

Package ‘glottoTrees’

March 1, 2025

Type Package
Title Phylogenetic Trees in Linguistics
Version 0.1.12
Author Erich Round
Maintainer Erich Round <e.round@surrey.ac.uk>
Description Preparation of phylogenetic trees by adapting resources from glottolog.com.
License CC BY 4.0
Encoding UTF-8
LazyData true
Imports ape,
dplyr,
phytools,
stringr,
tidyr
Collate 'branch_lengths.R'
'data.R'
'input_checks.R'
'labels.R'
'metadata.R'
'plotting.R'
'topology.R'
'zzz.R'
RoxygenNote 7.3.2
Suggests testthat
Depends R (>= 3.5)

Contents

abridge_labels	2
add_tip	3
apply_duplicate_suffixes	4
assemble_rake	5
assemble_supertree	6
clone_tip	7
collapse_node	8

convert_to_tip	9
extract_glottocode	9
get_glottolog_families	10
get_glottolog_languages	11
get_glottolog_phylo_geo	11
get_glottolog_trees	12
glottolog_geography	13
glottolog_trees	14
keep_as_tip	15
keep_tip	15
move_node	16
move_tip	17
nonbranching_nodes	17
plot_glotto	18
relabel_node	19
relabel_tip	20
relabel_with_names	20
remove_clade	21
remove_tip	22
rescale_branches	23
rescale_branches_constrained	23
rescale_branches_exp	24
rescale_deepest_branches	25
ultrametricize	25
which_tree	26

Index	27
--------------	-----------

abridge_labels	<i>Shorten labels to a glottocode substring</i>
----------------	---

Description

Shortens tip and node labels to a glottocode substring within them. If any labels lack a glottocode substring, a warning is given.

Usage

```
abridge_labels(phy)
```

Arguments

phy	A phylo or multiPhylo object, containing one or more trees to manipulate.
-----	---

Details

Glottocodes comprise four lowercase letters (or b10b or 3adt) followed by four numbers, and are only identified if they are initial in the string or are preceded by <.

Also recognizes and retains duplicate suffixes, i.e., a hyphen followed by one or more numerals at the end of the string (see [apply_duplicate_suffixes](#)).

Value

A phylo or multiPhylo object, the manipulated tree(s).

Examples

```
library(ape)

tree <- get_glottolog_trees("Koreanic")
plot(tree)
nodelabels(tree$node.label)
tree2 <- abridge_labels(tree)
plot(tree2)
nodelabels(tree2$node.label)

# Retain duplicate suffixes:
tree3 <- clone_tip(tree, "Jollado<chol1278>", n = 2, subgroup = TRUE)
tree3a <- apply_duplicate_suffixes(tree3)
plot(tree3a)
nodelabels(tree3a$node.label)
tree4 <- abridge_labels(tree3a)
plot(tree4)
nodelabels(tree4$node.label)

# A warning is issued if any label does not contain a glottocode
supertree <- assemble_supertree() # contains nodes without glottocodes
supertree2 <- abridge_labels(assemble_supertree())

# Applied to a `multiPhylo` object:
trees <- get_glottolog_trees(c("Kartvelian", "Basque"))
trees2 <- abridge_labels(trees)
plot(trees[[1]])
nodelabels(trees[[1]]$node.label)
plot(trees2[[1]])
nodelabels(trees2[[1]]$node.label)
plot(trees[[2]])
nodelabels(trees[[2]]$node.label)
plot(trees2[[2]])
nodelabels(trees2[[2]]$node.label)
```

add_tip

Add tips to a tree

Description

Add one or more tips below a parent node specified by its label.

