Package ‘nLTT’

July 22, 2025

Type Package

Title Calculate the NLTT Statistic

Version 1.4.10

Description Provides functions to calculate the normalised Lineage-Through-Time (nLTT) statistic, given two phylogenetic trees. The nLTT statistic measures the difference between two Lineage-Through-Time curves, where each curve is normalised both in time and in number of lineages.

License GPL-2

Imports ape, coda, testit

Suggests TreeSim, DDD, ggplot2, Hmisc, knitr, lintr, microbenchmark, plyr, reshape2, rmarkdown, TESS, testthat, DAISIE

NeedsCompilation no

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

URL `https://github.com/thijsjanzen/nLTT`

BugReports `https://github.com/thijsjanzen/nLTT/issues`

Author Thijs Janzen [aut, cre],
Richèl J.C. Bilderbeek [aut] (ORCID: <`https://orcid.org/0000-0003-1107-7049`>),
Pedro Neves [ctb]

Maintainer Thijs Janzen <`thijsjanzen@gmail.com`>

Repository CRAN

Date/Publication 2025-05-06 13:00:02 UTC

Contents

`nLTT-package` *Package providing functions to visualize the normalized Lineage-Through-Time statistic, and calculate the difference between two nLTT curves*

Description

This package provides a function to visualize the normalized Lineage-Through-Time (nLTT) statistic, where the number of lineages relative to the maximum number of lineages in a phylogenetic tree is plotted against the relative time between the most common recent ancestor and the present. Furthermore the package provides a function to calculate the difference between two nLTT curves, including two different distance measurements.

Updates: Version 1.4.9: improve DESCRIPTION
Version 1.4.8: minor improvements to code
Version 1.4.7: Fixed noSuggest error, to comply _R_CHECK_DEPENDS_ONLY_
Version 1.4.6: Fixed testing, to comply _R_CHECK_LENGTH_1_CONDITION_.
Version 1.4.4: Added support for phylogenies with extinct lineages.
Version 1.4.3: Added support for log transformation before normalization.

Version 1.4: Added the following four functions: get_branching_times, get_n_lineages, get_norm_brts and get_norm_n. Furthermore, vignette building has improved, and the underlying code base has been polished up as well.
Version 1.3.1: Added walkthrough vignette, and updated several typos in the manual
Version 1.3: Version 1.3 adds a lot of extended functionality: firstly, we have added functions to calculate, and plot, the average nLTT across a number of phylogenies. Furthermore, we have added vignettes, and we have added a GitHub repository. On the GitHub repository the vignettes are separately accessible through the wiki. Lastly we have added an extra option to the nLTT functions, where the user can specify if the used trees are rooted, or not. Under the hood, some changes have been made as well, the majority of the code is now conforming to the lintR code conventions, and we have written formalized tests that check correctness of all code (code coverage 100

Version 1.2.1: updated comments and coding style to adhere to the general coding rules. Backwards compatibility has been favoured for the nLTT stat functions. ABC related functions are no longer backwards compatible (variable names have been changed to adhere to coding style).
Version 1.2: added an "exact" nLTT function. This function is faster for small trees, and provides an exact measurement of the nLTT function. Comparison between "old" and "exact" estimates show that these are highly correlated, although the "exact" values are slightly higher than the "old" values. The "exact" function should generally be preferred, unless dealing with extremely large trees (500+ tips) in which case the old function is much faster.

Version 1.2: updated the example for the ABC_SMC_nLTT function, prior generating and prior density functions are now more realistic

Version 1.1.1: fixed a minor bug in the ABC_SMC_nLTT function
Version 1.1.1: removed some intermediate output in ABC_SMC_nLTT function
Version 1.1: Made a universal nLTT function called "nLTTstat", with argument "distanceMethod", this serves as a more elegant wrapper for the functions "normLTTdiffABS" and "normLTTdiffSQ"
Version 1.1: Updated references in the manual

Author(s)

Thijs Janzen

Maintainer: Thijs Janzen <thijsjanzen@gmail.com>

References

Janzen,T. Hoehna,S., Etienne,R.S. (2015) Approximate Bayesian Computation of diversification rates from molecular phylogenies: introducing a new efficient summary statistic, the nLTT. Methods in Ecology and Evolution. <doi:10.1111/2041-210X.12350>

`abc_smc_nltt` *A function to perform Approximate Bayesian Computation within a Sequential Markov Chain (ABC-SMC), for diversification analysis of phylogenetic trees.*

Description

This function performs ABC-SMC as described in Toni 2009 for given diversification model, provided a phylogenetic tree. ABC-SMC is not limited to only using the normalized LTT as statistic.

