Package 'TreeSummarizedExperiment'

October 17, 2020

Type Package

Title TreeSummarizedExperiment: a S4 Class for Data with Tree Structures

Version 1.4.8

Description TreeSummarizedExperiment has extended SingleCellExperiment to include hierarchical information on the rows or columns of the rectangular data.

Depends R(>= 3.6.0), SingleCellExperiment, S4Vectors (>= 0.23.18)

License GPL (>=2)

Encoding UTF-8

LazyData true

biocViews DataRepresentation, Infrastructure

Imports methods, BiocGenerics, utils, ape, rlang, dplyr, SummarizedExperiment

VignetteBuilder knitr

Suggests ggtree, ggplot2, BiocStyle, knitr, rmarkdown, testthat

RoxygenNote 7.1.1

Collate 'aboutLoop.R' 'classValid.R' 'allClass.R' 'aggValue.R' 'allGenerics.R' 'changeTree.R' 'classAccessor.R' 'data.R' 'deprecate_Fun.R' 'tree_addLabel.R' 'tree_asLeaf.R' 'tree_convertNode.R' 'tree_countLeaf.R' 'tree_countNode.R' 'tree_distNode.R' 'tree_findAncestor.R' 'tree_findChild.R' 'tree_findDescendant.R' 'tree_findSibling.R' 'tree_isLeaf.R' 'tree_matTree.R' 'tree_printNode.R' 'tree_shareNode.R' 'tree_showNode.R' 'tree_signalNode.R' 'tree_toTree.R' 'tree_trackNode.R' 'tree_unionLeaf.R'

git_url https://git.bioconductor.org/packages/TreeSummarizedExperiment

git_branch RELEASE_3_11

git_last_commit 03e0f94

git_last_commit_date 2020-09-02

Date/Publication 2020-10-16

Author Ruizhu Huang [aut, cre] (<https://orcid.org/0000-0003-3285-1945>), Felix G.M. Ernst [ctb] (<https://orcid.org/0000-0001-5064-0928>)

Maintainer Ruizhu Huang <ruizhuRH@gmail.com>

R topics documented:

addLabel 2
aggValue 3
asLeaf 5
changeTree 6
countLeaf 7
countNode 8
detectLoop 9
distNode 10
findAncestor 10
findChild 11
findOS 12
findSibling 13
isLeaf 14
LinkDataFrame-class 15
LinkDataFrame-constructor 15
matTree 16
printNode 17
resolveLoop 18
rowLinks 19
shareNode 21
showNode 22
signalNode 23
tinyTree 24
toTree 25
trackNode 26
transNode 27
TreeSummarizedExperiment-class 28
TreeSummarizedExperiment-constructor 29
unionLeaf 31

Index **32**

addLabel *add labels to nodes of a tree*

Description

`addLabel` adds labels to the node of a tree (`phylo` object)

Usage

```
addLabel(tree, label = NULL)
```

Arguments

`tree` A phylo object

`label` A character vector as the label of tree. The label is passed to nodes that are sorted by their node number in ascending order. The default is NULL, and nodes are labeled by using their node numbers (convert node numbers from numeric values to characters)

Value

a phylo object

Author(s)

Ruizhu Huang

Examples

```
data(tinyTree)
library(ggtree)

# PLOT tree
# The node labels are in orange texts and the node numbers are in blue
ggtree(tinyTree,branch.length = 'none')+
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)

# change labels
nodes <- showNode(tree = tinyTree, only.leaf = FALSE)
tt <- addLabel(tree = tinyTree, label = LETTERS[nodes])

ggtree(tt, branch.length = 'none')+
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)
```

aggValue *Perform data aggregations based on the available tree structures*

Description

aggValue aggregates values on the leaf nodes of a tree to a specific arbitrary level of the tree. The level is specified via the nodes of the tree. Users could decide on which dimension (row or column) and how should the aggregation be performed.

Usage

```
aggValue(
  x,
  rowLevel = NULL,
  rowBlock = NULL,
  colLevel = NULL,
  colBlock = NULL,
  FUN = sum,
  assay = NULL,
  message = FALSE
)
```

Arguments

x A TreeSummarizedExperiment object.

rowLevel A numeric (node numbers) or character (node labels) vector. It provides the level on the tree that data is aggregated to. The aggregation is on the row dimension. The default is rowLevel = NULL, and no aggregation is performed.

rowBlock A column name in the rowData to separate the aggregation.

colLevel A numeric (node numbers) or character (node labels) vector. It provides the level on the tree that data is aggregated to. The aggregation is on the column dimension. The default is colLevel = NULL, and no aggregation is performed.

colBlock A column name in the colData to separate the aggregation.

FUN A function to be applied on the aggregation. It's similar to the FUN in apply

assay A integer scalar or string indicating which assay of x to use in the aggregation.

message A logical value. The default is TRUE. If TRUE, it will print out the running process.

Value

A TreeSummarizedExperiment object or a matrix. The output has the same class of the input x.