Usage

```
add_tip(phy, label, parent_label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing tip labels.
parent_label	A character string containing the label of the parent node.

Details

The length of the branches, between the added tips and their parent node is set equal to the longest of the original branches directly below node n.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("LeftMay"))
tree <- ultrametricize(rescale_branches_exp(tree))
plot_glotto(tree)

# Attach one or more new tips to a tree:
tree2 <- add_tip(tree, label = "rockypeak", parent_label = "iter1240")
plot_glotto(tree2)
tree3 <- add_tip(tree, label = c("bo", "kaumifi"), parent_label = "bopa1235")
plot_glotto(tree3)

# Move tips by using remove_tip() and add_tip():
tree4 <- remove_tip(tree, "amap1240")
tree4a <- add_tip(tree4, "amap1240", parent_label = "left1242")
plot_glotto(tree4a)
```

apply_duplicate_suffixes

Apply duplicate suffixes to tips and nodes

Description

Suffixes are applied to ensure tip labels and node labels are not duplicates. Suffixes have the form -1, -2, -3, ...

Usage

```
apply_duplicate_suffixes(phy)
```

Arguments

phy A phylo object, the tree whose labels are to have copy suffixes applied.

Details

The function recognizes existing duplicate suffixes and deals with them in one of two ways. If a label has n duplicates that are already suffixed -1, -2, ... -n, then the suffixes are not changed. Under any other conditions, old suffixes are removed and new ones applied.

Suffixation of tips and of nodes are handled independently of one another.

Value

A phylo object, the same tree but with suffixes applied to the labels.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("Koreanic"))
plot_glotto(tree)
tree2 <- clone_tip(tree, "chol1278", n = 2, subgroup = TRUE)
plot_glotto(tree2)
# Technically, tree2 is ill-formed because it has duplicate tip labels.
# Note how this causes problems if we try to clone one of them, since it
# is unclear which should be cloned:
## Not run:
tree2a <- clone_tip(tree2, "chol1278")

## End(Not run)

# Suffixation of duplicate tips
tree3 <- apply_duplicate_suffixes(tree2)
plot_glotto(tree3)

# Once they are suffixed, these tips can be cloned successfully:
tree4 <- clone_tip(tree3, c("chol1278-2", "chol1278-3"), subgroup = TRUE)
plot_glotto(tree4)
# Suffixing is applied across all tips that share a glottocode, and
# separately, across all nodes that share a glottocode:
tree5 <- apply_duplicate_suffixes(tree4)
plot_glotto(tree5)
```

assemble_rake

Bind trees as a high-level rake

Description

Takes a multiPhylo object containing multiple trees and combines them into a single tree with a rake structure at its root, below which each tree appears on its own branch.

Usage

```
assemble_rake(phy)
```

Arguments

phy A multiphylo object containing the trees to be combined.

Value

A phylo object, a single tree.

Examples

```
library(ape)

arnhem_hypothesis <-
  c("Gunwinyguan", "Mangarrayi-Maran", "Maningrida",
    "Kungarakany", "Gaagudju")
trees <- get_glottolog_trees(arnhem_hypothesis)
simple_rake <- assemble_rake(trees)
rake_in_rake <- assemble_rake(c(assemble_rake(trees[1:3]), trees[4:5]))
plot(simple_rake)
plot(rake_in_rake)

# If `phy` contains only one tree, a warning is issued.
mono_rake <- assemble_rake(trees[3])
plot(mono_rake)
```

assemble_supertree	<i>Create a glottolog super-tree</i>
--------------------	--------------------------------------

Description

Combining glottolog family trees into one large tree. Families can be assembled directly below a rake structure at the root, or can be grouped, so that the root first branches into groups, and the families then branch out below the group nodes.

Usage

```
assemble_supertree(macro_groups, glottolog_version)
```

Arguments

macro_groups	A list of character vectors, in which each vector contains the names of one or more macroareas which define a group. Alternatively, setting macro_groups to NULL causes the tree to be assembled without groups.
glottolog_version	A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

Grouping is controlled by the macro_groups parameter. Groups can comprise a single glottolog macroarea, or multiple macroareas. Current macroareas are Africa, Australia, Eurasia, North America, Papunesia and South America. Setting macro_groups to NULL causes the tree to be assembled without groups.

Examples

