Package ‘treeio’

January 8, 2024

Title Base Classes and Functions for Phylogenetic Tree Input and Output

Version 1.26.0

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 (>= 0.4.5), utils, stats

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

VignetteBuilder knitr

ByteCompile true

License Artistic-2.0

Encoding UTF-8


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

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

RoxygenNote 7.2.3

git_url https://git.bioconductor.org/packages/treeio
git_branch  RELEASE_3_18
git_last_commit  cce8f0a
git_last_commit_date  2023-10-24
Repository  Bioconductor 3.18
Date/Publication  2024-01-07

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

R topics documented:

  treeio-package .................................................. 3
  as.treedata.phylo ............................................... 4
  find.hclust ................................................... 4
  get.placements ............................................... 5
  get.tree ....................................................... 6
  get.treetext .................................................. 7
  getNodeNum ................................................... 7
  is.ggtree ..................................................... 8
  jplace-class .................................................. 8
  label_branch_paml .............................................. 9
  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
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://github.com/YuLab-SMU/treeio(devel)
- https://docs.ropensci.org/treeio/(docs)
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

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)

describe run on 2023-04-10
get.tree

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

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**

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

**Arguments**

- `object` treedata object
- `...` additional parameter

**Value**

phylo object

---

### getNodeNum

**getNodeNum**

**Description**

calculate total number of nodes

**Usage**

```r
getNodeNum(tree)
```

**Arguments**

- `tree` tree object

**Value**

number

**Author(s)**

Guangchuang Yu
**jplace-class**

**Examples**

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

**is.ggtree**

```r
is.ggtree
```

**Description**

test whether input object is produced by ggtree function

**Usage**

```r
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](https://guangchuangyu.github.io)
**label_branch_paml**

---

**Description**

label branch for PAML to infer selection pressure using branch model

**Usage**

```r
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**

---

**Description**

site mask

**Usage**

```r
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.
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

Description
print information of a list of treedata objects

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

Value
updated tree object

Author(s)
Guangchuang Yu
**raxml2nwk**

**Arguments**
- `x` a list of treedata objects
- ... no used

**Value**
- message

---

**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))

Description
read beast/mrbayes/mega Nexus output
read beast/mrbayes/mega newick file format

Usage
read.beast(file)
read.mrbayes(file)
read.beast.newick(file)
read.mega(file)

Arguments
file newick file

Value
treedata object
treedata object

Author(s)
Guangchuang Yu https://guangchuangyu.github.io
Bradley R Jones
Examples

```r
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) ;
  )
)
```

Description

read baseml output

Usage

```r
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

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

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

Description
read HYPHY output

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

Arguments
nwk
tree file in nwk format, one of hyphy output
ancseqancestral 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

**Description**
parse sequences from hyphy output

**Usage**
```r
read.hyphy.seq(file)
```

**Arguments**
- `file`: output of hyphy ancestral sequence inference; nexus format

**Value**
DNAbin object

**Author(s)**
Guangchuang Yu

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

### read.iqtree

**Description**
parse IQ-TREE output

**Usage**
```r
read.iqtree(file)
```

**Arguments**
- `file`: IQ-TREE Newick text

**Value**
treedata object
**read.jplace**

**Author(s)**
Guangchuang Yu

**Description**
read jplace file

**Usage**
read.jplace(file)

**Arguments**
- **file**
  - jplace file

**Value**
jplace instance

**Author(s)**
Guangchuang Yu

**Examples**

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

---

**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

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

**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

**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

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

---

### 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](https://guangchuangyu.github.io)

**Examples**

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

---

**Description**

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

**Usage**

```r
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](https://guangchuangyu.github.io)

**Examples**

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

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
read.phylip.tree

**Author(s)**
Guangchuang Yu

**References**

---

**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

**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

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"))
**Description**

parse RAxML bootstrapping analysis output

**Usage**

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

**Arguments**

- `file` RAxML bootstrapping analysis output

**Value**

treedata object

**Author(s)**

Guangchuang Yu

**Examples**

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

---

**Description**

read.treeqza

**Usage**

```r
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'.

---
rename_taxa

Value

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

Examples

```r
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
t2 <- read.treeqza(qzafile2)
tr2
t3 <- 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
```

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

Description

rename tip label of phylogenetic tree

Usage

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

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

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 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

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

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

write.beast.newick

Description

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

write beasts 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 "cmd"). 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.jetree

Description

Export treedata object to json tree file

Usage

write.jetree(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, 26
  treeio-package, 3
.data, 26
.data (reexports), 26
%>% (reexports), 26
%>% (reexports), 26
%>%, 26
ancestor, 26
ancestor (reexports), 26
as.phylo, 26
as.phylo (reexports), 26
as.treedata, 26
as.treedata (reexports), 26
as.treedata.phylo, 4
as_tibble, 26
as_tibble (reexports), 26
child, 26
child (reexports), 26
drop.tip, 26
drop.tip (reexports), 26
find.hclust, 4
full_join, 26
full_join (reexports), 26
get.data, 26
get.data (reexports), 26
get.fields, 26
get.fields (reexports), 26
get.placements, 5
get.tree, 6
get.treetext, 7
getNodeNum, 7
inner_join, 26
inner_join (reexports), 26
is.ggtree, 8
is.rooted, 26
is.rooted (reexports), 26
isTip, 26
isTip (reexports), 26
jplace-class, 8
label_branch_paml, 9
mask, 9
merge_tree, 10
MRCA, 26
MRCA (reexports), 26
Nnode, 26
Nnode (reexports), 26
Nnode2 (getNodeNum), 7
nodeid, 26
nodeid (reexports), 26
nodelab, 26
nodelab (reexports), 26
Ntip, 26
Ntip (reexports), 26
offspring, 26
offspring (reexports), 26
parent, 26
parent (reexports), 26
print.treedataList, 10
raxml2nwk, 11
read.astral, 11
read.beast, 12
read.codeml, 13
read.codeml_mlc, 14
read.fasta, 14
read.hyphy, 15
INDEX

read.hyphy.seq, 16
read.iqtree, 16
read.jplace, 17
read.jtree, 17
read.mctree, 18
read.mega(read.beast), 12
read.mega_tabular, 19
read.mrbayes(read.beast), 12
read.newick, 19
read.nextstrain.json, 20
read.nexus, 26
read.nexus(reexports), 26
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.tree, 26
read.tree(reexports), 26
read.treeqza, 25
reexports, 26
rename_taxa, 26
rescale_tree, 27
root, 26
root(reexports), 26
rootnode, 26
rootnode(reexports), 26
rtree, 26
rtree(reexports), 26
spt, 28
tibble, 26
tibble(reexports), 26
treedata, 26
treedata(reexports), 26
treecore(treeio-package), 3
treeio-package, 3
write.beast, 29
write.beast.newick, 29
write.jtree, 30
write.nexus, 26
write.nexus(reexports), 26
write.tree, 26
write.tree(reexports), 26