Author(s)

Ruizhu HUANG

See Also

TreeSummarizedExperiment

Examples

```
# assays data
set.seed(1)
toyTable <- matrix(rnbinom(20, size = 1, mu = 10), nrow = 5)
colnames(toyTable) <- paste(rep(LETTERS[1:2], each = 2),
                            rep(1:2, 2), sep = "_")
rownames(toyTable) <- paste("entity", seq_len(5), sep = "")

toyTable

# the column data
colInf <- DataFrame(gg = c(1, 2, 3, 3),
                    group = rep(LETTERS[1:2], each = 2),
                    row.names = colnames(toyTable))
colInf

# the toy tree
library(ape)
set.seed(4)
treeC <- rtree(4)
treeC$node.label <- c("All", "GroupA", "GroupB")

library(ggtree)
ggtree(treeC, size = 2) +
   geom_text2(aes(label = node), color = "darkblue",
           hjust = -0.5, vjust = 0.7, size = 6) +
    geom_text2(aes(label = label), color = "darkorange",
               hjust = -0.1, vjust = -0.7, size = 6)


tse <- TreeSummarizedExperiment(assays = list(toyTable),
                                colData = colInf,
                                colTree = treeC,
                                colNodeLab = treeC$tip.label,
                                metadata = list(test = 1:4))

aggCol <- aggValue(x = tse, colLevel = c("GroupA", "GroupB"),
FUN = sum)

assays(aggCol)[[1]]
```

asLeaf *change internal nodes to leaf nodes*

Description

asLeaf updates a phylo tree by changing the specified internal nodes to leaf nodes. In other words, the descendant nodes of the specified internal nodes are removed.

Usage

```
asLeaf(tree, node)
```

Arguments

tree	A phylo object.
node	A numeric or character vector. It specifies internal nodes that are changed to leaves via their node labels or numbers.

Value

A phylo object.

Examples

```
library(ggtree)
data(tinyTree)
ggtree(tinyTree, ladderize = FALSE) +
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7) +
    geom_hilight(node = 18) +
    geom_point2()

# remove the blue branch
NT1 <- asLeaf(tree = tinyTree, node = 16)
```

```
ggtree(NT1, ladderize = FALSE) +
    geom_text2(aes(label = label), color = "darkorange",
               hjust = -0.1, vjust = -0.7) +
    geom_point2()

# if mergeSingle = TRUE, the node (Node_17) is removed.
NT2 <- asLeaf(tree = tinyTree, node = c(15, 13))
# or use the ape::drop.tip
# NT3 <-  ape::drop.tip(phy = tinyTree, tip = 4:5)
# all.equal(NT2, NT3)

ggtree(NT2, ladderize = FALSE) +
    geom_text2(aes(label = label), color = "darkorange",
               hjust = -0.1, vjust = -0.7) +
    geom_point2()
```

changeTree *Change the row or column tree*

Description

changeTree changes a row or column tree in a TreeSummarizedExperiment object.

Usage

```
changeTree(
  x,
  rowTree = NULL,
  rowNodeLab = NULL,
  colTree = NULL,
  colNodeLab = NULL
)
```

Arguments

x	A TreeSummarizedExperiment object
rowTree	A phylo object. A new row tree.
rowNodeLab	A character string. It provides the labels of nodes that the rows of assays tables corresponding to. If NULL (default), the row names of the assays tables are used.
colTree	A phylo object. A new column tree.
colNodeLab	A character string. It provides the labels of nodes that the columns of assays tables corresponding to. If NULL (default), the column names of the assays tables are used.

Value

A TreeSummarizedExperiment object

Author(s)

Ruizhu Huang

Examples

```
library(ape)
set.seed(1)
treeR <- ape::rtree(10)

# the count table
count <- matrix(rpois(160, 50), nrow = 20)
rownames(count) <- paste0("entity", 1:20)
colnames(count) <- paste("sample", 1:8, sep = "_")
# The sample information
sampC <- data.frame(condition = rep(c("control", "trt"),
                                    each = 4),
                    gender = sample(x = 1:2, size = 8,
                                    replace = TRUE))
rownames(sampC) <- colnames(count)
# build a TreeSummarizedExperiment object
tse <- TreeSummarizedExperiment(assays = list(count),
                                colData = sampC,
                                rowTree = treeR,
                                rowNodeLab = rep(treeR$tip.label, each =2))

treeR2 <- drop.tip(phy = treeR, tip = c("t10", "t9", "t8"))

use <- changeTree(x = tse, rowTree = treeR2)
use
```

countLeaf *count the number of leaf nodes*

Description

countLeaf calculates the number of leaves on a phylo tree.

Usage

```
countLeaf(tree)
```

Arguments

tree	A phylo object

Value

a numeric value

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)

data(tinyTree)

ggtree(tinyTree, branch.length = 'none') +
geom_text2(aes(label = label), hjust = -0.3) +
geom_text2(aes(label = node), vjust = -0.8,
hjust = -0.3, color = 'blue')


(n <- countLeaf(tinyTree))
```

countNode *count the number of nodes*

Description

countNode calculates the number of nodes on a phylo tree.

Usage

```
countNode(tree)
```

Arguments

tree	A phylo object

Value

a numeric value

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)

data(tinyTree)

ggtree(tinyTree, branch.length = 'none') +
geom_text2(aes(label = label), hjust = -0.3) +
geom_text2(aes(label = node), vjust = -0.8,
hjust = -0.3, color = 'blue')


(n <- countLeaf(tinyTree))
```

detectLoop *Detect loops* detectLoop *detects loops*

Description

Detect loops detectLoop detects loops

Usage

```
detectLoop(tax_tab)
```

Arguments

tax_tab a data frame where columns store hierarchical levels. The columns from the left to the right correspond nodes from the root to the leaf.

