

Package ‘OpenTreeChronograms’

October 12, 2022

Type Package

Title Open Tree of Life Chronograms

Version 2022.1.28

Maintainer Luna L. Sanchez Reyes <sanchez.reyes.luna@gmail.com>

Description Chronogram database constructed from Open Tree of Life's phylogenetic store.

License GPL (>= 2)

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 2.10)

RoxygenNote 7.1.2

Imports ape, geiger, knitcitations, paleotree, plyr, rotl, stringr,
taxize, treebase, usethis

NeedsCompilation no

Author Brian O'Meara [aut],
Luna L. Sanchez Reyes [aut, cre]

Repository CRAN

Date/Publication 2022-06-22 07:10:10 UTC

R topics documented:

.get_ott_lineage	2
check_ott_input	3
classification_paths_from_taxonomy	3
clean_ott_chronogram	7
clean_taxon_info_children	8
clean_tnrs	8
date_with_pbdb	9
extract_ott_ids	10
get_fossil_range	11
get_opentree_chronograms	11

get_ott_children	12
get_ott_clade	13
get_ott_lineage	14
get_valid_children	15
is_good_chronogram	16
is_phylo	16
make_all_associations	17
make_contributor_cache	17
make_otoI_associations	18
make_overlap_table	18
make_treebase_associations	19
make_treebase_cache	19
map_nodes_ott	20
opentree_chronograms	21
phylo_has_brlen	22
phylo_tiplabel_space_to_underscore	22
phylo_tiplabel_underscore_to_space	23
problematic_chronograms	23
recover_mrcaott	24
summarize_fossil_range	24
tnrs_match	25
tree_from_taxonomy	26
update_all_cached	27
update_datelife_cache	28

Index 30

`.get_ott_lineage` *Get the lineage of a set of taxa. `.get_ott_lineage` uses `rotl::taxonomy_taxon_info()` with `include_lineage = TRUE`.*

Description

Get the lineage of a set of taxa. `.get_ott_lineage` uses `rotl::taxonomy_taxon_info()` with `include_lineage = TRUE`.

Usage

```
.get_ott_lineage(input_ott_match)
```

Arguments

`input_ott_match`
An Output of `check_ott_input` function.

Value

A `taxonomy_taxon_info` object

check_ott_input	<i>Check input for other functions</i>
-----------------	--

Description

check_ott_input is currently used in functions [get_ott_clade\(\)](#), [get_ott_children\(\)](#).

Usage

```
check_ott_input(input = NULL, ott_ids = NULL)
```

Arguments

input	Optional. A character vector of names.
ott_ids	If not NULL, it takes this argument and ignores input. A numeric vector of ott ids obtained with rotl::taxonomy_taxon_info() or rotl::tnrs_match_names() or tnrs_match() .

Details

By default, it uses the ott_id argument if it is not NULL.

Value

A named numeric vector of valid Open Tree Taxonomy (OTT) ids.

classification_paths_from_taxonomy	<i>Gets classification paths for a vector of taxa</i>
------------------------------------	---

Description

This uses the taxize package's wrapper of the Global Names Resolver to get taxonomic paths for the vector of taxa you pass in. Sources is a vector of source labels in order (though it works best if everything uses the same taxonomy, so we recommend doing just one source). You can see options by doing [taxize::gnr_datasources\(\)](#). Our default is Catalogue of Life

Usage

```
classification_paths_from_taxonomy(taxa, sources = "Catalogue of Life")
```

Arguments

taxa	Vector of taxon names
sources	Vector of names of preferred sources; see taxize::gnr_datasources() . Currently supports 100 taxonomic resources, see details.

Details

Taxonomies supported by taxize::gnr_datasources()

1. Catalogue of Life
2. Wikispecies
3. ITIS
4. NCBI
5. Index Fungorum
6. GRIN Taxonomy for Plants
7. Union 4
8. The Interim Register of Marine and Nonmarine Genera
9. World Register of Marine Species
10. Freebase
11. GBIF Backbone Taxonomy
12. EOL
13. Passiflora vernacular names
14. Inventory of Fish Species in the Wami River Basin
15. Pheasant Diversity and Conservation in the Mt. Gaoligongshan Region
16. Finding Species
17. Birds of Lindi Forests Plantation
18. Nemertea
19. Kihansi Gorge Amphibian Species Checklist
20. Mushroom Observer
21. TaxonConcept
22. Amphibia and Reptilia of Yunnan
23. Common names of Chilean Plants
24. Invasive Species of Belgium
25. ZooKeys
26. COA Wildlife Conservation List
27. AskNature
28. China: Yunnan, Southern Gaoligongshan, Rapid Biological Inventories Report No. 04
29. Native Orchids from Gaoligongshan Mountains, China
30. Illinois Wildflowers
31. Coleorrhyncha Species File
32. /home/dimus/files/dwca/zoological names.zip
33. Peces de la zona hidrogeográfica de la Amazonia, Colombia (Spreadsheet)
34. Eastern Mediterranean Syllidae
35. Gaoligong Shan Medicinal Plants Checklist

