

R documentation

of ‘./man/add_data.Rd’ etc.

July 13, 2020

add_data

Set data on either probe-node matrix at a new or existing node

Description

At a given node `n_new`, this function adds data `d` to the either the methyl probe-node matrix or p-value probe-node matrix (depending on the `matrix` parameter). Make sure that the sizes of `d` and the probe-node matrix are compatible before calling this function.

Usage

```
add_data(icceTree, n_new, d, matrix = 0)
```

Arguments

<code>icceTree</code>	icceTree data structure
<code>n_new</code>	node number
<code>d</code>	column matrix of methylation status for a node across many probes
<code>matrix</code>	numerical value indicating which probe-node matrix to add data to (0 == methyl states, 1 == p-values)

Value

probe-node matrix with data `d` stored at node `n_new`

Examples

```
tree <- add_data(icceTree, 12, probe_node_matrix['6'], matrix=0)
```

```
betas_to_consensus_vector
```

Convert betas matrix to consensus vectors of probes for each cell group

Description

This function creates a consensus vector for each group using the betas matrix. Sex cells and unwanted cell groups given the function are removed first. Then the cells are fit to a bimodal distribution and finally compiled into consensus vectors for each cell group.

Usage

```
betas_to_consensus_vector(
  reference_file,
  series_matrix_file,
  betas,
  group_names,
  remove_names
)
```

Arguments

series_matrix_file	directory location of where given series matrix file is located
betas	matrix (rows are cell names, columns are probe names)
group_names	list of cell names in betas
remove_names	list of cell names in betas
reference	reference file

Value

consensus vector for each cell in group names

Examples

```
consensus_vector <- betas_to_consensus_vector(reference, 'GSE110554/samples_GEOLoadSeriesMatrix.rds',
betas, c('NK', "Monocytes", ...), c('A.mix', 'B.mix', ...))
```

```
build_icceTree
```

Fully generate probe-node matrix with available data

Description

This function fully generates the probe-node matrix with a given tree. It performs ancestral tree reconstruction using Fitch's algorithm.

Usage

```
build_icceTree(consensus_vector, tree)
```

Arguments

```
consensus_vector      consensus vector data frame for each cell in group names
tree                  ape::phylo tree
```

Value

prone-node matrix storing the methylation status across all probes and nodes on a tree

Examples

```
icceTree <- build_icceTree(consensus_vector, tree)
tree <- icceTree$tree
probe_node_matrix <- icceTree$probe_node_matrix
```

```
create_consensus_vector
```

Create consensus vector of probes for each group

Description

This function creates consensus vector for each group using majority ruling: across all probes, whichever category (methylated/unmethylated) is represented in more than 2/3's of the sample will be saved. Otherwise, -1 for ambiguous is saved. A probabilistic model will supplement this in the future.

Usage

```
create_consensus_vector(betas, group_names)
```

Arguments

```
betas                matrix (rows are cell names, columns are probe names)
group_names          list of cell group names in betas matrix
```

Value

consensus vector for each cell in group names

Examples

```
consensus_vector <- create_consensus_vector(betas, c('NK', "Monocytes",
...))
```

create_node_matrix	Create node matrix used for tree walking functions
--------------------	--

Description

Returns a matrix which tracks where recursive functions are in a tree structure. This is passed to most recursive functions in this package.

Usage

```
create_node_matrix(tree)
```

Arguments

tree	ape::phylo tree
------	-----------------

Value

node matrix with one column for ancestors and another column for descendants

Examples

```
node_matrix <- create_node_matrix(tree)
```

cut_tree	Recursively walk through a tree and structure remove descendants of given node.
----------	---

Description

This function will recursively walk through a given tree and remove all of the descendants of a given node, making the given node a new leaf.

Usage

```
cut_tree(node, root, node_matrix, i)
```

Arguments

node	node number
root	root of non-proper subtree
node_matrix	node matrix obtained from create_node_matrix function
i	row in node matrix, starting at 1

Value

None

Examples

```
cut_tree(node, get_root(tree), node_matrix, 1)
```

`fit_betas_to_distribution`*Categorize betas based on bimodal distribution*

Description

This function uses a given bimodal distribution generated within cell types in order to categorize betas as either methylated (1), unmethylated (0), or ambiguous (-1).

