

Package ‘PRDATR’

November 6, 2019

Title Computing probabilistic distances between phylogenetic models of continuous trait evolution

Version 0.0.0.9000

Description

PRDATR includes a suite of function for computing distances between phylogenetic models.

License What license it uses

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

Function_ComputeDistances	1
Function_ComputeDistances_1Network_1Tree	3
Function_ComputeDistances_2Networks	4
Function_ComputePairwiseDistances_SimulateTips	5
Function_ComputePairWiseModelDistances_BM	6
Function_ComputePairWiseModelDistances_OU	7
Function_ConductSimExperimentOU_ModelTestBM_OU	8
Function_FitCont_TreeSet_OU	9
Function_FitCont_TreeSet_BM	10
Index	12

Function_ComputeDistances

*Function_ComputeDistances: function to compute the Hellinger and
Kl distance for two probabilisty phylogenetic models under continuous
trait evolution*

Description

This function returns a vector containing the Hellinger and KL distances between two tree models

Usage

```
Function_ComputeDistances(list.Model_01, list.Model_02)
```

Arguments

- `list.Model_01` List containing the following (1) `handle.Phylogeny`, (2) `string.Model = "BM", "OU", "EB", "lambda", "kappa", "delta"`, (3) `vector.Z = vector of mean (ancestral) state values`, and (4) `vector.Model_01_Theta = vector containing relevant parameters for the models`
- `list.Model_02` List containing the following (1) `handle.Phylogeny`, (2) `string.Model = "BM", "OU", "EB", "lambda", "kappa", "delta"`, (3) `vector.Z = vector of mean (ancestral) state values`, and (4) `vector.Model_01_Theta = vector containing relevant parameters for the models`

Value

`vector.Distances` Vector containing the distances computed between the two focal tree models

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)

#####
# Specifity example tree (Fig. 8, Felsenstein 1985) used for demonstrating model distances #
#####
string.Figure01_Felsenstein1985_Tree <- "(((Species_7:1.635983031,Species_8:0.8079384444):1.801052391,(Spec
handle.Figure01_Felsenstein1985_Tree <- read.tree(text = string.Figure01_Felsenstein1985_Tree)

#####
# Set a vector containing parameters for Model 01 #
#####
vector.Model_02_Theta <- c(1, 1)
names(vector.Model_02_Theta) <- c("Sig2", "alpha")
vector.Model_01_Theta <- c(1)
names(vector.Model_01_Theta) <- c("Sig2")

list.Model01_BM <- list(handle.Phylogeny = handle.Figure01_Felsenstein1985_Tree,
                        string.Model = "BM",
                        vector.Z = rep(0, length(handle.Figure01_Felsenstein1985_Tree$tip.label)),
                        vector.Theta = vector.Model_01_Theta)

list.Model02_BM <- list(handle.Phylogeny = handle.Figure01_Felsenstein1985_Tree,
                        string.Model = "OU",
                        vector.Z = rep(0, length(handle.Figure01_Felsenstein1985_Tree$tip.label)),
                        vector.Theta = vector.Model_02_Theta)

Function_ComputeDistances(list.Model_01 = list.Model01_BM, list.Model_02 = list.Model02_BM)
```

Function_ComputeDistances_1Network_1Tree

Function_ComputeDistances_1Network_1Tree: function to compute probabilistic distances between a network and a bifurcating tree

Description

This function returns a vector containing the Hellinger and KL distances between two tree models

Usage

```
Function_ComputeDistances_1Network_1Tree(list.Model_01_Network,
list.Model_02_Tree)
```

Arguments

list.Model_01_Network

List containing the following (1) handle.Phylogeny, (2) string.Model = "BM", "OU", "EB", "lambda", "kappa", "delta", (3) vector.Z = vector of mean (ancestral) state values, and (4) vector.Model_01_Theta = vector containing relevant parameters for the models

list.Model_02_Tree

List containing the following (1) handle.Phylogeny, (2) string.Model = "BM", "OU", "EB", "lambda", "kappa", "delta", (3) vector.Z = vector of mean (ancestral) state values, and (4) vector.Model_01_Theta = vector containing relevant parameters for the models

Value

vector.Distances Vector containing the distances computed between the two focal tree models

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)
library(BMhyd)

#####
# Simulate a network #
#####
handle.SimulatedNetwork <- SimulateNetwork(ntax.nonhybrid=5, ntax.hybrid=3,
flow.proportion=0.5, origin.type='clade', birth = 1, death = 0.5, sample.f = 0.5,
tree.height = 1, allow.ghost=FALSE)

#####
# Set the first network model #
```

```
#####
list.Model01_BM <- list(handle.Phylogeny = handle.SimulatedNetwork$phy,
                        handle.Flow = handle.SimulatedNetwork$flow,
                        string.Model = "BM",
                        vector.Z = rep(0, 8),
                        numeric.Sig2 = 1)

#####
# Set the second tree model #
#####
list.Model02_BM <- list(handle.Phylogeny = handle.SimulatedNetwork$phy,
                        string.Model = "BM",
                        vector.Z = rep(0, 8),
                        numeric.Sig2 = 1)

#####
# Compute distances #
#####
Function_ComputeDistances_1Network_1Tree(list.Model_01_Network = list.Model01_BM, list.Model_02_Tree = list.
```