Usage

```
abc_smc_nltt(
    tree, statistics, simulation_function, init_epsilon_values,
    prior_generating_function, prior_density_function,
    number_of_particles = 1000, sigma = 0.05, stop_rate = 1e-05
  )
```

Arguments

`tree`	an object of class `"phylo"`; the tree upon which we want to fit our diversification model
`statistics`	A vector containing functions that take a tree as an argument and return a single scalar value (the statistic).
`simulation_function`	A function that implements the diversification model and returns an object of class `"phylo"`.
`init_epsilon_values`	A vector containing the initial threshold values for the summary statistics from the vector `statistics`.
`prior_generating_function`	Function to generate parameters from the prior distribution of these parameters (e.g. a function returning lambda and mu in case of the birth-death model)
`prior_density_function`	Function to calculate the prior probability of a set of parameters.
`number_of_particles`	Number of particles to be used per iteration of the ABC-SMC algorithm.
`sigma`	Standard deviation of the perturbance distribution (perturbance distribution is a gaussian with mean 0).
`stop_rate`	If the acceptance rate drops below `stopRate`, stop the ABC-SMC algorithm and assume convergence.

Value

A matrix with `n` columns, where `n` is the number of parameters you are trying to estimate.

Author(s)

Thijs Janzen

References

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M.P.H. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. Journal of the Royal Society Interface, 6(31), 187-202.

Examples

```
  ## Not run:

  prior_gen <- function() {
    return( rexp(n=2, rate=0.1) )
  }

  prior_dens <- function(val) {
    return( dexp( val[1], rate = 0.1) * dexp( val[2], rate = 0.1) )
  }

  require(TESS)

  treeSim <- function(params) {
    t <- TESS.sim.age(n=1, lambda = params[1], mu = params[2], age = 10)[[1]]
    return(t)
  }

  obs <- treeSim(c(0.5,0.1))

  statWrapper <- function(tree1) {
    return( nLTTstat_exact(tree1, obs, "abs"))
  }

  stats <- c(statWrapper)

  results <-  abc.smc.nltt(
    obs, stats, treeSim, init_epsilon_values = 0.2,
    prior_generating_function = prior_gen,
    prior_density_function = prior_dens,
    number_of_particles = 1000, sigma = 0.05, stop_rate = 1e-5
  )


## End(Not run) # end of dontrun
```

---

check_input_event_times

Checks that event times are correct

Description

Checks `event_times` and `event_times2` are of the appropriate class and have expected characteristics for correct calculation of NLTT in `nltt_diff_exact_extinct`.

Usage

```
check_input_event_times(event_times, event_times2, time_unit)
```

Arguments

`event_times`	event times of the first phylogeny
`event_times2`	event times of the second phylogeny
`time_unit`	the time unit of the branching times • "ago: " the branching times are postive, as these are in time units ago • "since: " the branching times are negative, as these are in time units since present

Value

Nothing. Throws error with helpful error message if `event_times` and `event_times2` are not correct.

Author(s)

Pedro Neves and Richèl Bilderbeek and Thijs Janzen

`check_phylogenies` *Check if the input is a valid collection of one or more phylogenies*

Description

Will stop if not

Usage

```
check_phylogenies(phylogenies)
```

Arguments

`phylogenies`	a collection of one or more phylogenies, where the phylogenies are of type phylo. This collection can both be a list of phylo or a multiphylo.

`check_step_type` *Check if the step type is valid*

Description

Will stop if not

Usage

```
check_step_type(step_type)
```

Arguments

`step_type`	when between two points, where the second point has both a higher x and y coordinat, which y coordinat to follow. 'step_type' can be: • lower maintain the y-coordinat of the leftmost point • upper already use the y-coordinat of the rightmost point

---

`check_time_unit` *Check if the time unit is valid*

Description

Will stop if not

Usage

```
check_time_unit(time_unit)
```

Arguments

`time_unit`	the time unit of the branching times • "ago: " the branching times are postive, as these are in time units ago • "since: " the branching times are negative, as these are in time units since present

Author(s)

Richèl J.C. Bilderbeek

`default_params_doc` *This function does nothing. It is intended to inherit is parameters' documentation.*

Description

This function does nothing. It is intended to inherit is parameters' documentation.