```
# Supertree whose first order branches are the glottolog macroareas
supertree <- assemble_supertree()
supertree_v.4.3 <- assemble_supertree(glottolog_version = "4.3")
```

```
# Supertree whose first order branches are glottolog families
supertree <- assemble_supertree(macro_groups = NULL)

# Supertree whose first order branches are the African & Eurasian macroareas
supertree <- assemble_supertree(macro_groups = list("Africa", "Eurasia"))

# Supertree whose first order branches are the glottolog macroareas, but
# with the Americas combined:
supertree <- assemble_supertree(
  macro_groups = list("Africa", "Australia", "Eurasia", "Papunesia",
    c("South America", "North America"))
)
```

clone_tip

*Clone tips***Description**

Clones tips as sisters of the original. Optionally, places the new clones and the original in their own subgroup, in which case the node for the new subgroup is assigned the same label as the original tip.

Usage

```
clone_tip(phy, label, n = 1, subgroup = FALSE)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing tip labels.
n	A numeric vector. The number of clones to make.
subgroup	A logical. Whether to create a subgroup containing the new clones and their original.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <-
  rescale_branches_exp(abridge_labels(get_glottolog_trees("Tangkic")))
plot_glottotree(tree)
tree2 <- clone_tip(tree, "nyan1300")
plot_glottotree(tree2)

tree3 <- clone_tip(tree, "nyan1300", subgroup = TRUE)
plot_glottotree(tree3)

# Add suffixes to labels, to keep all labels distinct
tree3a <- apply_duplicate_suffixes(tree3)
```

```

plot_glottto(tree3a)

tree4 <- clone_tip(tree, "lard1243", n = 3)
plot_glottto(tree4)

tree5 <- clone_tip(tree, "lard1243", n = 3, subgroup = TRUE)
plot_glottto(tree5)

tree6 <- clone_tip(tree, c("lard1243", "nyan1300"), n = 2, subgroup = TRUE)
plot_glottto(tree6)
tree6a <- apply_duplicate_suffixes(tree6)
plot_glottto(tree6a)

## Not run:
# Returns error if any element of `label` is not in `phy`
tree7 <- clone_tip(tree, c("lard1243", "xxxx1234"))

## End(Not run)

```

collapse_node

*Collapse one or more nodes rootwards***Description**

Collapse one or more nodes rootwards

Usage

```
collapse_node(phy, label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing node labels.

Value

A phylo object containing the modified tree.

Examples

```

library(ape)
tree <-
  rescale_branches_exp(abridge_labels(get_glottolog_trees("Tangkic")))
plot_glottto(tree)
tree2 <- collapse_node(tree, "gang1267")
plot_glottto(tree2)

```

convert_to_tip	<i>Convert nodes to tips</i>
----------------	------------------------------

Description

For nodes identified by their label, remove the clade that they dominate and replace it with a tip of the same name.

Usage

```
convert_to_tip(phy, label, warn = TRUE)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing node or tip labels.
warn	A logical, whether to issue warning when label contains tip labels.

Details

Any labels included in label which are both node labels and tip labels are regarded as node label, and that node will be removed and replaced by a tip.

Any labels included in label which are tip labels only are ignored.

Value

A phylo object containing the modified tree.

Examples

```
tree1 <- abridge_labels(get_glottolog_trees("GreatAndamanese"))
plot_glotto(tree1)
tree2 <- convert_to_tip(tree1, label = c("okol1242", "sout2683"))
plot_glotto(tree2)
```

extract_glottocode	<i>Extract glottocode substrings</i>
--------------------	--------------------------------------

Description

From a character vector, extracts the first glottocode from each element.

Usage

```
extract_glottocode(label)
```

Arguments

label	A string
-------	----------

Details

Glottocodes comprise four lowercase letters (or b10b or 3adt) followed by four numbers, and are only identified if they are initial in the string or are preceded by <.

Also recognizes and retains duplicate suffixes, i.e., a hyphen followed by one or more numerals at the end of the string (see [apply_duplicate_suffixes](#)).

Value

A string

Examples

