Package 'CollessLike'

March 26, 2018

Type Package

Title Methods to Compute Distribution and Percentile of 3 Balance Indices of Phylogenetic Trees

Version 0.0.6911

Date 2018-03-20

Author Arnau Mir, Francesc Rossello, Lucia Rotger

Maintainer Lucia Rotger <lucia.rotger@uib.es>

Description Computation of Colless-Like, Sackin and cophenetic balance indices of a phylogenetic tree and study of the distribution of these balance indices under the alpha-gamma model.

License GPL (>= 2)

LazyData true

Depends R (>= 3.3.0)

Imports ape, igraph

RoxygenNote 6.0.1

Encoding UTF-8

NeedsCompilation no

R topics documented:

`CollessLike-package` *Methods to Compute Distribution and Percentile of 3 Balance Indices of Phylogenetic Trees*

Description

Computation of Colless-Like, Sackin and cophenetic balance indices of a phylogenetic tree and study of the distribution of these balance indices under the alpha-gamma model.

Details

Package:	CollessLike
Type:	Package
Title:	Methods to Compute Distribution and Percentile of 3 Balance Indices of Phylogenetic Trees
Version:	0.0.6911
Date:	2018-03-20
Author:	Arnau Mir, Francesc Rossello, Lucia Rotger
Maintainer:	Lucia Rotger <lucia.rotger@uib.es>
Description:	Computation of Colless-Like, Sackin and cophenetic balance indices of a phylogenetic tree and study of
License:	GPL (>= 2)
LazyData:	true
Depends:	R (>= 3.3.0)
Imports:	ape, igraph
RoxygenNote:	6.0.1
Encoding:	UTF-8

Index of help topics:

```
CollessLike-package     Methods to Compute Distribution and Percentile
                        of 3 Balance Indices of Phylogenetic Trees
a.g.model               Generates a random tree
balance.indices         Computes Colles-like, Sackin and cophenetic
                        balance indices of a phylogenetic tree.
colless.like.index      Computes the Colless-like balance index of a
                        phylogenetic tree
cophen.index            Computes the cophenetic balance index of a
                        phylogenetic tree
distribution            Plot the distribution of Colless-Like, Sackin
                        and cophenetic normalized balance indices under
                        the alpha-gamma model and computes the
                        percentile of a tree from previous
                        distributions.
indices.simulation      Generates random trees and compute their
                        balance indices
sackin.index            Computes the Sackin balance index of a
                        phylogenetic tree
```

Author(s)

Arnau Mir, Francesc Rossello, Lucia Rotger

Maintainer: Lucia Rotger <lucia.rotger@uib.es>

References

B. Chen, D. Ford, M. Winkel, A new family of Markov branching trees: the alpha-gamma model. *Electr. J. Probab.* **14** (2009), 400-430.

A. Mir, F. Rossello, L. Rotger, A Colless-like balance index for multifurcating phylogenetic trees.

A. Mir, F. Rossello, L. Rotger, A new balance index for phylogenetic trees. *Mathematical Biosciences* **241** (2013), 125-136.

M. J. Sackin, "Good" and "bad" phenograms. *Sys. Zool*, **21** (1972), 225-226.

Examples

```
# A random phylogenetic tree of 5 leaves between all trees with 5 leaves
# following the alpha-gamma model with alpha=0.5 and gamma=0.3
a.g.tree = a.g.model(5,0.5,0.3)
# To compute the percentile of that tree of the Colless-Like,
# Sackin and cophenetic normalized balance indices under the alpha-gamma
# model with alpha=0.5 and gamma=0.3, and a distribution plot.
#distribution(a.g.tree,0.5,0.3,db.path=getwd())
# For a percentile plot set the parameter percentile.plot as TRUE
#distribution(a.g.tree,0.5,0.3,db.path=getwd(),percentile.plot=TRUE)

# Computation of the Colless-Like, Sackin and cophenetic balance indices
# of a sample of 50 trees that follow the alpha-gamma distribution
# with alpha=0.5 and gamma=0.3 with 5 leaves.
  indices.data = indices.simulation(5,0.5,0.3,50)
# Computation of the percentile of the random tree using the previous
# generated sample
  distribution(a.g.tree,0.5,0.3,set.indices=indices.data)
```

`a.g.model` *Generates a random tree*

Description

Given alpha, gamma and the number of leaves n, generates a random phylogenetic tree between all trees with n leaves following the alpha-gamma model.

Usage

```
a.g.model(n, alpha, gamma)
```

Arguments

n	the number of leaves in the tree.
alpha	parametrer of the alpha-gamma model, between 0 and 1.
gamma	parametrer of the alpha-gamma model, between 0 and alpha.