36. birds_of_tanzania
37. AmphibiaWeb
38. tanzania_plant_specimens
39. Papahānaumokuākea Marine National Monument
40. Taiwanese IUCN species list
41. BioPedia
42. AnAge
43. Embioptera Species File
44. Global Invasive Species Database
45. Sendoya S., Fernández F. AAT de hormigas (Hymenoptera: Formicidae) del Neotrópico 1.0 2004 (Spreadsheet)
46. Flora of Gaoligong Mountains
47. ARKive
48. True Fruit Flies (Diptera, Tephritidae) of the Afrotropical Region
49. 3i - Typhlocybinae Database
50. CATE Sphingidae
51. ZooBank
52. Diatoms
53. AntWeb
54. Endemic species in Taiwan
55. Dermaptera Species File
56. Mantodea Species File
57. Birds of the World: Recommended English Names
58. New Zealand Animalia
59. Blattodea Species File
60. Plecoptera Species File
61. /home/dimus/files/dwca/clemens.zip
62. Coreoidea Species File
63. Freshwater Animal Diversity Assessment - Normalized export
64. Catalogue of Vascular Plant Species of Central and Northeastern Brazil
65. Wikipedia in EOL
66. Database of Vascular Plants of Canada (VASCAN)
67. Phasmida Species File
68. OBIS
69. USDA NRCS PLANTS Database
70. Catalog of Fishes
71. Aphid Species File

72. The National Checklist of Taiwan
73. Psocodea Species File
74. FishBase
75. 3i - Typhlocybinae Database
76. Belgian Species List
77. EUNIS
78. CU*STAR
79. Orthoptera Species File
80. Bishop Museum
81. IUCN Red List of Threatened Species
82. BioLib.cz
83. Tropicos - Missouri Botanical Garden
84. nlbif
85. The International Plant Names Index
86. Index to Organism Names
87. uBio NameBank
88. Arctos
89. Checklist of Beetles (Coleoptera) of Canada and Alaska. Second Edition.
90. The Paleobiology Database
91. The Reptile Database
92. The Mammal Species of The World
93. BirdLife International
94. Checklist da Flora de Portugal (Continental, Açores e Madeira)
95. FishBase Cache
96. Silva
97. Open Tree of Life Reference Taxonomy
98. iNaturalist
99. The Interim Register of Marine and Nonmarine Genera
100. Gymno

Value

A list with resolved taxa (a tibble, from `taxize::gnr_resolve`) and a vector of taxa not resolved

clean_ott_chronogram *Clean up some issues with Open Tree of Life chronograms For now it 1) checks unmapped taxa and maps them with tnrs_match.phylo, 2) roots the chronogram if unrooted*

Description

Clean up some issues with Open Tree of Life chronograms For now it 1) checks unmapped taxa and maps them with tnrs_match.phylo, 2) roots the chronogram if unrooted

Usage

```
clean_ott_chronogram(phy)
```

Arguments

phy A phylo object.

Details

There is no limit to the number of names that can be queried and matched.

The output will preserve all elements from original input phylo object and will add

phy\$mapped A character vector indicating the state of mapping of phy\$tip.labels:

original Tnrs matching was not attempted. Original labeling is preserved.

ott Matching was manually made by a curator in Open Tree of Life.

tnrs Tnrs matching was attempted and successful with no approximate matching. Original label is replaced by the matched name.

approximated Tnrs matching was attempted and successful but with approximate matching. Original labeling is preserved.

unmatched Tnrs matching was attempted and unsuccessful. Original labeling is preserved.

phy\$original.tip.label A character vector preserving all original labels.

phy\$ott_ids A numeric vector with ott id numbers of matched tips. Unmatched and original tips will be NaN.

if tips are duplicated, tnrs will only be run once (avoiding increases in function running time) but the result will be applied to all duplicated tip labels

Value

An object of class data frame or phylo, with the added class match_names.

NULL

NULL

clean_taxon_info_children

Identify, extract and clean taxonomic children names from a `taxonomy_taxon_info()` output.

Description

clean_taxon_info_children eliminates all taxa that will give problems when trying to retrieve an induced subtree from Open Tree of Life.

