# Package 'treeio'

November 12, 2026

**Title** Base Classes and Functions for Phylogenetic Tree Input and Output

**Version** 1.36.1

**Description** 'treeio' is an R package to make it easier to import and store phylogenetic tree with associated data; and to link external data from different sources to phylogeny. It also supports exporting phylogenetic tree with heterogeneous associated data to a single tree file and can be served as a platform for merging tree with associated data and converting file formats.

**Depends** R (>= 3.6.0)

**Imports** ape, dplyr, jsonlite, magrittr, methods, rlang, stats, tibble, tidytree (>= 0.4.5), utils, yulab.utils (>= 0.1.6)

**Suggests** Biostrings, cli, ggplot2, ggtree, igraph, knitr, rmarkdown, phangorn, prettydoc, purrr, testthat, tidyr, vroom, xml2, yaml

**VignetteBuilder** knitr

**ByteCompile** true

**License** Artistic-2.0

**Encoding** UTF-8

**URL** https://yulab-smu.top/contribution-tree-data/

**BugReports** https://github.com/YuLab-SMU/treeio/issues

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

**RoxygenNote** 7.3.2

**git_url** https://git.bioconductor.org/packages/treeio

**git_branch** RELEASE_3_23

**git_last_commit** 098376f

**git_last_commit_date** 2026-04-30

**Repository** Bioconductor 3.23

**Date/Publication** 2026-11-11

**Author** Guangchuang Yu [aut, cre] (ORCID:
<https://orcid.org/0000-0002-6485-8781>),
Tommy Tsan-Yuk Lam [ctb, ths],
Shuangbin Xu [ctb] (ORCID: <https://orcid.org/0000-0003-3513-5362>),
Bradley Jones [ctb],
Casey Dunn [ctb],
Tyler Bradley [ctb],
Konstantinos Geles [ctb]

**Maintainer** Guangchuang Yu <guangchuangyu@gmail.com>

# Contents

treeio-package . . . 3
as.treedata.phylo . . . 4
find.hclust . . . 4
get.placements . . . 5
get.tree . . . 6
get.treetext . . . 6
getNodeNum . . . 7
is.ggtree . . . 7
jplace-class . . . 8
label_branch_paml . . . 8
mask . . . 9
merge_tree . . . 10
print.treedataList . . . 10
raxml2nwk . . . 11
read.astral . . . 11
read.beast . . . 12
read.codeml . . . 13
read.codeml_mlc . . . 14
read.fasta . . . 14
read.hyphy . . . 15
read.hyphy.seq . . . 16
read.iqtree . . . 16
read.jplace . . . 17
read.jtree . . . 17
read.mcmctree . . . 18
read.mega_tabular . . . 19
read.newick . . . 19
read.nextstrain.json . . . 20
read.nhx . . . 20
read.paml_rst . . . 21
read.phylip . . . 22
read.phylip.seq . . . 22
read.phylip.tree . . . 23
read.phyloxml . . . 23
read.r8s . . . 24

read.raxml . . . . . 25
read.treeqza . . . . . 25
read.treetime . . . . . 26
reexports . . . . . 27
rename_taxa . . . . . 27
rescale_tree . . . . . 28
spt . . . . . 28
write.beast . . . . . 29
write.beast.newick . . . . . 30
write.jplace . . . . . 31
write.jtree . . . . . 31

**Index** **33**

---

`treeio-package` *treeio: Base Classes and Functions for Phylogenetic Tree Input and Output*

---

## Description

'treeio' is an R package to make it easier to import and store phylogenetic tree with associated data; and to link external data from different sources to phylogeny. It also supports exporting phylogenetic tree with heterogeneous associated data to a single tree file and can be served as a platform for merging tree with associated data and converting file formats.

