trees

Examples

```
# first simulate a species tree
mu <- 0.5
lambda <- 1.0
nt <- 6
tr <- sim_stBD(sbr = lambda, sdr = mu, numbsim = 1, n_tips = nt)
```

```
# for a locus tree with 100 genes sampled per locus tree
gene_br <- 0.1
gene_dr <- 0.02
transfer_rate <- 0.2
locus_tree <- sim_ltBD(species_tree = tr[[1]],
                 gbr = gene_br,
                 gdr = gene_dr,
                 lgtr = transfer_rate,
                 num_loci = 1)
effect_popsize <- 1e6
gene_tree_obj <- sim_mlc(locus_tree[[1]],
                                     effect_popsize,
                                     num_reps = 20)
```

`sim_stBD` *Simulates species trees using constant rate birth-death process*

Description

Forward simulates to a number of tips. This function does so using the general algorithm of Hartmann et al. 2010. Short for simulate species tree under birth-death process.

Usage

```
sim_stBD(sbr, sdr, numbsim, n_tips, gsa_stop_mult = 10L)

sim_sptree_bdp(sbr, sdr, numbsim, n_tips, gsa_stop_mult = 10)
```

Arguments

`sbr`	species birth rate (i.e. speciation rate)
`sdr`	species death rate (i.e. extinction rate)
`numbsim`	number of species trees to simulate
`n_tips`	number of tips to simulate to
`gsa_stop_mult`	number of tips to simulate the GSA tip to

Value

List of objects of the tree class (as implemented in APE)

References

K. Hartmann, D. Wong, T. Stadler. Sampling trees from evolutionary models. Syst. Biol., 59(4): 465-476, 2010.

T. Stadler. Simulating trees on a fixed number of extant species. Syst. Biol., 60: 676-684, 2011.

Examples

```
mu <- 0.5 # death rate
lambda <- 2.0 # birth rate
numb_replicates <- 10
numb_extant_tips <- 4
# simulate trees under the GSA so first simulates a tree with
# numb_extant_tips * 100 tips counting each time we have a tree with 10 tips
# then randomly picks one of those trees

tree_list <- sim_stBD(sbr = lambda,
                sdr = mu,
                numbsim = numb_replicates,
                n_tips = numb_extant_tips)
```

`sim_stBD_t` *Simulates species tree using constant rate birth-death process to a time*

Description

Forward simulates a tree until a provided time is reached.

Usage

```
sim_stBD_t(sbr, sdr, numbsim, t)

sim_sptree_bdp_time(sbr, sdr, numbsim, t)
```

Arguments

`sbr`	species birth rate (i.e. speciation rate)
`sdr`	species death rate (i.e. extinction rate)
`numbsim`	number of species trees to simulate
`t`	time to simulate to

Value

List of objects of the tree class (as implemented in APE)

References

K. Hartmann, D. Wong, T. Stadler. Sampling trees from evolutionary models. Syst. Biol., 59(4): 465-476, 2010.

T. Stadler. Simulating trees on a fixed number of extant species. Syst. Biol., 60: 676-684, 2011.

Examples

```
mu <- 0.5 # death rate
lambda <- 2.0 # birth rate
numb_replicates <- 10
time <- 1

tree_list <- sim_stBD_t(sbr = lambda,
                sdr = mu,
                numbsim = numb_replicates,
                t = time)
```

`str.multiCophy` *Retrieve the structure of a class multiCophy*

Description

Retrieves the structure of the class multiCophy

Usage

```
## S3 method for class 'multiCophy'
str(object, ...)
```

Arguments

`object`	An object of class multiCophy
`...`	Potential further arguments to generic str class

Value

NULL value

Examples

```
h_lambda <- 1.0
h_mu <- 0.3
c_lambda <- 0.0
s_lambda <- 1.0
s_mu <- 0.3
s_her <- 0.0
host_symb_sets <-  sim_cophylo_bdp(hbr = h_lambda,
                              hdr = h_mu,
                              sbr = s_lambda,
                              cosp_rate = c_lambda,
                              sdr = s_mu,
                              host_exp_rate = s_her,
                              time_to_sim = 1.0,
                              numbsim = 2)
str(host_symb_sets)
```

Index