Value

a data frame

Author(s)

Ruizhu Huang

Examples

```
df <- data.frame(A = rep("a", 8),
                 B = rep (c("b1", "b2", "b3", "b4"), each = 2),
                 C = paste0("c", c(1, 2, 2, 3:7)),
                 D = paste0("d", 1:8))

# The result means that a loop is caused by 'b1' and 'b2' in column 'B' and
# 'c2' in column 'C' (a-b1-c2; a-b2-c2)
detectLoop(tax_tab = df)

df <- data.frame(R1 = rep("A", 6),
                     R2 = c("B1", rep("B2", 4), "B3"),
                     R3 = c("C1", "C2", "C3", NA, NA, NA),
                     R4 = c("D1", "D2", "D3", NA, NA, NA),
                     R5 = paste0("E", 1:6))
detectLoop(tax_tab = df)

df <- data.frame(R1 = rep("A", 7),
                     R2 = c("B1", rep("B2", 4), "B3", "B3"),
                     R3 = c("C1", "C2", "C3", "", "", "", ""),
                     R4 = c("D1", "D2", "D3", "", "", "", ""),
                     R5 = paste0("E", 1:7))
detectLoop(tax_tab = df)

df <- data.frame(R1 = rep("A", 7),
                     R2 = c("B1", rep("B2", 4), "B3", "B3"),
                     R3 = c("C1", "C2", "C3", NA, NA, NA, NA),
                     R4 = c("D1", "D2", "D3", NA, NA, NA, NA),
                     R5 = paste0("E", 1:7))
detectLoop(tax_tab = df)
```

distNode *Calculate the distance between any two nodes on the tree*

Description

distNode is to calculate the distance between any two nodes on a phylo tree

Usage

```
distNode(tree, node)
```

Arguments

tree	A phylo object.
node	A numeric or character vector of length two.

Value

A numeric value.

Examples

```
library(ggtree)
data(tinyTree)
ggtree(tinyTree) +
    geom_text2(aes(label = node), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = branch.length), color = "darkblue",
               vjust = 0.7)


distNode(tree = tinyTree, node = c(10, 11))
distNode(tree = tinyTree, node = c(12, 13))
distNode(tree = tinyTree, node = c(13, 15))
distNode(tree = tinyTree, node = c(12, 14))
```

findAncestor *Find the ancestors of specified nodes*

Description

findAncestor finds the ancestor in the nth generation above specified nodes.

Usage

```
findAncestor(tree, node, level, use.alias = FALSE)
```

Arguments

`tree` A phylo object

`node` A vector of node numbers or node labels

`level` A vector of numbers to define nth generation before the specified nodes

`use.alias` A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix `"Node_"` to the node number if the node is an internal node or adding a prefix `"Leaf_"` if the node is a leaf node.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when `use.alias = FALSE`, and have the alias of node label as name when `use.alias = TRUE`.

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)
data(tinyTree)
ggtree(tinyTree, branch.length = 'none') +
 geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
 geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)

 findAncestor(tree = tinyTree, node = c(18, 13), level = 1)
```

`findChild` *Find the children*

Description

`findChild` finds children of an internal node.

Usage

```
findChild(tree, node = 11, use.alias = FALSE)
```

Arguments

`tree` A phylo object.

`node` An internal node. It could be the node number or the node label.

`use.alias` A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix `"alia_"` to the node number.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when `use.alias = FALSE`, and have the alias of node label as name when `use.alias = TRUE`.

Author(s)

Ruizhu Huang

Examples

```
data(tinyTree)

library(ggtree)
ggtree(tinyTree) +
geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7) +
geom_hilight(node = 17, fill = 'steelblue', alpha = 0.5) +
geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7)

(tips <- findChild(tree = tinyTree, node = 17))
```

`findOS` *Find descendants (or offsprings)*

Description

`findDescendant` finds descendants of a node.

Usage

```
findOS(tree, node, only.leaf = TRUE, self.include = FALSE, use.alias = FALSE)

findDescendant(
  tree,
  node,
  only.leaf = TRUE,
  self.include = FALSE,
  use.alias = FALSE
)
```

Arguments

`tree`	A phylo object.
`node`	An internal node. It could be the node number or the node label.
`only.leaf`	A logical value, TRUE or FALSE. The default is TRUE. If default, only the leaf nodes in the descendant nodes would be returned.
`self.include`	A logical value, TRUE or FALSE. The default is FALSE. If TRUE, the node specified in **node** is included and the leaf node itself is returned as its descendant.

`use.alias` A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix `"alias_"` to the node number.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when `use.alias = FALSE`, and have the alias of node label as name when `use.alias = TRUE`.

Author(s)

Ruizhu Huang

Examples

```
data(tinyTree)

library(ggtree)
ggtree(tinyTree) +
geom_text2(aes(label = node), color = "darkblue",
              hjust = -0.5, vjust = 0.7) +
geom_hilight(node = 17, fill = 'steelblue', alpha = 0.5) +
geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7)

