Package 'MonoPhy'

July 21, 2025

Type Package

Title Explore Monophyly of Taxonomic Groups in a Phylogeny

Version 1.3.2

Date 2024-10-16

Description Requires rooted phylogeny as input and creates a table of genera, their monophyly-status, which taxa cause problems in monophyly etc. Different information can be extracted from the output and a plot function allows visualization of the results in a number of ways. ``MonoPhy: a simple R package to find and visualize monophyly issues." Schwery, O. & O'Meara, B.C. (2016) <doi:10.7717/peerj-cs.56>.

Depends ape, phytools

Imports phangorn, RColorBrewer

License GPL-3

Suggests knitr, testthat, paleotree, rmarkdown, taxize

VignetteBuilder knitr

NeedsCompilation no

Author Orlando Schwery [aut, cre],
Brian C. O'Meara [aut, ctb],
Peter Cowman [ctb]

Maintainer Orlando Schwery <schwery.macroevo@pm.me>

Repository CRAN

Date/Publication 2024-10-17 11:40:07 UTC

Contents

MonoPhy-package *Explore Monophyly of Taxonomic Groups in a Phylogeny.*

Description

Requires rooted phylogeny as input and creates a table of taxa, their monophyly-status, which other taxa cause problems in monophyly (as intruders or outliers) etc. Different information can be extracted from the output and a plot function allows visualization of the results in a number of ways.

Details

Package:	MonoPhy
Type:	Package
Version:	1.3
Date:	2021-02-15
License:	GPL-3

The package allows to explore monophyly in phylogenetic trees in a quick and simple way. AssessMonophyly determines whether taxa in a phylogeny are monophyletic. CollapseMonophyletics creates a tree objects where monophyletic groups are collapsed, GetSummaryMonophyly extracts a summary table from the results, GetResultMonophyly extracts the main result table, GetIntruderTaxa, GetIntruderTips, GetOutlierTaxa and GetOutlierTips extract names of taxa which interfere with the monophyly of another or the same taxon respectively and GetAncNodes a table of MRCA nodes. PlotMonophyly finally, allows visualizing the results in a couple of different ways.

Author(s)

Orlando Schwery, Brian O'Meara, Peter Cowman

Maintainer: Orlando Schwery <oschwery@vols.utk.edu>

See Also

AssessMonophyly, CollapseMonophyletics, GetAncNodes, GetIntruderTaxa, GetIntruderTips, GetOutlierTaxa, GetOutlierTips GetResultMonophyly, GetSummaryMonophyly, MonophylyData, PlotMonophyly

AssessMonophyly *Assesses monophyly (or lack of it) of taxonomic groups in a phylogeny.*

Description

Requires rooted phylogeny as input and creates a table of taxa, their monophyly-status, which taxa cause problems in monophyly (as intruders or outliers) etc. Output can be accessed with related functions (see section 'see also') and a plot function can be used to plot the results in a number of ways.

Usage

```
AssessMonophyly (tree, taxonomy=NULL, verbosity=5, outliercheck=TRUE,
outlierlevel=0.5, taxizelevel= NULL, taxizedb='ncbi', taxizepref='ncbi',
taxask=FALSE, taxverbose=FALSE)
```

Arguments

tree	An object of type 'phy', a rooted phylogeny. Multifurcating trees are accepted, but will be dealt with in a conservative manner (i.e. if different taxa share a multifurcation, they will be considered non-monophyletic) If tip labels are in the format 'genus_species', the function can extract the genus names and check their monophyly. If tip labels are in another format or if the monophyly of other taxonomic groups should be tested, a taxonomy file (see 'taxonomy') is required.
taxonomy	A data frame (e.g. an imported .csv file) with at least two columns and one row per tip in the tree. If a header is specified, the names therein will be used in the function output. Column one contains the tip labels, column two and higher the names of the taxonomic units the respective tip belongs to. The order of tip names in the file can be different from the order of the tip labels in the tree, but they have to contain the exact same names. If taxonomic levels are unknown for certain tips, they can be coded as NAs (but they cannot stay empty). Those tips will be considered when assessing monophyly of other groups, but the monophyly of the NA group will not be assessed. Alternatively, specifying taxonomy as 'taxize' allows to use the package with the same name to download taxonomic names from online databases. Default is NULL.
verbosity	An integer, default is 5. Determines how many outlier/intruder taxa should be listed by name in the result table (it will list up to this many names and then add 'and X more').
outliercheck	If TRUE (default), the function will differenciate intruders or outliers as cause for non-monophyly. If the descendants of a taxon's MRCA contain less actual members of that taxon than specified under 'outlierlevel', the function will try to find a 'core clade', which is above this threshold. Taxon members outside of this core clade are then considered outliers and only the intruders within the core clade will actually be scored as intruders.

