Package ‘GeRnika’

January 20, 2025

Type Package

Title Simulation, Visualization and Comparison of Tumor Evolution Data

Version 1.0.0

Description Simulating, visualizing and comparing tumor clonal data by using simple commands. This aims at providing a tool to help researchers to easily simulate tumor data and analyze the results of their approaches for studying the composition and the evolutionary history of tumors.

Encoding UTF-8

Imports data.tree, purrr, reshape2, DiagrammeR, colorspace, dplyr, methods, vctrs, magrittr, stats

Depends R (>= 4.00)

RoxygenNote 7.3.2

VignetteBuilder knitr, rmarkdown, knitcitations, ggpubr, ggplot2

Suggests knitr, rmarkdown, knitcitations, ggpubr, ggplot2

NeedsCompilation no

LazyData TRUE

Author Aitor Sánchez-Ferrera [cre, aut] (<https://orcid.org/0000-0001-6127-0686>),
Maitena Tellaetxe-Abete [aut] (<https://orcid.org/0000-0003-1894-4547>),
Borja Calvo [aut] (<https://orcid.org/0000-0001-9969-9664>)

Maintainer Aitor Sánchez-Ferrera <`aitor.sanchezf@ehu.eus`>

Repository CRAN

Date/Publication 2024-09-04 13:30:09 UTC

Contents

`.distribute_freqs` *Distribute frequencies among clone and its children clones*

Description

This function distributes frequencies among the clones in a multifurcation of a phylogenetic tree. It uses a Dirichlet distribution to generate random proportions for a clone and its children, and then normalizes the children's proportions to the parent's proportion. This is an internal function used by `calc_clone_proportions`.

Usage

```
.distribute_freqs(B, clone_idx, clone_proportions, selection)
```

Arguments

`B`	A matrix representing the mutation relationships between the nodes in the phylogenetic tree (B matrix).
`clone_idx`	An integer representing the index of the clone whose own proportion and its children's proportions are going to be updated.

`clone_proportions`	A numeric vector representing the proportions of each clone in the phylogenetic tree.
`selection`	A character string representing the evolutionary mode the tumor follows. This should be either "positive" or "neutral".

Value

A numeric vector representing the updated proportions of each clone in the phylogenetic tree.

`add_noise` *Add noise to the VAF values in an F matrix*

Description

This function adds noise to the variant allele frequency (VAF) values in an F matrix, simulating the effect of sequencing errors. The noise is modeled as a negative binomial distribution for the depth of the reads and a binomial distribution for both the variant allele counts and the mismatch counts.

Usage

```
add_noise(F_matrix, depth, overdispersion)
```

Arguments

`F_matrix`	A matrix representing the true VAF values of a series of mutations in a set of samples (F matrix).
`depth`	A numeric value representing the mean depth of sequencing.
`overdispersion`	A numeric value representing the overdispersion parameter for the negative binomial distribution used to simulate the depth of sequencing.

Value

A matrix containing noisy VAF values of a series of mutations in a set of samples.

Examples

```
# Calculate the noisy VAF values of a series of mutations in a set of samples, given the true
# VAF values in the F matrix F_true, a depth of 30 and an overdispersion of 5

# Create an instance of a tumor with 50 clones,
# 10 samples, k = 5, positive selection without noise
F_true <- create_instance(
  n = 50,
  m = 10,
  k = 5,
  selection = "positive",
  noisy = FALSE)$F
```

```
# Then we add the noise using a depth of 30 and an overdispersion of 5.
noisy_F <- add_noise(F_true, 30, 5)
```

`B_mats` *A set of 10 trios of B matrices for experimenting with the methods of* `GeRnika`

Description

A list of lists composed by 10 trios of B matrices; a real B matrix, a B matrix got by using the GRASP method and another one as a result of an ILS. These matrices can be used as examples for the methods of `GeRnika`.

Usage

```
B_mats
```

Format

A list of lists composed by 10 trios of B matrices.