## Author(s)

**Maintainer**: Guangchuang Yu <guangchuangyu@gmail.com> (ORCID)

Other contributors:

- Tommy Tsan-Yuk Lam <tylam.tommy@gmail.com> [contributor, thesis advisor]
- Shuangbin Xu <xshuangbin@163.com> (ORCID) [contributor]
- Bradley Jones <brj1@sfu.ca> [contributor]
- Casey Dunn <casey_dunn@brown.edu> [contributor]
- Tyler Bradley <tcb85@drexel.edu> [contributor]
- Konstantinos Geles <konstantinos.geles@studenti.unicz.it> [contributor]

## See Also

Useful links:

- https://yulab-smu.top/contribution-tree-data/
- Report bugs at https://github.com/YuLab-SMU/treeio/issues

`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

`find.hclust` *find the hierarchical cluster analysis among the nodes of graph based on the length of all the shortest paths in the graph.*

## Description

find the hierarchical cluster analysis among the nodes of graph based on the length of all the shortest paths in the graph.

## Usage

```
find.hclust(
  x,
  graph.mst = FALSE,
  weights = NULL,
  hclust.method = "average",
  ...
)
```

### Arguments

| | |
|---|---|
| x | a igraph object |
| graph.mst | logical whether obtain the minimum spanning tree first then find.hclust, default is FALSE. |
| weights | a numeric vector giving edge weights or a character. If this is NULL and the graph has a weight edge attribute, then the attribute is used. If this is NA then no weights are used even if the graph has a weight attribute. If this is a character, the graph has the edge attribute which is numeric, then it will be used, default is NULL. |
| hclust.method | the agglomeration method to be used, This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). |
| ... | additional parameters |

### Value

hclust object

### Examples

```
library(igraph)
set.seed(123)
g <- igraph::sample_gnp(100, .1) %>%
     set_edge_attr(name='weight', value=abs(rnorm(E(.),3)))
tr1 <- find.hclust(g, weights = NA)
tr2 <- find.hclust(g)
tr3 <- find.hclust(g, graph.mst = TRUE)
```

---

## get.placements *get.placements*

---

### 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, ...)
```

### 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 *Class "jplace" This class stores phylogenetic placements*

---

### 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 *label_branch_paml*

---

### 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` *mask*

---

### 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 *merge_tree*

### 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

## print.treedataList *print*

### Description

print information of a list of treedata objects

### Usage

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

### Arguments

| | |
|---|---|
| x | a list of treedata objects |
| ... | 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 *read.astral*

## Description

parse ASTRAL output newick text

## Usage

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

## Arguments

| | |
|---|---|
| file | ASTRAL Newick file |

## Value

treedata object

## Author(s)

Guangchuang Yu

## Examples

```
tt <- paste0(
  "((species1,(species2,species3)'[pp1=0.75;pp2=0.24;pp3=0.01]':",
  "1.2003685744180805)'[pp1=0.98;pp2=0.02;pp3=0]':0.9679599282730038,",
  "((species4,species5)'[pp1=0.88;pp2=0.11;pp3=0.01]':1.2454851536484994))"
)
read.astral(textConnection(tt))
```

---

`read.beast` *read.beast*

---

## Description

read beast/mrbayes/mega Nexus output

read beast/mrbayes/mega newick file format

## Usage

```
read.beast(file, threads = 1, verbose = FALSE)

read.mrbayes(file, threads = 1, verbose = FALSE)

read.beast.newick(file, threads = 1, verbose = FALSE)

read.mega(file, threads = 1, verbose = FALSE)
```

## Arguments

| | |
|---|---|
| `file` | newick file |
| `threads` | number of threads for multithreading (default: 1) |
| `verbose` | set TRUE to log progress (default: FALSE) |

## Value

treedata object

treedata object

## Author(s)

Guangchuang Yu https://guangchuangyu.github.io

Bradley R Jones

## Examples

