

Package ‘treeio’

April 16, 2019

Title Base Classes and Functions for Phylogenetic Tree Input and Output

Version 1.6.2

Description Base classes and functions for parsing and exporting phylogenetic trees. 'treeio' supports parsing analysis findings from commonly used software packages, allows linking external data to phylogeny and merging tree data obtained from different sources. It also supports exporting phylogenetic tree with heterogeneous associated data to a single tree file.

Depends R (>= 3.4.0)

Imports ape, dplyr, jsonlite, magrittr, methods, rlang, rvcheck, tibble, tidytree (>= 0.1.7), utils

Suggests ggplot2, ggtree, knitr, prettydoc, testthat, tidy

VignetteBuilder knitr

ByteCompile true

License Artistic-2.0

URL <https://guangchuangyu.github.io/software/treeio>

BugReports <https://github.com/GuangchuangYu/treeio/issues>

Encoding UTF-8

LazyData true

biocViews Software, Annotation, Clustering, DataImport, DataRepresentation, Alignment, MultipleSequenceAlignment

RoxygenNote 6.1.1

git_url <https://git.bioconductor.org/packages/treeio>

git_branch RELEASE_3_8

git_last_commit 4fb11e8

git_last_commit_date 2019-01-25

Date/Publication 2019-04-15

Author Guangchuang Yu [aut, cre] (<<https://orcid.org/0000-0002-6485-8781>>),
Tommy Tsan-Yuk Lam [ctb, ths],
Casey Dunn [ctb],
Bradley Jones [ctb],
Tyler Bradley [ctb]

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

R topics documented:

as.treedata.phylo	3
drop.tip	3
get.placements	4
get.tree	5
get.treetext	5
getNodeNum	6
is.ggtree	6
jplace-class	7
label_branch_paml	7
mask	8
merge_tree	8
Nnode.treedata	9
phyPML	9
print.beastList	10
raxml2nwk	10
read.astral	11
read.beast	11
read.codeml	12
read.codeml_mlc	12
read.fasta	13
read.hyphy	14
read.hyphy.seq	14
read.iqtree	15
read.jplace	15
read.jtree	16
read.newick	16
read.nhx	17
read.paml_rst	17
read.phylip	18
read.phylip.seq	19
read.phylip.tree	19
read.phyloT	20
read.r8s	20
read.raxml	21
rename_taxa	21
taxa_rename	22
tree_subset	22
write.beast	23
write.jplace	24
write.jtree	25

as.treedata.phylo *as.treedata*

Description

convert phylo to treedata

Usage

```
## S3 method for class 'phylo'
as.treedata(tree, boot = NULL, ...)
```

Arguments

tree	input tree, a phylo object
boot	optional, can be bootstrap value from ape::boot.phylo
...	additional parameters

Details

converting phylo object to treedata object

Author(s)

guangchuang yu

drop.tip *drop.tip method*

Description

drop.tip method
drop.tip method

Usage

```
drop.tip(object, tip, ...)
```

```
drop.tip(object, tip, ...)
```

```
## S4 method for signature 'phylo'
drop.tip(object, tip, ...)
```

Arguments

object	A treedata or phylo object
tip	a vector of mode numeric or character specifying the tips to delete
...	additional parameters

Value

updated object

Author(s)

Casey Dunn <http://dunnlab.org> and Guangchuang Yu <https://guangchuangyu.github.io>

Source

drop.tip for phylo object is a wrapper method of ape::drop.tip from the ape package. The documentation you should read for the drop.tip function can be found here: [drop.tip](#)

See Also

[drop.tip](#)

Examples

```
nhxfile <- system.file("extdata/NHX", "ADH.nhx", package="treeio")
nhx <- read.nhx(nhxfile)
drop.tip(nhx, c("ADH2", "ADH1"))
```

get.placements	<i>get.placements</i>
----------------	-----------------------

Description

access placement information

Usage

```
get.placements(tree, ...)  
  
## S3 method for class 'jplace'  
get.placements(tree, by = "best", ...)
```

Arguments

tree	tree object
...	additional parameters
by	one of 'best' and 'all'