Usage

```
default_params_doc(dt, phylogenies, step_type, time_unit)
```

Arguments

`dt`	The timestep resolution, a value bigger than zero and less or equal to one. 1/dt is the number of points that will be evaluated
`phylogenies`	a collection of one or more phylogenies, where the phylogenies are of type phylo. This collection can both be a list of phylo or a multiphylo.
`step_type`	when between two points, where the second point has both a higher x and y coordinat, which y coordinat to follow. 'step_type' can be: • lower maintain the y-coordinat of the leftmost point • upper already use the y-coordinat of the rightmost point
`time_unit`	the time unit of the branching times • "ago: " the branching times are postive, as these are in time units ago • "since: " the branching times are negative, as these are in time units since present

Note

This is an internal function, so it should be marked with `@noRd`. This is not done, as this will disallow all functions to find the documentation parameters

Author(s)

Richèl J.C. Bilderbeek

`exampleTrees` *example trees to test the functionality of the package*

Description

100 phylogenetic trees of class `phylo`, generated using the sim.globalBiDe.age function from the `TESS` package, with lambda = 0.3, mu = 0.1, age = 10.

Usage

```
data(exampleTrees)
```

Format

A list containing objects of class `phylo`.

Examples

```
data(exampleTrees);
obs <- exampleTrees[[1]];
nltt_plot(obs);
```

`get_average_nltt_matrix` *Get the average nLTT from a collection of phylogenies*

Description

Get the average nLTT from a collection of phylogenies

Usage

```
get_average_nltt_matrix(phylogenies, dt = 0.001)
```

Arguments

`phylogenies`	the phylogenies, supplied as either a list or a multiPhylo object, where the phylogenies are of type 'phylo'
`dt`	The timestep resolution, where 1/dt is the number of points evaluated

Value

A matrix of timepoints with the average number of (normalized) lineages through (normalized) time

Author(s)

Richèl J.C. Bilderbeek

Examples

```
get_average_nltt_matrix(c(ape::rcoal(10), ape::rcoal(20)))
```

`get_branching_times` *Collect the branching times from the stem age*

Description

Collect the branching times from the stem age

Usage

```
get_branching_times(phylogeny)
```

Arguments

`phylogeny`	a phylogeny of class 'phylo'

Value

branching times, in time units before the present

Author(s)

Richèl Bilderbeek

Examples

```
phylogeny <- ape::read.tree(text = "((a:2,b:2):1,c:3);")
phylogeny$root.edge <- 2 # nolint ape variable name
  all.equal(as.vector(nLTT::get_branching_times(phylogeny)), c(5, 3, 2))
```

`get_nltt_values` *Get the nLTT values in time*

Description

Collect the nLTT values in time over all phylogenies in the long form.

Usage

```
get_nltt_values(phylogenies, dt)
```

Arguments

`phylogenies`	the phylogenies, supplied as either a list or a multiPhylo object, where the phylogenies are of type 'phylo'
`dt`	The timestep resolution, where 1/dt is the number of points evaluated

Value

A dataframe of timepoints with the nLTT value of each phylogeny in time

Author(s)

Richèl Bilderbeek

See Also

Usenltts_diff to compare nLTT statistic between one focal tree and a set of one or more other trees

Examples

```
# Create some random phylogenies
phylogeny1 <- ape::rcoal(10)
phylogeny2 <- ape::rcoal(20)
phylogeny3 <- ape::rcoal(30)
phylogeny4 <- ape::rcoal(40)
phylogeny5 <- ape::rcoal(50)
phylogeny6 <- ape::rcoal(60)
phylogeny7 <- ape::rcoal(70)
phylogenies <- c(phylogeny1, phylogeny2, phylogeny3,
  phylogeny4, phylogeny5, phylogeny6, phylogeny7
)

# Obtain the nLTT values
dt <- 0.2
nltt_values <- get_nltt_values(phylogenies, dt = dt)
```