Usage

```
fit_betas_to_distribution(betas, bi)
```

Arguments

<code>betas</code>	matrix (rows are cell names, columns are probe names)
<code>bi</code>	matrix specifying μ_0 , μ_1 , σ_0 , σ_1 across all cell types

Value

betas matrix categorized based on bimodal distribution

Examples

```
betas <- fit_betas_to_distribution(betas, bi)
```

`fit_betas_to_distribution_p`*Categorize betas based on bimodal distribution and return the p-value of each categorization.*

Description

This function uses a given bimodal distribution generated within cell types in order to categorize betas as either methylated (1) or unmethylated (0) in the betas structure. It also returns the p-value of each categorization in the `betas_p` structure.

Usage

```
fit_betas_to_distribution_p(betas, bi)
```

Arguments

<code>betas</code>	matrix (rows are cell names, columns are probe names)
<code>bi</code>	matrix specifying μ_0 , μ_1 , σ_0 , σ_1 across all cell types

Value

betas matrix categorized based on bimodal distribution

Examples

```
betas <- fit_betas_to_distribution(betas, bi)
```

```
generate_bimodal_distribution
```

Generate bimodal distribution on cell types

Description

This function generates a bimodal distribution on the betas matrix data within each cell type

Usage

```
generate_bimodal_distribution(betas)
```

Arguments

betas matrix (rows are cell names, columns are probe names)

Value

matrix specifying mu0, mu1, sigma0, sigma1 across all cell types

Examples

```
bi <- generate_bimodal_distribution(betas)
```

```
generate_branch_changes
```

Generate list of relevant probes for gain, loss, and change in methylation status

Description

This function returns relevant probe lists for each node that exhibits either a gain or a loss in methylation status. It also returns three sets of branch labels: number of methylation changes, number of methylation gains, and number of methylation losses. These labels can be given to the show_tree function as an edge labels parameter to visualize this data.

Usage

```
generate_branch_changes(icceTree)
```

Arguments

icceTree icceTree data structure

probe_node_matrix

probe-node matrix storing the methylation status across all probes and nodes on a tree

Value

a structure containing a list of probes with differentially methylated probes for each node (gain and loss stored separately), a list of branch labels for the number of total changes in methylation, a list of branch labels for the number of gains in methylation, and a list of branch labels for the number of losses in methylation

Examples

```
probe_information <- generate_branch_changes(icceTree)
relevant_probes <- probe_information$relevant_probes
total_changes_label <- probe_information$total_changes_label
meth_gains_label <- probe_information$meth_gains_label
meth_losses_label <- probe_information$meth_losses_label
```

generate_p_values	<i>Work on this...</i>
-------------------	------------------------

Description

Given...

Usage

```
generate_p_values(x, mu0, mu1, nsamples = 10)
```

Arguments

x	data structure
mu0	name of variable/file
mu1	directory to which the data should be saved
nsamples	number of cell types

Value

None

Examples

```
save_data(betas, 'betas', getwd())
```

get_leaves	<i>Get the leaves of a tree</i>
------------	---------------------------------

Description

Given a tree, this function returns its leaves.

Usage

```
get_leaves(icceTree)
```

Arguments

icceTree	icceTree data structure
----------	-------------------------

Value

a list of nodes representing the leaves of the tree

Examples

```
leaves <- get_leaves(icceTree)
```

get_probes_of_gene	<i>Get probes along a given gene</i>
--------------------	--------------------------------------

Description

This function returns a list of probes along a particular gene using the given reference file.

Usage

```
get_probes_of_gene(reference, gene)
```

Arguments

reference	reference file
gene	gene name

Value

list of probes along a particular gene

Examples

```
gene_probes <- get_probes_of_gene(reference, gene)
```

get_reference	<i>Return reference data frame</i>
---------------	------------------------------------

Description

This function returns the contents of a given reference file as a data frame. The rows of the data frame are the probe names, allowing for quick indexing.

Usage

```
get_reference(reference_file)
```

Arguments

reference_file file location of reference file

Value

reference file

Examples

```
reference <- get_reference('~references/hg38/annotation/R/
EPIC.hg38.manifest.rds')
```

get_relevant_genes	<i>Find relevant genes for gain and loss of methylation</i>
--------------------	---

Description

This function returns a list of relevant gene for each node that exhibit a gain or loss in methylation status.

Usage

```
get_relevant_genes(icceTree, relevant_probes, reference)
```

Arguments

icceTree	icceTree data structure
relevant_probes	list of differentially methylated probes for each node
reference	reference file

Value

list of differentially methylated genes for each node

Examples

```
relevant_genes <- get_relevant_genes(icceTree, relevant_probes, reference)
```

get_root	<i>Get root of a tree</i>
----------	---------------------------

Description

Returns the root of a given tree structure.