Value

An igraph object that represents the generated phylogenetic tree.

Author(s)

Lucia Rotger

References

B. Chen, D. Ford, M. Winkel, A new family of Markov branching trees: the alpha-gamma model. *Electr. J. Probab.* **14** (2009), 400-430.

Examples

```
# A phylogenetic tree with 10 leaves and parameters alpha=0.8 and gamma=0.1
tree = a.g.model(10,0.8,0.1)
plot(tree,layout=layout.reingold.tilford(tree,root=which(degree(tree,mode="in")==0)))
```

---

balance.indices   *Computes Colles-like, Sackin and cophenetic balance indices of a phylogenetic tree.*

Description

Given a phylogenetic tree, computes Colles-like, Sackin and cophenetic balance indices of that tree.

Usage

```
balance.indices(tree, norm = FALSE)
```

Arguments

`tree`	a single phylogenetic tree. It can be entered as a string in Newick format, as a 'phylo' object (ape package) or as an 'igraph' object (igraph package).
`norm`	a logical variable that indicates whether the indices should be normalized or not.

Details

The Colless-like index is the generalization of the Colless' index for non-binary trees (see Mir et al. , 2017).

The Sackin's index is computed as the sum of the number of ancestors for each leave of the tree (see Mir et al. , 2013).

The cophenetic index is computed as the sum of the depths of the least common ancestor (LCA) of every pair of leaves of the tree(see Sackin et al, 1972).

Value

A numeric vector with the three computed balance indices of the tree: `Colless-like`, `Sackin` and `Cophenetic` values.

Author(s)

Lucia Rotger

References

A. Mir, F. Rossello, L.Rotger, A Colless-like balance index for multifurcating phylogenetic trees.

A. Mir, F. Rossello, L.Rotger, A new balance index for phylogenetic trees. *Mathematical Biosciences* **241** (2013), 125-136.

M. J. Sackin, "Good" and "bad" phenograms. *Sys. Zool*, **21** (1972), 225-226.

Examples

```
# Computation of the Colless-Like, Sackin and Cophenetic
# balance indices of trees entered in newick format:
balance.indices("(1,2,3,4,5);")
balance.indices("(1,(2,(3,(4,5))));")

# Computation of the Colless-Like, Sackin and Cophenetic
# balance indices of a tree entered as a phylo object:
require(ape)
random.tree = rtree(5,rooted=TRUE)
balance.indices(random.tree)

# Computation of the Colless-Like, Sackin and Cophenetic
# balance indices of a tree entered as a igraph object.
# The tree is randomly generated from all trees with 5
# leaves following the alpha-gamma model with alpha=0.5
# and gamma=0.3.
a.g.tree = a.g.model(5,0.5,0.3)
balance.indices(a.g.tree)

# All of them can be normalized (values between 0 and 1)
balance.indices("(1,2,3,4,5);",norm=TRUE)
balance.indices("(1,(2,(3,(4,5))));",norm=TRUE)
balance.indices(random.tree,norm=TRUE)
balance.indices(a.g.tree,norm=TRUE)
```

`colless.like.index` *Computes the Colless-like balance index of a phylogenetic tree*

Description

Given a phylogenetic tree, computes the Colless-like balance index of that phylogenetic tree.

Usage

```
colless.like.index(tree, f.size = "ln", diss = "MDM", norm = FALSE)
```

Arguments

`tree`	a single phylogenetic tree. It can be entered as a string in Newick format, as a 'phylo' object (ape package) or as an 'igraph' object (`igraph` package).
`f.size`	function to compute the f-size of the tree. See (Mir et al. , 2017) for details. Its default value is "ln" for f(n)=ln(n+e). Other value can be "exp" (f(n)=exp(n)). It can also be a user-defined function but in this case, the index cannot be normalized

`diss`	by default, the dissimilarity used to compute the balance index. See (Mir et al. , 2017) for details. Its default value is MDM (mean deviation from the median). Other values can be set as "sd" (sample standard deviation) or "var" (sample variance) . It can also be a user-defined function but in this case the index cannot be normalized.
`norm`	a logical object indicating if the indices should be normalized or not.

Details

The Colless-Like balance index is the generalization of the Colless balance index (see Colless,1982) for non-binary trees.

Given a function that computes the f-size of a tree and a dissimarity function that computes the difference of the f-sizes of the subtrees rooted at the children of every internal node of the tree, the Colless-Like index is defined as the sum of these dissimilarities for all internal nodes of the tree. (Mir et al. , 2017)

By default, the f-size function is `f(n)=exp(n)` and the dissimilarity is the mean deviation from the median (MDM). It is possible to change them by specifying it with the parameters `f.size` and `diss`, with "exp" the f-size would be `f(n)=exp(n)`, and with "var" (or "sd") the dissimilarity would be the sample variance (or the sample standard deviation). It is also possible to set a new function for both parameters, see "References".

