Package 'SpeciesTopoTestR'

September 30, 2020

Title Likelihood-based tests of species topologies in R

Version 0.0.0.9000

Description What the package does (one paragraph).

License What license it uses

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

Compute.GeneTree_Likelihoods

Compute.GeneTree_Likelihoods: function to compute gene tree likelihoods using STELL

Description

This function returns a vector of gene tree likelihoods given a set of gene trees and a particular species tree

Usage

```
Compute.GeneTree_Likelihoods(
  handle.SpeciesTree,
  handle.GeneTrees,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree`
Phylo object of the species tree

`handle.GeneTrees`
Phylo object of gene trees

`string.PathDir` String of the path to a parent directory used for simulating gene trees

Value

vector.GeneTreeProbs Vector of gene tree likelihoods for each gene tree provided

Examples

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

##################################
# Generate example species trees #
##################################
handle.SpeciesTree1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")

#################################################
# Simlate a set of gene trees for species tree1 #
#################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,
                                                           string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 100)


vector.GeneTree_Likelihoods <- Compute.GeneTree_Likelihoods(handle.SpeciesTree = handle.SpeciesTree1,
                                   handle.GeneTrees = handle.Simulated_GeneTrees,
                                           string.PathDir = '~/Desktop/')
```

Compute.GeneTree_Likelihoods_SpeciesNetwork

Compute.GeneTree_Likelihoods_SpeciesNetwork: function to compute gene tree likelihoods using PhyloNet for a species topology and/or network

Description

This function returns a vector of gene tree likelihoods given a set of gene trees and a particular species tree (networks included)

Usage

```
Compute.GeneTree_Likelihoods_SpeciesNetwork(
  string.SpeciesNetwork,
  handle.GeneTrees,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork`
String of the species network in Rich newick format (can be read by dendroscope)

`handle.GeneTrees`
Phylo object of gene trees

`string.PathDir` String of the path to a parent directory used for simulating gene trees

Value

vector.GeneTreeProbs Vector of gene tree likelihoods for each gene tree provided

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"
string.SpeciesNetwork_2 <- "((A:1,B:1):1,(C:1,D:1):1);"
```

```
####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNe
                                                          string.PathDir = '~/Desktop/',
                                                        numeric.NumberOfGeneTrees = 100)

############################
# Optimize species network #
############################
handle.Optimized_Network1 <- Optimize.Network(string.SpeciesNetwork = string.SpeciesNetwork,
                                           handle.GeneTrees = handle.SimulatedGeneTrees,
                                              string.PathDir = '~/Desktop/')


#################################################################
# Compute gene tree likelihoods given optimized species network #
#################################################################
vector.GeneTreeLikelihoods_SpeciesNetwork <- Compute.GeneTree_Likelihoods_SpeciesNetwork(string.SpeciesNetw
                                                           handle.GeneTrees = handle.SimulatedGeneTrees,
                                                           string.PathDir = '~/Desktop/')
```

Conduct.KH_1	*Conduct.KH_1: function to conduct the KH_1 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_1 STAR test for two input species tree topologies

Usage

```
Conduct.KH_1(
  handle.SpeciesTree1,
  handle.SpeciesTree2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree1`
: Phylogenetic tree defining the first species topology

`handle.SpeciesTree2`
: Phylogenetic tree defining the second species topology

`numeric.NumberOfReps`
: Number of bootstrap replicates to analyze

`string.PathDir`
: String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
: Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

Examples

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

#################################
# Generate example species trees #
#################################
handle.SpeciesTree1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree1$edge.length <- handle.SpeciesTree1$edge.length*0.2
handle.SpeciesTree2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")

################################################
# Simlate a set of gene trees for species tree1 #
################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,
                                                                  string.PathDir = '~/Desktop/',
                                                                 numeric.NumberOfGeneTrees = 10)

##############################################################
# Conduct KH_1 STAR test for the two species tree topologies #
##############################################################
Conduct.KH_1(handle.SpeciesTree1 = handle.SpeciesTree2,
             handle.SpeciesTree2 = handle.SpeciesTree1,
             handle.InputGeneTrees = handle.Simulated_GeneTrees,
             numeric.NumberOfReps = 100,
             string.PathDir = '~/Desktop/')
```

`Conduct.KH_1N` *Conduct.KH_1N: function to conduct the KH_1N STAR test given two distinct species topologies (includes one or more network topologies) and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_1 STAR test for two input species tree topologies

Usage

```
Conduct.KH_1N(
  string.SpeciesNetwork1,
  string.SpeciesNetwork2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork1`
Strint defining the first species topology (can be network or bifurcating)

`string.SpeciesNetwork2`
Strint defining the first species topology (can be network or bifurcating)

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"
string.SpeciesNetwork_2 <- "((A:1,B:1):1,(C:1,D:1):1);"


####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNe
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)


handle.KH_1N_Results <- Conduct.KH_1N(string.SpeciesNetwork1 = string.SpeciesNetwork,
                                      string.SpeciesNetwork2 = string.SpeciesNetwork_2,
                                      handle.InputGeneTrees = handle.SimulatedGeneTrees,
                                      numeric.NumberOfReps = 3,
                                      string.PathDir = '~/Desktop/')
```