Trio 1 B_real, B_grasp and B_opt (matrices composed by 5 clones)

Trio 2 B_real, B_grasp and B_opt (matrices composed by 5 clones)

Trio 3 B_real, B_grasp and B_opt (matrices composed by 5 clones)

Trio 4 B_real, B_grasp and B_opt (matrices composed by 5 clones)

Trio 5 B_real, B_grasp and B_opt (matrices composed by 5 clones)

Trio 6 B_real, B_grasp and B_opt (matrices composed by 10 clones)

Trio 7 B_real, B_grasp and B_opt (matrices composed by 10 clones)

Trio 8 B_real, B_grasp and B_opt (matrices composed by 10 clones)

Trio 9 B_real, B_grasp and B_opt (matrices composed by 10 clones)

Trio 10 B_real, B_grasp and B_opt (matrices composed by 10 clones)

Source

Local source; as a result of the Grasp and the ILS methods used for solving the Clonal Deconvolution and Evolution Problem (CDEP).

`B_to_phylotree` *Create a* `Phylotree` *object from a* `B` *matrix.*

Description

This function creates a `Phylotree` class object from a `B` matrix.

Usage

```
B_to_phylotree(B, labels = NA)
```

Arguments

`B`	A square matrix that represents the phylogenetic tree.
`labels`	An optional vector containing the tags of the genes in the phylogenetic tree. `NA` by default.

Value

A `Phylotree` class object.

Examples

```
# Create a B matrix instance
# composed by 10 subpopulations of
# clones
B <- create_instance(
       n = 10,
       m = 4,
       k = 1,
       selection = "neutral")$B

# Create a new 'Phylotree' object
# on the basis of the B matrix
phylotree <- B_to_phylotree(B = B)

# Generate the tags for the genes of
# the phyogenetic tree
tags <- LETTERS[1:nrow(B)]

# Create a new 'Phylotree' object
# on the basis of the B matrix and
# the list of tags
phylotree_tags <- B_to_phylotree(
                    B = B,
                    labels = tags)
```

`calc_clone_proportions` *Calculate clone proportions for a tumor*

Description

This function calculates the proportions of each clone in a phylogenetic tree, following a given evolutionary mode (positive selection or neutral evolution).

Usage

```
calc_clone_proportions(B, selection)
```

Arguments

`B`	A matrix representing the mutation relationships between the nodes in the phylogenetic tree (B matrix).
`selection`	A character string representing the evolutionary mode the tumor follows. This should be either "positive" or "neutral".

Value

A data frame with two columns: 'clone_idx', which contains the clone identifiers, and 'proportion', which contains the calculated proportions of each clone.

Examples

```
# Calculate clone proportions for a tumor phylogenetic tree represented by a B matrix
# and following a positive selection model

# Create a mutation matrix for a phylogenetic tree with 10 nodes and k = 2
B_mat <- create_B(10, 2)

# Calculate the clone proportions following a positive selection model
clone_proportions <- calc_clone_proportions(B_mat, "positive")
```

`combine_trees` *Get consensus tree between two phylogenetic trees*

Description

Returns a graph representing the consensus tree between two phylogenetic trees.

Usage

```
combine_trees(
  phylotree_1,
  phylotree_2,
  palette = GeRnika::palettes$Simpsons,
  labels = FALSE
)
```

Arguments

`phylotree_1`	A `Phylotree` class object.
`phylotree_2`	A `Phylotree` class object.
`palette`	A vector composed by the hexadecimal code of three colors. "The Simpsons" palette used as default.
`labels`	A boolean, if `TRUE` the resulting graph will be plotted with the tags of the genes in the phylogenetic trees instead of their mutation index. `FALSE` by default.

Value

a `dgr_graph` object representing the consensus graph between `phylotree_1` `phylotree_2`.