Usage

```
get_root(icceTree)
```

Arguments

icceTree icceTree data structure

Value

node corresponding the the root of the tree

Examples

```
root <- get_root(icceTree)
```

grow_off_leaf	<i>Grow a tree at a given leaf node</i>
---------------	---

Description

Add a pair of leaves branching off of a given leaf node on a tree. Assigns the two newly added leaves labels from node_names. The old leaf where the two new branches were add is also assigned as an internal node.

Usage

```
grow_off_leaf(icceTree, leaf, node_names)
```

Arguments

icceTree icceTree data structure
leaf node number
node_names list of two node names

Value

new tree topology—two branches added to given leaf and labeled accordingly

Examples

```
icceTree <- grow_off_leaf(icceTree, 1, c('Fibroblasts', 'Fibrocytes'))
```

icce_tree	<i>Create icce tree object</i>
-----------	--------------------------------

Description

Returns icce tree object given a probe-node matrix, a probe-node p-values and a ape::phylo tree

Usage

```
icce_tree(probe_node_matrix, probe_node_matrix_p, tree)
```

Arguments

probe_node_matrix	probe-node matrix storing the methylation status across all probes and nodes on a tree
probe_node_matrix_p	probe-node matrix storing the p-value of each node across all probes on a tree
tree	ape::phylo tree

Value

icce tree object

Examples

```
reconstruct_internal_nodes(get_root(tree), node_matrix, 1)
```

inherit_parental_state	<i>Inherit parental states to ambiguous nodes</i>
------------------------	---

Description

This function will recursively walk through a given tree and assigns parental states to ambiguous children nodes

Usage

```
inherit_parental_state(root, node_matrix, i)
```

Arguments

root	root of non-proper subtree
node_matrix	matrix obtained from create_node_matrix function
i	row in node matrix, start at 1

Value

None

Examples

```
inherit_parental_state(get_root(tree), node_matrix, 1)
```

```
initialize_probe_node_matrix
```

Initialize the probe-node matrix

Description

This function simply initializes a probe-node matrix across given nodes and probes with zeros.

Usage

```
initialize_probe_node_matrix(consensus_vector, tree)
```

Arguments

consensus_vector

consensus vector data frame for each cell in group names

tree

ape::phylo tree

Value

initialized probe_node_matrix

Examples

```
probe_node_matrix <- initialize_probe_node_matrix(consensus_vector,
  upgma_tree(consensus_vector))
```

```
ninternal_nodes
```

Get number of internal nodes on a tree

Description

Returns the number of internal nodes on the given tree.

Usage

```
ninternal_nodes(icceTree)
```

Arguments

icceTree

icceTree data structure

Value

number of internal nodes on a given tree

Examples

```
number_of_internal_nodes = ninternal_nodes(icceTree)
```

nleaves	<i>Get number of leaves of a tree</i>
---------	---------------------------------------

Description

Returns the number of leaves on a given tree.

Usage

```
nleaves(icceTree)
```

Arguments

icceTree	icceTree data structure
----------	-------------------------

Value

number of leafs of a given tree

Examples

```
number_of_leaves = nleaves(icceTree)
```

nnodes	<i>Get total number of nodes on a tree</i>
--------	--

Description

Returns the number of nodes on a given tree.

Usage

```
nnodes(icceTree)
```

Arguments

icceTree	icceTree data structure
----------	-------------------------

Value

number of nodes on a tree

Examples

```
number_of_nodes = nnodes(icceTree)
```

nprobes	<i>Get number of probes in consensus vector</i>
---------	---

Description

Returns the number of probes in a list of consensus vectors

Usage

```
nprobes(consensus_vector)
```

Arguments

consensus_vector
consensus vector data frame for each cell in group names

Value

number of probes

Examples

```
number_of_probes <- nprobes(consensus_vector)  
... some visualization
```

read_IDAT	<i>Load betas data</i>
-----------	------------------------

Description

Using sesame, this function loads betas data using IDAT files stored in a given directory

Usage

```
read_IDAT(IDAT_dir)
```

Arguments

IDAT_dir directory location of where given IDAT file is located

Value

betas matrix

Examples

```
betas <- read_IDAT('~/.20191212_GEO_datasets/GSE110554')
```

```
reconstruct_internal_nodes
```

Reconstruct internal nodes

Description

This function will recursively walk through a given tree and reconstruct all of its internal states using Fitch's algorithm

Usage

```
reconstruct_internal_nodes(root, node_matrix, i)
```

Arguments

root	root of non-proper subtree
node_matrix	matrix obtained from create_node_matrix function
i	row in node matrix, start at 1

Value

None

Examples

```
reconstruct_internal_nodes(get_root(tree), node_matrix, 1)
```

```
relabel_node_matrix
```

Relabel node in probe-node matrix

Description

Relabel node label n_old with node label n_new in either probe-node matrix (depending on the matrix parameter).