```
extract_glottocode("DongoKresh<dong1296>-1-")
extract_glottocode(c("DongoKresh<dong1296>-1-", "Goro-Golo<orlo1238>"))

# Duplicate suffixes are recognised and retained
extract_glottocode(c("Goro-Golo<orlo1238>-1", "Goro-Golo<orlo1238>-2"))
```

```
get_glottolog_families
```

Simple family metadata

Description

Returns a dataframe of metadata on glottolog's language families.

Usage

```
get_glottolog_families(glottolog_version)
```

Arguments

glottolog_version

A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

Returned columns are: tree, tree_name, n_tips, n_nodes and main_macroarea.

Examples

```
head(get_glottolog_families())
head(get_glottolog_families(glottolog_version = "4.3"))
```

`get_glottolog_languages`*Simple language metadata*

Description

Returns a dataframe of metadata on glottolog's languages.

Usage

```
get_glottolog_languages(glottolog_version)
```

Arguments

`glottolog_version`

A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

Returned columns are: glottocode, isocodes, name, name_in_tree, position, tree and tree_name.

Examples

```
head(get_glottolog_languages())  
head(get_glottolog_languages(glottolog_version = "4.3"))
```

`get_glottolog_phylo_geo`*Extended glottolog metadata*

Description

Returns a dataframe of glottolog geographical and phylogenetic metadata.

Usage

```
get_glottolog_phylo_geo(glottolog_version)
```

Arguments

`glottolog_version`

A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

Returned columns are: glottocode, isocodes, name, level, vertex_type, vertex_label, vertex_name, macroarea, latitude, longitude, family_glottocode, family_name and tree.

Examples

```
head(get_glottolog_phylo_geo())
head(get_glottolog_phylo_geo(glottolog_version = "4.3"))
```

get_glottolog_trees	<i>Glottolog trees by version</i>
---------------------	-----------------------------------

Description

Returns a multiPhylo object containing all, or a requested subset, of the glottolog trees.

Usage

```
get_glottolog_trees(family, glottolog_version)
```

Arguments

family	A character vector. Elements are names of glottolog families whose trees are to be returned. If family is left unspecified, all trees are returned.
glottolog_version	A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

By default, trees are returned from the most recent version of glottolog. Alternatively, an older version of glottolog can be specified.

Value

A phylo object containing one glottolog tree, or a multiPhylo object containing multiple glottolog trees.

Examples

```
library(ape)
tree_totonacan <- get_glottolog_trees(family = "Totonacan")
tree_totonacan_v4.3 <- get_glottolog_trees("Totonacan", "4.3")
plot(tree_totonacan)
plot(tree_totonacan_v4.3)
trees <- get_glottolog_trees(family = c("Caddoan", "Tangkic"))
plot(trees[[1]])
plot(trees[[2]])
```

glottolog_geography *Geographical data from glottolog, versions 4.0 - 5.0*

Description

Datasets of geographical information about the languages in glottolog.

Usage

glottolog_geography_v4.0

glottolog_geography_v4.1

glottolog_geography_v4.2

glottolog_geography_v4.3

glottolog_geography_v4.4

glottolog_geography_v4.5

glottolog_geography_v4.6

glottolog_geography_v4.7

glottolog_geography_v4.8

glottolog_geography_v5.0

Format

A dataframe:

glottocode glottocode of the lect

name name of the lect

isocodes the ISO-639-3 code of the lect

level "language" or "dialect"

macroarea glottolog's geographical macroarea

latitude

longitude

v4.0: A dataframe of 20,049 rows

v4.1: A dataframe of 20,290 rows

v4.2: A dataframe of 20,752 rows

v4.3: A dataframe of 20,930 rows

v4.4: A dataframe of 21,329 rows

v4.5: A dataframe of 21,508 rows

v4.6: A dataframe of 21,652 rows

v4.7: A dataframe of 21,746 rows

v4.8: A dataframe of 21,957 rows

v5.0: A dataframe of 22,111 rows

Source

<https://glottolog.org/meta/downloads>

glottolog_trees	<i>Trees from glottolog, versions 4.0 - 5.0</i>
-----------------	---

Description

multiPhylo objects, which provide a representation of the phylogenetic relationships of the languages in glottolog.

Usage

glottolog_trees_v4.0

glottolog_trees_v4.1

glottolog_trees_v4.2

glottolog_trees_v4.3

glottolog_trees_v4.4

glottolog_trees_v4.5

glottolog_trees_v4.6

glottolog_trees_v4.7

glottolog_trees_v4.8

glottolog_trees_v5.0

Format

v4.0: A multiPhylo object of 421 trees

v4.1: A multiPhylo object of 421 trees

v4.2: A multiPhylo object of 422 trees

v4.3: A multiPhylo object of 418 trees

v4.4: A multiPhylo object of 420 trees

v4.5: A multiPhylo object of 420 trees

v4.6: A multiPhylo object of 420 trees

v4.7: A multiPhylo object of 422 trees

v4.8: A multiPhylo object of 422 trees

v5.0: A multiPhylo object of 423 trees

Source

<https://glottolog.org/meta/downloads>

keep_as_tip	<i>Keep tips and convert nodes to tips</i>
-------------	--

Description

Nominate which tips and nodes are to be kept as tips. Others tips are removed. If any element of label is both a node label and tip label, it will be treated as referring to the tip, not the node.

Usage