`Conduct.KH_2` *Conduct.KH_2: function to conduct the KH_2 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_2 STAR test for two input species tree topologies

Usage

```
Conduct.KH_2(
  handle.SpeciesTree1,
  handle.SpeciesTree2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree1`
Phylogenetic tree defining the first species topology

`handle.SpeciesTree2`
Phylogenetic tree defining the second species topology

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

Examples

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

#################################
# Generate example species trees #
#################################
handle.SpeciesTree1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree1$edge.length <- handle.SpeciesTree1$edge.length*0.01
handle.SpeciesTree2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")
```

```
##################################################
# Simlate a set of gene trees for species tree1 #
##################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,
                                                                   string.PathDir = '~/Desktop/',
                                                                 numeric.NumberOfGeneTrees = 100)

###############################################################
# Conduct KH_2 STAR test for the two species tree topologies #
###############################################################
Conduct.KH_2(handle.SpeciesTree1 = handle.SpeciesTree1,
             handle.SpeciesTree2 = handle.SpeciesTree2,
             handle.InputGeneTrees = handle.Simulated_GeneTrees,
             numeric.NumberOfReps = 100,
             string.PathDir = '~/Desktop/')
```

Conduct.KH_2N   *Conduct.KH_2N: function to conduct the KH_2N STAR test given two distinct species topologies (one or both are networks) and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_2N STAR test for two input species tree topologies

Usage

```
Conduct.KH_2N(
  string.SpeciesNetwork1,
  string.SpeciesNetwork2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork1`
String defining the first species topology (can be network or bifurcating)

`string.SpeciesNetwork2`
String defining the first species topology (can be network or bifurcating)

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"
string.SpeciesNetwork_2 <- "((A:1,B:1):1,(C:1,D:1):1);"


####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNe
                                                          string.PathDir = '~/Desktop/',
                                                          numeric.NumberOfGeneTrees = 5)

Conduct.KH_2N(string.SpeciesNetwork1 = string.SpeciesNetwork,
              string.SpeciesNetwork2 = string.SpeciesNetwork_2,
              handle.InputGeneTrees = handle.SimulatedGeneTrees,
              numeric.NumberOfReps = 3,
              string.PathDir = '~/Desktop/')
```

Conduct.KH_3 *Conduct.KH_3: function to conduct the KH_2 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_2 STAR test for two input species tree topologies

Usage

```
Conduct.KH_3(
  handle.SpeciesTree1,
  handle.SpeciesTree2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree1`
Phylogenetic tree defining the first species topology

`handle.SpeciesTree2`
: Phylogenetic tree defining the second species topology

`numeric.NumberOfReps`
: Number of bootstrap replicates to analyze

`string.PathDir`
: String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
: Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

`Conduct.KH_3N` *Conduct.KH_3N: function to conduct the KH_2N STAR test given two distinct species topologies (one or both are networks) and a set of input gene trees*

Description

This function returns a list containing p-values of the KH_2N STAR test for two input species tree topologies

Usage

```
Conduct.KH_3N(
  string.SpeciesNetwork1,
  string.SpeciesNetwork2,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork1`
: String defining the first species topology (can be network or bifurcating)

`string.SpeciesNetwork2`
: String defining the first species topology (can be network or bifurcating)

`numeric.NumberOfReps`
: Number of bootstrap replicates to analyze

`string.PathDir`
: String defining the path to a parent directory used for conduct KH_1 STAR test

`handle.GeneTrees`
: Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) twosided pvalue, (2) upper p-values, (3) lower p-values, and (4) a vector of the bootstrapped test statistics delta

Conduct.SH_1 *Conduct.SH_1: function to conduct the SH_1 STAR test given a set of input (all plausible, including the ML) species topologies*

Description

This function returns a list containing p-values of the SH_1 STAR test for a set of input topologies

Usage

```
Conduct.SH_1(
  handle.SpeciesTopologies,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTopologies`
List of class multiPhylo containing the plausible topologies to be tested

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct SH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) matrix.Change_Delta_BS_Results, (2) vector.Pvalues, and (3) vector.Observed_Delta_ML.

Examples

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

#################################
# Generate example species trees #
#################################
handle.SpeciesTree1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree1$edge.length <- handle.SpeciesTree1$edge.length * 0.1
handle.SpeciesTree2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree3 <- read.tree(text = "(D:1.0,(B:0.5,(A:1.0,C:1.0):1.5):0.5);")
handle.SpeciesTopologies <- list()
class(handle.SpeciesTopologies) <- "multiPhylo"
handle.SpeciesTopologies[[1]] <- handle.SpeciesTree3
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2
handle.SpeciesTopologies[[3]] <- handle.SpeciesTree1



#################################################
# Simlate a set of gene trees for species tree1 #
#################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)

################################
# Conduct SH_1 STAR test for S #
################################
Conduct.SH_1(handle.SpeciesTopologies = handle.SpeciesTopologies,
             handle.InputGeneTrees = handle.Simulated_GeneTrees,
             numeric.NumberOfReps = 100,
             string.PathDir = '~/Desktop/')
```

Conduct.SH_1N *Conduct.SH_1N: function to conduct the SH_1N STAR test given a set of input (all plausible, including the ML) species topologies (can include networks)*