Usage

```
clean_taxon_info_children(
  taxon_info,
  invalid = c("barren", "extinct", "uncultured", "major_rank_conflict",
             "incertae_sedis", "unplaced", "conflict", "environmental", "not_otu", "hidden",
             "hybrid")
)
```

Arguments

taxon_info	An output of <code>rotl::taxonomy_taxon_info()</code> .
invalid	A character vector of "flags", i.e., characteristics that are used by Open Tree of Life Taxonomy to detect invalid taxon names.

Value

A list with valid children unique OTT names, OTT ids and taxonomic ranks.

clean_tnrs

Eliminates unmatched (NAs) and invalid taxa from a `rotl::tnrs_match_names()` or `tnrs_match()` output. Useful to get ott ids to retrieve an induced synthetic Open Tree of Life. Needed because using `include_suppressed = FALSE` in `rotl::tnrs_match_names()` does not drop all invalid taxa.

Description

Eliminates unmatched (NAs) and invalid taxa from a `rotl::tnrs_match_names()` or `tnrs_match()` output. Useful to get ott ids to retrieve an induced synthetic Open Tree of Life. Needed because using `include_suppressed = FALSE` in `rotl::tnrs_match_names()` does not drop all invalid taxa.

Usage

```
clean_tnrs(
  tnrs,
  invalid = c("barren", "extinct", "uncultured", "major_rank_conflict", "incertae",
             "unplaced", "conflict", "environmental", "not_otu"),
  remove_nonmatches = FALSE
)
```

Arguments

`tnrs` A data.frame, usually an output from `tnrs_match()` or `rotl::tnrs_match_names()`, but see details.

`invalid` A character string with flags to be removed from final object.

`remove_nonmatches` Boolean, whether to remove unsuccessfully matched names or not.

Details

Input can be any data frame or named list that relates taxa stored in an element named "unique" to a validity category stored in "flags".

Value

A data.frame or named list (depending on the input) with valid taxa only.

date_with_pbdb *Date with Paleobiology Database and paleotree.*

Description

This will take a topology, look up information about fossils for taxa on the tree, and use `paleotree::timePaleoPhy()` to compute branch lengths.

Usage

```
date_with_pbdb(phy, recent = FALSE, assume_recent_if_missing = TRUE)
```

Arguments

`phy` A phylo object.

`recent` If TRUE, forces the minimum age to be zero for any taxon

`assume_recent_if_missing` If TRUE, any taxon missing from PBDB is assumed to be recent.

Value

A dated tree.

Examples

```

taxa <- c(
  "Archaeopteryx", "Pinus", "Quetzalcoatlus", "Homo sapiens",
  "Tyrannosaurus rex", "Megatheriidae", "Metasequoia", "Aedes", "Panthera"
)
phy <- tree_from_taxonomy(taxa, sources = "The Paleobiology Database")$phy
# end donttest

```

extract_ott_ids	<i>Get OTT ids from a character vector containing species names and OTT ids.</i>
-----------------	--

Description

Get OTT ids from a character vector containing species names and OTT ids.

Usage

```
extract_ott_ids(x, na.rm = TRUE)
```

```
## Default S3 method:
```

```
extract_ott_ids(x, na.rm = TRUE)
```

Arguments

x	A character vector of taxon names, or a phylo object with tip names containing OTT ids.
na.rm	A logical value indicating whether NA values should be stripped from the output.

Value

An object of class numeric containing OTT ids only.

NULL

Examples

```

canis <- rotl::tnrs_match_names("canis")
canis_taxonomy <- rotl::taxonomy_subtree(canis$ott_id)
my_ott_ids <- extract_ott_ids(x = canis_taxonomy$tip_label)
# Get the problematic elements from input
canis_taxonomy$tip_label[attr(my_ott_ids, "na.action")]

```

get_fossil_range	<i>Get the ages for a taxon from PBDB</i>
------------------	---

Description

This uses the Paleobiology Database's API to gather information on the ages for all specimens of a taxon. It will also look for all descendants of the taxon. It fixes name misspellings if possible.