```
keep_as_tip(phy, label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing tip and node labels.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("Tangkic"))
plot_glotto(tree)

tree2 <- keep_as_tip(tree, c("lard1243", "kaya1319", "nyan1300", "gang1267"))
plot_glotto(tree2)
```

keep_tip	<i>Keep tips</i>
----------	------------------

Description

Nominate which tips are to be kept. Others are removed.

Usage

```
keep_tip(phy, label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing tip labels.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("Tangkic"))
plot_glottotree(tree)

tree2 <- keep_tip(tree, c("lard1243", "kang1283", "kaya1319"))
plot_glottotree(tree2)
```

move_node	<i>Move a node</i>
-----------	--------------------

Description

Move one node to a position dominated by a new parent node.

Usage

```
move_node(phy, label, parent_label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character string containing the label of the node to move.
parent_label	A character string containing the label of the new parent node.

Details

The branch length above the moved node remains unchanged.

If moving the node would result in any other node(s) having no descendant tips, then those other nodes are removed.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("LeftMay"))
tree <- ultrametricize(rescale_branches_exp(tree))
plot_glottotree(tree)

tree2 <- move_node(tree, "iter1240", parent_label = "left1242")
plot_glottotree(tree2)
```

move_tip	<i>Move a tip</i>
----------	-------------------

Description

Move one tip to a new parent node.

Usage

```
move_tip(phy, label, parent_label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character string containing the tip label.
parent_label	A character string containing the label of the parent node.

Details

In the tip's new position, the length of the branch above it is the same as the longest branch above any new sister of the tip.

If moving the tip would result in any node(s) having no descendant tips, then those nodes are removed.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("LeftMay"))
tree <- ultrametricize(rescale_branches_exp(tree))
plot_glotto(tree)

tree2 <- move_tip(tree, "amap1240", parent_label = "left1242")
plot_glotto(tree2)
```

nonbranching_nodes	<i>Find nodes with a single child</i>
--------------------	---------------------------------------

Description

Find nodes with a single child

Usage

```
nonbranching_nodes(phy)
```

Arguments

phy A phylo object

Value

A vector of node labels

plot_glotto	<i>Plot a tree in downward, linguistic style</i>
-------------	--

Description

An attempt is made to choose a reasonable plot width, height and offset of the labels from the tree's tips. If the choices are not satisfactory, manipulate them using positive or negative values of extra_width, extra_height and extra_offset.

Usage