`outlierlevel` If 'outliercheck' is set to TRUE, this argument defines the threshold for the outlier search. If a clade contains a fraction of actual taxon members that is lower than the fraction specified here, outliers will be defined. Enter a value between 0 and 1, default is 0.5.

`taxizelevel` If taxonomy is set to 'taxize', it will download the taxonomic level specified here. Default is NULL.

`taxizedb` If taxonomy is set to 'taxize', the desired taxonomic levels from the specified database. Either 'itis' or 'ncbi' or 'both'; is default is 'ncbi'. If 'both' is chosen, double entries will be discarded and the database result with did actually yeld an entry will be kept. Note that using 'both' can lead to conflicts within 'taxize'.

`taxizepref` If taxonomy is set to 'taxize' and taxizedb to 'both', either 'itis' or 'ncbi' can be set to be preferred if both databases have records for the tip in question. Default is 'ncbi'.

`taxask` If taxonomy is set to 'taxize', the called function 'tax_name' may find several potential matches in the queried databases. If 'taxask' is set to TRUE, the function will stop and prompt the user to pick an entry, if set to FALSE (which is default), it will pick the first entry and continue.

`taxverbose` If taxonomy is set to 'taxize', the called function 'tax_name' can display the progress, i.e. the names that are being queried and the success of the query. The default for this is FALSE, in which case nothing will be displayed while querrying.

Details

The function uses `getMRCA` from ape and `getDescendants` from phytools to determine if a genus or other taxonomic group is monophyletic or not and subsets the two to determine which taxa cause non-monophyly. From the result object, different output items (see Values) can be accessed using a set of related functions and the result can be visualized using `PlotMonophyly`.

Value

The output object of the function is a list containing the results as lists and data frames. If several taxonomic levels are analyzed, each will get a list-level containing its respective results. The different objects contained are:

`IntruderTaxa` List of the names of the taxonomic groups assessed to be non-monophyletic, each containing a character string with the names of the taxa which are intruders (i.e. interfere with the monophyly of that respective taxon). Can be accessed using `GetIntruderTaxa`.

`IntruderTips` List of the names of the taxonomic groups assessed to be non-monophyletic, each containing a character string with the names of species/tips which are intruders (i.e. interfere with the monophyly of that taxon). Can be accessed using `GetIntruderTips`.

`OutlierTaxa` Vector of the names of the taxonomic groups which were inferred to have outlier tips). Can be accessed using `GetOutlierTaxa`.

`OutlierTips` List of the names of the taxonomic groups assessed to be non-monophyletic, each containing a character string with the names of species/tips which were

inferred to be outliers (i.e. interfere with the monophyly of that taxon by being placed far from its core clade). Can be accessed using GetOutlierTips.

result — Data frame containing the main results. Rows are the taxonomic groups used, columns are 'Monophyly' ('Yes', 'No' or 'Monotypic'), 'MRCA' (node number of inferred ancestor), '#Tips' (number of tips assigned to this taxon),'Delta-Tips' (number of tips which share this ancestral node but do not belong to the same taxon), '#Intruders'(how many other taxa interfere with the monophyly of the taxon in quesiton) and 'Intruders' (names of interfering taxa, how many of these are written out is determined by the argument 'verbosity'). If the argument outliercheck was set to TRUE when running 'AssessMonophyly', the table will additionally include '#Outliers' (number of tips which are placed outside the core clade for that group) and 'Outliers' (names of outlier taxa, the number of which also depending on the argument 'verbosity'). The whole data frame can be accessed using GetResultMonophyly, the MRCA nodes can be accessed using GetAncNodes.

summary — Data frame containing two rows with the number of taxa (e.g. genera) and tips (e.g. species) for the total tree, and which were inferred to be monophyletic, non-monophyletic, monotypic, intruders and (if applicable) outliers. Can be accessed using GetSummaryMonophyly.