Usage

```
get_fossil_range(taxon, recent = FALSE, assume_recent_if_missing = TRUE)
```

Arguments

taxon	The scientific name of the taxon you want the range of occurrences of
recent	If TRUE, forces the minimum age to be zero
assume_recent_if_missing	If TRUE, any taxon missing from pbdb is assumed to be recent

Value

a data.frame of max_ma and min_ma for the specimens

get_opentree_chronograms	<i>Get all chronograms from Open Tree of Life database</i>
--------------------------	--

Description

Get all chronograms from Open Tree of Life database

Usage

```
get_opentree_chronograms(max_tree_count = "all")
```

```
get_oto1_chronograms(max_tree_count = "all")
```

Arguments

max_tree_count	Default to "all", it gets all available chronograms. For testing purposes, a numeric value indicating the max number of trees to be cached.
----------------	---

Value

A list of 4 elements:

authors A list of lists of author names of the original studies that published chronograms currently stored in the Open Tree of Life database.

curators A list of lists of curator names that uploaded chronograms to the Open Tree of Life database.

studies A list of study identifiers from original studies that published chronograms currently stored in the Open Tree of Life database.

trees A multiPhylo object storing the chronograms from Open Tree of Life database.

update A character vector indicating the time when the database object was last updated.

version A character vector indicating when the chronogram database opentree_chronograms object was last updated. Format is year.month.day

get_ott_children	<i>Use this instead of <code>rotl::tol_subtree()</code> when taxa are not in synthesis tree and you still need to get all species or an induced OpenTree subtree</i>
------------------	--

Description

Use this instead of `rotl::tol_subtree()` when taxa are not in synthesis tree and you still need to get all species or an induced OpenTree subtree

Usage

```
get_ott_children(input = NULL, ott_ids = NULL, ott_rank = "species", ...)
```

Arguments

input	Optional. A character vector of names.
ott_ids	If not NULL, it takes this argument and ignores input. A numeric vector of ott ids obtained with <code>rotl::taxonomy_taxon_info()</code> or <code>rotl::tnrs_match_names()</code> or <code>tnrs_match()</code> .
ott_rank	A character vector with the ranks you wanna get lineage children from.
...	Other arguments to pass to <code>get_valid_children()</code> .

Value

A data.frame object.

Examples

```

# An example with the dog genus:

# It is currently not possible to get an OpenTree subtree of a taxon that is
# missing from the OpenTree synthetic tree.
# The dog genus is not monophyletic in the OpenTree synthetic tree, so in
# practice, it has no node to extract a subtree from.
# Get the Open Tree Taxonomy identifier (OTT id) for "Canis":
tnrs <- tnrs_match("Canis")

## Not run:
rotl::tol_subtree(tnrs$ott_id[1])
#> Error: HTTP failure: 400
#> [/v3/tree_of_life/subtree] Error: node_id was not found (broken taxon).

## End(Not run) # end dontrun

ids <- tnrs$ott_id[1]
names(ids) <- tnrs$unique_name
children <- get_ott_children(ott_ids = ids) # or
# children <- get_ott_children(input = "Canis")
str(children)
ids <- children$Canis$ott_id
names(ids) <- rownames(children$Canis)
tree_children <- rotl::tol_induced_subtree(ott_ids = ids)
plot(tree_children, cex = 0.3)

# An example with flowering plants:

# orders <- get_ott_children(input = "magnoliophyta", ott_rank = "order")
# Get the number of orders of flowering plants that we have
# sum(orders$Magnoliophyta$rank == "order")

# end donttest

```

get_ott_clade

Get the Open Tree of Life Taxonomic identifiers (OTT ids) and name of one or several given taxonomic ranks from one or more input taxa.

Description

Get the Open Tree of Life Taxonomic identifiers (OTT ids) and name of one or several given taxonomic ranks from one or more input taxa.

Usage

```
get_ott_clade(input = NULL, ott_ids = NULL, ott_rank = "family")
```

Arguments

input	Optional. A character vector of names.
ott_ids	If not NULL, it takes this argument and ignores input. A numeric vector of ott ids obtained with <code>rotl::taxonomy_taxon_info()</code> or <code>rotl::tnrs_match_names()</code> or <code>tnrs_match()</code> .
ott_rank	A character vector with the ranks you wanna get lineage children from.

Value

A list of named numeric vectors with OTT ids from input and all requested ranks.

get_ott_lineage	<i>Get the Open Tree of Life Taxonomic identifier (OTT id) and name of all lineages from one or more input taxa.</i>
-----------------	--

Description

Get the Open Tree of Life Taxonomic identifier (OTT id) and name of all lineages from one or more input taxa.

Usage

```
get_ott_lineage(input = NULL, ott_ids = NULL)
```

Arguments

input	Optional. A character vector of names.
ott_ids	If not NULL, it takes this argument and ignores input. A numeric vector of ott ids obtained with <code>rotl::taxonomy_taxon_info()</code> or <code>rotl::tnrs_match_names()</code> or <code>tnrs_match()</code> .

Value

A list of named numeric vectors of ott ids from input and all the clades it belongs to.