```
plot_glotto(
  phy,
  nodelabels = TRUE,
  extra_width = 0,
  extra_height = 0,
  extra_offset = 0,
  srt = 0,
  adj = NULL,
  ...
)
```

Arguments

phy A phylo object.

nodelabels A logical, whether to plot node labels.

extra_width A numeric, extra width to add beyond the default.

extra_height A numeric, extra height to add beyond the default.

extra_offset A numeric, extra offset to add beyond the default.

srt A numeric giving how much the labels are rotated in degrees.

adj A numeric specifying the justification of the text strings of the labels: 0 (left-justification), 0.5 (centering), or 1 (right-justification).

... Additional arguments passed to plot().

Details

By default, tip labels will run vertically, except in the case that all tip labels are a maximum of one character long, in which case they are oriented horizontally.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("Tangkic"))
plot_glotto(tree)
tree2 <- rescale_deepest_branches(tree, 7)
plot_glotto(tree2)
plot_glotto(tree2, srt = 90, adj = 0.5)
```

relabel_node	<i>Change node labels</i>
--------------	---------------------------

Description

Replaces existing tip labels with new labels.

Usage

```
relabel_node(phy, label, new_label)
```

Arguments

phy	A phylo object.
label	A vector of strings, the node labels to be replaced.
new_label	A vector of strings, the same length as label. The corresponding replacement labels.

Examples

```
library(ape)

tree <- relabel_with_names(get_glottolog_trees("Tangkic"))
tree2 <- apply_duplicate_suffixes(clone_tip(tree, "Yangkaal", subgroup = TRUE))
plot(tree2)
tree3 <- relabel_tip(tree2, c("Kangkalita", "Yangkaal-1", "Yangkaal-2"),
                           c("Gangalidda", "Yangkaal", "Yangarella"))
plot(tree3)
nodelabels(tree3$node.label)
tree4 <- relabel_node(tree3, c("Yangkaal", "Kayardild-Yangkaal"),
                      c("Yangkaralda", "Kayardild-Yangkaralda"))
plot(tree4)
nodelabels(tree4$node.label)
```

relabel_tip	<i>Change tip labels</i>
-------------	--------------------------

Description

Replaces existing tip labels with new labels.

Usage

```
relabel_tip(phy, label, new_label)
```

Arguments

phy	A phylo object.
label	A vector of strings, the tip labels to be replaced.
new_label	A vector of strings, the same length as label. The corresponding replacement labels.

Examples

```
library(ape)

tree <- relabel_with_names(get_glottolog_trees("Tangkic"))
plot(tree)
tree2 <- relabel_tip(tree, "Kangkalita", "Gangalidda")
plot(tree2)

tree3 <- clone_tip(tree, "Yangkaal", subgroup = TRUE)
plot(tree3)
tree4 <- apply_duplicate_suffixes(tree3)
plot(tree4)
tree5 <- relabel_tip(tree4, c("Kangkalita", "Yangkaal-1", "Yangkaal-2"),
                           c("Gangalidda", "Yangkaal", "Yangarella"))
plot(tree5)
```

relabel_with_names	<i>Change labels from glottocodes to names</i>
--------------------	--

Description

Looks up glottolog language names corresponding to glottocodes and replaces tip and node labels, which contain glottocodes, with the appropriate names.

Usage

```
relabel_with_names(phy, glottolog_version)
```

Arguments

`phy` A phylo object, the tree to manipulate.

`glottolog_version` A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Details

Also recognizes and retains duplicate suffixes, i.e., a hyphen followed by one or more numerals at the end of the string (see [apply_duplicate_suffixes](#)).

Labels without glottocodes are left unchanged and a warning is given. The version of glottolog to use for look-up can be controlled with `glottolog_version`.