`get_norm_brts` *Collect the normalized branching times from the stem age*

Description

Collect the normalized branching times from the stem age

Usage

```
get_norm_brts(phylogeny)
```

Arguments

`phylogeny`	a phylogeny of class 'phylo'

Value

branching times, in time units before the present

Author(s)

Richèl Bilderbeek

Examples

```
phylogeny <- ape::read.tree(text = "((a:2,b:2):1,c:3);")
phylogeny$root.edge <- 2 # nolint ape variable name
  all(nLTT::get_branching_times(phylogeny) == c(5, 3, 2))
```

`get_norm_n` *Collect the normalized number of lineages from the stem age*

Description

Collect the normalized number of lineages from the stem age

Usage

```
get_norm_n(phylogeny)
```

Arguments

`phylogeny`	a phylogeny of class 'phylo'

Value

branching times, in time units before the present

Author(s)

Richèl Bilderbeek

Examples

```
phylogeny <- ape::read.tree(text = "((a:2,b:2):1,c:3);")
phylogeny$root.edge <- 2 # nolint ape variable name
  all.equal(as.vector(nLTT::get_branching_times(phylogeny)), c(5, 3, 2))
```

`get_n_lineages` *Collect the number of lineages from the stem age*

Description

Collect the number of lineages from the stem age

Usage

```
get_n_lineages(phylogeny)
```

Arguments

`phylogeny`	a phylogeny of class 'phylo'

Value

number of lineages, will go from 1 to the number of tips, if there is a stem, will go from 2 to the number of tips if there is no stem

Author(s)

Richèl Bilderbeek

Examples

```
phylogeny <- ape::read.tree(text = "((a:2,b:2):1,c:3);")
  all.equal(as.vector(nLTT::get_n_lineages(phylogeny)), c(2, 3))
phylogeny$root.edge <- 2 # nolint ape variable name
  all.equal(as.vector(nLTT::get_n_lineages(phylogeny)), c(1, 2, 3))
```

`get_phylogeny_nltt_matrix` *Extract the nLTT matrix from a phylogeny*

Description

Extract the nLTT matrix from a phylogeny

Usage

```
get_phylogeny_nltt_matrix(phylogeny)
```

Arguments

`phylogeny`	A phylogeny of type phylo

Value

a matrix

Author(s)

Richèl Bilderbeek

`mcmc_nltt` *function, using a Monte Carlo Markov Chain*

Description

function, using a Monte Carlo Markov Chain

Usage

```
mcmc_nltt(
  phy,
  likelihood_function,
  parameters,
  logtransforms,
  iterations,
  burnin = round(iterations/3),
  thinning = 1,
  sigma = 1
)
```

Arguments

`phy`	phylo Vector of weights
`likelihood_function`	function Function that calculates the likelihood of our diversification model, given the tree. function should be of the format function(parameters, phy).
`parameters`	vector Initial parameters to start the chain.
`logtransforms`	scalar Whether to perform jumps on logtransformed parameters (TRUE) or not (FALSE)
`iterations`	scalar Length of the chain
`burnin`	scalar Length of the burnin, default is 30% of iterations
`thinning`	scalar Size of thinning, default = 1
`sigma`	scalar Standard deviation of the jumping distribution, which is N(0, sigma).

Value

mcmc An MCMC object, as used by the package "coda".

`nLTTstat` *Calculate the difference between two normalized Lineage-Through-Time curves, given two phylogenetic trees.*

Description

This function takes two ultrametric phylogenetic trees, calculates the normalized Lineage-Through-Time statistic for both trees and then calculates the difference between the two statistics.

Usage

```
nLTTstat(tree1, tree2, distance_method = "abs", ignore_stem = TRUE,
               log_transform = FALSE)
```

Arguments

`tree1`	an object of class `"phylo"`
`tree2`	an object of class `"phylo"`
`distance_method`	Chosen measurement of distance between the two nLTT curves, options are (case sensitive): - "abs": use the absolute distance - "squ": use the squared distance;
`ignore_stem`	a boolean whether to ignore the stem length
`log_transform`	a boolean wether to log-transform the number of lineages before normalization

Value

The difference between the two nLTT statistics

Author(s)

Thijs Janzen

Examples

```
data(exampleTrees)
nltt_plot(exampleTrees[[1]])
nltt_lines(exampleTrees[[2]], lty=2)
nLTTstat(
  exampleTrees[[1]], exampleTrees[[2]],
  distance_method = "abs", ignore_stem = TRUE)
```