Function_ComputeDistances_2Networks

Function_ComputeDistances_2Networks: function to compute probabilistic distances between two network models

Description

This function returns a vector containing the Hellinger and KL distances between two tree models

Usage

```
Function_ComputeDistances_2Networks(list.Model_01_Network,
                                     list.Model_02_Network)
```

Arguments

```
list.Model_01_Network
  List containing network model with (1) phylogenetic tree (2) table of migratio
  flow (3) vector.Z and (4) Sig2 parameter

list.Model_02_Network
  List containing network model with (1) phylogenetic tree (2) table of migratio
  flow (3) vector.Z and (4) Sig2 parameter
```

Value

vector.Distances Vector containing the distances computed between the two focal tree models

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)
library(BMhyd)

#####
# Specific list for network model 01 #
#####
list.Model_01_Network <- list(handle.Phylogeny = handle.SimulatedNetwork$phy,
                              handle.Flow = handle.SimulatedNetwork$phy,
                              vector.Z = rep(0, length(network$phy$tip.label)),
                              numeric.Sig2 = 1)

list.Model_02_Network <- list(handle.Phylogeny = handle.SimulatedNetwork$phy,
                              handle.Flow = handle.SimulatedNetwork$phy,
                              vector.Z = rep(0, length(network$phy$tip.label)),
                              numeric.Sig2 = 2)

#####
# Compute distances #
#####
Function_ComputeDistances_2Networks(list.Model_01_Network = list.Model_01_Network,
                                     list.Model_02_Network = list.Model_02_Network)
```

Function_ComputePairwiseDistances_SimulateTips

Function_ComputePairwiseDistances_SimulateTips: function to generate pairwise distance computations

Description

This function returns a list with (1) matrix.PairwiseDistances_H, (2) matrix.PairwiseDistances_KL, and (3) vector.PairwiseDistances_H

Usage

```
Function_ComputePairwiseDistances_SimulateTips(numeric.NumberReps,
        numeric.n, numeric.birth, string.Model, vector.Theta)
```

Arguments

numeric.NumberReps	Number of simulation replicates
numeric.n	Number of tips
numeric.birth	Birth Rate
string.Model	String for evolutionary model

Value

List list with (1) matrix.PairwiseDistances_H, (2) matrix.PairwiseDistances_KL, and (3) vector.PairwiseDistances_H

Examples

```
#####
# Simulate datasets with n = 3 #
#####
vector.Theta <- c(1, 1)
names(vector.Theta) <- c("Sig2", "alpha")
list.RESULTS_n3_b1_a1 <- Function_ComputePairwiseDistances_SimulateTips(numeric.NumberReps = 100,
                                                                    numeric.n = 3, numeric.birth = 1,
                                                                    string.Model = "OU",
                                                                    vector.Theta = vector.Theta)$vector.PairwiseDistances_H
```

Function_ComputePairWiseModelDistances_BM

Function_ComputePairWiseModelDistances_BM: function to compute pairwise model distances between a set of trees given their model parameter estimates

Description

This function returns a list of pairwise model distances

Usage

```
Function_ComputePairWiseModelDistances_BM(phylogeny.TreeSet,
      matrix.ModelParams)
```

Arguments

```
phylogeny.TreeSet
      Set of phylogenetic trees
matrix.ModelParams
      Matrix of model parameters (BM) for each tree in the given set
```

Value

List List of pairwise model distances

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)
```

```
#####
# Simulate random trees #
#####
handle.SimulatedRandomTrees <- rmtree(N = 10, n = 10)
vector.SimulatedDataset <- fastBM(tree = handle.SimulatedRandomTrees[[1]], a = 0, mu = 1)

#####
# Fit models to trees #
#####
handle.FittedPhylogeneticModels <- Function_FitCont_TreeSet_BM(phylogeny.TreeSet = handle.SimulatedRandomTrees,
vector.SimulatedDataset)

#####
# Compute pairwise distances #
#####
handle.Results <- Function_ComputePairWiseModelDistances_BM(phylogeny.TreeSet = handle.SimulatedRandomTrees,
```

Function_ComputePairWiseModelDistances_OU

Function_ComputePairWiseModelDistances_OU: function to pairwise distances between a set of trees and fitted OU models

