Package ‘Apoderoides’

June 19, 2024

Type Package

Title Prioritize and Delete Erroneous Taxa in a Large Phylogenetic Tree

Version 2.0.1

Date 2024-06-17

Description Finds, prioritizes and deletes erroneous taxa in a phylogenetic tree. This package calculates scores for taxa in a tree. Higher score means the taxon is more erroneous. If the score is zero for a taxon, the taxon is not erroneous. This package also can remove all erroneous taxa automatically by iterating score calculation and pruning taxa with the highest score.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL `https://github.com/Sa-to-shi-A-o-ki/Apoderoides`

BugReports `https://github.com/Sa-to-shi-A-o-ki/Apoderoides/issues`

Depends R (>= 3.5.0)

Imports ape, Rcpp, RcppProgress

LinkingTo Rcpp,RcppProgress

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Satoshi Aoki [aut, cph, cre]

Maintainer Satoshi Aoki <`aokis1ll1@gmail.com`>

Repository CRAN

Date/Publication 2024-06-19 07:30:22 UTC

Contents

`Apoderoides-internal` *Internal Apoderoides Functions*

Description

Internal Apoderoides functions

Details

These are not to be called by the user.

Value

Different values, depending on the function.

`autoDeletion` *autoDeletion*

Description

Iterate calc.Score() and deleteAnomaly() until all the tree tips have 0 score or the number of the tips becomes three or lower.

Usage

```
autoDeletion(tree,OTUrankData=NULL,show_progress=TRUE,num_threads=1,prior="MRCA")
```

Arguments

`tree`	A phylogenetic tree to be checked. This is loaded by ape::read.tree() from a file.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience, and calculates for genera. When this is not NULL, the function calculates based on the upper rank in this list.
`show_progress`	If TRUE, calculation progress is shown on the R console.
`num_threads`	A positive integer to specify the number of threads to calculate.
`prior`	"MRCA" or "centroid". This argument defines the prioritized score when scores based on MRCA and centroid are equal.

Value

A list of the length four. The first element is a list of phylogenetic tree from which erroneous taxa are deleted. The second is a character vector of deleted taxa. The third and fourth are a list of lists showing the transition of the scores based on the centroids and MRCAs, respectively. See calc.Score about the contents of the third and fourth elements.

Examples

```
data(testTree)
data(testRankList)
#calculate scores for the rank in the list, and delete all the erroneous tips
#this takes tens of seconds for calculation
result<-autoDeletion(testTree,testRankList)
#tree without erroneos tips
result[[1]]
#deleted tips
result[[2]]
#scores based on the centroids during iteration of score calculation and tip deletion
result[[3]]
#scores based on the MRCAs during iteration of score calculation and tip deletion
result[[4]]
```

`calc.Score` *calc.Score*

Description

Calculate scores of a phylogenetic tree to find and prioritize erroneous taxa to delete.

Usage

```
calc.Score(tree,OTUrankData=NULL,
allRankNames=NULL,allCores=NULL,dropIndex=NULL,
sort=TRUE,show_progress=TRUE,num_threads=1)
```

Arguments

`tree`	A phylogenetic tree to be checked. This is loaded by ape::read.tree() from a file.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience, and calculate the score for genera. When this is not NULL, the function returns scores based on the upper rank in this list.
`allRankNames`	This can be omitted. This is a unique character vector of the upper ranks of the tree tips. If given, the calculation will be a little faster.
`allCores`	This can be omitted. This is a list of numeric vectors of the centroids or MRCAs of ranks. If given, the calculation will be a little faster.
`dropIndex`	This can be omitted. A numeric vector of indices of tree tips. The tree tips indicated by this dropIndex will be removed from the score calculation.
`sort`	If TRUE, the calculation result is sorted by descending order of the total score.
`show_progress`	If TRUE, calculation progress is shown on the R console.
`num_threads`	A positive integer to specify the number of threads to calculate the scores.

Value

A list containing two matrices of characters. The first one is the score based on the centroids, and the second is that based on the MRCAs. The following explains the columns in the matrix.

`OTU`	The name of tree tip.
`perCladeOTUScore`	The final score calculated by "sum" divided by the number of OTUs with the same "#clade".
`sum`	The sum of "intruder" and "outlier" for the OTU.
`intruder`	The intruder score showing how many ranks the OTU intruding into.
`outlier`	The outlier score showing how the OTU is far away from the core clade of the belonging rank.
`#clade`	The clade number. Monophyletic OTUs with the same rank has the same #clade.