Examples

```
# Load the predefined B matrices of the package
B_mats <- GeRnika::B_mats


B_real <- B_mats[[2]]$B_real
B_opt <- B_mats[[2]]$B_opt


# Generate the tags for the genes of
# the phyogenetic tree
tags <- LETTERS[1:nrow(B_real)]


# Instantiate two \code{Phylotree} class objects on
# the basis of the B matrices
phylotree_real <- B_to_phylotree(
                    B = B_real,
                    labels = tags)

phylotree_opt <- B_to_phylotree(
                    B = B_opt,
                    labels = tags)


# Create the consensus tree between phylotree_real
# and phylotree_opt
consensus <- combine_trees(
```

```
                phylotree_1 = phylotree_real,
                phylotree_2 = phylotree_opt)


# Render the consensus tree
DiagrammeR::render_graph(consensus)


# Load another palette
palette_1 <- GeRnika::palettes$Lancet


# Create the consensus tree between phylotree_real
# and phylotree_opt using tags and another palette
consensus_tag <- combine_trees(
                    phylotree_1 = phylotree_real,
                    phylotree_2 = phylotree_opt,
                    palette = palette_1,
                    labels = TRUE)


# Render the consensus tree using tags and the
# selected palette
DiagrammeR::render_graph(consensus_tag)
```

`create_B` *Create tumor phylogenetic tree topology*

Description

This function generates a mutation matrix (B matrix) for a tumor phylogenetic tree with a given number of nodes. This matrix represents the topology and it is created randomly, with the probability of a node to be chosen as a parent of a new node being proportional to the number of its ascendants raised to the power of a constant ‘k‘.

Usage

```
create_B(n, k)
```

Arguments

`n`	An integer representing the number of nodes in the phylogenetic tree.
`k`	A numeric value representing the constant used to calculate the probability of a node to be chosen as a parent of a new node.

Value

A square matrix representing the mutation relationships between the nodes in the phylogenetic tree. Each row corresponds to a node, and each column corresponds to a mutation. The value at the i-th row and j-th column is 1 if the i-th node has the j-th mutation, and 0 otherwise.

Examples

```
# Create a mutation matrix for a phylogenetic tree with 10 nodes and k = 2
B <- create_B(10, 2)
```

`create_instance` *Create a tumor phylogenetic tree instance*

Description

This function generates a tumor phylogenetic tree instance, composed by a mutation matrix (B matrix), a matrix of true variant allele frequencies (F_true), a matrix of noisy variant allele frequencies (F), and a matrix of clone frequencies in samples (U).

Usage

```
create_instance(
  n,
  m,
  k,
  selection,
  noisy = TRUE,
  depth = 30,
  seed = Sys.time()
)
```

Arguments

`n`	An integer representing the number of clones.
`m`	An integer representing the number of samples.
`k`	A numeric value that determines the linearity of the tree topology. Also referred to as the topology parameter. Increasing values of this parameter increase the linearity of the topology. When ‘k‘ is set to 1, all nodes have equal probabilities of being chosen as parents, resulting in a completely random topology.
`selection`	A character string representing the evolutionary mode the tumor follows. This should be either "positive" or "neutral".
`noisy`	A logical value indicating whether to add noise to the frequency matrix. If ‘TRUE‘, noise is added to the frequency matrix. If ‘FALSE‘, no noise is added. ‘TRUE‘ by default.
`depth`	A numeric value representing the mean depth of sequencing. 30 by default.
`seed`	A numeric value used to set the seed for the random number generator. Sys.time() by default.

Details

The B matrix is a square matrix representing the mutation relationships between the clones in the tumor, or, in other words, it represents the topology of the phylogenetic tree. The F_true matrix represents the true variant allele frequencies of the mutations present in the tumor in a set of samples. The F matrix represents the noisy variant allele frequencies of the mutations in the same set of samples. The U matrix represents the frequencies of the clones in the tumor in the set of samples.

Value

A list containing four elements: 'F', a matrix representing the noisy frequencies of each mutation in each sample; 'B', a matrix representing the mutation relationships between the clones in the tumor; 'U', a matrix that represents the frequencies of the clones in the tumor in the set of samples; and 'F_true', a matrix representing the true frequencies of each mutation in each sample.