Examples

```
taxa <- c("Homo", "Bacillus anthracis", "Apis", "Salvia")
# lin <- get_ott_lineage(taxa)

# Look up an unknown OTT id:
get_ott_lineage(ott_id = 454749)

# end donttest
```

get_valid_children *Extract valid children from given taxonomic name(s) or Open Tree of Life Taxonomic identifiers (OTT ids) from a taxonomic source.*

Description

Extract valid children from given taxonomic name(s) or Open Tree of Life Taxonomic identifiers (OTT ids) from a taxonomic source.

Usage

```
get_valid_children(input = NULL, ott_ids = NULL, taxonomic_source = "ncbi")
```

Arguments

input Optional. A character vector of names.

ott_ids If not NULL, it takes this argument and ignores input. A numeric vector of ott ids obtained with `rotl::taxonomy_taxon_info()` or `rotl::tnrs_match_names()` or `tnrs_match()`.

taxonomic_source A character vector with the desired taxonomic sources. Options are "ncbi", "gbif" or "irmng". Any other value will retrieve data from all taxonomic sources. The function defaults to "ncbi".

Details

GBIF and other taxonomies contain deprecated taxa that are not marked as such in the Open Tree of Life Taxonomy. We are relying mainly in the NCBI taxonomy for now.

Value

A named list containing valid taxonomic children of given taxonomic name(s).

Examples

```
# genus Dictyophyllidites with ott id = 6003921 has only extinct children
# in cases like this the same name will be returned

tti <- rotl::taxonomy_taxon_info(6003921, include_children = TRUE)
gvc <- get_valid_children(ott_ids = 6003921)

# More examples:

get_valid_children(ott_ids = 769681) # Psilotopsida
get_valid_children(ott_ids = 56601) # Marchantiophyta
```

is_good_chronogram *Check if a tree is a valid chronogram.*

Description

Check if a tree is a valid chronogram.

Usage

```
is_good_chronogram(phy)
```

Arguments

phy A phylo object.

Value

TRUE if it is a valid tree.

is_phylo *Checks if phy is a phylo object and/or a chronogram.*

Description

Checks if phy is a phylo object and/or a chronogram.

Usage

```
is_phylo(phy = NULL, brlen = FALSE, dated = FALSE)
```

Arguments

phy A phylo object.
brlen Boolean. If TRUE it checks if phylo object has branch lengths.
dated Boolean. If TRUE it checks if phylo object is ultrametric.

Value

Nothing

make_all_associations *Find all authors and where they have deposited their trees*

Description

Find all authors and where they have deposited their trees

Usage

```
make_all_associations(outputfile = "depositorcache.RData")
```

Arguments

outputfile Path including file name. NULL to prevent saving.

Value

a data.frame of "person" and "urls".

make_contributor_cache
Create a cache from Open Tree of Life

Description

Create a cache from Open Tree of Life

Usage

```
make_contributor_cache(outputfile = "contributorcache.RData")
```

Arguments

outputfile Path including file name

Value

List containing author and curator results

`make_otol_associations`*Associate Open Tree of Life authors with studies*

Description

Associate Open Tree of Life authors with studies

Usage

```
make_otol_associations()
```

Value

`data.frame` with author last name, author first and other names, and comma delimited URLs for Open Tree of Life studies

`make_overlap_table`*Create an overlap table*

Description

Create an overlap table

Usage

```
make_overlap_table(results_table)
```

Arguments

`results_table` An "author.results" or "curator.results" `data.frame`

Value

A `data.frame` with information on curators and what clades they've worked on

make_treebase_associations

Associate TreeBase authors with studies

Description

Associate TreeBase authors with studies

Usage

```
make_treebase_associations()
```

Value

data.frame with author last name, author first and other names, and comma delimited URLs for TreeBase studies

make_treebase_cache *Create a cache from TreeBase*

Description

Create a cache from TreeBase

Usage

```
make_treebase_cache(outputfile = "treebasecache.RData")
```

Arguments

outputfile Path including file name

Value

List containing author and curator results

map_nodes_ott	<i>Add Open Tree of Life Taxonomy to tree nodes.</i>
---------------	--

Description

Add Open Tree of Life Taxonomy to tree nodes.

Usage

```
map_nodes_ott(tree)
```

Arguments

tree A tree either as a newick character string or as a phylo object.

Value

A phylo object with "nodelabels".