(tips <- findDescendant(tree = tinyTree, node = c(17), only.leaf = TRUE))
```

`findSibling` *find the sibling node*

Description

`findSibling` is to find the sibling node of an node node.

Usage

```
findSibling(tree, node, use.alias = FALSE)
```

Arguments

`tree` A phylo object.

`node` A numeric or character vector. Node labels or node numbers.

`use.alias` A logical value, TRUE or FALSE. The default is FALSE, and the original node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix `"Node_"` to the node number if the node is an internal node or adding a prefix `"Leaf_"` if the node is a leaf node.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when `use.alias = FALSE`, and have the alias of node label as name when `use.alias = TRUE`.

Examples

```
library(ggtree)
data(tinyTree)
ggtree(tinyTree, branch.length = 'none') +
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)


 findSibling(tree = tinyTree, node = 17)
 findSibling(tree = tinyTree, node = c(13, 17))
```

`isLeaf` *To test whether the specified nodes are leaf nodes*

Description

`isLeaf` is to test wheter some specified nodes are leaf nodes of a tree.

Usage

```
isLeaf(tree, node)
```

Arguments

`tree`	A phylo object.
`node`	A numeric or character vector. Node labels or node numbers.

Value

a logical vector with the same length as the input `node`.

Author(s)

Ruizhu HUANG

Examples

```
data(tinyTree)
library(ggtree)

# PLOT tree
# The node labels are in orange texts and the node numbers are in blue
ggtree(tinyTree,branch.length = 'none')+
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)

isLeaf(tree = tinyTree, node = c(5, 4, 18))
isLeaf(tree = tinyTree, node = c("t4", "t9", "Node_18" ))
```

`LinkDataFrame-class` *LinkDataFrame: A S4 class extended from DataFrame An S4 class LinkDataFrame*

Description

The **LinkDataFrame** is extended from the class **DataFrame** to include at least four columns `nodeLab`, `nodeLab_alias`, `nodeNum`, and `isLeaf`.

Constructor

See `LinkDataFrame-constructor` for constructor functions.

`LinkDataFrame-constructor` *Construct a LinkDataFrame Construct a LinkDataFrame object*

Description

Construct a LinkDataFrame Construct a LinkDataFrame object

Usage

```
LinkDataFrame(nodeLab, nodeLab_alias, nodeNum, isLeaf, ...)
```

Arguments

`nodeLab`	A character vector
`nodeLab_alias`	A character vector
`nodeNum`	A numeric vector
`isLeaf`	A logical vector
`...`	All arguments accepted by `DataFrame-class`.

Value

A LinkDataFrame object

See Also

`LinkDataFrame` `DataFrame`

Examples

```
(ld <- LinkDataFrame(nodeLab = letters[1:5],
                     nodeLab_alias = LETTERS[1:5],
                     nodeNum = 1:5,
                     isLeaf = TRUE,
                     right = 1:5))
```

---

matTree *Transform a phylo object into a matrix.*

Description

matTree transforms a phylo tree into a matrix. The entry of the matrix is node number. Each row represents a path connecting a leaf node and the root. The columns are arranged in the order as the path passing the nodes to reach the root.

Usage

```
matTree(tree)
```

Arguments

tree	A phylo object

Value

A matrix

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)

data(tinyTree)
ggtree(tinyTree, branch.length = 'none') +
 geom_text2(aes(label = node))


# each row of the matrix representing a path.
# the first column is leaf nodes; the last non-NA value in a row is the root
mat <- matTree(tree = tinyTree)
```

printNode *To print out the node labels*

Description

nodeLabel is to print out the node labels of a phylo tree.

Usage

```
printNode(tree, type = c("leaf", "internal", "all"))
```

Arguments

tree	A phylo object.
type	A character value choose from leaf, all, and internal. If leaf, the output is a data frame including only leaf nodes; if internal, the output is a data frame including only internal nodes; if all, the output is a data frame including all nodes.

Value

a data frame

Author(s)

Ruizhu HUANG

Examples

```
data(tinyTree)
library(ggtree)

# PLOT tree
# The node labels are in orange texts and the node numbers are in blue
ggtree(tinyTree,branch.length = 'none')+
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)

(pn1 <- printNode(tinyTree, type = "leaf"))
(pn2 <- printNode(tinyTree, type = "internal"))
(pn3 <- printNode(tinyTree, type = "all"))
```

`resolveLoop` *Resolve loops* `resolveLoop` *resolve loops by adding suffix to the child node. The suffix is "_i" where 'i' is a number. Please see examples.*

Description

Resolve loops `resolveLoop` resolve loops by adding suffix to the child node. The suffix is "_i" where 'i' is a number. Please see examples.

Usage

```
resolveLoop(tax_tab)
```

Arguments

`tax_tab`	a data frame where columns store hierarchical levels. The columns from the left to the right correspond nodes from the root to the leaf.