Value

placement tibble

`get.tree`*get.tree*

Description

access phylo slot

Usage

```
get.tree(x, ...)
```

Arguments

x	tree object
...	additional parameters

Value

phylo object

Author(s)

guangchuang yu

`get.treetext`*get.treetext method*

Description

access tree text (newick text) from tree object

Usage

```
get.treetext(object, ...)
```

```
## S4 method for signature 'treedata'  
get.treetext(object)
```

Arguments

object	treedata object
...	additional parameter

Value

phylo object

`getNodeNum`*getNodeNum*

Description

calculate total number of nodes

Usage`getNodeNum(tree)``Nnode2(tree)`**Arguments**

`tree` tree object

Value

number

Author(s)

Guangchuang Yu

Examples

```
getNodeNum(rtree(30))
Nnode2(rtree(30))
```

`is.ggtree`*is.ggtree*

Description

test whether input object is produced by ggtree function

Usage`is.ggtree(x)`**Arguments**

`x` object

Value

TRUE or FALSE

Author(s)

guangchuang yu

jplace-class	<i>Class "jplace" This class stores phylogenetic placements</i>
--------------	---

Description

Class "jplace" This class stores phylogenetic placements

Slots

phylo phylo object for tree structure
 treetext newick tree string
 data associated data
 extraInfo extra information, reserve for merge_tree
 file tree file
 placements reserve for jplace file to store placement information
 info extra information, e.g. metadata, software version etc.

Author(s)

guangchuang yu <https://guangchuangyu.github.io>

label_branch_paml	<i>label_branch_paml</i>
-------------------	--------------------------

Description

label branch for PAML to infer selection pressure using branch model

Usage

```
label_branch_paml(tree, node, label)
```

Arguments

tree	phylo object
node	node number
label	label of branch, e.g. #1

Value

updated phylo object

Author(s)

guangchuang yu

mask	<i>mask</i>
------	-------------

Description

site mask

Usage

```
mask(tree_object, field, site, mask_site = FALSE)
```

Arguments

tree_object	tree object
field	selected field
site	site
mask_site	if TRUE, site will be masked. if FALSE, selected site will not be masked, while other sites will be masked.

Value

updated tree object

Author(s)

Guangchuang Yu

merge_tree	<i>merge_tree</i>
------------	-------------------

Description

merge two tree object

Usage

```
merge_tree(obj1, obj2)
```

Arguments

obj1	tree object 1
obj2	tree object 2

Value

tree object

Author(s)

Guangchuang Yu

Nnode.treedata	<i>Nnode</i>
----------------	--------------

Description

number of nodes

Usage

```
## S3 method for class 'treedata'
Nnode(phy, internal.only = TRUE, ...)
```

Arguments

phy	treedata object
internal.only	whether only count internal nodes
...	additional parameters

Value

number of nodes

Author(s)

guangchuang yu

Examples

```
Nnode(rtree(30))
```

phyPML	<i>treeAnno.pml</i>
--------	---------------------

Description

tree annotation of sequence substitution by comparing to parent node

Usage

```
phyPML(pmlTree, type = "ml")
```

Arguments

pmlTree	tree in pml object, output of phangorn::optim.pml
type	one of 'ml' and 'bayes' for inferring ancestral sequences

Value

phangorn object

Author(s)

Yu Guangchuang

```
print.beastList      print
```

Description

print information of a list of beast trees

Usage

```
## S3 method for class 'beastList'
print(x, ...)
```

Arguments

```
x          a list of beast object
...        no used
```

Value

message

```
raxml2nwk      raxml2nwk
```

Description

convert raxml bootstrap tree to newick format

Usage

```
raxml2nwk(infile, outfile = "raxml.tree")
```

Arguments

```
infile      input file
outfile     output file
```

Value

newick file

Author(s)

Guangchuang Yu

read.astral	<i>read.astral</i>
-------------	--------------------

Description

parse ASTRAL output newick text

Usage

```
read.astral(file)
```

Arguments

file ASTRAL Newick file

Value

treedata object

Author(s)