Examples

```
## Not run:  
  
# Load the Open Tree chronograms database cached in datelife:  
utils::data(opentree_chronograms)  
  
# Get the small chronograms (i.e., chronograms with less than ten tips) to generate a pretty plot:  
small <- opentree_chronograms$trees[unlist(sapply(opentree_chronograms$trees, ape::Ntip)) < 10]  
  
# Now, map the Open Tree taxonomy to the nodes of the first tree  
phy <- map_nodes_ott(tree = small[[1]])  
# and plot it:  
# plot_phylo_all(phy)  
library(ape)  
plot(phy)  
nodelabels(phy$node.label)  
  
## End(Not run) # end dontrun
```

opentree_chronograms *Open Tree of Life Chronogram database in R*

Description

Now storing >200 chronograms from Open Tree of Life

Usage

```
opentree_chronograms
```

Format

A list of four elements, containing data from OpenTree of Life chronograms

authors A list of lists of author names of the original studies that published chronograms in the Open Tree of Life database.

curators A list of lists of curator names that uploaded chronograms to the Open Tree of Life database.

studies A list of study identifiers.

trees A multiPhylo object storing the chronograms from Open Tree of Life database.

update A character vector indicating the time when the database object was last updated.

version A character vector indicating when the chronogram database opentree_chronograms object was last updated. Format is year.month.day

Details

```
Generated with devtools::install_github("ropensci/rotl", ref = devtools::github_pull("137"))
remotes::install_github("ROpenSci/bibtex") opentree_chronograms <- get_opentree_chronograms()
opentree_chronograms$update_time <- Sys.time() opentree_chronograms$version <- format(Sys.time(),
"%Y.%m.%d") usethis::use_data(opentree_chronograms, overwrite = T, compress="xz") and up-
dated with update_datelife_cache()
```

Source

<http://opentreeoflife.org>

phylo_has_brlen *Check if a tree has branch lengths*

Description

Check if a tree has branch lengths

Usage

```
phylo_has_brlen(phy)
```

Arguments

phy A phylo object.

Value

A TRUE or FALSE

phylo_tiplabel_space_to_underscore
Convert spaces to underscores in trees.

Description

Convert spaces to underscores in trees.

Usage

```
phylo_tiplabel_space_to_underscore(phy)
```

Arguments

phy A phylo object.

Value

A phylo object.

 phylo_tiplabel_underscore_to_space

Convert underscores to spaces in trees.

Description

Convert underscores to spaces in trees.

Usage

```
phylo_tiplabel_underscore_to_space(phy)
```

Arguments

phy A phylo object.

Value

A phylo object.

problematic_chronograms

Problematic chronograms from Open Tree of Life.

Description

Problematic chronograms from Open Tree of Life.

Usage

```
problematic_chronograms
```

Format

A list of trees with unmapped taxa

Details

Before we developed tools to clean and map tip labels for our cached trees we found some trees that were stored with unmapped tip labels we extracted them and saved them to be used for testing functions. Generated with `problematic_chronograms <- opentree_chronograms$trees[sapply(sapply(opentree_chronograms$trees, "[", "tip.label"), function(x) any(grepl("not.mapped", x)))]` `usethis::use_data(problematic_chronograms)` `opentree_chronograms` object from commit <https://github.com/phylo-tastic/datelife/tree/be894448f6fc437241cd0916fab45e84> `["", "tip.label"), function(x) any(grepl("not.mapped", x)))]`: `R:%22,%20%22tip.label%22,%20function(x)%20any(grepl(%22`

Source

<http://opentreeoflife.org>

recover_mrcaott	<i>Get an mrcaott tag from an OpenTree induced synthetic tree and get its name and ott id</i>
-----------------	---

Description

Get an mrcaott tag from an OpenTree induced synthetic tree and get its name and ott id

Usage

```
recover_mrcaott(tag)
```

Arguments

tag	A character vector with the mrca tag
-----	--------------------------------------

Value

A numeric vector with ott id from original taxon named with the corresponding ott name

summarize_fossil_range	<i>Summarize taxon age from PBDB to just a single min and max age</i>
------------------------	---

Description

This uses the Paleobiology Database's API to gather information on the ages for all specimens of a taxon. It will also look for all descendants of the taxon. It fixes name misspellings if possible. It is basically a wrapper for `get_fossil_range`.

