

# Package ‘treeio’

October 23, 2020

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

**Version** 1.13.1

**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, tibble, tidytree (>= 0.3.0), utils

**Suggests** Biostrings, ggplot2, ggtree, igraph, knitr, phangorn, prettydoc, testthat, tidyr, vroom, xml2

**VignetteBuilder** knitr

**ByteCompile** true

**License** Artistic-2.0

**Encoding** UTF-8

**URL** <https://github.com/YuLab-SMU/treeio> (devel),  
<https://docs.ropensci.org/treeio/> (docs),  
<https://yulab-smu.top/treedata-book/> (book)

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

**LazyData** true

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

**RoxygenNote** 7.1.1

**git\_url** <https://git.bioconductor.org/packages/treeio>

**git\_branch** master

**git\_last\_commit** d373c61

**git\_last\_commit\_date** 2020-09-14

**Date/Publication** 2020-10-22

**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],  
 Shuangbin Xu [ctb] (<<https://orcid.org/0000-0003-3513-5362>>),  
 Konstantinos Geles [ctb]

**Maintainer** Guangchuang Yu <[guangchuangyu@gmail.com](mailto: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
isTip . . . . .	7
jplace-class . . . . .	7
label_branch_paml . . . . .	8
mask . . . . .	8
merge_tree . . . . .	9
Nnode.treedata . . . . .	9
print.treedataList . . . . .	10
raxml2nwk . . . . .	10
read.astral . . . . .	11
read.beast . . . . .	12
read.codeml . . . . .	12
read.codeml_mlc . . . . .	13
read.fasta . . . . .	14
read.hyphy . . . . .	14
read.hyphy.seq . . . . .	15
read.iqtree . . . . .	15
read.jplace . . . . .	16
read.jtree . . . . .	16
read.mega_tabular . . . . .	17
read.newick . . . . .	17
read.nhx . . . . .	18
read.paml_rst . . . . .	18
read.phylip . . . . .	19
read.phylip.seq . . . . .	19
read.phylip.tree . . . . .	20
read.phyloxml . . . . .	20
read.r8s . . . . .	21
read.raxml . . . . .	21
rename_taxa . . . . .	22
rescale_tree . . . . .	23
root.phylo . . . . .	23
tree_subset . . . . .	24
write.beast . . . . .	25

<i>as.treedata.phylo</i>	3
write.jtree . . . . .	26

**Index** 27

*as.treedata.phylo*      *as.treedata*

**Description**

convert phylo to treedata

**Usage**

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

```
## S3 method for class 'pml'
as.treedata(tree, type = "ml", ...)
```

**Arguments**

tree	input tree, a phylo object
boot	optional, can be bootstrap value from ape::boot.phylo
...	additional parameters
type	one of 'ml' and 'bayes' for inferring ancestral sequences

**Details**

converting phylo object to treedata object

**Author(s)**

Guangchuang Yu  
Yu Guangchuang

*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

*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, ...)
```

```
## 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

---

isTip	<i>isTip</i>
-------	--------------

---

**Description**

whether the node is a tip

**Usage**

```
isTip(.data, .node, ...)

## S3 method for class 'tbl_tree'
isTip(.data, .node, ...)

## S3 method for class 'phylo'
isTip(.data, .node, ...)

## S3 method for class 'treedata'
isTip(.data, .node, ...)
```

**Arguments**

.data	phylo, treedata or tbl_tree object
.node	node number
...	additional parameters

**Value**

logical value

**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

*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

---

Nnode.treedata

*Nnode*

---

**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))
```

---

```
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

---

<code>read.astral</code>	<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 <- 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