Guangchuang Yu

Examples

```
tt <- "((species1,(species2,species3)'[pp1=0.75;pp2=0.24;pp3=0.01]':1.2003685744180805)'[pp1=0.98;pp2=0.02;pp3=0.01]':1.2003685744180805)";
read.astral(textConnection(tt))
```

read.beast	<i>read.beast</i>
------------	-------------------

Description

read beast output

Usage

```
read.beast(file)
```

```
read.mrbayes(file)
```

Arguments

file beast file

Value

beast object

Author(s)

Guangchuang Yu <https://guangchuangyu.github.io>

Examples

```
file <- system.file("extdata/BEAST", "beast_mcc.tree", package="treeio")
read.beast(file)
```

read.codeml	<i>read.codeml</i>
-------------	--------------------

Description

read baseml output

Usage

```
read.codeml(rstfile, mlcfile, tree = "mlc", type = "Joint")
```

Arguments

rstfile	rst file
mlcfile	mlc file
tree	one of 'mlc' or 'rst'
type	one of 'Marginal' or 'Joint'

Value

A treedata object

Author(s)

Guangchuang Yu

Examples

```
rstfile <- system.file("extdata/PAML_Codeml", "rst", package="treeio")
mlcfile <- system.file("extdata/PAML_Codeml", "mlc", package="treeio")
read.codeml(rstfile, mlcfile)
```

read.codeml_mlc	<i>read.codeml_mlc</i>
-----------------	------------------------

Description

read mlc file of codeml output

Usage

```
read.codeml_mlc(mlcfile)
```

Arguments

mlcfile	mlc file
---------	----------

Value

A `codeml_mlc` object

Author(s)

ygc

Examples

```
mlcfile <- system.file("extdata/PAML_Codeml", "mlc", package="treeio")
read.codeml_mlc(mlcfile)
```

<code>read.fasta</code>	<i>read.fasta</i>
-------------------------	-------------------

Description

read FASTA file

Usage

```
read.fasta(fasta)
```

Arguments

`fasta` fasta file

Details

This function supports both DNA or AA sequences

Value

DNABin or AABin object

Author(s)

guangchuang yu

read.hyphy *read.hyphy*

Description

read HYPHY output

Usage

```
read.hyphy(nwk, ancseq, tip.fasfile = NULL)
```

Arguments

nwk	tree file in nwk format, one of hyphy output
ancseq	ancestral sequence file in nexus format, one of hyphy output
tip.fasfile	tip sequence file

Value

A hyphy object

Author(s)

Guangchuan Yu <https://guangchuangyu.github.io>

Examples

```
nwk <- system.file("extdata/HYPHY", "labelledtree.tree", package="treeio")
ancseq <- system.file("extdata/HYPHY", "ancseq.nex", package="treeio")
read.hyphy(nwk, ancseq)
```

read.hyphy.seq *read.hyphy.seq*

Description

parse sequences from hyphy output

Usage

```
read.hyphy.seq(file)
```

Arguments

file	output of hyphy ancestral sequence inference; nexus format
------	--

Value

DNABin object

Author(s)

guangchuang yu

Examples

```
ancseq <- system.file("extdata/HYPHY", "ancseq.nex", package="treeio")  
read.hyphy.seq(ancseq)
```

read.iqtree *read.iqtree*

Description

parse IQ-TREE output

Usage

```
read.iqtree(file)
```

Arguments

file IQ-TREE Newick text

Value

treedata object

Author(s)

Guangchuang Yu

read.jplace *read.jplace*

Description

read jplace file

Usage

```
read.jplace(file)
```

Arguments

file jplace file

Value

jplace instance

Author(s)

ygc

Examples

```
jp <- system.file("extdata", "sample.jplace", package="treeio")
read.jplace(jp)
```

`read.jtree`*read.jtree*

Description

Import tree data from jtree file, which is JSON-based text and probably output by write.jtree

Usage

```
read.jtree(file)
```

Arguments

file tree file

Value

treedata object

Author(s)

guangchuang yu

`read.newick`*read.newick*

Description

read newick tree

Usage

```
read.newick(file, node.label = "label", ...)
```

Arguments

file newick file
node.label parse node label as 'label' or 'support' value
... additional parameter, passed to 'read.tree'