Usage

```
relabel_node_matrix(icceTree, n_old, n_new, matrix = 2)
```

Arguments

icceTree	icceTree data structure
n_old	node number
n_new	node number
matrix	numerical value indicating which probe-node matrix to add data to (0 == methyl states, 1 == p-values)

Value

relabeled probe_node_matrix with node label n_old with node label n_new

Examples

```
probe_node_matrix <- relabel_node_matrix(icceTree, 7, 15, matrix=2)
```

relabel_tree	<i>Relabel node on tree</i>
--------------	-----------------------------

Description

Relabel node label n_old with node label n_new on tree.

Usage

```
relabel_tree(icceTree, n_old, n_new)
```

Arguments

icceTree	icceTree data structure
n_old	node number
n_new	node number

Value

relabeled tree with node label n_old with node label n_new

Examples

```
icceTree <- relabel_tree(icceTree, 7, 15)
```

remove_NA_probes	<i>Removes NA probes</i>
------------------	--------------------------

Description

Returns consensus vector with NA probes removed

Usage

```
remove_NA_probes(consensus_vector)
```

Arguments

consensus_vector	consensus vector data frame for each cell in group names
------------------	--

Value

consensus vector without NA probes

Examples

```
consensus_vector <- remove_NA_probes(consensus_vector)
```

```
remove_branches_from_node
```

Cut tree at specified node

Description

Given a tree and a selected node, this function removes the branches stemming off of that node, thus making that node a leaf. It returns this altered tree.

Usage

```
remove_branches_from_node(icceTree, node)
```

Arguments

icceTree	icceTree data structure
node	node number

Value

new tree topology—node is now a leaf and all of node's descendants are removed.

Examples

```
icceTree <- remove_branches_from_node(icceTree, node=8)
```

```
remove_cell_types
```

Removes specific cell types from betas matrix

Description

This function removes specified group names. Common names include MIX or are typically heterogeneous mixtures of cells.

Usage

```
remove_cell_types(betas, remove_names)
```

Arguments

betas	matrix (rows are cell names, columns are probe names)
remove_names	list of cell names in betas

Value

betas matrix without specified cell names

Examples

```
betas <- remove_cell_types(betas, c('A.mix', 'B.mix', ...))
```

```
remove_intermediate_probes
```

Removes intermediate probes

Description

Returns consensus vector with intermediate probes removed.

Usage

```
remove_intermediate_probes(consensus_vector)
```

Arguments

consensus_vector

consensus vector data frame for each cell in group names

Value

consensus vector without intermediate probes

Examples

```
consensus_vector <- remove_intermediate_probes(consensus_vector)
... some visualization
```

```
remove_sex_chr
```

Removes probes on sex chromosomes

Description

This function removes probes located on sex chromosomes because these probes have a disproportionate global methylation status

Usage

```
remove_sex_chr(reference, betas)
```

Arguments

reference reference file

betas matrix (rows are probe names, columns are cell names)

Value

betas matrix without probes located on sex chromosomes.

Examples

```
betas <- remove_sex_chr(reference, betas)
```

remove_unapplicable_probes

Removes all inapplicable probes

Description

Returns consensus vector with NA, intermediate, and zero parsimony probes removed

Usage

```
remove_unapplicable_probes(consensus_vector)
```

Arguments

consensus_vector

consensus vector data frame for each cell in group names

Value

consensus vector without inapplicable probes

Examples

```
consensus_vector <- remove_unapplicable_probes(consensus_vector)
```

remove_zero_parsimony_probes

Removes zero parsimony probes

Description

Returns consensus vector with zero parsimony probes removed

Usage

```
remove_zero_parsimony_probes(consensus_vector)
```

Arguments

consensus_vector

consensus vector data frame for each cell in group names

Value

consensus vector without zero parsimony probes

Examples

```
consensus_vector <- remove_zero_parsimony_probes(consensus_vector)
... some visualization
```

save_data	<i>Saves specified data with variable name to working directory</i>
-----------	---

Description

Given data, a variable name, and a location for that data to be stored, this function simply saves data.