```
file <- system.file("extdata/BEAST", "beast_mcc.tree", package="treeio")
read.beast(file)
file <- system.file("extdata/MrBayes", "Gq_nxs.tre", package="treeio")
read.mrbayes(file)
tr <- read.beast.newick(
        textConnection(
          '(a[&rate=1]:2,(b[&rate=1.1]:1,c[&rate=0.9]:1)[&rate=1]:1);'
        )
)
```

---

`read.codeml` *read.codeml*

---

## 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 *read.codeml_mlc*

## Description

read mlc file of codeml output

## Usage

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

## Arguments

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

## Value

A codeml_mlc object

## Author(s)

Guangchuang Yu

## Examples

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

read.fasta *read.fasta*

## Description

read FASTA file

## Usage

```
read.fasta(fasta, type = "auto")
```

## Arguments

| | |
|---|---|
| fasta | fasta file |
| type | sequence type of the input file, one of 'NT' or 'AA'. Default is 'auto' and guess the sequence type automatically |

### 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)

Guangchuang 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)

Guangchuang Yu

### 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.mcmctree` *read.mcmctree*

---

## Description

read MCMCTree output Tree

## Usage

```
read.mcmctree(file, force.ultrametric = FALSE)
```

## Arguments

| | |
|---|---|
| `file` | the output tree file of MCMCTree |
| `force.ultrametric` | logical whether convert the tree to be ultrametric, if it is not ultrametric, default is FALSE. When the tree is ultrametric, branch times will be calculated automatically. |

## Value

treedata object

## Examples

```
file <- system.file("extdata/MCMCTree", "mcmctree_output.tree", package="treeio")
tr <- read.mcmctree(file)
tr
```

`read.mega_tabular` *read.mega_tabular*

## Description

parse tabular output of MEGA

## Usage

```
read.mega_tabular(file)
```

## Arguments

| | |
|---|---|
| `file` | MEGA tabular 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.nextstrain.json *read.nextstrain.json*

### Description

read.nextstrain.json

### Usage

```
read.nextstrain.json(x)
```

### Arguments

| | |
|---|---|
| x | the json tree file of auspice from nextstrain. |

### Value

treedata object

### Author(s)

Shuangbin Xu

### Examples

```
file1 <- system.file("extdata/nextstrain.json", "minimal_v2.json", package="treeio")
tr <- read.nextstrain.json(file1)
tr
```

## 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.phylip` *read.phylip*

## Description

parsing phylip tree format

## Usage

```
read.phylip(file)
```

## Arguments

| | |
|---|---|
| `file` | phylip file |

## Value

an instance of 'phylip'

## Author(s)

Guangchuang Yu

## Examples

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

---

### Description

read.phyloxml

### Usage

```
read.phyloxml(file)
```

### Arguments

| | |
|---|---|
| file | phyloxml file |

## Value

treedata class or treedataList class

## Examples

```
xmlfile1 <- system.file("extdata/phyloxml", "test_x2.xml", package="treeio")
px1 <- read.phyloxml(xmlfile1)
px1
xmlfile2 <- system.file("extdata/phyloxml", "phyloxml_examples.xml", package="treeio")
px2 <- read.phyloxml(xmlfile2)
px2
```

---

# read.r8s *read.r8s*

---

## 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 *read.raxml*

### 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)
```

## read.treeqza *read.treeqza*

### Description

read.treeqza

### Usage

```
read.treeqza(treeqza, node.label = "label", ...)
```

### Arguments

| | |
|---|---|
| treeqza | the qiime2 output file contained tree file. |
| node.label | parse node label as 'label' or 'support' value. |
| ... | additional parameter, passed to 'read.tree'. |

## Value

phylo tree object or treedata object when node.label was parsed 'support'.

