Package ‘treebase’

May 8, 2026

Type Package

Title Discovery, Access and Manipulation of 'TreeBASE' Phylogenies

Version 0.1.5

Description Interface to the API for 'TreeBASE' <http://treebase.org> from 'R.' 'TreeBASE' is a repository of user-submitted phylogenetic trees (of species, population, or genes) and the data used to create them.

License CC0

Encoding UTF-8

URL https://docs.ropensci.org/treebase/,
https://github.com/ropensci/treebase

BugReports https://github.com/ropensci/treebase/issues

Depends R (>= 2.15), ape

Imports XML, RCurl, methods, utils, httr

Suggests testthat, knitr, rmarkdown

RoxygenNote 7.3.1

VignetteBuilder knitr

NeedsCompilation no

Author Carl Boettiger [aut, cre],
Duncan Temple Lang [aut]

Maintainer Carl Boettiger <cboettig@gmail.com>

Repository CRAN

Date/Publication 2024-02-16 06:20:02 UTC

Contents

`cache_treebase` *A function to cache the phylogenies in treebase locally*

Description

A function to cache the phylogenies in treebase locally

Usage

```
cache_treebase(
  file = paste("treebase-", Sys.Date(), ".rda", sep = ""),
  pause1 = 3,
  pause2 = 3,
  attempts = 10,
  max_trees = Inf,
  only_metadata = FALSE,
  save = TRUE
)
```

Arguments

`file`	filename for the cache, otherwise created with datestamp
`pause1`	number of seconds to hesitate between requests
`pause2`	number of seconds to hesitate between individual files
`attempts`	number of attempts to access a particular resource
`max_trees`	maximum number of trees to return (default is Inf)
`only_metadata`	option to only return metadata about matching trees
`save`	logical indicating whether to save a file with the resuls.

Details

it's a good idea to let this run overnight

Value

saves a cached file of treebase

Examples

```
## Not run:
 treebase <- cache_treebase()

## End(Not run)
```

`download_metadata` *Download the metadata on treebase using the OAI-MPH interface*

Description

Download the metadata on treebase using the OAI-MPH interface

Usage

```
download_metadata(
  query = "",
  by = c("all", "until", "from"),
  curl = getCurlHandle()
)
```

Arguments

`query`	a date in format yyyy-mm-dd
`by`	return all data "until" that date, "from" that date to current, or "all"
`curl`	if calling in series many times, call getCurlHandle() first and then pass the return value in here. Avoids repeated handshakes with server.

Details

query must be#' download_metadata(2010-01-01, by="until") all isn't a real query type, but will return all trees regardless of date

Examples

```
## Not run:
Near <- search_treebase("Near", "author", max_trees=1)
 metadata(Near[[1]]$S.id)
## or manualy give a sudy id
metadata("2377")

### get all trees from a certain depostition date forwards ##
m <- download_metadata("2009-01-01", by="until")
## extract any metadata, e.g. publication date:
dates <- sapply(m, function(x) as.numeric(x$date))
hist(dates, main="TreeBase growth", xlab="Year")

### show authors with most tree submissions in that date range
```

```
authors <- sapply(m, function(x){
   index <- grep( "creator", names(x))
     x[index]
})
a <- as.factor(unlist(authors))
head(summary(a))

## Show growth of TreeBASE
all <- download_metadata("", by="all")
dates <- sapply(all, function(x) as.numeric(x$date))
hist(dates, main="TreeBase growth", xlab="Year")

## make a barplot submission volume by journals
journals <- sapply(all, function(x) x$publisher)
J <- tail(sort(table(as.factor(unlist(journals)))),5)
b<- barplot(as.numeric(J))
text(b, names(J), srt=70, pos=4, xpd=T)

## End(Not run)
```

`drop_nontrees` *drop errors from the search*

Description

drop errors from the search

Usage

```
drop_nontrees(tr)
```

Arguments

`tr`	a list of phylogenetic trees returned by search_treebase

Details

primarily for the internal use of search_treebase, but may be useful

Value

the list of phylogenetic trees returned successfully

`dryad_metadata` *Search the dryad metadata archive*

Description

Search the dryad metadata archive

Usage

```
dryad_metadata(study.id, curl = getCurlHandle())
```

Arguments

`study.id`	the dryad identifier
`curl`	if calling in series many times, call getCurlHandle() first and then pass the return value in here. Avoids repeated handshakes with server.