**Usage**

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

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

```
read.mega(file)
```

**Arguments**

file                    beast file

**Value**

treedata 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                    *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)
```

**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)**

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	<i>read.hyphy.seq</i>
----------------	-----------------------

---

**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	<i>read.iqtree</i>
-------------	--------------------

---

**Description**

parse IQ-TREE output

**Usage**

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

**Arguments**

file                    IQ-TREE Newick text

**Value**

treedata object

**Author(s)**

Guangchuang Yu

---

read.jplace	<i>read.jplace</i>
-------------	--------------------

---

**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	<i>read.jtree</i>
------------	-------------------

---

**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.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.nhx	<i>read.nhx</i>
----------	-----------------

---

**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	<i>read.paml_rst</i>
---------------	----------------------

---

**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	<i>read.phylip</i>
-------------	--------------------

---

**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	<i>read.phylip.seq</i>
-----------------	------------------------

---

**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.phyloxml	<i>read.phyloxml</i>
---------------	----------------------

---

**Description**

parse tree from phyloxml file

**Usage**

read.phyloxml(file)

**Arguments**

file            phyloxml file

**Value**

phylo or multiPhylo object

**Author(s)**

Guangchuang Yu

---

read.phyloxml	<i>read.phyloxml</i>
---------------	----------------------

---

**Description**

read.phyloxml

**Usage**

read.phyloxml(file)

**Arguments**

file            phyloxml file

**Value**

treedata class or multitreedata class

**Examples**

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

---

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

---

**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	<i>rescale_tree</i>
--------------	---------------------

---

**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

---

root.phylo	<i>root</i>
------------	-------------

---

**Description**

re-root a tree

**Usage**

```
## S3 method for class 'phylo'
root(phy, outgroup, node = NULL, resolve.root = TRUE, ...)
```

```
## S3 method for class 'treedata'
root(phy, outgroup, node = NULL, resolve.root = TRUE, ...)
```

**Arguments**

phy	tree object
outgroup	a vector of mode numeric or character specifying the new outgroup
node	node to reroot
resolve.root	a logical specifying whether to resolve the new root as a bifurcating node
...	additional parameters passed to ape::root.phylo

**Value**

rerooted tree

**Author(s)**

Guangchuang Yu

---

tree\_subset

*Subset tree objects by related nodes*

---

**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,  
  group_name = "group",  
  root_edge = TRUE  
)  
  
## S3 method for class 'phylo'  
tree_subset(  
  tree,  
  node,  
  levels_back = 5,  
  group_node = TRUE,  
  group_name = "group",  
  root_edge = TRUE  
)  
  
## S3 method for class 'treedata'  
tree_subset(  
  tree,  
  node,  
  levels_back = 5,  
  group_node = TRUE,  
  group_name = "group",  
  root_edge = 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
group_name	group name (default 'group') for storing grouping information if group_node = TRUE
root_edge	If TRUE (by default), set root.edge to path length of original root to the root of subset tree

### 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", "phyllog.nhx", package="treeio")
nhx <- read.nhx(nhxfile)
write.beast(nhx)
```

---

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](#), 7
- [as.treedata.phylo](#), 3
- [as.treedata.paml](#) ([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
- [isTip](#), 7
- [jplace-class](#), 7
- [label\\_branch\\_paml](#), 8
- [mask](#), 8
- [merge\\_tree](#), 9
- [Nnode.treedata](#), 9
- [Nnode2](#) ([getNodeNum](#)), 6
- [print.treedataList](#), 10
- [raxml2nwk](#), 10
- [read.astral](#), 11
- [read.beast](#), 12
- [read.codeml](#), 12
- [read.codeml\\_mlc](#), 13
- [read.fasta](#), 14
- [read.hyphy](#), 14
- [read.hyphy.seq](#), 15
- [read.iqtree](#), 15
- [read.jplace](#), 16
- [read.jtree](#), 16
- [read.mega](#) ([read.beast](#)), 12
- [read.mega\\_tabular](#), 17
- [read.mrbayes](#) ([read.beast](#)), 12
- [read.newick](#), 17
- [read.nhx](#), 18
- [read.paml\\_rst](#), 18
- [read.phylip](#), 19
- [read.phylip.seq](#), 19
- [read.phylip.tree](#), 20
- [read.phyloxml](#), 20
- [read.r8s](#), 21
- [read.raxml](#), 21
- [rename\\_taxa](#), 22
- [rescale\\_tree](#), 23
- [root.phylo](#), 23
- [root.treedata](#) ([root.phylo](#)), 23
- [tree\\_subset](#), 24
- [write.beast](#), 25
- [write.jtree](#), 26