Value

A numeric value.

Author(s)

Lucia Rotger

References

A. Mir, F. Rossello, L.Rotger, A Colless-like balance index for multifurcating phylogenetic trees.

D. H. Colless, Review of "Phylogenetics: the theory and practice of phylogenetic systematics". *Sys. Zool*, **31** (1982), 100–104.

Examples

```
# Computation of the Colless-Like balance index of trees
# entered in newick format:
colless.like.index("(1,2,3,4,5);")
colless.like.index("(1,(2,(3,(4,5))));")

# Computation of the Colless-Like balance index of trees
# entered as a phylo object:
require(ape)
random.tree = rtree(5,rooted=TRUE)
colless.like.index(random.tree)

# Computation of the Colless-Like balance index of a tree
# entered as a igraph object. The tree is randomly
# generated from all trees with 5 leaves following
# the alpha-gamma model with alpha=0.5 and gamma=0.3.
a.g.tree = a.g.model(5,0.5,0.3)
colless.like.index(a.g.tree)
```

```
# All of them can be normalized (values between 0 and 1)
colless.like.index("(1,2,3,4,5);",norm=TRUE)
colless.like.index("(1,(2,(3,(4,5))));",norm=TRUE)
colless.like.index(random.tree,norm=TRUE)
colless.like.index(a.g.tree,norm=TRUE)

# Computation of the Colless-Like balance index of the
# previous generated tree with f-size function f(n)=exp(n):
colless.like.index(a.g.tree,f.size="exp")

# Computation of the Colless-Like balance index of the
# previous generated tree that sets the sample variance
# and the sample standard deviation as dissimilarity.
colless.like.index(a.g.tree,diss="var")
colless.like.index(a.g.tree,diss="sd")

# Computation of the Colless-Like balance index of the
# previous generated tree with f-size function f(n)=exp(n)
# that sets the sample variance and the sample standard
# deviation as dissimilarity.
colless.like.index(a.g.tree,f.size="exp",diss="var")
colless.like.index(a.g.tree,f.size="exp",diss="sd")
```

`cophen.index` *Computes the cophenetic balance index of a phylogenetic tree*

Description

Given a phylogenetic tree, computes the cophenetic balance index of that phylogenetic tree.

Usage

```
cophen.index(tree, norm = FALSE)
```

Arguments

`tree`	a single phylogenetic tree. It can be entered as a string the Newick format, as a 'phylo' object (ape package) or as an 'igraph' object (`igraph` package).
`norm`	a logical variable that indicates whether the index should be normalized or not.

Details

The cophenetic index is computed as the sum of the depths of the least common ancestor (LCA) of every pair of leaves.

Value

A numeric value.

Author(s)

Lucia Rotger

References

A. Mir, F. Rossello, L.Rotger, A new balance index for phylogenetic trees. *Mathematical Biosciences* **241** (2013), 125-136.

Examples

```
# Computation of the cophenetic balance index of trees
# entered in newick format:
cophen.index("(1,2,3,4,5);")
cophen.index("(1,(2,(3,(4,5))));")

# Computation of the cophenetic balance index of trees
# entered as a phylo object:
require(ape)
random.tree = rtree(5,rooted=TRUE)
cophen.index(random.tree)

# Computation of the cophenetic balance index of a tree
# entered as a igraph object. The tree is randomly
# generated from all trees with 5 leaves following
# the alpha-gamma model with alpha=0.5 and gamma=0.3.
a.g.tree = a.g.model(5,0.5,0.3)
cophen.index(a.g.tree)

#All of them can be normalized (values between 0 and 1)
cophen.index("(1,2,3,4,5);",norm=TRUE)
cophen.index("(1,(2,(3,(4,5))));",norm=TRUE)
cophen.index(random.tree,norm=TRUE)
cophen.index(a.g.tree,norm=TRUE)
```

`distribution` *Plot the distribution of Colless-Like, Sackin and cophenetic normalized balance indices under the alpha-gamma model and computes the percentile of a tree from previous distributions.*

Description

Given alpha, gamma and a phylogenetic tree, plot the distribution of the Colless-Like, Sackin and cophenetic normalized balance indices under the alpha-gamma model and computes the percentile of that tree of the previous normalized balance indices under the alpha-gamma model.