Description

This function returns a list containing p-values of the Conduct.SH_1N STAR test for a set of input topologies (can include networks)

Usage

```
Conduct.SH_1N(
  list.SpeciesTopologiesNetworks,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`list.SpeciesTopologiesNetworks`
List of class (not multiPhylo) containing the strings defining all plausible topologies to be tested (can include networks). Topologies are defined in strings (not phylogenetic objects)

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct SH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) matrix.Change_Delta_BS_Results, (2) vector.Pvalues, and (3) vector.Observed_Delta_ML.

Examples

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

##################################
# Generate example species trees #
##################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.5,A:1.0):2,B:1.0):2,#H1:0::0.5);"
string.SpeciesNetwork_2 <- "(((((C:1.0,A:1.0):1)#H1:0::0.5,D:1.0):2,B:1.0):2,#H1:0::0.5);"
list.SpeciesNetworks <- list()
list.SpeciesNetworks[[1]] <- string.SpeciesNetwork
list.SpeciesNetworks[[2]] <- string.SpeciesNetwork_2

##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Specie
                                                          string.PathDir = '~/Desktop/',
                                                          numeric.NumberOfGeneTrees = 2)



#####################################
# Conduct SH_1N STAR using networks #
#####################################
Conduct.SH_1N(list.SpeciesTopologiesNetworks = list.SpeciesNetworks,
              handle.InputGeneTrees = handle.Simulated_GeneTreeSet,
              numeric.NumberOfReps = 2,
              string.PathDir = '~/Desktop/')
```

Conduct.SH_2 *Conduct.SH_2: function to conduct the SH_2 STAR test given a set of input (all plausible, including the ML) species topologies using the RELL method*

Description

This function returns a list containing p-values of the SH_2 STAR test for a set of input topologies

Usage

```
Conduct.SH_2(
  handle.SpeciesTopologies,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

handle.SpeciesTopologies
List of class multiPhylo containing the plausible topologies to be tested

numeric.NumberOfReps
Number of bootstrap replicates to analyze

string.PathDir String defining the path to a parent directory used for conduct SH_2 STAR test

handle.GeneTrees
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) matrix.Change_Delta_BS_Results, (2) vector.Pvalues, and (3) vector.Observed_Delta_ML.

Examples

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

#################################
# Generate example species trees #
#################################
handle.SpeciesTree1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree1$edge.length <- handle.SpeciesTree1$edge.length * 0.1
handle.SpeciesTree2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTree3 <- read.tree(text = "(D:1.0,(B:0.5,(A:1.0,C:1.0):1.5):0.5);")
handle.SpeciesTopologies <- list()
class(handle.SpeciesTopologies) <- "multiPhylo"
handle.SpeciesTopologies[[1]] <- handle.SpeciesTree3
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2
handle.SpeciesTopologies[[3]] <- handle.SpeciesTree1



#################################################
# Simlate a set of gene trees for species tree1 #
#################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,
                                                                    string.PathDir = '~/Desktop/',
                                                                  numeric.NumberOfGeneTrees = 100)

################################
# Conduct SH_2 STAR test for S #
################################
Conduct.SH_2(handle.SpeciesTopologies = handle.SpeciesTopologies,
             handle.InputGeneTrees = handle.Simulated_GeneTrees,
             numeric.NumberOfReps = 1000,
             string.PathDir = '~/Desktop/')
```

Conduct.SH_2N *Conduct.SH_2N: function to conduct the SH_1N STAR test given a set of input (all plausible, including the ML) species topologies (can include networks)*

Description

This function returns a list containing p-values of the Conduct.SH_2N STAR test for a set of input topologies (can include networks)

Usage

```
Conduct.SH_2N(
  list.SpeciesTopologiesNetworks,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`list.SpeciesTopologiesNetworks`
: List of class (not multiPhylo) containing the strings defining all plausible topologies to be tested (can include networks). Topologies are defined in strings (not phylogenetic objects)

`numeric.NumberOfReps`
: Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct SH_2 STAR test

`handle.GeneTrees`
: Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) matrix.Change_Delta_BS_Results, (2) vector.Pvalues, and (3) vector.Observed_Delta_ML.

Examples

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

##############################
```

```
# Generate example species trees #
#################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.5,A:1.0):2,B:1.0):2,#H1:0::0.5);"
string.SpeciesNetwork_2 <- "(((((C:1.0,A:1.0):1)#H1:0::0.5,D:1.0):2,B:1.0):2,#H1:0::0.5);"
list.SpeciesNetworks <- list()
list.SpeciesNetworks[[1]] <- string.SpeciesNetwork
list.SpeciesNetworks[[2]] <- string.SpeciesNetwork_2

##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Specie
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)



#####################################
# Conduct SH_2N STAR using networks #
#####################################
Conduct.SH_2N(list.SpeciesTopologiesNetworks = list.SpeciesNetworks,
              handle.InputGeneTrees = handle.Simulated_GeneTreeSet,
              numeric.NumberOfReps = 100,
              string.PathDir = '~/Desktop/')
```

`Conduct.SOWH_1` *Conduct.SOWH_1: function to conduct the SOWH_1 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the SOWH_1 STAR test