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)

# Replace full glottolog labels with names
tree <- get_glottolog_trees("Kresh-Aja")
plot(tree)
nodelabels(tree$node.label)
tree2 <- relabel_with_names(tree)
plot(tree2)
nodelabels(tree2$node.label)

# Replace abridged labels with names
tree3 <- abridge_labels(tree)
plot(tree3)
nodelabels(tree3$node.label)
tree4 <- relabel_with_names(tree3)
plot(tree4)
nodelabels(tree4$node.label)

# Use names from earlier glottolog version:
tree5 <- relabel_with_names(tree, glottolog_version = "4.3")
plot(tree5)
nodelabels(tree5$node.label)
```

remove_clade

Remove clades

Description

From a tree, remove clades, identified by the labels of their deepest node.

Usage

```
remove_clade(phy, label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing node labels.

Value

A phylo object containing the modified tree.

remove_tip	<i>Remove tips</i>
------------	--------------------

Description

From a tree, remove tips identified by their node labels.

Usage

```
remove_tip(phy, label)
```

Arguments

phy	A phylo object. The tree to manipulate.
label	A character vector containing tip labels.

Value

A phylo object containing the modified tree.

Examples

```
library(ape)

tree <- abridge_labels(get_glottolog_trees("Tangkic"))
plot_glotto(tree)

tree2 <- remove_tip(tree, c("kang1283", "kaya1319"))
plot_glotto(tree2)
```

rescale_branches	<i>Set all branch length to 1</i>
------------------	-----------------------------------

Description

Sets all branch lengths in a tree to the same length.

Usage

```
rescale_branches(phy, length = 1)
```

Arguments

phy	A phylo object, the tree to manipulate.
length	A numeric stating the branch length.

Details

By default, sets all branch lengths to 1.

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)
tree <- abridge_labels(get_glottolog_trees("Tangkic"))
tree2 <- clone_tip(tree, "nyan1300", n = 2, subgroup = TRUE)
plot_glottolog(tree2)
tree3 <- rescale_branches(tree2)
plot_glottolog(tree3)
```

rescale_branches_constrained	<i>Exponentialize branch lengths using constraints</i>
------------------------------	--

Description

Sets the branches furthest from the root to `shortest_length` and then scales other branches exponentially in such a way that the total tree has a height of `tree_height`.

Usage

```
rescale_branches_constrained(phy, tree_height, shortest_length)
```

Arguments

phy	A phylo object, the tree to manipulate.
tree_height	A positive numeric, the desired height of the whole tree.
shortest_length	A positive numeric, the desired length of the shortest branches, i.e., the branches farthest from the root.

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)
tree <- abridge_labels(get_glottolog_trees("Siouan"))
plot_glottotree(tree)
tree2 <- rescale_branches_constrained(tree, tree_height = 2000, shortest_length = 200)
plot_glottotree(tree2)
tree3 <- rescale_branches_constrained(tree, tree_height = 2000, shortest_length = 100)
plot_glottotree(tree3)
```

rescale_branches_exp *Exponentialize branch lengths*

Description

Sets the deepest branches to length 1/2, the next deepest to 1/4, the next to 1/8, etc., all multiplied by the parameter length.

Usage

```
rescale_branches_exp(phy, length = 1)
```

Arguments

phy	A phylo object, the tree to manipulate.
length	A positive numeric, a multiplier for the exponential branch lengths 1/2, 1/4, 1/8...

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)
tree <- abridge_labels(get_glottolog_trees("Siouan"))
plot_glottotree(tree)
tree2 <- rescale_branches_exp(tree)
plot_glottotree(tree2)
```

rescale_deepest_branches
Set length of deepest branches

Description

Sets lengths of branches immediately below the root to the same, user-specified length.

Usage

```
rescale_deepest_branches(phy, length = 1)
```

Arguments

phy A phylo object, the tree to manipulate.
length A numeric stating the branch length.

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)
arnhem_hypothesis <-
  c("Gunwinyguan", "Mangarrayi-Maran", "Maningrida",
    "Kungarakany", "Gaagudju")
tree <- assemble_rake(abridge_labels(get_glottolog_trees(arnhem_hypothesis)))
plot_glottotree(tree)
# tree now contains five language families. All branch lengths are 1.
# Set the deepest branch lengths to 5, implying a great genealogical
# distance between the families within the tree.
tree2 <- rescale_deepest_branches(tree, length = 5)
plot_glottotree(tree2)
```

ultrametricize *Ultrametricize tree by stretching final edges*

Description

Alters branches ending in a tip in such a way that all tips are equidistant from the root. Does this by lengthening branches above all but the existing, most-distance tip(s).

