Package 'SpeciesTopoTestR'

September 30, 2020

Title	Likelihood-based	tests of	species	topol	logies	in F	(
Version	n 0.00 9000						

version 0.0.0.9000

 $\label{package does one paragraph).} \textbf{Description} \ \ What the package does (one paragraph).$

License What license it uses

Encoding UTF-8 **LazyData** true **RoxygenNote** 7.1.1

R topics documented:

Compute.GeneTree_Likelihoods
Compute.GeneTree_Likelihoods_SpeciesNetwork
Conduct.KH_1
Conduct.KH_1N
Conduct.KH_2
Conduct.KH_2N
Conduct.KH_3
Conduct.KH_3N
Conduct.SH_1
Conduct.SH_1N
Conduct.SH_2
Conduct.SH_2N
Conduct.SOWH_1
Conduct.SOWH_1N
Conduct.SOWH_2
Conduct.SOWH_2N
Conduct.SOWH_2v2
Conduct.SpeciesTopoTestR
Estimate.GeneTrees_IqTree
Optimize.BranchLengths
Optimize.Network
Optimize.NetworkSearch
Optimize.TopologySearch
Resample.Bootstrap_GeneTree_Replicates_NP
Simulate.GeneSequenceAlignments
Simulate.GeneTrees_From_SpeciesNetwork
Simulate.GeneTrees_From_SpeciesNetwork_MS
Simulate.GeneTrees_From_SpeciesTree
Simulate.GeneTrees_From_SpeciesTree2

Index 37

```
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

vector.GeneTree_Likelihoods <- Compute.GeneTree_Likelihoods(handle.SpeciesTree = handle.SpeciesTree1,</pre>

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

4 Conduct.KH_1

```
# Simlate a set of gene trees for this species tree #
 handle.SimulatedGeneTrees <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNetwork
                                              string.PathDir = '~/Desktop/',
                                            numeric.NumberOfGeneTrees = 100)
 # Optimize species network #
 handle.Optimized_Network1 <- Optimize.Network(string.SpeciesNetwork = string.SpeciesNetwork,</pre>
                                  handle.GeneTrees = handle.SimulatedGeneTrees,
                                    string.PathDir = '~/Desktop/')
 # Compute gene tree likelihoods given optimized species network #
 vector.GeneTreeLikelihoods_SpeciesNetwork <- Compute.GeneTree_Likelihoods_SpeciesNetwork(string.SpeciesNetwork)</pre>
                                              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. Species Tree 1

Phylogenetic tree defining the first species topology

handle. Species Tree 2

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

Conduct.KH_1N 5

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
```

Description

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

and a set of input gene trees

distinct species topologies (includes one or more network topologies)

Usage

```
Conduct.KH_1N(
   string.SpeciesNetwork1,
   string.SpeciesNetwork2,
   handle.InputGeneTrees,
```

6 Conduct.KH_1N

```
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. Species Network <- "((((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.SpeciesNetwork
                                               string.PathDir = '~/Desktop/',
                                               numeric.NumberOfGeneTrees = 10)
handle.KH_1N_Results <- Conduct.KH_1N(string.SpeciesNetwork1 = string.SpeciesNetwork,</pre>
                               string.SpeciesNetwork2 = string.SpeciesNetwork_2,
                               handle.InputGeneTrees = handle.SimulatedGeneTrees,
```

numeric.NumberOfReps = 3, string.PathDir = '~/Desktop/') Conduct.KH_2 7

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

8 Conduct.KH_2N

```
# 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
```

Description

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

gene trees

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

Conduct.KH_3

Examples

```
##################
# Load depends #
##################
library(SpeciesTopoTestR)
library(ape)
# Generate example species network #
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.SpeciesNetwork
                                          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

10 Conduct.KH_3N

```
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

 ${\it string.} Species Network 1$

String defining the first species topology (can be network or bifurcating)

string.SpeciesNetwork2

String defining the first species topology (can be network or bifurcating)

 ${\tt numeric.NumberOfReps}$

Number of bootstrap replicates to analyze

 ${\tt string.PathDir} \ \ {\tt String} \ defining \ the \ path \ to \ a \ parent \ directory \ used \ for \ conduct \ KH_1 \ STAR \ test \ handle. GeneTrees$

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

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.

