

Package ‘treeio’

October 14, 2021

Title Base Classes and Functions for Phylogenetic Tree Input and Output

Version 1.16.2

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, rmarkdown, phangorn, prettydoc, testthat, tidyr, vroom, xml2, yaml

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>

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

RoxygenNote 7.1.1

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

git_branch RELEASE_3_13

git_last_commit 5d5bfb8

git_last_commit_date 2021-08-17

Date/Publication 2021-10-14

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:

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

as.treedata.phylo 3

read.treeqza	25
rename_taxa	25
rescale_tree	26
root.treedata	27
tree_subset	27
write.beast	29
write.beast.newick	30
write.jtree	31

Index 32

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	<i>drop.tip method</i>
----------	------------------------

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	<i>get.tree</i>
----------	-----------------

Description

access phylo slot

Usage

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

Arguments

x	tree object
...	additional parameters

Value

phylo object

Author(s)

Guangchuang Yu

get.treetext	<i>get.treetext method</i>
--------------	----------------------------

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	<i>getNodeNum</i>
------------	-------------------

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	<i>is.ggtree</i>
-----------	------------------

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

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	<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

<code>phy</code>	treedata object
<code>internal.only</code>	whether only count internal nodes
<code>...</code>	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

<code>x</code>	a list of treedata objects
<code>...</code>	no used

Value

message

`raxml2nwk`*raxml2nwk*

Description

convert raxml bootstrap tree to newick format

Usage

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

Arguments

<code>infile</code>	input file
<code>outfile</code>	output file

Value

newick file

Author(s)

Guangchuang Yu

`read.astral`*read.astral*

Description

parse ASTRAL output newick text

Usage

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

Arguments

<code>file</code>	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)

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

```
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)
tree <- read.beast.newick(textConnection('(a[&rate=1]:2,(b[&rate=1.1]:1,c[&rate=0.9]:1)[&rate=1]:1);'))
```

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)

Guangchuang Yu

Examples

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

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

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)
```

<code>read.iqtree</code>	<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

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.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")

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

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	<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

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	<i>read.treeqza</i>
--------------	---------------------

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

```
qzaf1 <- system.file("extdata/qiime2treeqza", "fasttree-tree.qza", package="treeio")
qzaf2 <- system.file("extdata/qiime2treeqza", "iqtree-tree.qza", package="treeio")
qzaf3 <- system.file("extdata/qiime2treeqza", "raxml-cat-tree.qza", package="treeio")
tr1 <- read.treeqza(qzaf1)
tr1
tr2 <- read.treeqza(qzaf2)
tr2
tr3 <- read.treeqza(qzaf3)
tr3
# parse node label as 'support' value.
qzaf4 <- system.file("extdata/qiime2treeqza", "raxml-cat-bootstrap-tree.qza", package="treeio")
tr4 <- read.treeqza(qzaf4, node.label="support")
tr4
```

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

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

root.treedata	<i>root</i>
---------------	-------------

Description

re-root a tree

Usage

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

Arguments

phy	tree object
outgroup	a vector of mode numeric or character specifying the new outgroup
node	node to reroot
edgelabel	a logical value specifying whether to treat node labels as edge labels and thus eventually switching them so that they are associated with the correct edges.
...	additional parameters passed to ape::root.phylo

Value

rerooted treedata

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,
  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", "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)
```

<code>write.jtree</code>	<i>write.jtree</i>
--------------------------	--------------------

Description

Export treedata object to json tree file

Usage

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

Arguments

<code>treedata</code>	treedata object
<code>file</code>	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
- as.treedata.phylo, 3
- as.treedata.pml (as.treedata.phylo), 3
- drop.tip, 4, 4
- drop.tip, phylo (drop.tip), 4
- drop.tip, phylo-method (drop.tip), 4
- drop.tip, treedata (drop.tip), 4
- drop.tip, treedata-method (drop.tip), 4
- get.placements, 5
- get.tree, 5
- get.treetext, 6
- get.treetext, treedata-method (get.treetext), 6
- getNodeNum, 6
- is.ggtree, 7
- isTip, 7
- jplace-class, 8
- label_branch_paml, 9
- mask, 9
- merge_tree, 10
- Nnode.treedata, 10
- Nnode2 (getNodeNum), 6
- print.treedataList, 11
- raxml2nwk, 12
- read.astral, 12
- read.beast, 13
- read.codeml, 14
- read.codeml_mlc, 14
- read.fasta, 15
- read.hyphy, 16
- read.hyphy.seq, 16
- read.iqtree, 17
- read.jplace, 18
- read.jtree, 18
- read.mega (read.beast), 13
- read.mega_tabular, 19
- read.mrbayes (read.beast), 13
- read.newick, 19
- read.nhx, 20
- read.paml_rst, 20
- read.phylip, 21
- read.phylip.seq, 22
- read.phylip.tree, 22
- read.phyloxml, 23
- read.r8s, 23
- read.raxml, 24
- read.treeqza, 25
- rename_taxa, 25
- rescale_tree, 26
- root.treedata, 27
- tree_subset, 27
- write.beast, 29
- write.beast.newick, 30
- write.jtree, 31