## Examples

```
qzafile1 <- system.file("extdata/qiime2treeqza", "fasttree-tree.qza", package="treeio")
qzafile2 <- system.file("extdata/qiime2treeqza", "iqt-tree.qza", package="treeio")
qzafile3 <- system.file("extdata/qiime2treeqza", "raxml-cat-tree.qza", package="treeio")
tr1 <- read.treeqza(qzafile1)
tr1
tr2 <- read.treeqza(qzafile2)
tr2
tr3 <- read.treeqza(qzafile3)
tr3
# parse node label as 'support' value.
qzafile4 <- system.file("extdata/qiime2treeqza", "raxml-cat-bootstrap-tree.qza", package="treeio")
tr4 <- read.treeqza(qzafile4, node.label="support")
tr4
```

---

# read.treetime *read.timetree*

---

## Description

read timetree output

## Usage

```
read.treetime(file)

read.timetree(file)
```

## Arguments

| | |
|---|---|
| file | the output tree file of timetree |

## Value

treedata object

## reexports *Objects exported from other packages*

### Description

These objects are imported from other packages. Follow the links below to see their documentation.

**ape** as.phylo, is.rooted, Nnode, Ntip, read.nexus, read.tree, root, rtree, write.nexus, write.tree

**dplyr** full_join, inner_join

**magrittr** %<>%, %>%

**rlang** .data

**tibble** as_tibble, tibble

**tidytree** ancestor, as.phylo, as.treedata, child, drop.tip, get.data, get.fields, isTip, MRCA, nodeid, nodelab, offspring, parent, rootnode, treedata

## rename_taxa *rename_taxa*

### Description

rename tip label of phylogenetic tree

### Usage

```
rename_taxa(tree, data, key = 1, value = 2)
```

### Arguments

| | |
|---|---|
| tree | tree object, either treedata or phylo |
| data | data frame |
| key | column in data that match tip label (use 1st column by default) |
| value | column in data for rename tip label (use 2nd column by default) |

### Value

tree object

### Author(s)

Guangchuang Yu

## Examples

```
tree <- rtree(3)
d <- data.frame(old = paste0('t', 1:3), new = LETTERS[1:3])
rename_taxa(tree, d)
rename_taxa(tree, d, old, new)
```

---

`rescale_tree` *rescale_tree*

---

## Description

rescale branch length of tree object

## Usage

```
rescale_tree(tree_object, branch.length)
```

## Arguments

| | |
|---|---|
| `tree_object` | tree object |
| `branch.length` | numerical features (e.g. dN/dS) |

## Value

update tree object

## Author(s)

Guangchuang Yu

---

`spt` *spt method*

---

## Description

spt method

## Usage

```
spt(x, from, to, weights = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | a igraph object |
| from | a specific node of network. |
| to | other nodes of the network, length of it must be larger than 2. |
| weights | a numeric vector giving edge weights or a character. If this is NULL and the graph has a weight edge attribute, then the attribute is used. If this is NA then no weights are used even if the graph has a weight attribute. If this is a character, the graph has the edge attribute which is numeric, then it will be used, default is NULL. |
| ... | additional parameters |

### Value

phylo object

### Examples

```
library(igraph)
set.seed(123)
g <- igraph::sample_gnp(100, .1) %>%
     set_edge_attr(name='weight', value=abs(rnorm(E(.),3)))
tr1 <- spt(g, from = 6, to=V(g), weights = 'weight')
tr1
tr2 <- spt(g, from = 6, to = V(g), weights = NA)
tr2
```

---

## 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 = NULL)
```

### Arguments

| | |
|---|---|
| treedata | treedata object, list of treedata, phylo, or list of phylo |
| file | output file. If file = "", print the output content on screen |
| translate | whether to translate taxa labels |
| tree.name | names of the trees, NULL to use existing tree names |

## 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.beast.newick` *write.beast.newick*

---

## Description

Export `treedata` object to BEAST Newick file. This is useful for making BEAST starting trees with metadata

## Usage

```
write.beast.newick(
  treedata,
  file = "",
  append = FALSE,
  digits = 10,
  tree.prefix = ""
)
```

## Arguments

| | |
|---|---|
| `treedata` | `treedata` object |
| `file` | output file. If file = "", print the output content on screen |
| `append` | logical. Only used if the argument 'file' is the name of file (and not a connection or "lcmd"). If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of file. |
| `digits` | integer, the indicating the number of decimal places, default is 10. |
| `tree.prefix` | character the tree prefix, default is "". |