Value

a data frame

Author(s)

Ruizhu Huang

Examples

```
# example 1
df <- data.frame(A = rep("a", 8),
                 B = rep (c("b1", "b2", "b3", "b4"), each = 2),
                 C = paste0("c", c(1, 2, 2, 3:7)),
                 D = paste0("d", 1:8))

# The result means that a loop is caused by 'b1' and 'b2' in column 'B' and
# 'c2' in column 'C' (a-b1-c2; a-b2-c2)
resolveLoop(tax_tab = df)

# example 2
taxTab <- data.frame(R1 = rep("A", 5),
                     R2 = c("B1", rep("B2", 3), ""),
                     R3 = c("C1", "C2", "C3", "", ""),
                     R4 = c("D1", "D2", "D3", "", ""),
                     R5 = paste0("E", 1:5))

resolveLoop(tax_tab = taxTab)
# example 3
taxTab <- data.frame(R1 = rep("A", 6),
                     R2 = c("B1", rep("B2", 4), ""),
                     R3 = c("C1", "C2", "C3", "", "", ""),
                     R4 = c("D1", "D2", "D3", "", "", ""),
                     R5 = paste0("E", 1:6))

resolveLoop(tax_tab = taxTab)
```

```
 # example 3
taxTab <- data.frame(
            R1 = rep("A", 5),
            R2 = c("B1", rep("B2", 3), "B3"),
            R3 = c("C1", "C2", "C3", NA, NA),
            R4 = c("D1", "D2", "D3", NA, NA),
            R5 = paste0("E", 1:5))
resolveLoop(tax_tab = taxTab)
```

rowLinks *TreeSummarizedExperiment-accessors*

Description

All accessor functions that work on SingleCellExperiment should work on **TreeSummarizedExperiment**. Additionally, new accessors rowLinks colLinks, rowTree and colTree accessor function are available for **TreeSummarizedExperiment**.

Usage

```
rowLinks(x)

## S4 method for signature 'TreeSummarizedExperiment'
rowLinks(x)

colLinks(x)

## S4 method for signature 'TreeSummarizedExperiment'
colLinks(x)

rowTree(x)

## S4 method for signature 'TreeSummarizedExperiment'
rowTree(x)

colTree(x)

## S4 method for signature 'TreeSummarizedExperiment'
colTree(x)

## S4 method for signature 'TreeSummarizedExperiment,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'TreeSummarizedExperiment'
rownames(x) <- value

## S4 replacement method for signature 'TreeSummarizedExperiment'
colnames(x) <- value
```

```
subsetByNode(x, rowNode, colNode)

## S4 method for signature 'TreeSummarizedExperiment'
subsetByNode(x, rowNode, colNode)
```

Arguments

`x`	A TreeSummarizedExperiment object
`i, j`	The row, column index to subset `x`. The arguments of the subset function `[]`
`...`	The argument from the subset function `[]`
`drop`	A logical value, TRUE or FALSE. The argument from the subset function `[]`
`value`	the new rownames or colnames as a `character` value. See `BiocGenerics`.
`rowNode`	A vector of nodes that are used to subset rows. One could use the node number, the node label or the node alias to specify nodes, but not a mixture of them.
`colNode`	A vector of nodes that are used to subset columns. One could use the node number, the node label or the node alias to specify nodes, but not a mixture of them.

Value

Elements from `TreeSummarizedExperiment`.

Author(s)

Ruizhu HUANG

See Also

`TreeSummarizedExperiment` `SingleCellExperiment`

Examples

```
# the assay table
set.seed(1)
y <- matrix(rnbinom(300,size=1,mu=10),nrow=10)
colnames(y) <- paste(rep(LETTERS[1:3], each = 10), rep(1:10,3), sep = "_")
rownames(y) <- tinyTree$tip.label

# the row data
rowInf <- DataFrame(var1 = sample(letters[1:3], 10, replace = TRUE),
                    var2 = sample(c(TRUE, FALSE), 10, replace = TRUE))
# the column data
colInf <- DataFrame(gg = factor(sample(1:3, 30, replace = TRUE)),
                    group = rep(LETTERS[1:3], each = 10))

# the tree structure on the rows of assay tables
data("tinyTree")

# the tree structure on the columns of assay tables
sampTree <- ape::rtree(30)
sampTree$tip.label <- colnames(y)

# create the TreeSummarizedExperiment object
```

```
toy_tse <- TreeSummarizedExperiment(assays = list(y),
                                    rowData = rowInf,
                                    colData = colInf,
                                    rowTree = tinyTree,
                                    colTree = sampTree)

## extract the rowData
(rowD <- rowData(x = toy_tse))

## extract the colData
(colD <- colData(x = toy_tse))

## extract the linkData
# on rows
(rowL <- rowLinks(x = toy_tse))
# on columns
(colL <- colLinks(x = toy_tse))

 ## extract the treeData
# on rows
(rowT <- rowTree(x = toy_tse))
# on columns
(colT <- colTree(x = toy_tse))
```

shareNode *Find the share node*

Description

shareNode is to find the node where the specified nodes first meet.

Usage

```
shareNode(tree, node, use.alias = FALSE)
```

Arguments

tree	A phylo object.
node	A vector of node numbers or node labels.
use.alias	A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix "Node_" to the node number if the node is an internal node or adding a prefix "Leaf_" if the node is a leaf node.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when use.alias = FALSE, and have the alias of node label as name when use.alias = TRUE.