Usage

```
ultrametricize(phy)
```

Arguments

phy A phylo object, the tree to manipulate.

Value

A phylo object, the manipulated tree.

Examples

```
library(ape)
tree <- rescale_branches_exp(abridge_labels(get_glottolog_trees("Siouan")))
plot_glottotree(tree)
tree2 <- ultrametricize(tree)
plot_glottotree(tree2)
```

which_tree	<i>Tree numbers of glottolog families</i>
------------	---

Description

Returns the tree number of one or more glottolog families.

Usage

```
which_tree(family, glottolog_version)
```

Arguments

family	A character vector. Elements are names of glottolog families whose trees are to be returned. If family is left unspecified, all trees are returned.
glottolog_version	A character string, specifying which glottolog version to use. Currently available options are '4.0' through to '4.8' and '5.0'. If no value is specified then the newest available version is used.

Value

A named vector of integers, giving the tree numbers and the family names as the vector names.

Examples

```
which_tree("Caddoan")
which_tree(c("Caddoan", "Tangkic"), glottolog_version = "4.3")
# If some family names are unrecognized, a warning is issued
which_tree(c("Caddoan", "Zzz"), glottolog_version = "4.4")
## Not run:
# If no family names are recognized, an error results
which_tree()
which_tree("Zzz")

## End(Not run)
```

Index

* datasets

- glottolog_geography, [13](#)
- glottolog_trees, [14](#)
- abridge_labels, [2](#)
- add_tip, [3](#)
- apply_duplicate_suffixes, [2](#), [4](#), [10](#), [21](#)
- assemble_rake, [5](#)
- assemble_supertree, [6](#)
- clone_tip, [7](#)
- collapse_node, [8](#)
- convert_to_tip, [9](#)
- extract_glottocode, [9](#)
- get_glottolog_families, [10](#)
- get_glottolog_languages, [11](#)
- get_glottolog_phylo_geo, [11](#)
- get_glottolog_trees, [12](#)
- glottolog_geography, [13](#)
- glottolog_geography_v4.0
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.1
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.2
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.3
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.4
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.5
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.6
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.7
 - (glottolog_geography), [13](#)
- glottolog_geography_v4.8
 - (glottolog_geography), [13](#)
- glottolog_geography_v5.0
 - (glottolog_geography), [13](#)
- glottolog_trees, [14](#)
- glottolog_trees_v4.0 (glottolog_trees),
[14](#)
- glottolog_trees_v4.1 (glottolog_trees),
[14](#)
- glottolog_trees_v4.2 (glottolog_trees),
[14](#)
- glottolog_trees_v4.3 (glottolog_trees),
[14](#)
- glottolog_trees_v4.4 (glottolog_trees),
[14](#)
- glottolog_trees_v4.5 (glottolog_trees),
[14](#)
- glottolog_trees_v4.6 (glottolog_trees),
[14](#)
- glottolog_trees_v4.7 (glottolog_trees),
[14](#)
- glottolog_trees_v4.8 (glottolog_trees),
[14](#)
- glottolog_trees_v5.0 (glottolog_trees),
[14](#)
- keep_as_tip, [15](#)
- keep_tip, [15](#)
- move_node, [16](#)
- move_tip, [17](#)
- nonbranching_nodes, [17](#)
- plot_glotto, [18](#)
- relabel_node, [19](#)
- relabel_tip, [20](#)
- relabel_with_names, [20](#)
- remove_clade, [21](#)
- remove_tip, [22](#)
- rescale_branches, [23](#)
- rescale_branches_constrained, [23](#)
- rescale_branches_exp, [24](#)
- rescale_deepest_branches, [25](#)
- ultrametricize, [25](#)
- which_tree, [26](#)