Package 'TreeSummarizedExperiment'

November 21, 2019

Type Package

Title TreeSummarizedExperiment: a S4 Class for Data with Tree Structures

Version 1.2.0

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, utils, ape, dplyr, SummarizedExperiment

VignetteBuilder knitr

Suggests ggtree, BiocStyle, knitr, rmarkdown, testthat

RoxygenNote 6.1.1

Collate 'classValid.R' 'allClass.R' 'aggValue.R' 'allGenerics.R' 'classAccessor.R' 'data.R' 'tree_countLeaf.R' 'tree_countNode.R' 'tree_distNode.R' 'tree_findAncestor.R' 'tree_findOS.R' 'tree_findSibling.R' 'tree_isLeaf.R' 'tree_matTree.R' 'tree_printNode.R' 'tree_pruneTree.R' 'tree_shareNode.R' 'tree_signalNode.R' 'tree_toTree.R' 'tree_trackNode.R' 'tree_transNode.R'

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

git_branch RELEASE_3_10

git_last_commit 96ad84f

git_last_commit_date 2019-10-29

Date/Publication 2019-11-20

Author Ruizhu Huang [aut, cre] (<https://orcid.org/0000-0003-3285-1945>)

Maintainer Ruizhu Huang <ruizhuRH@gmail.com>

R topics documented:

`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, colLevel = NULL, FUN = sum,
  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.
`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.
`FUN`	A function to be applied on the aggregation. It's similar to the `FUN` in `apply`
`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)

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

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

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))
```

`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)
```

findOS *Find descendants (or offsprings)*

Description

findOS finds descendants of a node.

Usage

```
findOS(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 <- findOS(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: An S4 class extended (indirectly) from DataFrame*

An S4 class LinkDataFrame

Description

The **LinkDataFrame** class is extended (indirectly) from the **DataFrame** class 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-class DataFrame-class

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 = "leaf")
```

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"))
```

`pruneTree` *remove branches of a phylo tree*

Description

`pruneTree` is to remove branches that the specified leaves are in from a `phylo` tree

Usage

```
pruneTree(tree, rmLeaf, mergeSingle = TRUE)
```

Arguments

`tree`	A phylo object.
`rmLeaf`	A numeric or character vector. The labels or numbers of leaves to be removed.
`mergeSingle`	A logical value, TRUE or FALSE. If TRUE, the internal node that has only one child will be omitted. For example, a tree has structure as A-[B,C-D] (A has two children B, and C. C has a child D). If `mergeSingle = TRUE`, the node C is removed and D become the child of A. If `mergeSingle = FALSE`, the node C is kept. Another example: the tree has a structure as A-[B,C-[D,E]] (The root is A, and A has two children B, and C. C has two children D, and E.) If mergeSingle = TRUE, nothing changes because there is no internal node with only one child.

Value

A phylo object.

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) +
    geom_hilight(node = 18) +
    geom_point2()

# remove the blue branch
NT1 <- pruneTree(tree = tinyTree, rmLeaf = c(4, 5),
                 mergeSingle = FALSE)

ggtree(NT1, branch.length = "none") +
    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 <- pruneTree(tree = tinyTree, rmLeaf = c(4, 5),
                 mergeSingle = TRUE)
# or use the ape::drop.tip
# NT3 <-  ape::drop.tip(phy = tinyTree, tip = 4:5)
# all.equal(NT2, NT3)

ggtree(NT2, branch.length = "none") +
    geom_text2(aes(label = label), color = "darkorange",
                hjust = -0.1, vjust = -0.7) +
    geom_point2()
```

`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]
```

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

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)
```

---

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, cache = FALSE)
```

Arguments

data	A data frame or matrix.
cache	A logical value, TRUE or FALSE. The default is FALSE. If TRUE, the output 'phylo' has 6 elements (edge, tip.label, edge.length, Nnode, node.label, and cache). The **cache** is a list that has the length equals to the number of internal node, and each of its element stores the descendant leaves. The list is named with the alias labels of internal nodes. The alias labels are created by prefixing the node numbers with alias_

Details

The last column is used as the leaf nodes

Value

a phylo object

Author(s)

Ruizhu HUANG

Examples

```
taxTab <- data.frame(R1 = rep("A", 5),
R2 = c("B1", rep("B2", 4)),
R3 = c("C1", "C2", "C3", "C3", "C4"))

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)
```

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

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

Usage

```
transNode(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
transNode(tinyTree, node = c(11, 2, 4, 15))

transNode(tree = tinyTree, node = c("Node_16", "Node_11"))
transNode(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 to pass to the SummarizedExperiment constructor.
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-class. 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-class 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)
```

Index