TipStates — Data frame with the columns 'Tip', 'Taxon' and 'Status', containing the tip labels, their associated taxa and monophyly status (monophyletic, non-monophyletic, intruder or outlier) respectively. This data frame will be used by the function PlotMonophyly.

Author(s)

Orlando Schwery

See Also

GetAncNodes, GetIntruderTaxa, GetIntruderTips, GetOutlierTaxa, GetOutlierTips,GetResultMonophyly, GetSummaryMonophyly, MonophylyData, PlotMonophyly, MonoPhy-package

Examples

```
data(Ericactree)  # load tree
solution <- AssessMonophyly(Ericactree)  # run analysis
GetSummaryMonophyly(solution)  # extract summary table from output

#use custom taxonomic level
data(Ericactree)  # load tree
data(Ericactribes)  # load taxonomy file
solutiontribes <- AssessMonophyly(Ericactree, taxonomy=Ericactribes)  # run analysis
GetSummaryMonophyly(solutiontribes)  # extract summary table from output
```

`CollapseMonophyletics` *Get tree with collapsed monophyletic groups*

Description

Creates a new tree object with all monophyletic groups collapsed to one tip each, based on the output of 'AssessMonophyly'.

Usage

```
CollapseMonophyletics(solution, tree, taxlevels = 1, ladderize = TRUE)
```

Arguments

`solution`	Object with saved output of the 'AssessMonophyly' function.
`tree`	An object of type 'phy', rooted phylogeny (the same which was used to obtain the solution before).
`taxlevels`	An integer or name corresponding to the desired taxonomic level (i.e. the number of its column in the taxonomy table, not counting the tip names). Default is 1.
`ladderize`	If TRUE will ladderize tree before collapsing clades. Default is TRUE.

Details

Can be used after 'AssessMonophyly' is run to obtain a tree object where all groups that were inferred as monophyletic at a given taxonomic level will be collapsed to single tips and relabeled according to the name of their group. The resulting tree will have the same topology as seen when plotting the original tree with setting 'monocoll=TRUE' in 'PlotMonophyly'.

Value

A tree object of type 'phy'.

Note

The code is largely the same as 'PlotMonophyly' uses to create a plot with collapsed monophyletic groups.

Author(s)

Orlando Schwery, Peter Cowman

See Also

`AssessMonophyly`, `PlotMonophyly`, `MonoPhy-package`

Examples

```
data(Ericactree)
solution <- AssessMonophyly(Ericactree, verbosity=5)
collapsetree <- CollapseMonophyletics(solution, Ericactree, taxlevels = 1, ladderize = TRUE)
```

GetAncNodes *Get MRCA nodes of taxonomic groups from 'AssessMonophyly' output*

Description

Print MRCA node numbers of taxonomic groups from the output of 'AssessMonophyly', either the whole list or for specific groups.

Usage

```
GetAncNodes(solution, taxa = NULL, taxlevels='ALL')
```

Arguments

solution	Object with saved output of the 'AssessMonophyly' function.
taxa	Vector containing taxon names (genera or whichever taxonomic unit was used). Default is 'NULL'
taxlevels	Either an integer corresponding to the desired taxonomic level (i.e. the number of its column in the taxonimy table, not counting the tip names), the column name in the header of the taxonomy file, or 'ALL' (which is the default).

Details

Can be used after 'AssessMonophyly' is run to extract MRCA nodes of taxa from it. The argument 'taxa' allows to limit the output to one or several taxa of interest. The argument 'taxlevels' allows to limit the output if several taxonomic levels were used; 'ALL' is default, a specific level can be selected by entering its number or name instead.

Value

List of data frames.