Description

This function returns a list of pairwise distances under the OU models

Usage

```
Function_ComputePairWiseModelDistances_OU(phylogeny.TreeSet,
matrix.ModelParams)
```

Arguments

```
phylogeny.TreeSet
    Set of phylogenetic trees
matrix.ModelParams
    Matrix of model parameters (OU) for each tree in the given set
```

Value

List List of pairwise model distances

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)
```

```
#####
# Simulate random trees #
#####
handle.SimulatedRandomTree <- rmtree(N = 1, n = 10)
handle.SimulatedRandomTrees <- list()
class(handle.SimulatedRandomTrees) <- "multiPhylo"
handle.SimulatedRandomTrees[[1]] <- handle.SimulatedRandomTrees[[2]] <- handle.SimulatedRandomTrees[[3]] <- handle.SimulatedRandomTrees[[4]] <- handle.SimulatedRandomTrees[[5]] <- handle.SimulatedRandomTrees[[6]] <- handle.SimulatedRandomTrees[[7]] <- handle.SimulatedRandomTrees[[8]] <- handle.SimulatedRandomTrees[[9]] <- handle.SimulatedRandomTrees[[10]]
vector.SimulatedDataset <- fastBM(tree = handle.SimulatedRandomTrees[[1]], a = 0, sig2 = 1, alpha = 2)

#####
# Fit models to trees #
#####
handle.FittedPhylogeneticModels <- Function_FitCont_TreeSet_OU(phylogeny.TreeSet = handle.SimulatedRandomTrees)

#####
# Compute pairwise distances #
#####
handle.Results <- Function_ComputePairWiseModelDistances_OU(phylogeny.TreeSet = handle.SimulatedRandomTrees,
```

Function_ConductSimExperimentOU_ModelTestBM_OU

Function_ConductSimExperimentOU_ModelTestBM_OU: function to compute p-values for likelihood ratio test and compute distances

Description

This function returns a matrix.SimulationResults for the simulation results

Usage

```
Function_ConductSimExperimentOU_ModelTestBM_OU(list.Model_Simulation_OU,
numeric.NumberOfReps)
```

Arguments

```
list.Model_Simulation_OU
List containing components of simulation model
```

Value

vector.Distances Vector containing the distances computed between the two focal tree models

Examples

```
#####
# Load depends #
#####
#library(ape)
```



```

#library(phytools)
#library(geiger)
#library(vioplplot)
#library(gaussDiff)
#library(MASS)

#####
# Build simulation model #
#####
phylogeny.RandomTree <- pbtree(b = 1, d = 0, n = 50)
vector.Model_Simulation_Theta <- c(1, 10)
names(vector.Model_Simulation_Theta) <- c("Sig2", "alpha")

list.Model_Simulation_OU <- list(handle.Phylogeny = phylogeny.RandomTree,
                                string.Model = "OU",
                                vector.Z = rep(0, length(phylogeny.RandomTree$tip.label)),
                                vector.Theta = vector.Model_Simulation_Theta)

handle.Sim_Results <- Function_ConductSimExperimentOU_ModelTestBM_OU(list.Model_Simulation_OU = list.Model_S
#####
# Compare this to #
#####
#####
# Build simulation model #
#####
phylogeny.RandomTree <- pbtree(b = 1, d = 0, n = 100)
vector.Model_Simulation_Theta <- c(1, 0.0001)
names(vector.Model_Simulation_Theta) <- c("Sig2", "alpha")

list.Model_Simulation_OU <- list(handle.Phylogeny = phylogeny.RandomTree,
                                string.Model = "OU",
                                vector.Z = rep(0, length(phylogeny.RandomTree$tip.label)),
                                vector.Theta = vector.Model_Simulation_Theta)

handle.Sim_Results <- Function_ConductSimExperimentOU_ModelTestBM_OU(list.Model_Simulation_OU = list.Model_S

```

Function_FitCont_TreeSet_OU

Function_FitCont_TreeSet_OU: function to estimate OU model parameters for each tree in a given set

Description

This function returns a matrix of fitted model parameters (assuming OU) for a given set of trees

Usage

```
Function_FitCont_TreeSet_OU(phylogeny.TreeSet, vector.InputData)
```

Arguments

phylogeny.TreeSet
List of phylogenetic trees

vector.InputData
Vector of continuous trait data

Value

vector.Distances Vector containing the distances computed between the two focal tree models

Examples

```
#####
# Load depends #
#####
library(ape)
library(geiger)
library(gaussDiff)