`nLTTstat_exact` *Calculate the exact difference between two normalized Lineage-Through-Time curves, given two phylogenetic trees.*

Description

This function takes two ultrametric phylogenetic trees, calculates the normalized Lineage-Through-Time statistic for both trees and then calculates the exact difference between the two statistics. Whereas the function `nLTTstat` uses an approximation to calculate the difference (which is faster for large trees), the function `nLTTstat_exact` calculates the exact difference, and should generally be preferred. Although the estimates are highly similar, `nLTTstat_exact` tends to return slightly higher values.

Usage

```
nLTTstat_exact(tree1, tree2, distance_method = "abs",
                  ignore_stem = TRUE, log_transform = FALSE)
```

Arguments

`tree1`	an object of class `"phylo"`
`tree2`	an object of class `"phylo"`
`distance_method`	Chosen measurement of distance between the two nLTT curves, options are (case sensitive): - "abs": use the absolute distance. - "squ": use the squared distance
`ignore_stem`	a boolean whether to ignore the stem length
`log_transform`	a boolean wether to log-transform the number of lineages before normalization

Value

The exact difference between the two nLTT statistics

Author(s)

Thijs Janzen

Examples

```
data(exampleTrees)
nltt_plot(exampleTrees[[1]])
nltt_lines(exampleTrees[[2]], lty = 2)
nLTTstat_exact(
  exampleTrees[[1]],
  exampleTrees[[2]],
  distance_method = "abs",
  ignore_stem = TRUE
)
```

`nltts_diff` *Calculates the nLTT statistic between each phylogeny in a collection compared to a same focal/reference tree*

Description

Calculates the nLTT statistic between each phylogeny in a collection compared to a same focal/reference tree

Usage

```
nltts_diff(
  tree,
  trees,
  distance_method = "abs",
  ignore_stem = TRUE,
  log_transform = FALSE
)
```

Arguments

`tree`	One phylogenetic tree
`trees`	A collection of one or more phylogenetic trees
`distance_method`	(string) absolute, or squared distance?
`ignore_stem`	(logical) Should the phylogeny its stem be ignored?
`log_transform`	(logical) Should the number of lineages be log-transformed before normalization?

Value

the nLTT statistic values, as a numeric vector of the same length as `trees`

Author(s)

Richèl J.C. Bilderbeek

See Also

use `nltt_diff` to compare two phylogenies

Examples

```
tree <- ape::rcoal(4)
trees <- c(ape::rcoal(4), ape::rcoal(4))
nltts <- nltts_diff(tree, trees)
```

`nltts_plot` *Get the average nLTT from a collection of phylogenies*

Description

Get the average nLTT from a collection of phylogenies

Usage

```
nltts_plot(
  phylogenies,
  dt = 0.001,
  plot_nltts = FALSE,
  xlab = "Normalized Time",
  ylab = "Normalized Lineages",
  replot = FALSE,
  ...
)
```

Arguments

`phylogenies`	a collection of one or more phylogenies, where the phylogenies are of type phylo. This collection can both be a list of phylo or a multiphylo.
`dt`	The timestep resolution, a value bigger than zero and less or equal to one. 1/dt is the number of points that will be evaluated
`plot_nltts`	Also plot each nLLT line
`xlab`	Label on the x axis
`ylab`	Label on the y axis
`replot`	If false, start a clean plot. If true, plot the new data over the current
`...`	Plotting options

Value

Nothing

Author(s)

Richèl J.C. Bilderbeek

Examples

```
nltts_plot(c(ape::rcoal(10), ape::rcoal(10)))
nltts_plot(c(ape::rcoal(10), ape::rcoal(20)), dt = 0.1)
```

`nltt_diff` *Calculates the exact difference between the lineage through time curves of tree1 & tree2 (normalized in time and for the number of lineages)*

Description

Calculates the exact difference between the lineage through time curves of tree1 & tree2 (normalized in time and for the number of lineages)

Usage

```
nltt_diff(
  tree1,
  tree2,
  distance_method = "abs",
  ignore_stem = TRUE,
  log_transform = FALSE
)
```

Arguments

`tree1`	(phylo) First phylogenetic tree
`tree2`	(phylo) Second phylogenetic tree
`distance_method`	(string) absolute, or squared distance?
`ignore_stem`	logical Should the phylogeny its stem be ignored?
`log_transform`	(logical) Should the number of lineages be log-transformed before normalization?