Usage

```
save_data(data, variable_name, dir)
```

Arguments

data	data structure
variable_name	name of variable/file
dir	directory to which the data should be saved

Value

None

Examples

```
save_data(betas, 'betas', getwd())
```

show_gene_track	<i>Track the single probe resolution change of a gene</i>
-----------------	---

Description

Displays constructed tree with node labels for the proportion of each probe state for a selected gene. This is saved at the given filename.

Usage

```
show_gene_track(
  filename,
  tree,
  gene,
  thermo_prop_internal_nodes,
  thermo_prop_leaves
)
```

Arguments

filename	filename to which the plot will be saved
tree	ape::phylo tree
gene	gene name
thermo_prop_leaves	thermo proportion plot for leaves of all of the probe states
thermo_prop_node	thermo proportion plot for internal nodes of all of the probe states

Value

None

Examples

```
show_gene_track('tree_CD81', tree=icceTree$tree, gene='CD81',
thermo_prop_internal_nodes=thermo_prop_internal_nodes, thermo_prop_leaves=
thermo_prop_leaves)
... some visualization
```

show_heatmap	<i>Display heatmap of probes In progress</i>
--------------	--

Description

This is in progress

Usage

```
show_heatmap(betas, reference)
```

Arguments

betas	betas
reference	reference

Examples

```
head_map <- show_heatmap()
```

show_tree	<i>Display tree</i>
-----------	---------------------

Description

Displays constructed tree and saves it at the given filename.

Usage

```
show_tree(filename, tree, title, edge_labels)
```

Arguments

filename	filename to which the plot will be saved
tree	ape::phylo tree
title	title name
edge_labels	optional labels for each branch

Value

None

Examples

```
show_tree(filename='tree.pdf', tree=icceTree$tree, title='Hematopoietic Tree',
edge_labels=meth_gains_label)
```

track_gene	<i>Get proportion of probes methylated, unmethylated, ambiguous at each node across a tree</i>
------------	--

Description

This function returns the proportion of each probes that are methylated, unmethylated, ambiguous at each node across a tree.

Usage

```
track_gene(icceTree, reference, gene)
```

Arguments

icceTree	icceTree data structure
reference	reference file
gene	gene name
probe_node_matrix	matrix storing the methylation status across all probes and nodes on a tree

Value

a structure composed of two lists that contain methylation state proportions for internal nodes and methylation state proportions for leaves

Examples

```
gene_information <- track_gene(icceTree, reference, 'CD81')
thermo_prop_internal_nodes = gene_information$thermo_prop_internal_nodes
thermo_prop_leaves = gene_information$thermo_prop_leaves
```

upgma_tree	<i>Get UPGMA tree construction</i>
------------	------------------------------------

Description

This function uses the UPGMA algorithm to construct a phylogenetic tree using methylation data.

Usage

```
upgma_tree(consensus_vector)
```

Arguments

consensus_vector
consensus vector data frame for each cell in group names

Value

ape::phylo tree constructed using UPGMA algorithm

Examples

```
tree <- upgma_tree(consensus_vector)
```

Index

add_data, [1](#)

betas_to_consensus_vector, [2](#)

build_icceTree, [2](#)

create_consensus_vector, [3](#)

create_node_matrix, [4](#)

cut_tree, [4](#)

fit_betas_to_distribution, [5](#)

fit_betas_to_distribution_p, [5](#)

generate_bimodal_distribution, [6](#)

generate_branch_changes, [6](#)

generate_p_values, [7](#)

get_leaves, [8](#)

get_probes_of_gene, [8](#)

get_reference, [9](#)

get_relevant_genes, [9](#)

get_root, [10](#)

grow_off_leaf, [10](#)

icce_tree, [11](#)

inherit_parental_state, [11](#)

initialize_probe_node_matrix, [12](#)

ninternal_nodes, [12](#)

nleaves, [13](#)

nnodes, [13](#)

nprobes, [14](#)

read_IDAT, [14](#)

reconstruct_internal_nodes, [15](#)

relabel_node_matrix, [15](#)

relabel_tree, [16](#)

remove_branches_from_node, [17](#)

remove_cell_types, [17](#)

remove_intermediate_probes, [18](#)

remove_NA_probes, [16](#)

remove_sex_chr, [18](#)

remove_unapplicable_probes, [19](#)

remove_zero_parsimony_probes, [19](#)

save_data, [20](#)

show_gene_track, [20](#)

show_heatmap, [21](#)

show_tree, [22](#)

track_gene, [22](#)

upgma_tree, [23](#)