Usage

```
Conduct.SOWH_1(
  handle.SpeciesTree_1,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree_1`
Phylogenetic tree defining the first species topology

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct SOWH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) vector.Null_BS_Delta_Stat, (2) numeric.pValue, and (3) numeric.Delta_Observed

Examples

```
#'

################
# Load depends #
################
library(SpeciesTopoTestR)
library(ape)

#################################
# Generate example species trees #
#################################
handle.SpeciesTreeX1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTreeX1$edge.length <- handle.SpeciesTreeX1$edge.length * 0.01
handle.SpeciesTreeX2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")



#################################################
# Simlate a set of gene trees for species tree1 #
#################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTreeX1,
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)


################################
# Conduct SH_2 STAR test for S #
################################
Conduct.SOWH_1(handle.SpeciesTree_1 = handle.SpeciesTreeX2,
               handle.InputGeneTrees = handle.Simulated_GeneTrees,
               numeric.NumberOfReps = 100,
               string.PathDir = '~/Desktop/')
```

Conduct.SOWH_1N *Conduct.SOWH_1N: function to conduct the SOWH_1N STAR test given a species network topology*

Description

This function returns a list containing p-values of the SOWH_1 STAR test

Usage

```
Conduct.SOWH_1N(
  string.SpeciesNetwork_1,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  numeric.MaxReticulations,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork_1`
String defining the first species topology (can be network or bifurcating)

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`numeric.MaxReticulations`
Numeric defining the maximum number of reticulations for PhyloNet search

`string.PathDir` String defining the path to a parent directory used for conduct SOWH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) vector.Null_BS_Delta_Stat, (2) numeric.pValue, and (3) numeric.Delta_Observed

Examples

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

#################################
# Generate example species trees #
#################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.5,A:1.0):2,B:1.0):2,#H1:0::0.5);"
string.SpeciesNetwork_2 <- "(((((C:1.0,A:1.0):1)#H1:0::0.5,D:1.0):2,B:1.0):2,#H1:0::0.5);"

##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Specie
                                                           string.PathDir = '~/Desktop/',
                                                           numeric.NumberOfGeneTrees = 3)



######################################
# Conduct SOWH_1N STAR using network #
######################################
Conduct.SOWH_1N(string.SpeciesNetwork_1 = string.SpeciesNetwork_2,
                handle.InputGeneTrees = handle.Simulated_GeneTreeSet,
                numeric.NumberOfReps = 2,
                numeric.MaxReticulations = 1,
                string.PathDir = '~/Desktop/')
```

Conduct.SOWH_2 *Conduct.SOWH_2: function to conduct the SOWH_1 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the SOWH_1 STAR test

Usage

```
Conduct.SOWH_2(
  handle.SpeciesTree_1,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree_1`
Phylogenetic tree defining the first species topology

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.PathDir` String defining the path to a parent directory used for conduct SOWH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) vector.Null_BS_Delta_Stat, (2) numeric.pValue, and (3) numeric.Delta_Observed

Examples

```
#'


################
# Load depends #
################
library(SpeciesTopoTestR)
library(ape)

#################################
# Generate example species trees #
#################################
handle.SpeciesTreeX1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTreeX1$edge.length <- handle.SpeciesTreeX1$edge.length * 1
handle.SpeciesTreeX2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")
```

```
#################################################
# Simlate a set of gene trees for species tree1 #
#################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTreeX1,
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)


################################
# Conduct SOWH_2 STAR test for S #
################################
Conduct.SOWH_2(handle.SpeciesTree_1 = handle.SpeciesTreeX2,
              handle.InputGeneTrees = handle.Simulated_GeneTrees,
              numeric.NumberOfReps = 100,
              string.PathDir = '~/Desktop/')
```

Conduct.SOWH_2N *Conduct.SOWH_2N: function to conduct the SOWH_2N STAR test given a species network topology*

Description

This function returns a list containing p-values of the SOWH_1 STAR test

Usage

```
Conduct.SOWH_2N(
  string.SpeciesNetwork_1,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  numeric.MaxReticulations,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork_1`
String defining the first species topology (can be network or bifurcating)

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`numeric.MaxReticulations`
Numeric defining the maximum number of reticulations for PhyloNet search

`string.PathDir` String defining the path to a parent directory used for conduct SOWH_1 STAR test

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) vector.Null_BS_Delta_Stat, (2) numeric.pValue, and (3) numeric.Delta_Observed

Examples

```
#'
#'

################
# Load depends #
################
library(SpeciesTopoTestR)
library(ape)

##################################
# Generate example species trees #
##################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.5,A:1.0):2,B:1.0):2,#H1:0::0.5);"
string.SpeciesNetwork_2 <- "(((((C:1.0,A:1.0):1)#H1:0::0.5,D:1.0):2,B:1.0):2,#H1:0::0.5);"

##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Specie
                                                                       string.PathDir = '~/Desktop/',
                                                                       numeric.NumberOfGeneTrees = 4)