Usage

```
summarize_fossil_range(taxon, recent = FALSE, assume_recent_if_missing = TRUE)
```

Arguments

taxon	The scientific name of the taxon you want the range of occurrences of
recent	If TRUE, forces the minimum age to be zero
assume_recent_if_missing	If TRUE, any taxon missing from pbdb is assumed to be recent

Value

a single row data.frame of `max_ma` and `min_ma` for the specimens, with rowname equal to taxon input

tnrs_match	<i>Taxon name resolution service (tnrs) applied to a vector of names by batches</i>
------------	---

Description

Taxon name resolution service (tnrs) applied to a vector of names by batches

Usage

```
tnrs_match(input, reference_taxonomy, tip, ...)

## Default S3 method:
tnrs_match(input, reference_taxonomy = "otl", ...)

## S3 method for class 'phylo'
tnrs_match(input, reference_taxonomy = "otl", tip = NULL, ...)
```

Arguments

input	A character vector of taxon names, or a phylo object with tip names, to be matched to taxonomy.
reference_taxonomy	A character vector specifying the reference taxonomy to use for tnrs.
tip	A vector of mode numeric or character specifying the tips to match. If left empty all tips will be matched.
...	Arguments passed on to rotl::tnrs_match_names
context_name	name of the taxonomic context to be searched (length-one character vector or NULL). Must match (case sensitive) one of the values returned by tnrs_contexts . Default to "All life".
do_approximate_matching	A logical indicating whether or not to perform approximate string (a.k.a. "fuzzy") matching. Using FALSE will greatly improve speed. Default, however, is TRUE.
ids	A vector of ids to use for identifying names. These will be assigned to each name in the names array. If ids is provided, then ids and names must be identical in length.
include_suppressed	Ordinarily, some quasi-taxa, such as incertae sedis buckets and other non-OTUs, are suppressed from TNRS results. If this parameter is true, these quasi-taxa are allowed as possible TNRS results.

Details

There is no limit to the number of names that can be queried and matched.

The output will preserve all elements from original input phylo object and will add

phy\$mapped A character vector indicating the state of mapping of phy\$tip.labels:

original Tnrs matching was not attempted. Original labeling is preserved.

ott Matching was manually made by a curator in Open Tree of Life.

tnrs Tnrs matching was attempted and successful with no approximate matching. Original label is replaced by the matched name.

approximated Tnrs matching was attempted and successful but with approximate matching. Original labeling is preserved.

unmatched Tnrs matching was attempted and unsuccessful. Original labeling is preserved.

phy\$original.tip.label A character vector preserving all original labels.

phy\$ott_ids A numeric vector with ott id numbers of matched tips. Unmatched and original tips will be NaN.

if tips are duplicated, tnrs will only be run once (avoiding increases in function running time) but the result will be applied to all duplicated tip labels

Value

An object of class data frame or phylo, with the added class `match_names`.

NULL

NULL

Examples

```
tnrs_match(input = c("Mus"))
tnrs_match(input = c("Mus", "Mus musculus"))
tnrs_match(input = c("Mus", "Echinus", "Hommo", "Mus"))
```

tree_from_taxonomy *Gets a taxonomic tree from a vector of taxa*

Description

This uses the `taxize` package's wrapper of the Global Names Resolver to get taxonomic paths for the vector of `taxa` you pass in. Sources is a vector of source labels in order (though it works best if everything uses the same taxonomy, so we recommend doing just one source). You can see options by doing `taxize::gnr_datasources()`. Our default is Catalogue of Life. The output is a phylo object (typically with many singleton nodes if `collapse_singles` is `FALSE`: nodes with only one descendant (like "Homo" having "Homo sapiens" as its only descendant) but these singletons typically have `node.labels`

Usage

```
tree_from_taxonomy(
  taxa,
  sources = "Catalogue of Life",
  collapse_singles = TRUE
)
```

Arguments

taxa Vector of taxon names
sources Vector of names of preferred sources; see `taxize::gnr_datasources()`. Currently supports 100 taxonomic resources, see details.
collapse_singles If true, collapses singleton nodes

Value

A list containing a phylo object with resolved names and a vector with unresolved names

Examples

```

taxa <- c(
  "Homo sapiens", "Ursus arctos", "Pan paniscus", "Tyrannosaurus rex",
  "Ginkgo biloba", "Vulcan", "Klingon"
)
results <- tree_from_taxonomy(taxa)
print(results$unresolved) # The taxa that do not match
ape::plot.phylo(results$phy) # may generate warnings due to problems with singletons
ape::plot.phylo(ape::collapse.singles(results$phy), show.node.label = TRUE)
# got rid of singles, but this also removes a lot of the node.labels
# end donttest

```

update_all_cached *Update all data files as data objects for the package*

Description

This includes opentree chronograms, contributors, treebase and curators For speed, datelife caches chronograms and other information. Running this (within the checked out version of datelife) will refresh these. Then git commit and git push them back

Usage

```
update_all_cached()
```

Value

None

update_datelife_cache *Create an updated OpenTree chronograms database object*

Description

The function calls `get_opentree_chronograms()` to update the OpenTree chronograms database cached in datelife. It has the option to write the updated object as an .Rdata file, that will be independent of the opentree_chronograms data object that you can load with `data("opentree_chronograms", package = "datelife")`.