Examples

```
data(testTree)
#calculate scores for genus
calc.Score(testTree)
data(testRankList)
#calculate scores for the rank in the list
calc.Score(testTree,testRankList)
```

`deleteAnomaly` *deleteAnomaly*

Description

Delete tip(s) with the highest score from a tree.

Usage

```
deleteAnomaly(tree,scores,OTUrankData=NULL,drop=FALSE,prior="MRCA")
```

Arguments

`tree`	A phylogenetic tree to be checked. This is loaded by ape::read.tree() from a file.
`scores`	A list of scores calculated by calc.Score function.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience and that the score is calculated based on genera. When this is not NULL, the function assumes the score is calculated based on the upper rank in this list.
`drop`	Whether the dropped OTU(s) is included in the returned tree.
`prior`	"MRCA" or "centroid". This argument defines the prioritized score when scores based on MRCA and centroid are equal.

Value

A list of the length two. The first element is a vector of characters of deleted tip label(s). The second is a list of a phylogenetic tree without the deleted tip(s).

Examples

```
data(testTree)
data(testRankList)
#calculate scores for the rank in the list
score<-calc.Score(testTree,testRankList)
#delete tip with the highest score from tree
deleteAnomaly(testTree,score,testRankList)
```

`get.upperRank` *get.upperRank*

Description

Obtain upper rank of scientific names in data. When OTUrankData is not provided, this function returns genus names assuming the elements in data are scientific names connected by underlines like "Homo_sapiens". When OTUrankData is provided, this function searches data in OTUrankData[[1]] and returns OTUrankData[[2]] of the corresponding index.

Usage

```
get.upperRank(data,OTUrankData=NULL)
```

Arguments

`data`	A vector of characters.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience, and calculate the score for genera. When this is not NULL, the function returns scores based on the upper rank in this list.

Value

A vector of characters of upper rank.

Examples

```
#obtain genus name
get.upperRank(c("Oxalis_nipponica","Homo_sapiens"))
data(testTree)
data(testRankList)
#obtain higher rank names
get.upperRank(testTree$tip[1:3],testRankList)
```

`getAllCentroids` *getAllCentroids*

Description

Calculate all the centroids of ranks in the tree. The centroid of a rank is equivalent to S-centroid by Slater (1978).

Usage

```
getAllCentroids(tree,OTUrankData=NULL,show_progress=FALSE,num_threads=1)
```

Arguments

`tree`	A phylogenetic tree to be checked. This is loaded by ape::read.tree() from a file.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience, and calculate the centroids for genera. When this is not NULL, the function returns centroids based on the upper rank in this list.
`show_progress`	If TRUE, calculation progress is shown on the R console.
`num_threads`	A positive integer to specify the number of threads to calculate the scores.

Value

A list containing vectors of integers of centroid node number(s).

References

Slater P. J. 1978. Centers to centroids in graphs. Journal of Graph Theory 2: 209–222.

Examples

```
data(testTree)
#calculate centroids for genus
getAllCentroids(testTree)
data(testRankList)
#calculate centroids for the rank in the list
getAllCentroids(testTree,testRankList)
```

`getAllMRCAs` *getAllMRCAs*

Description

Calculate all the most recent common ancestors (MRCAs) of ranks in the tree. Unlike getMRCA() in ape package, this function returns a tip node number when the rank is monotypic.

Usage

```
getAllMRCAs(tree,OTUrankData=NULL)
```

Arguments

`tree`	A phylogenetic tree to be checked. This is loaded by ape::read.tree() from a file.
`OTUrankData`	A list composed of two character vectors. The first vector is tips of tree. The second vector is the upper rank of the tips. When this is NULL, the function assumes that all the tree tips are expressed as Genus_species like Homo_sapience, and calculate the MRCAs for genera. When this is not NULL, the function returns MRCAs based on the upper rank in this list.

Value

A list containing vectors of an MRCA node number.

Examples

```
data(testTree)
#calculate MRCAs for genus
getAllMRCAs(testTree)
data(testRankList)
#calculate MRCAs for the rank in the list
getAllMRCAs(testTree,testRankList)
```

`testRankList` *testRankList*

Description

Example data to test Apoderoides. testRankList is a list of two elements. The first element is the tip label of testTree, and the second element is corresponding family names of the tips.

Usage

```
data(testRankList)
```

`testTree` *testTree*

Description

Example data to test Apoderoides. testTree is a tree of land plants based on chlB gene.

Usage

```
data(testTree)
```

Index