Value

phylo or treedata object

Author(s)

guangchuang yu

read.nhx

read.nhx

Description

read nhx tree file

Usage

```
read.nhx(file)
```

Arguments

file nhx file

Value

nhx object

Author(s)

Guangchuang Yu <https://guangchuangyu.github.io>

Examples

```
nhxfile <- system.file("extdata/NHX", "ADH.nhx", package="treeio")
read.nhx(nhxfile)
```

read.paml_rst

read.paml_rst

Description

read rst file from paml (both baseml and codeml) output

Usage

```
read.paml_rst(rstfile, type = "Joint")
```

Arguments

rstfile rst file
type one of 'Marginal' or 'Joint'

Value

A treedata object

Author(s)

Guangchuang Yu <https://guangchuangyu.github.io>

Examples

```
rstfile <- system.file("extdata/PAML_Baseml", "rst", package="treeio")
read.paml_rst(rstfile)
```

read.phytip

read.phytip

Description

parsing phytip tree format

Usage

```
read.phytip(file)
```

Arguments

file phytip file

Value

an instance of 'phytip'

Author(s)

Guangchuang Yu

Examples

```
phyfile <- system.file("extdata", "sample.phy", package="treeio")
read.phytip(phyfile)
```

`read.phylip.seq` *read.phylip.seq*

Description

read aligned sequences from phylip format

Usage

`read.phylip.seq(file)`

Arguments

`file` phylip file, currently only sequential format is supported

Value

DNABin object

Author(s)

guangchuang yu

References

<http://evolution.genetics.washington.edu/phylip/doc/sequence.html>

`read.phylip.tree` *read.phylip.tree*

Description

parse tree from phylip file

Usage

`read.phylip.tree(file)`

Arguments

`file` phylip file

Value

phylo or multiPhylo object

Author(s)

guangchuang yu

read.phyloT	<i>read.phyloT</i>
-------------	--------------------

Description

parse output from phyloT

Usage

```
read.phyloT(file, ...)
```

Arguments

file	newick tree file
...	additional parameters to read.tree

Value

phylo object

Author(s)

guangchuang yu

References

<http://phylot.biobyte.de/>

read.r8s	<i>read.r8s</i>
----------	-----------------

Description

parse output from r8s

Usage

```
read.r8s(file)
```

Arguments

file	r8s output log file
------	---------------------

Value

multiPhylo object

Author(s)

Guangchuang Yu

Examples

```
read.r8s(system.file("extdata/r8s", "H3_r8s_output.log", package="treeio"))
```

read.raxml	<i>read.raxml</i>
------------	-------------------

Description

parse RAxML bootstrapping analysis output

Usage

```
read.raxml(file)
```

Arguments

file	RAxML bootstrapping analysis output
------	-------------------------------------

Value

treedata object

Author(s)

Guangchuang Yu

Examples

```
raxml_file <- system.file("extdata/RAxML", "RAxML_bipartitionsBranchLabels.H3", package="treeio")
read.raxml(raxml_file)
```

rename_taxa	<i>rename_taxa</i>
-------------	--------------------

Description

rename tip label of phylogenetic tree

Usage

```
rename_taxa(tree, data, key, value)
```

Arguments

tree	tree object, either treedata or phylo
data	data frame
key	column in data that match tip label
value	column in data for rename tip label

Value

tree object

Author(s)

Guangchuang Yu

taxa_rename	<i>taxa_rename</i>
-------------	--------------------

Description

rename taxa

Usage

```
taxa_rename(tree, name)
```

Arguments

tree	tree object
name	a two column data.frame contains original name in 1st column and new name in 2nd column

Value

updated tree object with new taxa name

Author(s)

guangchuang yu

tree_subset	<i>Subset tree objects by related nodes</i>
-------------	---

Description

This function allows for a tree object to be subset by specifying a node and returns all related nodes within a selected number of levels

Usage

```
tree_subset(tree, node, levels_back = 5, group_node = TRUE)
```

```
## S3 method for class 'phylo'
tree_subset(tree, node, levels_back = 5,
  group_node = TRUE)
```

```
## S3 method for class 'treedata'
tree_subset(tree, node, levels_back = 5,
  group_node = TRUE)
```

