Package ‘nLTT’

February 20, 2015

Type Package
Title Calculate The NLTT Statistic
Version 1.0
Date 2014-09-17
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 absolute 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, deSolve
Suggests TESS
NeedsCompilation no
Repository CRAN
Date/Publication 2015-01-19 19:02:14

R topics documented:

nLTT-package ................................................................. 2
ABC_SMC_nLTT ............................................................... 2
eampleTrees ................................................................. 4
MCMC_nLTT ................................................................. 5
nLTT.lines ................................................................. 6
nLTT.plot ................................................................. 7
normLTTdiffABS ............................................................ 8
normLTTdiffSQ ............................................................. 8

Index 10
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 two functions to calculate either the absolute or the squared difference between two nLTT curves.

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>nLTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Date</td>
<td>2014-09-17</td>
</tr>
<tr>
<td>License</td>
<td>GPL 2.0</td>
</tr>
</tbody>
</table>

**Author(s)**

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

**References**

T. Janzen, S. Hoehna, R.S. Etienne, 2015, Approximate Bayesian Computation of diversification rates from molecular phylogenies: introducing a new efficient summary statistic, the nLTT, Methods in Ecology and Evolution, in prep.

---

**ABC_SMC_nLTT**  
*A function to perform Approximate Bayesian Computation within an 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, 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 perturbance distribution (perturbance 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


Examples

# Not run:
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

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

---

**MCMC_nLTT**

*Code to perform Metropolis-Hastings MCMC for a diversification model, given a phylogenetic tree.*

---

**Description**

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

**Usage**

```r
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,sigma).

**Value**

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

**Author(s)**

Sebastian Hoehna & Thijs Janzen
Examples

```r
## 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

```r
nLTT.lines(phy, ...)
```

Arguments

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

Author(s)

Thijs Janzen
**Examples**

```r
data(exampleTrees);
nLTT.plot(exampleTrees[[1]]);
nLTT.lines(exampleTrees[[2]], lty=2);
```

---

**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**

```r
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**

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

$Calculate$ the absolute 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 absolute difference between the two statistics.

**Usage**

```r
normLTTdiffABS(tree1, tree2)
```

**Arguments**

- `tree1` an object of class "phylo";
- `tree2` an object of class "phylo";

**Value**

The absolute difference between the two nLTT statistics

**Author(s)**

Thijs Janzen

**Examples**

```r
data(exampleTrees);
nLTT.plot(exampleTrees[[1]]);
nLTT.lines(exampleTrees[[2]], lty=2);
normLTTdiffABS(exampleTrees[[1]], exampleTrees[[2]]);
```

---

normLTTdiffSQ

$Calculate$ the squared 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 squared difference between the two statistics.

**Usage**

```r
normLTTdiffSQ(tree1, tree2)
```
normLTTdiffSQ

Arguments

tree1 an object of class "phylo";
tree2 an object of class "phylo";

Value

The squared difference between the two nLTT statistics

Author(s)

Thijs Janzen

Examples

data(exampleTrees);
nLTT.plot(exampleTrees[[1]]);
nLTT.lines(exampleTrees[[2]],lty=2);
normLTTdiffSQ(exampleTrees[[1]],exampleTrees[[2]]);
Index

*Topic **Likelihood**
  MCMC_nLTT, 5
*Topic **MCMC**
  MCMC_nLTT, 5
*Topic `\textasciitilde$\text{kw}d1`
  nLTT.plot, 7
*Topic `\textasciitilde$\text{kw}d2`
  nLTT.plot, 7
*Topic **datasets**
  exampleTrees, 4

ABC_SMC_nLTT, 2

eampleTrees, 4

MCMC_nLTT, 5

nLTT (nLTT-package), 2
nLTT-package, 2
nLTT.lines, 6
nLTT.plot, 7
normLTTdiffABS, 8
normLTTdiffSQ, 8