Usage

```
distribution(tree, alpha = NA, gamma = NA, set.indices = NULL,
  new.simulation = FALSE, repetitions = 1000,
  legend.location = "topright", cex = 0.75, percentile.plot = FALSE,
  db.path = getwd())
```

Arguments

`tree`	a single phylogenetic tree. It can be entered as a string in Newick format, as a "phylo" object (ape package) or as an "igraph" object (igraph package).
`alpha`	parametrer of the alpha-gamma model, between 0 and 1.
`gamma`	parametrer of the alpha-gamma model, between 0 and alpha.
`set.indices`	If `NULL` (default) the values of the balance indices are taken from stored data or from a new simulated data (See "Details"). If not, it must be a 3-column data.frame with the three balance indices (Colles-like, Sackin, Cophenetic). See `indices.simulation`.
`new.simulation`	if `FALSE`(default) the values of the balance indices are taken from a data.frame entered by the user or from our database. If `TRUE`, the values of the balance indices are computed from a new simulation. See `indices.simulation`.
`repetitions`	if the value of the new.simulation parameter is `TRUE`, the number of trees to be generated.
`legend.location`	location of the legend. See "Details".
`cex`	expansion factor of the legend. See "Details".
`percentile.plot`	if `FALSE` (default), density plots of the normalized balance indices are shown. if `TRUE`, percentiles plots of the normalized balance indices are shown.
`db.path`	the current working directory. If our database is used, the db.path parameter should be the directory where the database is located.

Details

Two plots are available: one represents the percentile plots of the normalized balance indices (`percentile.plot=TRUE`), and the other one represents the density plots of the normalized balance indices (`percentile.plot=FALSE`).

The trees stored in our database have between 3 and 50 leaves and the values of the parameters `alpha` and `gamma` are in {0,0.1,...,1} such that `gamma` $\leq$ `alpha`. If the introduced parameters are not in the list, a new computation is done with them and a new dataset of trees is generated, and their indices are also computed. The number of trees generated can be modified by the parameter `repetitions` (see `indices.simulation` for more information). This computation may take some time, therefore you can computate them separately with `indices.simulation`, save their values and then call this function by setting the parameter `set.indices=NULL`.

The legend is placed with the `graphics` function `legend()`, so its location can be specified by setting `legend.position` to a single keyword from the list `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`, `"top"`, `"topright"`, `"right"` and `"center"`. The expansion factor for the legend is controlled by the parameter `cex`, by default `cex=1`. See `legend`.

Value

A numeric vector with the three percentiles.

Author(s)

Lucia Rotger

References

B. Chen, D. Ford, M. Winkel, A new family of Markov branching trees: the alpha-gamma model. *Electr. J. Probab.* **14** (2009), 400-430.

A. Mir, F. Rossello, L.Rotger, A Colless-like balance index for multifurcating phylogenetic trees.

A. Mir, F. Rossello, L.Rotger, A new balance index for phylogenetic trees. *Mathematical Biosciences* **241** (2013), 125-136.

M. J. Sackin, "Good" and "bad" phenograms. *Sys. Zool*, **21** (1972), 225-226.

See Also

`legend`, `indices.simulation`, `balance.indices`

Examples

```
#The parameter folder contains the location of the database
#If not specified folder=getwd()

## Different ways to introduce the tree
#From a newick string
distribution("(1,2,3,4,5);",0.5,0.3,db.path=folder)
distribution("(1,(2,(3,(4,5))));",0.5,0.3,db.path=folder)

#From a phylo object
require(ape)
random.tree = rtree(5,rooted=TRUE)
distribution(random.tree,0.5,0.3,db.path=folder)

#An example of a tree generated by the alpha-gamma model (igraph object)
a.g.tree = a.g.model(5,0.5,0.3)
distribution(a.g.tree,0.5,0.3,db.path=folder)

## Different indices data
# From our data base
distribution(a.g.tree,0.5,0.3,db.path=folder)

# From a data.frame generated by 'indices.simulation'
# ('Repetitions' set as 10 for a fast example)
indices.data = indices.simulation(5,0.5,0.3,10)
distribution(a.g.tree,0.5,0.3,set.indices=indices.data)

# Allow the function to do a new generation of data and compute their indices
distribution(a.g.tree,0.5,0.3,new.simulation=TRUE,repetitions=10)
# WARNING! it might take a long time, it depends on the parameters
# 'n' (number of leaves) and 'repetition' (number of repetitions)
```

`indices.simulation` *Generates random trees and compute their balance indices*

Description

Generates a list of trees according to the introduced parameters for the alpha-gamma model. Then, this 3 balance index are calulated: Colless-like, Sackin and Cophenetic.