```
##################
# 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. Species Tree 1\$ edge. length <- handle. Species Tree 1\$ 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()</pre>
class(handle.SpeciesTopologies) <- "multiPhylo"</pre>
handle.SpeciesTopologies[[1]] <- handle.SpeciesTree3</pre>
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2</pre>
```

12 Conduct.SH_1N

handle.InputGeneTrees = handle.Simulated_GeneTrees,

```
handle.SpeciesTopologies[[3]] <- handle.SpeciesTree1</pre>
```

numeric.NumberOfReps = 100, string.PathDir = '~/Desktop/')

Conduct.SH_1N

Conduct.SH_IN: function to conduct the SH_IN 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. Species Topologies Networks

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

Conduct.SH_2

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()</pre>
 list.SpeciesNetworks[[1]] <- string.SpeciesNetwork</pre>
 list.SpeciesNetworks[[2]] <- string.SpeciesNetwork_2</pre>
 ##############################
 # Simulate gene tree set #
 #############################
 handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.SpeciesNetwork)
                                                         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,
```

14 Conduct.SH_2

```
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.

```
#################
# 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. Species Tree 1\$ edge.length <- handle. Species Tree 1\$ 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()</pre>
class(handle.SpeciesTopologies) <- "multiPhylo"</pre>
handle. Species Topologies \hbox{\tt [[1]]} <- handle. Species Tree 3
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2</pre>
handle.SpeciesTopologies[[3]] <- handle.SpeciesTree1</pre>
# Simlate a set of gene trees for species tree1 #
handle.Simulated_GeneTrees <- Simulate.GeneTrees_From_SpeciesTree(handle.SpeciesTree = handle.SpeciesTree1,</pre>
                                                    string.PathDir = '~/Desktop/',
                                                  numeric.NumberOfGeneTrees = 100)
# Conduct SH_2 STAR test for S #
Conduct.SH_2(handle.SpeciesTopologies = handle.SpeciesTopologies,
```

Conduct.SH_2N 15

```
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

 ${\tt string.PathDir} \ \ String\ defining\ the\ path\ to\ a\ parent\ directory\ used\ for\ conduct\ SH_2\ STAR\ test\ handle.GeneTrees$

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.

16 Conduct.SOWH_1

```
# 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. Species Network\_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()</pre>
list.SpeciesNetworks[[1]] <- string.SpeciesNetwork</pre>
list.SpeciesNetworks[[2]] <- string.SpeciesNetwork_2</pre>
# Simulate gene tree set #
##############################
handle.Simulated_GeneTreeSet <- Simulate.GeneTrees_From_SpeciesNetwork(string.SpeciesNetwork = string.Species
                                                      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: function to conduct the SOWH_1 STAR test given

two distinct species topologies and a set of input gene trees

Description

Conduct.SOWH_1

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
```

Conduct.SOWH_1N 17

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. Species Tree X1\$ edge.length \ \leftarrow \ handle. Species Tree X1\$ 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
```

Description

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

given a species network topology

Usage

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

18 Conduct.SOWH_1N

```
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

Conduct.SOWH_1N(string.SpeciesNetwork_1 = string.SpeciesNetwork_2,

numeric.NumberOfReps = 2,
numeric.MaxReticulations = 1,
string.PathDir = '~/Desktop/')

handle.InputGeneTrees = handle.Simulated_GeneTreeSet,

Value

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

```
#'
#'
#################
# 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. Species Network\_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.SpeciesNetwork)
                                                 string.PathDir = '~/Desktop/',
                                                 numeric.NumberOfGeneTrees = 3)
# Conduct SOWH_1N STAR using network #
```

Conduct.SOWH_2

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

20 Conduct.SOWH_2N

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

Conduct.SOWH_2v2 21

Examples

```
#'
#################
# Load depends #
#################
library(SpeciesTopoTestR)
library(ape)
####################################
# Generate example species trees #
string. Species Network <- "(((((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.SpeciesNetwork)
                                                  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
)
```

22 Conduct.SOWH_2v2

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</pre>
\label{eq: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.MaxReticulations

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)

 ${\tt string.PathDir} \ \ {\tt String} \ {\tt defining} \ {\tt the} \ {\tt path} \ {\tt to} \ {\tt a} \ {\tt parent} \ {\tt directory} \ {\tt used} \ {\tt for} \ {\tt conduct} \ {\tt KH_1} \ {\tt STAR} \ {\tt test} \\$