Author(s)

Orlando Schwery

See Also

AssessMonophyly, MonoPhy-package

Examples

```
data(Ericactree)
solution <- AssessMonophyly(Ericactree)
GetAncNodes(solution=solution, taxa=c("Phyllodoce", "Vaccinium", "Erica"))
```

`GetIntruderTaxa` *Get taxa or tips which are intruders or outliers to a taxon from AssessMonophyly*

Description

Prints names of taxa (genera or whichever taxonomic unit used) or tip names (species, most likely) interfering with the monophyly of one or several (or all) groups, either as intruders (sharing a clade with said taxon) or outliers (being placed outside of the taxon's core clade)

Usage

```
GetIntruderTaxa(solution, taxa = NULL, taxlevels='ALL')
GetIntruderTips(solution, taxa = NULL, taxlevels='ALL')
GetOutlierTaxa(solution, taxlevels='ALL')
GetOutlierTips(solution, taxa = NULL, taxlevels='ALL')
```

Arguments

`solution`	Object with saved output of the 'AssessMonophyly' function.
`taxa`	Vector containing taxon names (genus or whichever taxonomic unit was used). Default NULL will return intruders/outliers for all taxa.
`taxlevels`	Either an integer corresponding to the desired taxonomic level (i.e. the number of its column in the taxonimy table, not counting the tip names), the column name in the header of the taxonomy file, or 'ALL', which is the default.

Details

Can be used after 'AssessMonophyly' is run to extract intruder/outlier taxa or intruder/outlier tips from it. The argument 'taxa' allows to limit the output to one or several taxa of interest. It is not available for GetOutlierTaxa, since this is just one vector with names anyway (a taxon can only be outlier of itself). The argument 'taxlevels' allows to limit the output if several taxonomic levels were used; 'ALL' is default, a single level can be selected by entering its number or name instead. Outliers can of course only be retrieved if the argument outliercheck was set to TRUE when running AssessMonophyly.

Value

List of character strings

Note

Noteworthy difference: intruder tips for e.g. Vaccinium will be tips of OTHER taxa that are placed inside the Vaccinium clade, whereas outlier tips for Vaccinium will be Vaccinium species that are placed outside of the Vaccinium core clade. Can also be accessed by simply typing 'solution$Taxlevel_i$IntruderTaxa' or 'solution$Taxlevel_i$IntruderTaxa$taxonname' and 'solution$Taxlevel_i$IntruderTips' or 'solution$Taxlevel_i$IntruderTips$taxonname' respectively (or whichever name the Taxlevels might have been given, e.g. based on the header); same for outliers.

Author(s)

Orlando Schwery

See Also

AssessMonophyly, GetAncNodes, MonoPhy-package

Examples

```
data(Ericactree)
solution <- AssessMonophyly(Ericactree)
GetIntruderTaxa(solution=solution)
GetOutlierTaxa(solution=solution)
GetIntruderTips(solution=solution, taxa=c("Phyllodoce", "Vaccinium", "Erica"))
GetOutlierTips(solution=solution, taxa=c("Vaccinium"))
```

GetResultMonophyly *Get main result table from 'AssessMonophyly' output*

Description

Extracts data frame with main result table or containing summary information such as number of tips, number of monophyletic taxa, etc. from the output of 'AssessMonophyly'.

Usage

```
GetResultMonophyly(solution, taxlevels='ALL')
GetSummaryMonophyly(solution, taxlevels='ALL')
```

Arguments

solution	Object with saved output of the 'AssessMonophyly' function.
taxlevels	Either an integer corresponding to the desired taxonomic level (i.e. the number of its column in the taxonimy table, not counting the tip names), the column name in the header of the taxonomy file, or 'ALL'.