Usage

```
indices.simulation(n, alpha = NA, gamma = NA, repetitions = 1000,
  norm = FALSE)
```

Arguments

`n`	the number of leaves of the tree.
`alpha`	parametrer of the alpha-gamma model, between 0 and 1.
`gamma`	parametrer of the alpha-gamma model, between 0 and alpha.
`repetitions`	the number of trees to generate.
`norm`	a logical object indicating if the indices should been normalized or not.

Details

Given a number of leaves, the function generates a tree with that number of leaves and computates the three indeces of balance (Colles-like, Sackin and Cophenetic with function `balance.indices`). This is done as many times as it is set by 'repetitions' parameter, and it generates a 3-column data.frame of indices.

The trees are generated according to the alpha-gamma model. These parameters can be specified by `alpha` and `gamma` parameters of the function. The following cases are distinguished:

- `alpha = NA` and `gamma = NA` : All the 66 combinations of `alpha` in { 0, 0.1, 0.2, ... ,0.9, 1 } and `gamma` in { 0, 0.1, ... ,`alpha` } are done.
- `alpha` in [0,1] and `gamma = NA` : Since `alpha` is fixed, all the combinations with that `alpha` and `gamma` in { 0, 0.1, ... ,`alpha` } are done.
- `alpha` in [0,1] and `gamma` in [0,`alpha`] : Both parameters are fixed. Then, only that combination is done.

Value

A 3-column data.frame with the Colless-like, Sackin and Cophenetic balance indices for every generated tree. If more than one data.frame has to be generated, then the returned value is a data.frame list (its names specify which alpha and gamma parameters have generated that data.frame, for instance "a0.5g0.3" indicates alpha=0.5 and gamma=0.3).

Author(s)

Lucia Rotger

References

B. Chen, D. Ford, M. Winkel, A new family of Markov branching trees: the alpha-gamma model. *Electr. J. Probab.* **14** (2009), 400-430.

A. Mir, F. Rossello, L.Rotger, A Colless-like balance index for multifurcating phylogenetic trees.

A. Mir, F. Rossello, L.Rotger, A new balance index for phylogenetic trees. *Mathematical Biosciences* **241** (2013), 125-136.

M. J. Sackin, "Good" and "bad" phenograms. *Sys. Zool*, **21** (1972), 225-226.

See Also

`balance.indices`

Examples

```
#('Repetitions' set as 100 for a fast example)
indices.table = indices.simulation(5,0.5,0.3,repetitions=10)
head(indices.table)

#Normalized indices (between 0 and 1)
indices.table = indices.simulation(5,0.5,0.3,repetitions=10,norm=TRUE)
head(indices.table)

#Without specifying alpha and gamma
indices.list = indices.simulation(5,repetitions=100)
#by default alpha=seq(0,1,0.1) and gamma=seq(0,alpha,0.1), thus
length(indices.list) #=66
#all the elements of the list have a name that identifies its parameters
indices.list$a0.5g0.3
indices.list$a0.7g0.2
```

`sackin.index` *Computes the Sackin balance index of a phylogenetic tree*

Description

Given a phylogenetic tree, computes the Sackin balance index of that phylogenetic tree.

Usage

```
sackin.index(tree, norm = FALSE)
```

Arguments

`tree`	a single phylogenetic tree. It can be entered as a string in Newick format, as a 'phylo' object (ape package) or as an 'igraph' object (igraph package).
`norm`	a logical variable that indicates whether the index should be normalized or not.

Details

The Sackin's index is computed as the sum of the number of ancestors for each leave of the tree.

Value

numeric value.

Author(s)

Lucia Rotger

References

M. J. Sackin, "Good" and "bad" phenograms. *Sys. Zool*, **21** (1972), 225-226.

Examples

```
# Computation of the Sackin balance index of trees
# entered in newick format:
sackin.index("(1,2,3,4,5);")
sackin.index("(1,(2,(3,(4,5))));")

# Computation of the Sackin balance index of trees
# entered as a phylo object:
require(ape)
random.tree = rtree(5,rooted=TRUE)
sackin.index(random.tree)

# Computation of the Sackin balance index of a tree
# entered as a igraph object. The tree is randomly
# generated from all trees with 5 leaves following
# the alpha-gamma model with alpha=0.5 and gamma=0.3.
a.g.tree = a.g.model(5,0.5,0.3)
sackin.index(a.g.tree)

#All of them can be normalized (values between 0 and 1)
sackin.index("(1,2,3,4,5);",norm=TRUE)
sackin.index("(1,(2,(3,(4,5))));",norm=TRUE)
sackin.index(random.tree,norm=TRUE)
sackin.index(a.g.tree,norm=TRUE)
```

Index