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);")</pre>
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")</pre>
##############################
# 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()</pre>
class(handle.SpeciesTopologies) <- "multiPhylo"</pre>
handle.SpeciesTopologies[[1]] <- handle.SpeciesTree1</pre>
handle.SpeciesTopologies[[2]] <- handle.SpeciesTree2</pre>
list.SpeciesTopologies <- list()</pre>
list.SpeciesTopologies[[1]] <- write.tree(phy = handle.SpeciesTree1, file = "")</pre>
list.SpeciesTopologies[[2]] <- write.tree(phy = handle.SpeciesTree2, file = "")</pre>
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

```
##################
# 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);")</pre>
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")</pre>
##############################
# 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()</pre>
class(handle.Converted_GeneTrees_T1) <- "multiPhylo"</pre>
for (i in 1:length(handle.Simulated_GeneTrees_T1)){
 handle.Simulated_GeneTrees_T1_i <- handle.Simulated_GeneTrees_T1[[i]]</pre>
 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")</pre>
list.SimulatedAlignments <- Simulate.GeneSequenceAlignments(list.Simulated.Gene.Tree.Set = handle.Converted_</pre>
                                                          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()</pre>
class(handle.Topologies) <- "multiPhylo"</pre>
handle.Topologies[[1]] <- handle.SpeciesTree1</pre>
handle.Topologies[[2]] <- handle.SpeciesTree2</pre>
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

Optimize.Network 27

```
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 dendro-scope)

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.SpeciesNetwork
                                          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

```
##################
# Load depends #
#################
library(SpeciesTopoTestR)
library(ape)
# Generate example species trees #
string. Species Network <- "((((C:1.0,D:1.0):1)\#H1:0::0.5,A:1.0):2,B:1.0):2,\#H1:0::0.5);"
string. Species Network\_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.SpeciesNetwork)
                                                   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 boot-strapped 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
```

Examples

vector.BaseFrequencies

```
##################
# 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);")</pre>
handle.SpeciesTree2 <- read.tree(text = "(A:3,(D:2,(C:1,B:1):1):1);")</pre>
##############################
# 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()</pre>
class(handle.Converted_GeneTrees_T1) <- "multiPhylo"</pre>
for (i in 1:length(handle.Simulated_GeneTrees_T1)){
 handle.Simulated_GeneTrees_T1_i <- handle.Simulated_GeneTrees_T1[[i]]</pre>
 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)
```

Vector of base equibrium frequencies, with names equal to the nucleotide state

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

```
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

```
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

```
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

Index

