

Package ‘nLTT’

December 7, 2015

Type Package

Title Calculate the NLTT Statistic

Version 1.1.1

Date 2015-12-04

Author Thijs Janzen

Maintainer Thijs Janzen <thijsjanzen@gmail.com>

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 deSolve, coda, ape

Suggests TESS

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-07 22:21:38

R topics documented:

nLTT-package	2
ABC_SMC_nLTT	3
exampleTrees	4
MCMC_nLTT	5
nLTT.lines	6
nLTT.plot	7
nLTTstat	8

Index	9
--------------	----------

nLTT-package	<i>Package providing functions to visualize the normalized Lineage-Through-Time statistic, and calculate the difference between two nLTT curves</i>
--------------	---

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 1.1.1: 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

Details

Package:	nLTT
Type:	Package
Version:	1.1
Date:	2015-04-21
License:	GPL 2.0

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	<i>A function to perform Approximate Bayesian Computation within an Sequential Markov Chain (ABC-SMC), for diversification analysis of phylogenetic trees.</i>
--------------	--

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, simFunc, initEpsilon,
             PRIOR_GEN, PRIOR_DENS, numParticles, sigma = 0.05, stopRate = 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).
simFunc	A function that implements the diversification model and returns an object of class "phylo".
initEpsilon	A vector containing the initial threshold values for the summary statistics from the vector statistics.
PRIOR_GEN	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_DENS	Function to calculate the prior probability of a set of parameters.
numParticles	Number of particles to be used per iteration of the ABC-SMC algorithm.
sigma	Standard deviation of the perturbation distribution (perturbation distribution is a gaussian with mean 0).
stopRate	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

```
##### Code to run the ABC-SMC algorithm would involve a lot
##### of dependencies on other packages, and be computationally demanding
##### here we sketch out an example, but do not execute it:

## Not run:

prior_gen <- function() {
  return( rlnorm(n=2,mean=0,sd=1) );
}

prior_dens <- function(val) {
  return( dlnorm(val[1],mean=0,sd=1) * dlnorm(val[1],mean=0,sd=1));
}

require(TESS);
treeSim <- function(params) {
  t <- sim.globalBiDe.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(nLTT(tree1,obs,"abs"));
}

stats <- c(statWrapper);

results <- ABC_SMC_nLTT(obs,stats,treeSim,initEpsilon = 0.2,
  PRIOR_GEN = prior_gen,PRIOR_DENS <- prior_dens,
  numParticles = 1000, sigma = 0.05, stopRate = 1e-5);

## End(Not run)
```

Description

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

Usage

```
data(exampleTrees)
```

Format

A list containing objects of class phylo.

Examples

```
data(exampleTrees);
obs <- exampleTrees[[1]];
nLTT.plot(obs);
```

MCMC_nLTT

Code to perform Metropolis-Hastings MCMC for a diversification model, given a phylogenetic tree. This function was used in the MEE paper to calculate the likelihood reference estimates.

Description

This function performs Metropolis-Hastings MCMC, where the user provides a likelihood function and a phylogenetic tree.

Usage

```
MCMC_nLTT(phy,likelihoodFunction, parameters, logTransforms,
iterations, burnin = round(iterations/3), thinning = 1,sigma=1)
```

Arguments

phy	an object of class "phylo"; the tree upon which we want to fit our diversification model
likelihoodFunction	Function that calculates the likelihood of our diversification model, given the tree. Function should me of the format function(parameters,phy).
parameters	Initial parameters to start the chain.
logTransforms	Whether to perform jumps on logtransformed parameters (TRUE) or not (FALSE)
iterations	Length of the chain
burnin	Length of the burnin, default is 30
thinning	Size of thinning, default = 1
sigma	Standard deviation of the jumping distribution, which is $N(0,\text{sigma})$.

Value

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

Author(s)

Sebastian Hoehna & Thijs Janzen

Examples

```
## MCMC examples are typically very slow ####
## Not run:

require(TESS);

obs <- sim.globalBiDe.age(n=1,lambda=0.5,mu=0.1,age=10)[[1]];

LL_BD <- function(params,phy) {
  lnL <- globalBiDe.likelihood(phy,lambda=params[1],mu=params[2],samplingProbability=1,log=TRUE);
  prior1 <- dunif(params[1],0,100,log=TRUE)
  prior2 <- dunif(params[2],0,100,log=TRUE);
  return(lnL + prior1+prior2);
}

require(coda);

mcmc_out <- MCMC_nLTT(obs,LL_BD,c(0.5,0.1),c(TRUE,TRUE),
  iterations=1000,burnin=100,thinning=10,sigma=1)
plot(mcmc_out);

## End(Not run)
```

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 lt.plot*

Description

This function uses a modified version of the `lt.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 $t(\text{MRCA}) = 0$ & $t(\text{present}) = 1$.

Usage

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

Arguments

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

Author(s)

Thijs Janzen

Examples

```
data(exampleTrees);
nLTT.plot(exampleTrees[[1]]);
```

nLTTstat	<i>Calculate the difference between two normalized Lineage-Through-Time curves, given two phylogenetic trees.</i>
----------	---

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, distanceMethod = "abs")
```

Arguments

tree1	an object of class "phylo";
tree2	an object of class "phylo";
distanceMethod	Chosen measurement of distance between the two nLTT curves, options are (case sensitive): <ul style="list-style-type: none">- "abs": use the absolute distance- "squ": use the squared distance;

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]], distanceMethod = "abs");
```


Index

*Topic **Likelihood**

MCMC_nLTT, 5

*Topic **MCMC**

MCMC_nLTT, 5

*Topic **\textasciitildekwd1**

nLTT.plot, 7

*Topic **\textasciitildekwd2**

nLTT.plot, 7

*Topic **datasets**

exampleTrees, 4

ABC_SMC_nLTT, 3

exampleTrees, 4

MCMC_nLTT, 5

nLTT (nLTT-package), 2

nLTT-package, 2

nLTT.lines, 6

nLTT.plot, 7

nLTTstat, 8