#####################################
# Conduct SOWH_2 STAR using network #
#####################################
Conduct.SOWH_2N(string.SpeciesNetwork_1 = string.SpeciesNetwork_2,
                    handle.InputGeneTrees = handle.Simulated_GeneTreeSet,
                    numeric.NumberOfReps = 3,
                    numeric.MaxReticulations = 1,
                    string.PathDir = '~/Desktop/')
```

Conduct.SOWH_2v2 *Conduct.SOWH_2v2: function to conduct the SOWH_1 STAR test given two distinct species topologies and a set of input gene trees*

Description

This function returns a list containing p-values of the SOWH_1 STAR test

Usage

```
Conduct.SOWH_2v2(
  handle.SpeciesTree_1,
  handle.InputGeneTrees,
  numeric.NumberOfReps,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree_1`
: Phylogenetic tree defining the first species topology

`numeric.NumberOfReps`
: Number of bootstrap replicates to analyze

`string.PathDir`
: String defining the path to a parent directory used for conduct SOWH_1 STAR test

`handle.GeneTrees`
: Phylo object containing a list of the input gene trees

Value

List Returns a list containing (1) vector.Null_BS_Delta_Stat, (2) numeric.pValue, and (3) numeric.Delta_Observed

Examples

```
#'


################
# Load depends #
################
library(SpeciesTopoTestR)
library(ape)

#################################
# Generate example species trees #
#################################
handle.SpeciesTreeX1 <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")
handle.SpeciesTreeX1$edge.length <- handle.SpeciesTreeX1$edge.length * 1
handle.SpeciesTreeX2 <- read.tree(text = "(C:1.0,(B:0.5,(A:1.0,D:1.0):1.5):0.5);")



################################################
# Simlate a set of gene trees for species tree1 #
################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTreeX1,
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)


################################
# Conduct SOWH_2 STAR test for S #
################################
Conduct.SOWH_2v2(handle.SpeciesTree_1 = handle.SpeciesTreeX2,
              handle.InputGeneTrees = handle.Simulated_GeneTrees,
              numeric.NumberOfReps = 100,
              string.PathDir = '~/Desktop/')
```

`Conduct.SpeciesTopoTestR`

Conduct.SpeciesTopoTestR: function to conduct an array of different likelihood-based tests of species topologies

Description

This function returns a list containing the results of a topology test

Usage

```
Conduct.SpeciesTopoTestR(
  handle.Topologies2Test,
  handle.GeneTrees,
  numeric.NumberOfReps,
  string.Test,
  numeric.Algorithm,
  boo.Networks,
  string.PathDir,
  numeric.MaxReticulations
)
```

Arguments

`handle.Topologies2Test`
List of topologies to be tested. Use format "multiPhylo" for bifurcating topologies, and a standard list for networks (each network is just a string)

`handle.GeneTrees`
Phylo object containing a list of the input gene trees

`numeric.NumberOfReps`
Number of bootstrap replicates to analyze

`string.Test` String defining the test to run, can be "KH", "SH", "SOWH"

`numeric.Algorithm`
Numeric specifiying the algorithm KH (1, 2 or 3), SH (1 or 2), SOWH (1 or 2)

`boo.Networks` Boolien specifiying whether the input topologies include networks (True) or not (False)

`string.PathDir` String defining the path to a parent directory used for conduct KH_1 STAR test

`numeric.MaxReticulations`
Number of maximum reticulating edges. Only used for the SOWH test with network topologies

Value

List Returns a list containing p-values and details for the assumed topology test

Examples

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

########################################################
# Define species topologies for simulation and testing #
########################################################
handle.SpeciesTree1 <- read.tree(text = "(A:3,(B:2,(C:1,D:1):1):1);")
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")


##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTrees_T1 <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree
                                                          string.PathDir = '~/Desktop/',
                                                          numeric.NumberOfGeneTrees = 3)

########################
# Run SpeciesTopoTestR #
########################
handle.SpeciesTopologies <- list()
class(handle.SpeciesTopologies) <- "multiPhylo"
handle.SpeciesTopologies[[1]] <- handle.SpeciesTree1
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2

list.SpeciesTopologies <- list()
list.SpeciesTopologies[[1]] <- write.tree(phy = handle.SpeciesTree1, file = "")
list.SpeciesTopologies[[2]] <- write.tree(phy = handle.SpeciesTree2, file = "")


Conduct.SpeciesTopoTestR(handle.Topologies2Test = list.SpeciesTopologies,
                         handle.GeneTrees = handle.Simulated_GeneTrees_T1,
                         numeric.NumberOfReps = 3,
                         string.Test = "SOWH",
                         numeric.Algorithm = 2,
                         boo.Networks = T,
                         string.PathDir = '~/Desktop/',
                         numeric.MaxReticulations = 1)
```

Estimate.GeneTrees_IqTree

Estimate.GeneTrees_IqTree: function to estimate a set of gene trees for an input list of nexus alignments (simulated with Simulate.Gene.Alignments)

Description

This function returns (1) a list of ML gene tree estimates obtained via IQ-Tree

Usage