```
* Network
                                                    Simulate.GeneTrees_From_SpeciesTree2,
    Conduct.SH_1N, 12
                                                        36
                                               * bayesian
    Conduct.SH_2N, 15
    Conduct.SOWH_1N, 17
                                                    Estimate.GeneTrees_IqTree, 24
                                                    Simulate.GeneSequenceAlignments,
    Conduct.SOWH_2N, 20
                                                        31
* SH,
                                               * coalescent,
    Conduct.SH_1N, 12
    Conduct.SH_2N, 15
                                                    Compute.GeneTree_Likelihoods, 2
* SWOH,
                                                    Compute.GeneTree_Likelihoods_SpeciesNetwork,
                                                        3
    Conduct.SOWH_1N, 17
                                                    Conduct.KH_1, 4
    Conduct.SOWH_2N, 20
* Species
                                                    Conduct.KH_1N, 5
    Compute.GeneTree_Likelihoods, 2
                                                    Conduct.KH_2, 7
    Compute.GeneTree_Likelihoods_SpeciesNetwork, Conduct.KH_2N, 8
        3
                                                    Conduct.KH_3, 9
    Conduct.KH_1, 4
                                                    Conduct.KH_3N, 10
    Conduct.KH_1N, 5
                                                    Conduct.SH_1, 11
    Conduct.KH_2, 7
                                                    Conduct.SH_1N, 12
    Conduct.KH_2N, 8
                                                    Conduct.SH_2, 13
    Conduct.KH_3, 9
                                                    Conduct. SH_2N, 15
                                                    Conduct.SOWH_1, 16
    Conduct.KH_3N, 10
    Conduct.SH_1, 11
                                                    Conduct.SOWH_1N, 17
    Conduct.SH_1N, 12
                                                    Conduct.SOWH_2, 19
    Conduct.SH_2, 13
                                                    Conduct. SOWH_2N, 20
    Conduct.SH_2N, 15
                                                    Conduct.SOWH_2v2, 21
    Conduct.SOWH_1, 16
                                                    Conduct.SpeciesTopoTestR, 23
    Conduct.SOWH_1N, 17
                                                    Optimize.BranchLengths, 26
    Conduct.SOWH_2, 19
                                                    Optimize. Network, 27
                                                    Optimize.NetworkSearch, 28
    Conduct.SOWH_2N, 20
    Conduct.SOWH_2v2, 21
                                                    Optimize. Topology Search, 30
    Conduct.SpeciesTopoTestR, 23
                                                    Resample.Bootstrap_GeneTree_Replicates_NP,
    Optimize.BranchLengths, 26
                                                    Simulate.GeneTrees_From_SpeciesNetwork,
    Optimize. Network, 27
    Optimize.NetworkSearch, 28
    Optimize. Topology Search, 30
                                                    Simulate.GeneTrees_From_SpeciesNetwork_MS,
    Resample.Bootstrap_GeneTree_Replicates_NP,
                                                    Simulate.GeneTrees_From_SpeciesTree,
    Simulate.GeneTrees_From_SpeciesNetwork,
                                                    Simulate.GeneTrees_From_SpeciesTree2,
                                                        36
    Simulate.GeneTrees_From_SpeciesNetwork_MS,
                                               * coalescent
    Simulate.GeneTrees_From_SpeciesTree,
                                                    Estimate.GeneTrees_IqTree, 24
        35
                                                    Simulate.GeneSequenceAlignments,
```

INDEX

31	Compute.GeneTree_Likelihoods_SpeciesNetwork,				
* empirical	3				
Estimate.GeneTrees_IqTree, 24	Conduct.KH_1,4				
Simulate.GeneSequenceAlignments,	Conduct.KH_1N, 5				
31	Conduct.KH_2, 7				
* estimation,	Conduct.KH_2N, 8				
Estimate.GeneTrees_IqTree, 24	Conduct.KH_3, 9				
Simulate.GeneSequenceAlignments,	Conduct.KH_3N, 10				
31	Conduct.SH_1, 11				
* inference,	Conduct.SH_1N, 12				
Estimate.GeneTrees_IqTree, 24	Conduct.SH_2, 13				
Simulate.GeneSequenceAlignments,	Conduct.SH_2N, 15				
31	Conduct.SOWH_1, 16				
* multispecies	Conduct.SOWH_2, 19				
Compute.GeneTree_Likelihoods, 2	Conduct.SOWH_2v2, 21				
Compute.GeneTree_Likelihoods_SpeciesNetwork					
3	Optimize.Network, 27				
Conduct.KH_1, 4	Optimize.NetworkSearch, 28				
Conduct.KH_1N, 5	Optimize.TopologySearch, 30				
Conduct.KH_2, 7	Resample.Bootstrap_GeneTree_Replicates_NP,				
Conduct.KH_2N, 8	30				
Conduct.KH_3, 9	Simulate.GeneTrees_From_SpeciesNetwork,				
Conduct.KH_3N, 10	33				
Conduct.SH_1, 11	Simulate.GeneTrees_From_SpeciesNetwork_MS,				
Conduct.SH_1N, 12	34				
Conduct.SH_2, 13	Simulate.GeneTrees_From_SpeciesTree,				
Conduct.SH_2N, 15	35				
Conduct.SOWH_1, 16	Simulate.GeneTrees_From_SpeciesTree2,				
Conduct.SOWH_1N, 17	36				
	phylogenetics				
Conduct.SOWH_2N, 20	Conduct.SpeciesTopoTestR, 23				
	phylogenomics,				
Conduct.SpeciesTopoTestR, 23	Conduct.SH_1N, 12				
Estimate.GeneTrees_IqTree, 24	Conduct.SH_2N, 15				
·	obly logenomics				
Optimize. Network, 27	Compute.GeneTree_Likelihoods, 2				
Optimize.NetworkSearch, 28	Compute.GeneTree_Likelihoods_SpeciesNetwork,				
Optimize. TopologySearch, 30	3				
Resample.Bootstrap_GeneTree_Replicates_NP,	Conduct.KH_1,4				
30	Conduct.KH_1N, 5				
Simulate.GeneSequenceAlignments,	Conduct.KH_2, 7				
31	Conduct.KH_2N, 8				
Simulate.GeneTrees_From_SpeciesNetwork,	Conduct.KH_3,9				
33	Conduct.KH_3N, 10				
Simulate.GeneTrees_From_SpeciesNetwork_MS,	Conduct.SH_1, 11				
34	Conduct.SH_2, 13				
Simulate.GeneTrees_From_SpeciesTree,	Conduct.SOWH_1, 16				
35	Conduct.SOWH_2, 19				
Simulate.GeneTrees_From_SpeciesTree2,	Conduct.SOWH_2v2, 21				
36	Optimize.BranchLengths, 26				
* phylogenetics,	Optimize. Network, 27				
Compute.GeneTree_Likelihoods, 2	Optimize.Network, 27 Optimize.NetworkSearch, 28				
osmpate. Serie i ce_Like i i loods, 2	op timize the their recar off, 20				

INDEX 39