#####
# Simulate random trees #
#####
handle.SimulatedRandomTrees <- rmtree(N = 10, n = 10)
vector.SimulatedDataset <- fastBM(tree = handle.SimulatedRandomTrees[[1]], a = 0, mu = 1)

handle.FittedPhylogeneticModels <- Function_FitCont_TreeSet_0U(phylogeny.TreeSet = handle.SimulatedRandomTrees, vector.SimulatedDataset)
```

Function_FitCont_TreeSet_BM

Function_FitCont_TreeSet_BM: function to estimate BM model parameters for each tree in a given set

Description

This function returns a matrix of fitted model parameters (assuming a BM) for a given set of trees

Usage

```
Function_FitCont_TreeSet_BM(phylogeny.TreeSet, vector.InputData)
```

Arguments

phylogeny.TreeSet
List of phylogenetic trees

vector.InputData
Vector of continuous trait data

Value

vector.Distances Vector containing the distances computed between the two focal tree models

Examples

```
#####  
# Load depends #  
#####  
library(ape)  
library(geiger)  
library(gaussDiff)  
  
#####  
# Simulate random trees #  
#####  
handle.SimulatedRandomTrees <- rmtree(N = 10, n = 10)  
vector.SimulatedDataset <- fastBM(tree = handle.SimulatedRandomTrees[[1]], a = 0, mu = 1)  
  
handle.FittedPhylogeneticModels <- Function_FitCont_TreeSet_BM(phylogeny.TreeSet = handle.SimulatedRandomTrees)
```

Index

*Topic **brownian**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)
 Function_ComputePairwiseDistances_SimulateTips, [5](#)
 Function_ComputePairWiseModelDistances_BM, [6](#)
 Function_ComputePairWiseModelDistances_OU, [7](#)
 Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
 Function_FitCont_TreeSet_0U, [9](#)
 Function_FitCont_TreeSet_BM, [10](#)

*Topic **continuous**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)
 Function_ComputePairwiseDistances_SimulateTips, [5](#)
 Function_ComputePairWiseModelDistances_BM, [6](#)
 Function_ComputePairWiseModelDistances_OU, [7](#)
 Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
 Function_FitCont_TreeSet_0U, [9](#)
 Function_FitCont_TreeSet_BM, [10](#)

*Topic **distance,**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)
 Function_ComputePairwiseDistances_SimulateTips, [5](#)
 Function_ComputePairWiseModelDistances_BM, [6](#)
 Function_ComputePairWiseModelDistances_OU, [7](#)

[7](#)
 Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
 Function_FitCont_TreeSet_0U, [9](#)
 Function_FitCont_TreeSet_BM, [10](#)

*Topic **distances,**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)
 Function_ComputePairwiseDistances_SimulateTips, [5](#)
 Function_ComputePairWiseModelDistances_BM, [6](#)
 Function_ComputePairWiseModelDistances_OU, [7](#)
 Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
 Function_FitCont_TreeSet_0U, [9](#)
 Function_FitCont_TreeSet_BM, [10](#)

*Topic **evolution**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)
 Function_ComputePairwiseDistances_SimulateTips, [5](#)
 Function_ComputePairWiseModelDistances_BM, [6](#)
 Function_ComputePairWiseModelDistances_OU, [7](#)
 Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
 Function_FitCont_TreeSet_0U, [9](#)
 Function_FitCont_TreeSet_BM, [10](#)

*Topic **model**

Function_ComputeDistances, [1](#)
 Function_ComputeDistances_1Network_1Tree, [3](#)
 Function_ComputeDistances_2Networks, [4](#)

- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_BM, [6](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)
- *Topic **motion**,
- Function_ComputeDistances, [1](#)
- Function_ComputeDistances_1Network_1Tree, [3](#)
- Function_ComputeDistances_2Networks, [4](#)
- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_BM, [6](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)
- *Topic **phylogenetic**
- Function_ComputeDistances, [1](#)
- Function_ComputeDistances_1Network_1Tree, [3](#)
- Function_ComputeDistances_2Networks, [4](#)
- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_BM, [6](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)
- *Topic **probabilistic**
- Function_ComputeDistances, [1](#)
- Function_ComputeDistances_1Network_1Tree, [3](#)
- Function_ComputeDistances_2Networks, [4](#)
- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_BM, [6](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)
- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)
- Function_ComputeDistances, [1](#)
- Function_ComputeDistances_1Network_1Tree, [3](#)
- Function_ComputeDistances_2Networks, [4](#)
- Function_ComputePairwiseDistances_SimulateTips, [5](#)
- Function_ComputePairWiseModelDistances_BM, [6](#)
- Function_ComputePairWiseModelDistances_OU, [7](#)
- Function_ConductSimExperimentOU_ModelTestBM_OU, [8](#)
- Function_FitCont_TreeSet_0U, [9](#)
- Function_FitCont_TreeSet_BM, [10](#)