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)
data(tinyTree)

# PLOT tree
ggtree(tinyTree, branch.length = 'none') +
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)


## find the node shared by provided node labels
shareNode(node = c('t4','t9'), tree = tinyTree,
          use.alias = FALSE)

shareNode(node = c('t10','Node_17'), tree = tinyTree,
          use.alias = FALSE)

## find the node shared by provided node numbers
shareNode(node = c(2, 3), tree = tinyTree)
```

---

showNode *Find nodes on the tree*

Description

showNode is to get nodes from the tree.

Usage

```
showNode(tree, only.leaf = FALSE, use.alias = FALSE)
```

Arguments

`tree` A phylo object.

`only.leaf` A logical value, TRUE or FALSE. The default is FALSE, all nodes are output; otherwise, leaves are output

`use.alias` A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix `"alias_"` to the node number.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when `use.alias = FALSE`, and have the alias of node label as name when `use.alias = TRUE`.

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)
data(tinyTree)

# PLOT tree
ggtree(tinyTree, branch.length = 'none') +
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)


## find the node shared by provided node labels
showNode(tree = tinyTree, only.leaf = TRUE,
          use.alias = FALSE)

showNode(tree = tinyTree, only.leaf = FALSE,
          use.alias = FALSE)
```

signalNode *find the optimal nodes to short result.*

Description

signalNode is to represent some nodes with their ancestor to make result as short as possible. The ancestors share exactly the same leaves as the original nodes.

Usage

```
signalNode(tree, node, use.alias = FALSE)
```

Arguments

tree	A tree (phylo object)
node	A vector of node numbers or node labels
use.alias	A logical value, TRUE or FALSE. The default is FALSE, and the node label would be used to name the output; otherwise, the alias of node label would be used to name the output. The alias of node label is created by adding a prefix "Node_" to the node number if the node is an internal node or adding a prefix "Leaf_" if the node is a leaf node.

Value

A vector of nodes. The numeric value is the node number, and the vector name is the corresponding node label. If a node has no label, it would have NA as name when use.alias = FALSE, and have the alias of node label as name when use.alias = TRUE.

Author(s)

Ruizhu Huang

Examples

```
data(tinyTree)
library(ggtree)

# PLOT tree
# The node labels are in orange texts and the node numbers are in blue
ggtree(tinyTree,branch.length = 'none')+
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7)

## find the node shared by provided node labels
signalNode(node = c('t4','t9'), tree = tinyTree)
signalNode(node = c('t4','t9'), tree = tinyTree)
signalNode(node = c('t10','Node_18', 't8'), tree = tinyTree,
 use.alias = FALSE)
signalNode(node = c('t10','Node_18', 't8'), tree = tinyTree,
 use.alias = TRUE)

## find the node shared by provided node numbers
signalNode(node = c(2, 3), tree = tinyTree)
signalNode(node = c(2, 3, 16), tree = tinyTree)
```

`tinyTree` *A simulated phylogenetic tree with 10 tips and 9 internal nodes*

Description

A random phylo object created using the function `rtree`

Usage

```
tinyTree
```

Format

A phylo object with 10 tips and 9 internal nodes:

Tip labels t1, t2, ..., t10.

Node labels Node_11, Node_12, ..., Node_19

toTree *Translate a data frame to a phylo object*

Description

toTree translates a data frame to a phylo object

Usage

```
toTree(data, column_order = NULL)
```

Arguments

data	A data frame or matrix.
column_order	A vector that includes the column names of data to reorder columns of data. Default is NULL, the original order of data is kept.

Details

The last column is used as the leaf nodes

Value

a phylo object

Author(s)

Ruizhu HUANG

Examples

```
library(ggtree)
# Example 1:
taxTab <- data.frame(R1 = rep("A", 5),
                     R2 = c("B1", rep("B2", 4)),
                     R3 = paste0("C", 1:5))
# Internal nodes: their labels are prefixed with colnames of taxTab
#  e.g., R2:B2
tree <- toTree(data = taxTab)
# viz the tree
 ggtree(tree) +
 geom_text2(aes(label = label), color = "red", vjust = 1) +
 geom_nodepoint()

# Example 2: duplicated rows in the 3rd and 4th rows
taxTab <- data.frame(R1 = rep("A", 5),
                     R2 = c("B1", rep("B2", 4)),
                     R3 = c("C1", "C2", "C3", "C3", "C4"))
# duplicated rows are removed with warnings
tree <- toTree(data = taxTab)

# Example 3: NA values in R2 column
# results: the internal node with the label 'R2:'
taxTab <- data.frame(R1 = rep("A", 5),
                     R2 = c("B1", rep("B2", 2), NA, "B2"),
                     R3 = c("C1", "C2", "C3", NA, "C4"))
tree <- toTree(data = taxTab)
# viz the tree
 ggtree(tree) +
 geom_text2(aes(label = label), color = "red", vjust = 1) +
 geom_nodepoint()

# Example 4: duplicated values in the leaf column (R4)
# Not allowed and give errors
# taxTab <- data.frame(R1 = rep("A", 5),
#                       R2 = c("B1", rep("B2", 3), "B3"),
#                       R3 = c("C1", "C2", "C3", "C3",NA),
#                       R4 = c("D1", "D2", "D3", NA, NA))

# Example 5: loops caused by missing values in B2-C4, B3-C4
taxTab <- data.frame(R1 = rep("A", 6),
                     R2 = c("B1", rep("B2", 4), "B3"),
                     R3 = c("C1", "C2", "C3", "C3", "C4", "C4"),
                     R4 = c("D1", "D2", "D3", "D3", "D4" , "D4"),
                     R5 = paste0("E", 1:6))
 # resolove loops before run to Tree
 # Suffix are adding to C4
 taxNew <- resolveLoop(taxTab)
 tree <- toTree(data = taxNew)

 # viz the tree
 ggtree(tree) +
 geom_text2(aes(label = label), color = "red", vjust = 1) +
 geom_nodepoint()
```