Usage

```
update_datelife_cache(
  write = TRUE,
  updated_name = "opentree_chronograms_updated",
  file_path = file.path(tempdir()),
  ...
)
```

Arguments

<code>write</code>	Defaults to TRUE, it saves an .Rdata file named indicated by argument name, containing available chronograms from Open Tree of Life. Saves to path indicated by argument path.
<code>updated_name</code>	Used if <code>write = TRUE</code> . Defaults to "opentree_chronograms_updated". A character vector of length one indicating the name to assign to both the updated OpenTree chronogram database object and the ".Rdata" file. For example, if <code>name = "my_database"</code> , the function will assign the updated chronogram database to an object named <code>my_database</code> and will write it to a file named "my_database.Rdata" in the path indicated by argument <code>file_path</code> .
<code>file_path</code>	Used if <code>write = TRUE</code> . A character vector of length 1 indicating the path to write the updated database ".Rdata" file to, excluding file name. Defaults to temporary directory obtained with <code>base::tempdir()</code> and formatted with <code>base::file.path()</code> .
<code>...</code>	Arguments passed on to <code>get_opentree_chronograms</code>
<code>max_tree_count</code>	Default to "all", it gets all available chronograms. For testing purposes, a numeric value indicating the max number of trees to be cached.

Value

A list of 4 elements:

authors A list of lists of author names of the original studies that published chronograms currently stored in the Open Tree of Life database.

curators A list of lists of curator names that uploaded chronograms to the Open Tree of Life database.

studies A list of study identifiers from original studies that published chronograms currently stored in the Open Tree of Life database.

trees A multiPhylo object storing the chronograms from Open Tree of Life database.

update A character vector indicating the time when the database object was last updated.

version A character vector indicating when the chronogram database opentree_chronograms object was last updated. Format is year.month.day

Index

- * **chronogram**
 - opentree_chronograms, 21
 - problematic_chronograms, 23
- * **dates**
 - opentree_chronograms, 21
- * **million**
 - opentree_chronograms, 21
- * **myrs**
 - opentree_chronograms, 21
- * **opentree**
 - opentree_chronograms, 21
 - problematic_chronograms, 23
- * **phylogeny**
 - opentree_chronograms, 21
- * **time**
 - opentree_chronograms, 21
- * **tnrs**
 - problematic_chronograms, 23
- * **tree**
 - problematic_chronograms, 23
- * **unmapped**
 - problematic_chronograms, 23
- * **years**
 - opentree_chronograms, 21
- .get_ott_lineage, 2
- base::file.path(), 28
- base::tempdir(), 28
- check_ott_input, 3
- classification_paths_from_taxonomy, 3
- clean_ott_chronogram, 7
- clean_taxon_info_children, 8
- clean_tnrs, 8
- date_with_pdb, 9
- extract_ott_ids, 10
- get_fossil_range, 11
- get_opentree_chronograms, 11, 28
- get_opentree_chronograms(), 28
- get_ottl_chronograms
 - (get_opentree_chronograms), 11
- get_ott_children, 12
- get_ott_children(), 3
- get_ott_clade, 13
- get_ott_clade(), 3
- get_ott_lineage, 14
- get_valid_children, 15
- get_valid_children(), 12
- is_good_chronogram, 16
- is_phylo, 16
- make_all_associations, 17
- make_contributor_cache, 17
- make_ottl_associations, 18
- make_overlap_table, 18
- make_treebase_associations, 19
- make_treebase_cache, 19
- map_nodes_ott, 20
- opentree_chronograms, 21
- paleotree::timePaleoPhy(), 9
- phylo_has_brlen, 22
- phylo_tiplabel_space_to_underscore, 22
- phylo_tiplabel_underscore_to_space, 23
- problematic_chronograms, 23
- recover_mrcaott, 24
- rotl::taxonomy_taxon_info(), 2, 3, 8, 12, 14, 15
- rotl::tnrs_match_names, 25
- rotl::tnrs_match_names(), 3, 8, 9, 12, 14, 15
- rotl::tol_subtree(), 12
- summarize_fossil_range, 24
- taxonomy_taxon_info(), 8

tnrs_contexts, [25](#)
tnrs_match, [25](#)
tnrs_match(), [3](#), [8](#), [9](#), [12](#), [14](#), [15](#)
tree_from_taxonomy, [26](#)

update_all_cached, [27](#)
update_datelife_cache, [28](#)