Value

a list object containing the study metadata

Examples

```
## Not run:
  dryad_metadata("10255/dryad.12")

## End(Not run)
```

`have_branchlength` *Simple function to identify which trees have branch lengths*

Description

Simple function to identify which trees have branch lengths

Usage

```
have_branchlength(trees)
```

Arguments

`trees`	a list of phylogenetic trees (ape/phylo format)

Value

logical string indicating which have branch length data

`metadata` *metadata.rda*

Description

Contains a cache of all publication metadata the search_metadata() to pull down when run on 2012-05-12.

Usage

```
metadata(phylo.md = NULL, oai.md = NULL)
```

Arguments

`phylo.md`	cached phyloWS (tree) metadata, (optional)
`oai.md`	cached OAI-PMH (study) metadata (optional)

Details

recreate with: `search_metadata()`

Value

a data frame of all available metadata, (as a data.table object) columns are: "Study.id", "Tree.id", "kind", "type", "quality", "ntaxa" "date", "publisher", "author", "title".

Examples

```
## Not run:
meta <- metadata()
meta[publisher %in% c("Nature", "Science") & ntaxa > 50 & kind == "Species Tree",]

## End(Not run)
```

`search_treebase` *A function to pull in the phyologeny/phylogenies matching a search query*

Description

A function to pull in the phyologeny/phylogenies matching a search query

Usage

```
search_treebase(
  input,
  by,
  returns = c("tree", "matrix"),
  exact_match = FALSE,
  max_trees = Inf,
  branch_lengths = FALSE,
  curl = getCurlHandle(),
  verbose = TRUE,
  pause1 = 0,
  pause2 = 0,
  attempts = 3,
  only_metadata = FALSE
)
```

Arguments

`input`	a search query (character string)
`by`	the kind of search; author, taxon, subject, study, etc (see list of possible search terms, details)
`returns`	should the fn return the tree or the character matrix?
`exact_match`	force exact matching for author name, taxon, etc. Otherwise does partial matching
`max_trees`	Upper bound for the number of trees returned, good for keeping possibly large initial queries fast
`branch_lengths`	logical indicating whether should only return trees that have branch lengths.
`curl`	the handle to the curl web utility for repeated calls, see the getCurlHandle() function in RCurl package for details.
`verbose`	logical indicating level of progress reporting
`pause1`	number of seconds to hesitate between requests
`pause2`	number of seconds to hesitate between individual files
`attempts`	number of attempts to access a particular resource
`only_metadata`	option to only return metadata about matching trees which lists study.id, tree.id, kind (gene,species,barcode) type (single, consensus) number of taxa, and possible quality score.

Value

either a list of trees (multiphylo) or a list of character matrices

Examples

```
## Not run:
## defaults to return phylogeny
```

```
Huelsenbeck <- search_treebase("Huelsenbeck", by="author")

## can ask for character matrices:
wingless <- search_treebase("2907", by="id.matrix", returns="matrix")

## Some nexus matrices don't meet read.nexus.data's strict requirements,
## these aren't returned
H_matrices <- search_treebase("Huelsenbeck", by="author", returns="matrix")

## Use Booleans in search: and, or, not
## Note that by must identify each entry type if a Boolean is given
HR_trees <- search_treebase("Ronquist or Hulesenbeck", by=c("author", "author"))

## We'll often use max_trees in the example so that they run quickly,
## notice the quotes for species.
dolphins <- search_treebase('"Delphinus"', by="taxon", max_trees=5)
## can do exact matches
humans <- search_treebase('"Homo sapiens"', by="taxon", exact_match=TRUE, max_trees=10)
## all trees with 5 taxa
five <- search_treebase(5, by="ntax", max_trees = 10)
## These are different, a tree id isn't a Study id.  we report both
studies <- search_treebase("2377", by="id.study")
tree <- search_treebase("2377", by="id.tree")
c("TreeID" = tree$Tr.id, "StudyID" = tree$S.id)
## Only results with branch lengths
## Has to grab all the trees first, then toss out ones without branch_lengths
Near <- search_treebase("Near", "author", branch_lengths=TRUE)

## End(Not run)
```

`treebase` *treebase.rda*

Description

Contains a cache of all phylogenies `cache_treebase()` function was able to pull down when run on 2012-05-14.

Details

recreate with: `cache_treebase()`

Index