Examples

```
# Create an instance of a tumor with 10 clones,
# 4 samples, k = 1, neutral evolution and
# added noise with depth = 500
I1 <- create_instance(
  n = 10,
  m = 4,
  k = 1,
  selection = "neutral",
  depth = 500)


# Create an instance of a tumor with 50 clones,
# 10 samples, k = 5, positive selection and
# added noise with depth = 500
I2 <- create_instance(
  n = 50,
  m = 10,
  k = 5,
  selection = "positive",
  noisy = TRUE,
  depth = 500)


# Create an instance of a tumor with 100 clones,
# 25 samples, k = 0, positive selection without
# noise
I3 <- create_instance(
  n = 100,
  m = 25,
  k = 0,
  selection = "positive",
  noisy = FALSE)
```

`create_phylotree` *Create a* `Phylotree` *object*

Description

This is the general constructor of the `Phylotree` S4 class.

Usage

```
create_phylotree(B, clones, genes, parents, tree, labels = NA)
```

Arguments

`B`	A square matrix that represents the phylogenetic tree.
`clones`	A numeric vector representing the clones in the phylogenetic tree.
`genes`	A numeric vector representing the genes in the phylogenetic tree.
`parents`	A numeric vector representing the parents of the clones in the phylogenetic tree.
`tree`	A `data.tree` object containing the tree structure of the phylogenetic tree.
`labels`	An optional vector containing the tags of the genes in the phylogenetic tree. `NA` by default.

Value

A `Phylotree` class object.

Examples

```
# Create a B matrix instance
# composed by 10 subpopulations of
# clones
B <- create_instance(
       n = 10,
       m = 4,
       k = 1,
       selection = "neutral")$B


# Create a new 'Phylotree' object
# on the basis of the B matrix
phylotree1 <- B_to_phylotree(B = B)


# Create a new 'Phylotree' object
# with the general constructor of
# the class
phylotree2 <- create_phylotree(
                B = B,
```

```
                clones = phylotree1@clones,
                genes = phylotree1@genes,
                parents = phylotree1@parents,
                tree = phylotree1@tree)


# Generate the tags for the genes of
# the phyogenetic tree
tags <- LETTERS[1:nrow(B)]


# Create a new 'Phylotree' object
# with the general constructor of
# the class using tags
phylotree_tags <- create_phylotree(
                    B = B,
                    clones = phylotree1@clones,
                    genes = phylotree1@genes,
                    parents = phylotree1@parents,
                    tree = phylotree1@tree,
                    labels = tags)
```

`create_U` *Calculate tumor clone frequencies in samples*

Description

This function calculates the frequencies of each clone in a set of samples, given the global clone proportions in the tumor and their spatial distribution.

Usage

```
create_U(B, clone_proportions, density_coords, m, x)
```

Arguments

`B`	A matrix representing the mutation relationships between the nodes in the phylogenetic tree (B matrix).
`clone_proportions`	A data frame with two columns: ’clone_idx’, which contains the clone identifiers, and ’proportion’, which contains the proportions of each clone in the tumor.
`density_coords`	A data frame where each column represents the density of a clone at different spatial coordinates.
`m`	An integer representing the number of samples taken from the tumor.
`x`	A numeric vector representing the spatial coordinates.

Value

A matrix where each row corresponds to a sample, and each column corresponds to a clone. The value at the i-th row and j-th column is the frequency of the j-th clone in the i-th sample.

Examples

```
# Calculate the frequencies of each clone in 10 samples taken from a tumor represented by the B
# matrix B_mat, with global clone proportions clone_proportions, spatial distribution
# density_coords, and spatial coordinates x

# Create random topology
B <- create_B(20, 3)

# Assign proportions to each clone following a neutral evolution model
clone_proportions <- calc_clone_proportions(B, "neutral")

# Place clones in 1D space
clones_space <- place_clones_space(B)
density_coords <- clones_space$spatial_coords
domain <- clones_space$x

# Create U matrix with parameter m=4
U <- create_U(B = B, clone_proportions = clone_proportions,
              density_coords = density_coords, m = 4, x = domain)
```

`equals` *Check if two phylogenetic trees are equal*

Description

Checks wether two phylogenetic trees are equivalent or not.