```
Estimate.GeneTrees_IqTree(
  list.Simulated.Nexus.Alignments,
  path.PathParentDir,
  string.Model
)
```

Arguments

`list.Simulated.Nexus.Alignments`
List of nexus alignments that have been simulated using Simulate.Gene.Alignments

`path.PathParentDir`
Path to parent directory used for gene tree estimation

`string.Model` String to specify substitution model for IQ-tree

Examples

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

########################################################
# Define species topologies for simulation and testing #
########################################################
handle.SpeciesTree1 <- read.tree(text = "(A:3,(B:2,(C:1,D:1):1):1);")
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")


##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTrees_T1 <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree
                                                          string.PathDir = '~/Desktop/',
                                                        numeric.NumberOfGeneTrees = 100)
############################################
# Convert to mutation units branch lengths #
############################################
handle.Converted_GeneTrees_T1 <- list()
class(handle.Converted_GeneTrees_T1) <- "multiPhylo"

for (i in 1:length(handle.Simulated_GeneTrees_T1)){

  handle.Simulated_GeneTrees_T1_i <- handle.Simulated_GeneTrees_T1[[i]]
 handle.Simulated_GeneTrees_T1_i$edge.length = handle.Simulated_GeneTrees_T1_i$edge.length* 0.00005
  handle.Converted_GeneTrees_T1[[i]] <- handle.Simulated_GeneTrees_T1_i
}

################################
# Simulate sequence alignments #
################################
vector.pi <- c(0.3, 0.2, 0.2, 0.3)
```

```
names(vector.pi) <- c("A", "C", "G", "T")
list.SimulatedAlignments <- Simulate.GeneSequenceAlignments(list.Simulated.Gene.Tree.Set = handle.Converted_
                                                       path.PathParentDir = '~/Desktop/',
                                                             numeric.Ratio = 4.6,
                                                             numeric.LocusLength = 1000,
                                                      vector.BaseFrequencies = vector.pi)

#######################
# Estimate gene trees #
#######################
handle.EstimatedGeneTrees <- Estimate.GeneTrees_IqTree(list.Simulated.Nexus.Alignments = list.SimulatedAlign
                                                       path.PathParentDir = '~/Desktop/',
                                                        string.Model = "HKY")

################
# Conduct KH_2 #
################
handle.Topologies <- list()
class(handle.Topologies) <- "multiPhylo"
handle.Topologies[[1]] <- handle.SpeciesTree1
handle.Topologies[[2]] <- handle.SpeciesTree2

Conduct.SpeciesTopoTestR(handle.Topologies2Test = handle.Topologies,
                        handle.GeneTrees = handle.EstimatedGeneTrees$list.MLE.GeneTrees,
                         numeric.NumberOfReps = 1000,
                         string.Test = "KH",
                         numeric.Algorithm = 2,
                         boo.Networks = F,
                         string.PathDir = '~/Desktop/')
```

Optimize.BranchLengths

Optimize.BranchLengths: function to optimize branch lengths (in coalescent units) of an input species tree topology given a set of input gene tree topologies

Description

This function returns a species tree with branch lengths in coalescent units, which have been optimized using the MSC and the STELLS2 algorithm

Usage

```
Optimize.BranchLengths(
  handle.SpeciesTree,
  handle.GeneTrees,
  numeric.Stells_algorthm,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree`
Phylo object defining the input species tree

`handle.GeneTrees`
Phylo object containing a list of gene trees

`numeric.Stells_algorthm`
Numeric defining with algorithm of STELLS to use: 0 or 1

`string.PathDir` String defining the path to a parent directory used for optimizing species tree

Value

handle.Optimized_SpeciesTree Species tree with branch lengths (in coalescent units) that has been optimized with STELLS

numeric.MaximizedLnL Numeric containing the maximum likelihood given the optimized branch lengths of the species tree

Examples

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

#################################
# Generate example species tree #
#################################
handle.SpeciesTree <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")

#####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree,
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)

#############################################################################
# Optimize branch lengths (in coalescent units) of given input species tree #
#############################################################################
handle.OptimizedBranchLengths <- Optimize.BranchLengths(handle.SpeciesTree = handle.SpeciesTree,
                                          handle.GeneTrees = handle.Simulated_GeneTrees,
                                                        numeric.Stells_algorthm = 1,
                                                        string.PathDir = '~/Desktop/')
```

`Optimize.Network` *Optimize.Network: function to optimize a species network topology using Phylonet*

Description

This function returns a species network (string) with branch lengths amd hybyridization edges that have been optimized using the MSC and the PhyloNet algorithm

Usage

```
Optimize.Network(string.SpeciesNetwork, handle.GeneTrees, string.PathDir)
```

Arguments

`string.SpeciesNetwork`
String of the species network in Rich newick format (can be read by dendroscope)

`handle.GeneTrees`
Phylo object containing a list of gene trees

`string.PathDir` String defining the path to a parent directory used for optimizing species tree

Value

string.Optimized_SpeciesNetwork Species network with branch lengths (in coalescent units) and hybridization edge that has been optimized with PhyloNet

numeric.MaximizedLnL Numeric containing the maximum likelihood given the optimized parameters of the species network

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"

####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNe
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)

###############################################
# Optimize network using simulated gene trees #
###############################################
handle.Optimized_Network <- Optimize.Network(string.SpeciesNetwork = string.SpeciesNetwork,
                                           handle.GeneTrees = handle.SimulatedGeneTrees,
                                             string.PathDir = '~/Desktop/')
```

Optimize.NetworkSearch

Optimize.NetworkSearch: function to find the ML network topology using PhyloNet

---

Description

This function returns an optimized network topology and its maximized likelihood

Usage