Value

(scalar) normalized Lineage-Through-Time difference between tree1 & tree2

Author(s)

Thijs Janzen

See Also

use `nltts_diff` to compare a collection of phylogenies to one focal/reference tree

---

`nltt_diff_exact` *Calculates the exact, difference between the lineage through time curves of tree1 & tree2 (normalized in time and for the number of lineages)*

Description

Calculates the exact, difference between the lineage through time curves of tree1 & tree2 (normalized in time and for the number of lineages)

Usage

```
nltt_diff_exact(
  tree1,
  tree2,
  distance_method = "abs",
  ignore_stem = TRUE,
  log_transform = FALSE
)
```

Arguments

`tree1`	(phylo) First phylogenetic tree
`tree2`	(phylo) Second phylogenetic tree
`distance_method`	(string) absolute, or squared distance?
`ignore_stem`	(logical) Should the phylogeny its stem be ignored?
`log_transform`	(logical) Should the number of lineages be log-transformed before normalization?

Value

(scalar) normalized Lineage-Through-Time difference between tree1 & tree2

Author(s)

Thijs Janzen

`nltt_diff_exact_brts` *Calculates the exact difference between the nLTT curves of the branching times*

Description

Calculates the exact difference between the nLTT curves of the branching times

Usage

```
nltt_diff_exact_brts(
  b_times,
  lineages,
  b_times2,
  lineages2,
  distance_method = "abs",
  time_unit = "since"
)
```

Arguments

`b_times`	branching times of the first phylogeny,
`lineages`	the number of lineages, usually one to the number of lineages
`b_times2`	branching times of the first phylogeny
`lineages2`	the number of lineages, usually one to the number of lineages
`distance_method`	how the difference between the two nLTTs is summed

	• "abs: " the absolute distance between the two nLTTs is summed • "squ: " the squared distance between the two nLTTs is summed
`time_unit`	the time unit of the branching times • "ago: " the branching times are postive, as these are in time units ago • "since: " the branching times are negative, as these are in time units since present

Author(s)

Thijs Janzen and Richèl Bilderbeek

`nltt_diff_exact_calc_extinct`

Calculates the exact difference between the nLTT curves of the event times. This includes extinction events.

Description

Calculates the exact difference between the nLTT curves of the event times. This includes extinction events.

Usage

```
nltt_diff_exact_calc_extinct(
  event_times,
  species_number,
  event_times2,
  species_number2,
  distance_method
)
```

Arguments

`event_times`	event times of the first phylogeny
`species_number`	the number of species at each event time of the first phylogeny
`event_times2`	event times of the second phylogeny
`species_number2`	the number of species at each event time of the second phylogeny
`distance_method`	(string) absolute, or squared distance?

Author(s)

Thijs Janzen and Richèl Bilderbeek and Pedro Neves

Examples

```
# Generate data
n <- 10
b_times_n <- (seq(1, n) / n)
lineages_n <- b_times_n
b_times2_n <- b_times_n * b_times_n
lineages2_n <- b_times2_n

# Calculate nLTT
out <- nLTT::nltt_diff_exact_calc_extinct(
  event_times = b_times_n,
  species_number  = lineages_n,
  event_times2 = b_times2_n,
  species_number2 = lineages2_n,
  distance_method = "abs"
)
#'
```

`nltt_diff_exact_extinct` *Calculates the exact difference between the nLTT curves of the event times. This includes extinction events.*

Description

Takes branching times such as (for example) as returned by the DDD package.

Usage

```
nltt_diff_exact_extinct(
  event_times,
  species_number,
  event_times2,
  species_number2,
  distance_method = "abs",
  time_unit = "since",
  normalize = TRUE
)
```

Arguments

`event_times`	event times of the first phylogeny
`species_number`	the number of species at each event time of the first phylogeny
`event_times2`	event times of the second phylogeny
`species_number2`	the number of species at each event time of the second phylogeny

`distance_method`	how the difference between the two nLTTs is summed • "abs: " the absolute distance between the two nLTTs is summed • "squ: " the squared distance between the two nLTTs is summed
`time_unit`	the time unit of the branching times • "ago: " the branching times are postive, as these are in time units ago • "since: " the branching times are negative, as these are in time units since present
`normalize`	should the output be normalized? Default is TRUE.