```
Optimize. Topology Search, 30
                                                        31
    Resample.Bootstrap_GeneTree_Replicates_NP,
                                                Compute.GeneTree_Likelihoods, 2
                                                Compute.GeneTree_Likelihoods_SpeciesNetwork,
    Simulate.GeneTrees_From_SpeciesNetwork,
    {\tt Simulate.GeneTrees\_From\_SpeciesNetwork\_MS\_Conduct.KH\_1,4}
                                                Conduct.KH_1N, 5
                                                Conduct.KH_2, 7
    Simulate.GeneTrees_From_SpeciesTree,
                                                Conduct.KH_2N, 8
                                                Conduct.KH_3, 9
    Simulate.GeneTrees_From_SpeciesTree2,
                                                Conduct.KH_3N, 10
        36
                                                Conduct.SH_1, 11
* species
                                                Conduct.SH_1N, 12
    Estimate.GeneTrees_IqTree, 24
                                                Conduct.SH_2, 13
    Simulate.GeneSequenceAlignments,
                                                Conduct.SH_2N, 15
        31
                                                Conduct.SOWH_1, 16
* tree,
                                                Conduct.SOWH_1N, 17
    Compute.GeneTree_Likelihoods, 2
    Compute.GeneTree_Likelihoods_SpeciesNetwork, ______Conduct.SOWH_2, 19
                                                Conduct.SOWH_2N, 20
                                                Conduct.SOWH_2v2, 21
    Conduct.KH_1, 4
                                                Conduct.SpeciesTopoTestR, 23
    Conduct.KH_1N, 5
    Conduct.KH_2, 7
                                                Estimate.GeneTrees_IqTree, 24
    Conduct.KH_2N, 8
    Conduct.KH_3, 9
                                                Optimize.BranchLengths, 26
    Conduct.KH_3N, 10
                                                Optimize.Network, 27
    Conduct.SH_1, 11
                                                Optimize. Network Search, 28
    Conduct.SH_1N, 12
                                                Optimize. Topology Search, 30
    Conduct.SH_2, 13
                                                Resample.Bootstrap_GeneTree_Replicates_NP,
    Conduct.SH_2N, 15
    Conduct.SOWH_1, 16
    Conduct.SOWH_1N, 17
                                                Simulate.GeneSequenceAlignments, 31
    Conduct.SOWH_2, 19
                                                Simulate.GeneTrees_From_SpeciesNetwork,
    Conduct.SOWH_2N, 20
    Conduct.SOWH_2v2, 21
                                                Simulate.GeneTrees_From_SpeciesNetwork_MS,
    Conduct.SpeciesTopoTestR, 23
    Optimize.BranchLengths, 26
                                                Simulate.GeneTrees_From_SpeciesTree,
    Optimize. Network, 27
    Optimize.NetworkSearch, 28
                                                Simulate.GeneTrees_From_SpeciesTree2,
    Optimize. Topology Search, 30
                                                        36
    Resample.Bootstrap_GeneTree_Replicates_NP,
    Simulate.GeneTrees_From_SpeciesNetwork,
    Simulate.GeneTrees_From_SpeciesNetwork_MS,
    Simulate.GeneTrees_From_SpeciesTree,
    {\tt Simulate.GeneTrees\_From\_SpeciesTree2},
        36
* tree
    Estimate.GeneTrees_IqTree, 24
    Simulate.GeneSequenceAlignments,
```