Arguments

tree	a tree object of class phylo
node	either a tip label or a node number for the given tree that will be the focus of the subsetted tree
levels_back	a number specifying how many nodes back from the selected node the subsetted tree should include
group_node	whether add grouping information of selected node

Details

This function will take a tree and a specified node from that tree and subset the tree showing all relatives back to a specified number of nodes. This function allows for a combination of ancestor and offspring to return a subsetted tree that is of class phylo. This allows for easy graphing of the tree with ggtree

Examples

```
## Not run:
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)

sub_tree <- tree_subset(tree, node = "A", levels_back = 3)
ggtree(sub_tree) + geom_tiplab() + geom_nodelab()

## End(Not run)

## Not run:
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)

sub_tree <- tree_subset(tree, node = "A", levels_back = 3)
ggtree(sub_tree) + geom_tiplab() + geom_nodelab()

## End(Not run)
```

write.beast

write.beast

Description

Export treedata object to BEAST NEXUS file. This function was adopted and modified from ape::write.nexus

Usage

```
write.beast(treedata, file = "", translate = TRUE,
            tree.name = "UNTITLED")
```

Arguments

treedata	treedata object
file	output file. If file = "", print the output content on screen
translate	whether translate taxa labels
tree.name	name of the tree

Value

output file or file content on screen

Author(s)

guangchuang yu

Examples

```
nhxfile <- system.file("extdata/NHX", "phyldog.nhx", package="treeio")
nhx <- read.nhx(nhxfile)
write.beast(nhx)
```

write.jplace

write.jplace

Description

generate jplace file

Usage

```
write.jplace(nwk, data, outfile)
```

Arguments

nwk	tree in newick format
data	annotation data
outfile	jplace output file

Value

jplace file

Author(s)

ygc

`write.jtree`

write.jtree

Description

Export treedata object to json tree file

Usage

```
write.jtree(treedata, file = "")
```

Arguments

`treedata` treedata object

`file` output file. If file = "", print the output content on screen

Value

output file or file content on screen

Author(s)

guangchuang yu

Index

*Topic classes

- [jplace-class, 7](#)
- [as.treedata.phylo, 3](#)
- [drop.tip, 3, 4](#)
- [drop.tip, phylo \(drop.tip\), 3](#)
- [drop.tip, phylo-method \(drop.tip\), 3](#)
- [drop.tip, treedata \(drop.tip\), 3](#)
- [drop.tip, treedata-method \(drop.tip\), 3](#)

- [get.placements, 4](#)
- [get.tree, 5](#)
- [get.treetext, 5](#)
- [get.treetext, treedata-method \(get.treetext\), 5](#)
- [getNodeNum, 6](#)

- [is.ggtree, 6](#)

- [jplace-class, 7](#)

- [label_branch_paml, 7](#)

- [mask, 8](#)
- [merge_tree, 8](#)

- [Nnode.treedata, 9](#)
- [Nnode2 \(getNodeNum\), 6](#)

- [phyPML, 9](#)
- [print.beastList, 10](#)

- [raxml2nwk, 10](#)
- [read.astral, 11](#)
- [read.beast, 11](#)
- [read.codeml, 12](#)
- [read.codeml_mlc, 12](#)
- [read.fasta, 13](#)
- [read.hyphy, 14](#)
- [read.hyphy.seq, 14](#)
- [read.iqtree, 15](#)
- [read.jplace, 15](#)
- [read.jtree, 16](#)
- [read.mrbayes \(read.beast\), 11](#)

- [read.newick, 16](#)
- [read.nhx, 17](#)
- [read.paml_rst, 17](#)
- [read.phylip, 18](#)
- [read.phylip.seq, 19](#)
- [read.phylip.tree, 19](#)
- [read.phyloT, 20](#)
- [read.r8s, 20](#)
- [read.raxml, 21](#)
- [rename_taxa, 21](#)

- [taxa_rename, 22](#)
- [tree_subset, 22](#)

- [write.beast, 23](#)
- [write.jplace, 24](#)
- [write.jtree, 25](#)