```
Optimize.NetworkSearch(
  handle.GeneTrees,
  numeric.MaxReticulations,
  string.PathDir
)
```

Arguments

`handle.GeneTrees`
Phylo object containing a list of gene trees

`numeric.MaxReticulations`
Numeric defining the maximum number of reticulations for PhyloNet search

`string.PathDir` String defining the path to a parent directory used for optimizing species tree

`numeric.Stells_algorthm`
Numeric defining with algorithm of STELLS to use: 0 or 1

Value

handle.Optimized_SpeciesTree Species tree with branch lengths (in coalescent units) that has been optimized with STELLS

numeric.MaximizedLnL Numeric containing the maximum likelihood given the optimized branch lengths of the species tree

Examples

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

#################################
# Generate example species trees #
#################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.5,A:1.0):2,B:1.0):2,#H1:0::0.5);"
string.SpeciesNetwork_2 <- "(((((C:1.0,A:1.0):1)#H1:0::0.5,D:1.0):2,B:1.0):2,#H1:0::0.5);"

##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Specie
                                                          string.PathDir = '~/Desktop/',
                                                          numeric.NumberOfGeneTrees = 5)

####################################
# Optimize network topology search #
####################################
handle.OptimizedNetwork_Topology <- Optimize.NetworkSearch(handle.GeneTrees = handle.Simulated_GeneTreeSet,
                                                           numeric.MaxReticulations = 1,
                                                          string.PathDir = '~/Desktop/')
```

`Optimize.TopologySearch` *Optimize.TopologySearch: function to find the ML topology using STELLS*

Description

This function returns an optimized species topology and its maximized likelihood

Usage

```
Optimize.TopologySearch(handle.GeneTrees, string.PathDir)
```

Arguments

`handle.GeneTrees`
Phylo object containing a list of gene trees

`string.PathDir` String defining the path to a parent directory used for optimizing species tree

`numeric.Stells_algorthm`
Numeric defining with algorithm of STELLS to use: 0 or 1

Value

handle.Optimized_SpeciesTree Species tree with branch lengths (in coalescent units) that has been optimized with STELLS

numeric.MaximizedLnL Numeric containing the maximum likelihood given the optimized branch lengths of the species tree

`Resample.Bootstrap_GeneTree_Replicates_NP` *Resample.Bootstrap_GeneTree_Replicates_NP: function to conduct non-parametric bootstrapping (i.e., random sampling with replacement) for a set of input gene trees*

Description

This function returns a list containing multiple replicate sets of gene trees (each replicate is a bootstrapped dataset)

Usage

```
Resample.Bootstrap_GeneTree_Replicates_NP(
  handle.GeneTrees,
  numeric.NumberOfReps
)
```

Arguments

`handle.GeneTrees`
: MultiPhylo object of the input gene trees

`numeric.NumberOfReps`
: Numeric number of bootstrap replicates

Value

list.BootStrap_GeneTree_ReplicateSets List containing sets of gene trees bootstrap replicates

Examples

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

#################################
# Generate example species tree #
#################################
handle.SpeciesTree <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")

#####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree,
                                                                  string.PathDir = '~/Desktop/',
                                                                 numeric.NumberOfGeneTrees = 10)

#################################################
# Resample from example simulated gene tree set #
#################################################
list.Bootstrapped_GeneTreesSets <- Resample.Bootstrap_GeneTree_Replicates_NP(handle.GeneTrees = handle.Simul
```

`Simulate.GeneSequenceAlignments` *Simulate.GeneSequenceAlignments: function to simulate a set of sequence alignments for a list of gene trees*

Description

This function returns (1) a list of nexus alignments that have been simulated for the input gene trees

Usage

```
Simulate.GeneSequenceAlignments(
  list.Simulated.Gene.Tree.Set,
  path.PathParentDir,
  numeric.Ratio,
  numeric.LocusLength,
  vector.BaseFrequencies
)
```

Arguments

`list.Simulated.Gene.Tree.Set`
Multiphylo list containing all the gene trees

`path.PathParentDir`
Path to parent directory used for simulations

`numeric.Ratio` Kappa parameter for HKY substitution model

`numeric.LocusLength`
Length of each locus

`vector.BaseFrequencies`
Vector of base equibrium frequencies, with names equal to the nucleotide state

Examples

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

########################################################
# Define species topologies for simulation and testing #
########################################################
handle.SpeciesTree1 <- read.tree(text = "(A:3,(B:2,(C:1,D:1):1):1);")
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")


##########################
# Simulate gene tree set #
##########################
handle.Simulated_GeneTrees_T1 <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree
                                                                     string.PathDir = '~/Desktop/',
                                                                     numeric.NumberOfGeneTrees = 3)
############################################
# Convert to mutation units branch lengths #
############################################
handle.Converted_GeneTrees_T1 <- list()
class(handle.Converted_GeneTrees_T1) <- "multiPhylo"

for (i in 1:length(handle.Simulated_GeneTrees_T1)){

  handle.Simulated_GeneTrees_T1_i <- handle.Simulated_GeneTrees_T1[[i]]
 handle.Simulated_GeneTrees_T1_i$edge.length = handle.Simulated_GeneTrees_T1_i$edge.length* 0.00005
  handle.Converted_GeneTrees_T1[[i]] <- handle.Simulated_GeneTrees_T1_i
}

################################
# Simulate sequence alignments #
################################
vector.pi <- c(0.3, 0.2, 0.2, 0.3)
```

```
names(vector.pi) <- c("A", "C", "G", "T")
list.SimulatedAlignments <- Simulate.GeneSequenceAlignments(list.Simulated.Gene.Tree.Set = handle.Converted_
                                                            path.PathParentDir = '~/Desktop/',
                                                                  numeric.Ratio = 4.6,
                                                                  numeric.LocusLength = 100,
                                                           vector.BaseFrequencies = vector.pi)
```