`trackNode` *track the nodes of a phylo tree*

Description

`trackNode` track nodes of a phylo tree by adding the alias labels to them

Usage

```
trackNode(tree)
```

Arguments

`tree`	A phylo object

Value

a phylo object

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)

data(tinyTree)

ggtree(tinyTree, branch.length = 'none') +
    geom_text2(aes(label = label), hjust = -0.3) +
    geom_text2(aes(label = node), vjust = -0.8,
               hjust = -0.3, color = 'blue')

#check whether the node number and node label are matched
trackTree <- trackNode(tinyTree)
ggtree(trackTree, branch.length = 'none') +
    geom_text2(aes(label = label), hjust = -0.3) +
    geom_text2(aes(label = node), vjust = -0.8,
               hjust = -0.3, color = 'blue')
```

---

transNode *Transfer between node number and node label*

Description

`convertNode` does the transformation between the number and the label of a node on a tree

Usage

```
transNode(tree, node, use.alias = FALSE, message = FALSE)

convertNode(tree, node, use.alias = FALSE, message = FALSE)
```

Arguments

`tree`	A phylo object
`node`	A character or numeric vector representing tree node label(s) or tree node number(s)
`use.alias`	A logical value, TRUE or FALSE. This is an optional argument that only requried when the input `node` is a numeric vector. The default is FALSE, and the node label would be returned; otherwise, the alias of node label would be output. The alias of node label is created by adding a prefix `"alias_"` to the node number.
`message`	A logical value, TRUE or FALSE. The default is FALSE. If TRUE, message will show when a tree have duplicated labels for some internal nodes.

Value

a vector

Author(s)

Ruizhu Huang

Examples

```
library(ggtree)

data(tinyTree)

ggtree(tinyTree, branch.length = 'none') +
geom_text2(aes(label = label), hjust = -0.3) +
geom_text2(aes(label = node), vjust = -0.8,
hjust = -0.3, color = 'blue')

#check whether the node number and node label are matched
convertNode(tinyTree, node = c(11, 2, 4, 15))

convertNode(tree = tinyTree, node = c("Node_16", "Node_11"))
convertNode(tree = tinyTree, node = c("alias_16", "alias_11"))
```

TreeSummarizedExperiment-class

An S4 class TreeSummarizedExperiment

Description

The class **TreeSummarizedExperiment** is an extension class of standard SingleCellExperiment class. It has four more slots that are not in SingleCellExperiment class: rowTree, rowLinks colTree and colLinks. The hierarchical information of rows (columns) is stored in rowTree (colTree) and the link between the rows (columns) of assays tables and nodes of the tree is given in rowLinks (colLinks).

Details

The class **TreeSummarizedExperiment** is designed to store rectangular data for entities (e.g., microbes or cell types) (assays), information about the hiearchical structure (rowTree on rows; colTree on columns), and the mapping information between the tree nodes and the rows or the columns of the rectangular data. Users could provide the hiearchical structure of the rows, columns or both) of the assays tables, and the link data will be automatically generated in rowLinks, colData or both, respectively. It's required that the object in rowLinks or colLinks has the LinkDataFrame class. Please see the page LinkDataFrame for more details.

Slots

rowTree A phylo object or NULL. It gives information about the hiearchical structure of rows of assays tables.

colTree A phylo object or NULL. It gives information about the hiearchical structure of columns of assays tables.

rowLinks A LinkDataFrame. It gives information about the link between the nodes of the rowTree and the rows of assays tables.

colLinks A LinkDataFrame. It gives information about the link between the nodes of the colTree and the columns of assays tables.

... Other slots from SingleCellExperiment

Constructor

See TreeSummarizedExperiment-constructor for constructor functions.

Accessor

See TreeSummarizedExperiment-accessor for accessor functions.

See Also

TreeSummarizedExperiment TreeSummarizedExperiment-accessor SingleCellExperiment

---

TreeSummarizedExperiment-constructor

Construct a TreeSummarizedExperiment object

Description

TreeSummarizedExperiment constructs a TreeSummarizedExperiment object.

Usage