Details

Can be used after 'AssessMonophyly' is run to extract the results or summary table from it. In the results table, rows are the taxonomic groups used, columns are 'Monophyly' ('Yes', 'No' or 'Monotypic'), 'MRCA' (node number of inferred ancestor), '#Tips' (number of tips assigned to this taxon),'Delta-Tips' (number of tips which share this ancestral node but do not belong to the same taxon), '#Intruders'(how many other taxa interfere with the monophyly of the taxon in quesiton) and 'Intruders' (names of interfering taxa, how many of these are written out is determined by the argument 'verbosity'). If the argument outliercheck was set to TRUE when running 'AssessMonophyly', the table will additionally include '#Outliers' (number of tips which are placed outside the core clade for that group) and 'Outliers' (names of outlier taxa, the number of which also depending on the argument 'verbosity'). The summary table contains two rows with the number of taxa (e.g. genera) and tips (e.g. species) for the total tree, and which were inferred to be monophyletic, non-monophyletic, monotypic, intruders and (if applicable) outliers. The argument 'taxlevels' allows to limit the output if several taxonomic levels were used; 'ALL' is default, a specific level can be selected by entering its corresponding number instead.

Value

List of data frame(s) containing the main results/summary.

Note

Can also be accessed by simply typing 'solution$Taxlevel_i$result' or 'solution$Taxlevel_i$summary' (or whichever name the Taxlevels might have been given, e.g. based on the header) respectively.

Author(s)

Orlando Schwery

See Also

AssessMonophyly, MonoPhy-package

Examples

```
data(Ericactree)
solution <- AssessMonophyly(Ericactree, verbosity=5)
GetSummaryMonophyly(solution)
GetResultMonophyly(solution)
```

MonophylyData *Example dataset for the package MonoPhy.*

Description

Example dataset for the package MonoPhy. One phylogeny and two custom taxonomy files.

Usage

```
data(Ericactree)
data(Ericactribes)
data(Ericacsubfams)
```

Details

Ericactree is a phylogeny of the angiosperm family Ericaceae, as presented in Schwery et al. (2014), pruned to 77 species. Ericactribes is an example file for how to specify which taxonomic groups should be tested for monophyly (instead of genus names taken from the tip labels of the tree), here exemplified by tribes. In Ericacsubfams we have two columns, one with tribes and one with subfamilies. For the sake of the example, some taxa have been intentionally mislabelled in the taxonomy files: the tribes of the tips 'Vaccinium_vitis_ideae' and 'Zenobia_pulverulenta' have been swapped and 'Rhodothamnus_chamaecystus' was labelled to be a member of the 'Vaccinioideae'.

References

Schwery, O., Onstein, R.E., Bouchenak-Khelladi, Y., Xing, Y., Carter, R.J. and Linder, H.P. (2014), As old as the mountains: the radiations of the Ericaceae. New Phytologist. doi: 10.1111/nph.13234

See Also

MonoPhy-package

PlotMonophyly *Plot output of AssessMonophyly*

Description

Allows plotting different visualisations of the results of AssessMonophyly, based on its output

Usage

```
PlotMonophyly(solution, tree, taxlevels=1, plot.type='monophyly',
monocoll=FALSE, ladderize=TRUE, PDF=FALSE, PDF_filename='Monophylyplot.pdf',
PDF_width='auto', PDF_height='auto', mono.colour='PRGn',
tax.colour='rainbow', intrud.colour='rainbow', edge.width=3, cex=0.2,
adj.names='auto', adj.tips=0.5, label.offset='auto', type='phylogram', ...)
```

Arguments

solution	Object with saved output of the 'AssessMonophyly' function.
tree	An object of type 'phy', rooted phylogeny (the same which was used to obtain the solution before).
taxlevels	An integer or name corresponding to the desired taxonomic level (i.e. the number of its column in the taxonomy table, not counting the tip names). Default is 1.