## 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.newick(nhx)
```

---

## write.jplace *write.jplace*

---

### Description

Export jplace object to jplace file.

### Usage

```
write.jplace(x, outfile)
```

### Arguments

| | |
|---|---|
| x | a jplace object. |
| outfile | the output file name |

### Examples

```
jp <- system.file("extdata", "sample.jplace", package="treeio")
tr1 <- read.jplace(jp)
outfile <- tempfile()
write.jplace(tr1, outfile)
tr2 <- read.jplace(outfile)
tr2
```

---

## 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


∗ **classes**
    jplace-class, 8
∗ **internal**
    reexports, 27
    treeio-package, 3
.data, *27*
.data *(*reexports*)*, 27
%<>% *(*reexports*)*, 27
%>% *(*reexports*)*, 27
%<>%, *27*
%>%, *27*

ancestor, *27*
ancestor *(*reexports*)*, 27
as.phylo, *27*
as.phylo *(*reexports*)*, 27
as.treedata, *27*
as.treedata *(*reexports*)*, 27
as.treedata.phylo, 4
as_tibble, *27*
as_tibble *(*reexports*)*, 27

child, *27*
child *(*reexports*)*, 27

drop.tip, *27*
drop.tip *(*reexports*)*, 27

find.hclust, 4
full_join, *27*
full_join *(*reexports*)*, 27

get.data, *27*
get.data *(*reexports*)*, 27
get.fields, *27*
get.fields *(*reexports*)*, 27
get.placements, 5
get.tree, 6
get.treetext, 6
getNodeNum, 7

inner_join, *27*
inner_join *(*reexports*)*, 27
is.ggtree, 7
is.rooted, *27*
is.rooted *(*reexports*)*, 27
isTip, *27*
isTip *(*reexports*)*, 27

jplace-class, 8

label_branch_paml, 8

mask, 9
merge_tree, 10
MRCA, *27*
MRCA *(*reexports*)*, 27

Nnode, *27*
Nnode *(*reexports*)*, 27
Nnode2 *(*getNodeNum*)*, 7
nodeid, *27*
nodeid *(*reexports*)*, 27
nodelab, *27*
nodelab *(*reexports*)*, 27
Ntip, *27*
Ntip *(*reexports*)*, 27

offspring, *27*
offspring *(*reexports*)*, 27

parent, *27*
parent *(*reexports*)*, 27
print.treedataList, 10

raxml2nwk, 11
read.astral, 11
read.beast, 12
read.codeml, 13
read.codeml_mlc, 14
read.fasta, 14
read.hyphy, 15

read.hyphy.seq, 16
read.iqtree, 16
read.jplace, 17
read.jtree, 17
read.mcmctree, 18
read.mega (read.beast), 12
read.mega_tabular, 19
read.mrbayes (read.beast), 12
read.newick, 19
read.nextstrain.json, 20
read.nexus, *27*
read.nexus (reexports), 27
read.nhx, 20
read.paml_rst, 21
read.phylip, 22
read.phylip.seq, 22
read.phylip.tree, 23
read.phyloxml, 23
read.r8s, 24
read.raxml, 25
read.timetree (read.treetime), 26
read.tree, *27*
read.tree (reexports), 27
read.treeqza, 25
read.treetime, 26
reexports, 27
rename_taxa, 27
rescale_tree, 28
root, *27*
root (reexports), 27
rootnode, *27*
rootnode (reexports), 27
rtree, *27*
rtree (reexports), 27

spt, 28

tibble, *27*
tibble (reexports), 27
treedata, *27*
treedata (reexports), 27
treeio (treeio-package), 3
treeio-package, 3

write.beast, 29
write.beast.newick, 30
write.jplace, 31
write.jtree, 31
write.nexus, *27*
write.nexus (reexports), 27
write.tree, *27*
write.tree (reexports), 27