Author(s)

Pedro Neves and Richèl Bilderbeek and Thijs Janzen

Examples

```
# Generate data
n <- 10
b_times_n <- (seq(1, n) / n)
lineages_n <- b_times_n
b_times2_n <- b_times_n * b_times_n
lineages2_n <- b_times2_n

# Calculate nLTT
out <- nLTT::nltt_diff_exact_extinct(
  event_times = b_times_n,
  species_number  = lineages_n,
  event_times2 = b_times2_n,
  species_number2 = lineages2_n,
  time_unit = "ago",
  distance_method = "abs"
)
```

`nltt_diff_exact_norm_brts`

Calculates the exact difference between the nLTT curves of the branching times

---

Description

Calculates the exact difference between the nLTT curves of the branching times

Usage

```
nltt_diff_exact_norm_brts(
  b_times_n,
  lineages_n,
  b_times2_n,
  lineages2_n,
  distance_method
)
```

Arguments

`b_times_n`	branching times of the first phylogeny
`lineages_n`	the number of lineages, usually one to the number of lineages
`b_times2_n`	branching times of the first phylogeny
`lineages2_n`	the number of lineages, usually one to the number of lineages
`distance_method`	(string) absolute, or squared distance?

Author(s)

Thijs Janzen and Richèl Bilderbeek

`nltt_lines` *Normalized version of the ape function ltt.lines.*

Description

This is a modified version of the ape function ltt.lines: add the normalized Lineage-Through-Time statistic of a phylogenetic tree to an already existing plot

Usage

```
nltt_lines(phy, ...)
```

Arguments

`phy`	an object of class `"phylo"`
`...`	further graphical arguments that can be passed to `lines()`

Author(s)

Thijs Janzen

Examples

```
data(exampleTrees)
nltt_plot(exampleTrees[[1]])
nltt_lines(exampleTrees[[2]], lty=2)
```

`nltt_plot` *Normalized version of the ape function ltt.plot*

Description

This function uses a modified version of the ltt.plot function from `"ape"` to plot the normalized number of lineages through normalized time, where the number of lineages is normalized by dividing by the number of tips of the tree, and the time is normalized by the total time between the most common recent ancestor and the present, such that t(MRCA) = 0 & t(present) = 1.

Usage

```
nltt_plot(
  phy, xlab = "Normalized Time", ylab = "Normalized Lineages", ...)
```

Arguments

`phy`	an object of class `"phylo"`
`xlab`	a character string (or a variable of mode character) giving the label for the x-axis (default is "Normalized Time").
`ylab`	a character string (or a variable of mode character) giving the label for the y-axis (default is "Normalized Lineages").
`...`	further graphical arguments that can be passed to `plot()`

Author(s)

Thijs Janzen

Examples

```
data(exampleTrees)
nltt_plot(exampleTrees[[1]])
```

`stretch_nltt_matrix` *Stretch matrix 'm' with a timestep resolution of 'dt'.*

Description

Stretch matrix 'm' with a timestep resolution of 'dt'.

Usage

```
stretch_nltt_matrix(m, dt, step_type)
```

Arguments

`m`	A matrix of 2 columns and at least 2 rows
`dt`	The resulution, a value e [0.0001, 1]. If ’dt’ is set to a very small value, this function will stop
`step_type`	when between two points, where the second point has both a higher x and y coordinat, which y coordinat to follow. ’step_type’ can be: • lower maintain the y-coordinat of the leftmost point • upper already use the y-coordinat of the rightmost point

Value

The stretched matrix

Author(s)

Richèl J.C. Bilderbeek

Examples

```
m <- matrix( c(c(0.0, 1.0), c(0.5, 1.0)), ncol = 2, nrow = 2)
expected <- matrix(
  c(
    c(0.0, 0.5, 1.0),  # Timepoints
    c(0.5, 0.5, 1.0)   # Values
  ),
  ncol = 2, nrow = 3
)
result <- stretch_nltt_matrix(m = m, dt = 0.5, step_type = ”lower”)
all.equal(result, expected)
```

Index