Simulate.GeneTrees_From_SpeciesNetwork

Simulate.GeneTrees_From_SpeciesNetwork: function to simulate a set of gene trees given a species network using PhyloNet under the MSC

Description

This function returns a list of simulated gene trees that have been generated for a given input species network

Usage

```
Simulate.GeneTrees_From_SpeciesNetwork(
  string.SpeciesNetwork,
  numeric.NumberOfGeneTrees,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork`
String of the species network in Rich newick format (can be read by dendroscope)

`numeric.NumberOfGeneTrees`
Numeric number of gene trees to simulate

`string.PathDir` String of the path to a parent directory that will be used for simulating gene trees

Value

handle.SimulatedGeneTrees List of simulated gene trees

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"

###############################################
# Simlate a set of gene trees for this species tree #
```

```
#####################################################
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNe
                                                          string.PathDir = '~/Desktop/',
                                                         numeric.NumberOfGeneTrees = 10)
```

Simulate.GeneTrees_From_SpeciesNetwork_MS

Simulate.GeneTrees_From_SpeciesNetwork_MS: function to simulate a set of gene trees given a species network using PhyloNet under the MSC

Description

This function returns a list of simulated gene trees that have been generated for a given input species network

Usage

```
Simulate.GeneTrees_From_SpeciesNetwork_MS(
  string.SpeciesNetwork,
  numeric.NumberOfGeneTrees,
  string.PathDir
)
```

Arguments

`string.SpeciesNetwork`
String of the species network in Rich newick format (can be read by dendroscope)

`numeric.NumberOfGeneTrees`
Numeric number of gene trees to simulate

`string.PathDir` String of the path to a parent directory that will be used for simulating gene trees

Value

handle.SimulatedGeneTrees List of simulated gene trees

Examples

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

####################################
# Generate example species network #
####################################
string.SpeciesNetwork <- "(((((C:1.0,D:1.0):1)#H1:0::0.25,A:1.0):2,B:1.0):2,#H1:0::0.75);"

####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
```

```
handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork_MS(string.SpeciesNetwork = string.Specie
                                                                      string.PathDir = '~/Desktop/',
                                                                     numeric.NumberOfGeneTrees = 10)
```

Simulate.GeneTrees_From_SpeciesTree

Simulate.GeneTrees_From_SpeciesTree: function to simulate a set of gene trees given a species tree using HYBRID-LAMBDA under the MSC

Description

This function returns a list of simulated gene trees that have been generated for a given input species tree

Usage

```
Simulate.GeneTrees_From_SpeciesTree(
  handle.SpeciesTree,
  numeric.NumberOfGeneTrees,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree`
Phylo object of the input species tree

`numeric.NumberOfGeneTrees`
Numeric number of gene trees to simulate

`string.PathDir` String of the path to a parent directory that will be used for simulating gene trees

Value

handle.SimulatedGeneTrees List of simulated gene trees

Examples

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

#################################
# Generate example species tree #
#################################
handle.SpeciesTree <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")

#####################################################
# Simlate a set of gene trees for this species tree #
#####################################################
Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree,
                                    string.PathDir = '~/Desktop/',
                                    numeric.NumberOfGeneTrees = 10)
```

Simulate.GeneTrees_From_SpeciesTree2

Simulate.GeneTrees_From_SpeciesTree2: function to simulate a set of gene trees given a species tree using HYBRID-LAMBDA under the MSC

Description

This function returns a list of simulated gene trees that have been generated for a given input species tree

Usage

```
Simulate.GeneTrees_From_SpeciesTree2(
  handle.SpeciesTree,
  numeric.NumberOfGeneTrees,
  string.PathDir
)
```

Arguments

`handle.SpeciesTree`
Phylo object of the input species tree

`numeric.NumberOfGeneTrees`
Numeric number of gene trees to simulate

`string.PathDir` String of the path to a parent directory that will be used for simulating gene trees

Value

handle.SimulatedGeneTrees List of simulated gene trees

Examples

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

#################################
# Generate example species tree #
#################################
handle.SpeciesTree <- read.tree(text = "(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5);")

####################################################
# Simlate a set of gene trees for this species tree #
####################################################
Simulate.GeneTrees_From_SpeciesTree2(handle.SpeciesTree = handle.SpeciesTree,
                                    string.PathDir = '~/Desktop/',
                                    numeric.NumberOfGeneTrees = 10)
```

Index