plot.type	Either "monophyly", "taxonomy", "intruders" or "monoVStax", see details. Default is "monophyly".
monocoll	If TRUE will collapse all monophyletic taxa to one single tip labeled with the taxon name. Default is FALSE.
ladderize	If TRUE will ladderize tree before reconstruction and plotting. Default is TRUE.
PDF	If TRUE will print a pdf with lenght adjusted to number of tips instead of plotting in R. Default is FALSE.
PDF_filename	Used to customize the name of pdf file created when PDF=TRUE. Default is 'Monophylyplot.pdf'.
PDF_width	Used to customize the width of pdf file created when PDF=TRUE. Default is 'auto', which will scale the width automatically based on the size of the tree.
PDF_height	Used to customize the height of pdf file created when PDF=TRUE. Default is 'auto', which will scale the height automatically based on the size of the tree.
mono.colour	Colour vector for plot type "monophyly" (and left side of "monoVStax"). Default is "PRGn", which colours monophyletic taxa green, non-monophyletic (intruded) taxa light purple and intruders purple using colours as specified in the corresponding ColorBrewer palette. Other predefined palettes are 'RdBu' (blue and red), 'PuOr' (purple and orange), 'PiYG' (green and pink) and 'BrBG' (petrol and brown). Alternatively, the colours can be customized as a vector in the order: c('monophyletic', 'non-monophyletic/intruded', 'intruder/outlier').
tax.colour	Colour vector for plot type "taxonomy" (and right side of "monoVStax"). Default is "rainbow", which assigns every taxon a different colour using the command 'rainbow()'.
intrud.colour	Colour vector for plot type "intruder". Default is "rainbow", which assigns every intruder taxon a different colour using the command 'rainbow()', while monophyletic taxa are gray and non-monophyletic (intruded) taxa are black.
edge.width	Used to customize the thickness of the tree's branches. Default is 3.
cex	Used to customize the font size of the tip labels. Default is 0.2.
adj.names	Used to change the justification of the tip label text (0: left, 0.5: centered, 1: right). Default is 'auto', which will left-justify the names in all plot types except for 'monoVStax', in which they will be centered.
adj.tips	Used to change the justification of the tip label symbols (0: left, 0.5: centered, 1: right). Default is 0.5 (centered).
label.offset	Used to adjust the distance between the tips of the tree and the tip label text. Default is 'auto', which will scale the distance based on the size of the tree.
type	Used to determine the type of phylogeny, can either be "phylogram", "cladogram", "fan" or "radial". The type "unrooted" is not available, since the assessment of monophyly requires a rooted tree. The default is "phylogram".
...	Other plot.phylo arguments.

Details

Using the output object of AssessMonophyly, the result can be visualized in a couple of ways. Specifying the 'plot.type' "monophyly" plots a tree colouring the branches according to whether

the respective clades are monophyletic, non-monophyletic or intruders/outliers. 'taxonomy' simply colour codes the different taxa. 'intruders' colours monophyletic groups gray, non-monophyletic groups black and the intruders and outliers according to the taxonomic group they belong to. Finally, 'monoVStax', plots two trees in a mirrored fashion, with the 'monophyly' type on the left and the 'taxonomy' type on the right. Note that 'intruder' status will be used for monophyletic taxa that intrude others. It is important to remember that this is merely supposed to be a useful visualization and not a biologically meaningful reconstruction. The colour vectors must be of sufficient length if customized, i.e. three colours for mono.colour, as many as taxa for tax.colour and as many as intruder taxa for intrud.colour. Collapsing monophyletic taxa by specifying 'monocoll = TRUE' allows to focus on the problem zones and should especially be useful for larger phylogenies.

Value

Plots phylogeny or prints it to pdf.

Note

Currently, if outliers are checked for (by specifying outliercheck as TRUE when running 'AssessMonophyly'), the plotting function will treat outliers and intruders the same way. This may be changed in future versions.

Author(s)

Orlando Schwery

See Also

AssessMonophyly, MonoPhy-package

Examples

```
data(Ericactree)
solution <- AssessMonophyly(Ericactree)

PlotMonophyly(solution=solution, tree=Ericactree, plot.type='monophyly', ladderize=TRUE,
mono.colour='PRGn')
PlotMonophyly(solution=solution, tree=Ericactree, plot.type='taxonomy', ladderize=TRUE)

# especially for larger phylogenies it is recommended to print the plot to a pdf file instead of
#plotting, for easier inspection
# the argument "PDF" has to be set to "TRUE" for the example to actually output PDFs.
PlotMonophyly(solution=solution, tree=Ericactree, plot.type='monoVStax', ladderize=TRUE,
PDF=FALSE, mono.colour='PRGn', PDF_filename='MonoTaxplot.pdf')
PlotMonophyly(solution=solution, tree=Ericactree, plot.type='taxonomy', ladderize=TRUE,
PDF=FALSE, PDF_filename='Taxplot.pdf')
```

Index