```
TreeSummarizedExperiment(
  ...,
  rowTree = NULL,
  colTree = NULL,
  rowNodeLab = NULL,
  colNodeLab = NULL
)
```

Arguments

...	Arguments passed to the SummarizedExperiment constructor to fill the slots of the base class.
rowTree	A phylo object that provides hiearchical information of rows of assay tables.
colTree	A phylo object that provides hiearchical information of columns of assay tables.
rowNodeLab	A character string. It provides the labels of nodes that the rows of assays tables corresponding to. If NULL (default), the row names of the assays tables are used.
colNodeLab	A character string. It provides the labels of nodes that the columns of assays tables corresponding to. If NULL (default), the column names of the assays tables are used.

Details

The output TreeSummarizedExperiment object has very similar structure as the SingleCellExperiment. The differences are summarized be as below.

- **rowTree** A slot exists in TreeSummarizedExperiment but not in SingleCellExperiment. It stores the tree structure(s) that provide(s) hierarchical information of assays rows or columns or both.

- **rowData** If a phylo object is available in the slot treeData to provide the hiearchical information about the rows of the assays table, the rowData would be a LinkDataFrame-class instead of DataFrame. The data on the right side of the vertical line provides the link information between the assays rows and the tree phylo object, and could be accessed via linkData; The data on the left side is the original rowData like SingleCellExperiment object.
- **colData** Similar to the explanaition for **rowData** as above.

More details about the LinkDataFrame in the rowData or colData.

- nodeLab The labels of nodes on the tree.
- nodeLab_alias The alias of node labels on the tree.
- nodeNum The numbers of nodes on the tree.
- isLeaf It indicates whether the node is a leaf node or internal node.

Value

a TreeSummarizedExperiment object

Author(s)

Ruizhu HUANG

See Also

TreeSummarizedExperiment TreeSummarizedExperiment-accessor SingleCellExperiment

Examples

```
data("tinyTree")

# the count table
count <- matrix(rpois(100, 50), nrow = 10)
rownames(count) <- c(tinyTree$tip.label)
colnames(count) <- paste("C_", 1:10, sep = "_")

# The sample information
sampC <- data.frame(condition = rep(c("control", "trt"), each = 5),
                    gender = sample(x = 1:2, size = 10, replace = TRUE))
rownames(sampC) <- colnames(count)

# build a TreeSummarizedExperiment object
tse <- TreeSummarizedExperiment(assays = list(count),
                                colData = sampC,
                                rowTree = tinyTree)
```

unionLeaf *list leaf nodes that are the descendants of at least one specified node*

Description

unionLeaf list the leaf nodes that are the desendants of (at least one) specified nodes.

Usage

```
unionLeaf(tree, node)
```

Arguments

tree	A phylo object.
node	A numeric or character vector. It specifies internal nodes that are changed to leaves via their node labels or numbers.

Value

A phylo object.

Examples

```
library(ggtree)
data(tinyTree)
ggtree(tinyTree, ladderize = FALSE) +
    geom_text2(aes(label = label), color = "darkorange",
           hjust = -0.1, vjust = -0.7) +
    geom_text2(aes(label = node), color = "darkblue",
               hjust = -0.5, vjust = 0.7) +
    geom_hilight(node = 18) +
    geom_point2()

u1 <- unionLeaf(tree = tinyTree, node = c(19, 17))
u2 <- unionLeaf(tree = tinyTree, node = c(19, 17, 7))
(u3 <- unionLeaf(tree = tinyTree, node = c(11, 17, 7)))
```

Index

∗ **datasets**
  tinyTree, 24
[,TreeSummarizedExperiment,ANY,ANY,ANY-method
  (rowLinks), 19

addLabel, 2
aggValue, 3
apply, *4*
asLeaf, 5

BiocGenerics, *20*

changeTree, 6
colLinks (rowLinks), 19
colLinks,TreeSummarizedExperiment-method
  (rowLinks), 19
colnames<-,TreeSummarizedExperiment-method
  (rowLinks), 19
colTree (rowLinks), 19
colTree,TreeSummarizedExperiment-method
  (rowLinks), 19
convertNode (transNode), 27
countLeaf, 7
countNode, 8

DataFrame, *15*, *30*
detectLoop, 9
distNode, 10

findAncestor, 10
findChild, 11
findDescendant (findOS), 12
findOS, 12
findSibling, 13

isLeaf, 14

LinkDataFrame, *15*, *28*
LinkDataFrame
  (LinkDataFrame-constructor), 15
LinkDataFrame-class, 15
LinkDataFrame-constructor, 15

matTree, 16

printNode, 17

resolveLoop, 18
rowLinks, 19
rowLinks,TreeSummarizedExperiment-method
  (rowLinks), 19
rownames<-,TreeSummarizedExperiment-method
  (rowLinks), 19
rowTree (rowLinks), 19
rowTree,TreeSummarizedExperiment-method
  (rowLinks), 19
rtree, *24*

shareNode, 21
showNode, 22
signalNode, 23
SingleCellExperiment, *19*, *20*, *28–30*
subsetByNode (rowLinks), 19
subsetByNode,TreeSummarizedExperiment-method
  (rowLinks), 19
SummarizedExperiment, *29*

tinyTree, 24
toTree, 25
trackNode, 26
transNode, 27
TreeSummarizedExperiment, *4*, *20*, *29*, *30*
TreeSummarizedExperiment
  (TreeSummarizedExperiment-constructor),
  29
TreeSummarizedExperiment-accessor
  (rowLinks), 19
TreeSummarizedExperiment-class, 28
TreeSummarizedExperiment-constructor,
  29

unionLeaf, 31