Usage

```
equals(phylotree_1, phylotree_2)
```

Arguments

`phylotree_1`	A `Phylotree` class object.
`phylotree_2`	A `Phylotree` class object.

Value

A boolean, `TRUE` if they are equal and `FALSE` if not.

Examples

```
# Load the predefined B matrices of the package
B_mats <- GeRnika::B_mats


B_real <- B_mats[[2]]$B_real
B_opt <- B_mats[[2]]$B_opt


# Instantiate two \code{Phylotree} class objects on
# the basis of the B matrices
phylotree_real <- B_to_phylotree(
                    B = B_real)

phylotree_opt <- B_to_phylotree(
                    B = B_opt)


equals(phylotree_real, phylotree_opt)
```

`find_common_subtrees` *Find the set of common subtrees between two phylogenetic trees*

Description

Plots the common subtrees between two phylogenetic trees and prints the information about their similarities and their differences.

Usage

```
find_common_subtrees(phylotree_1, phylotree_2, labels = FALSE)
```

Arguments

`phylotree_1`	A `Phylotree` class object.
`phylotree_2`	A `Phylotree` class object.
`labels`	A boolean, if `TRUE` the rendered graph will be plotted with the tags of the genes in the phylogenetic trees instead of their gene index. `FALSE` by default.

Value

A plot of the common subtrees between two phylogenetic trees and the information about the distance between them based on their independent and common edges.

Examples

```
# Load the predefined B matrices of the package
B_mats <- GeRnika::B_mats


B_real <- B_mats[[2]]$B_real
B_opt <- B_mats[[2]]$B_opt


# Generate the tags for the genes of
# the phyogenetic tree
tags <- LETTERS[1:nrow(B_real)]


# Instantiate two Phylotree class objects on
# the basis of the B matrices using tags
phylotree_real <- B_to_phylotree(
                    B = B_real,
                    labels = tags)

phylotree_opt <- B_to_phylotree(
                    B = B_opt,
                    labels = tags)


# find the set of common subtrees between both
# phylogenetic trees
find_common_subtrees(
  phylotree_1 = phylotree_real,
  phylotree_2 = phylotree_opt)


# find the set of common subtrees between both
# phylogenetic trees using tags
find_common_subtrees(
  phylotree_1 = phylotree_real,
  phylotree_2 = phylotree_opt,
  labels = TRUE)
```

`get_clones` *Retrieve the clone indices for a set of gene indices*

Description

Retrieve the clone indices for a set of gene indices

Usage

```
get_clones(genes)
```

Arguments

`genes`	A vector of gene indices.

Value

A vector of clone indices.

`get_genes` *Retrieve the gene indices for the clones in a phylogenetic tree*

Description

Retrieve the gene indices for the clones in a phylogenetic tree

Usage

```
get_genes(B)
```

Arguments

`B`	A square matrix that represents the phylogenetic tree.

Value

A vector of gene indices.

`get_parents` *Get parent nodes in a phylogenetic tree.*

Description

This function retrieves the parent nodes for each node in a given phylogenetic tree.

Usage

```
get_parents(phylotree)
```

Arguments

`phylotree`	An object of the `Phylotree` class representing the phylogenetic tree.

Value

A vector of parent nodes.

`hyperparameters` *Hyperparameters for the methods of* `GeRnika`

Description

A data.frame containing the static values for the parameters used in the methods of `GeRnika`.

Usage

```
hyperparameters
```

Format

A data.frame containing different static values.

Overdispersion value = 0.5

Depth_sequencing value = 30

Source

local source; inspired on the optimal parameters for the methods of `GeRnika`.

`Node-class` *S4 class to represent a node in a phylogenetic tree.*

Description

S4 class to represent a node in a phylogenetic tree.

Slots

`Node` A node object

`palettes` *Palettes for the methods of* `GeRnika`

Description

A data.frame containing 3 default palettes for the parameters used in the methods of `GeRnika`.

Usage

```
palettes
```

Format

A data.frame containing 3 palettes.

Lancet #0099B444, #AD002A77, #42B540FF

NEJM #FFDC9177, #7876B188, #EE4C97FF

Simpsons #FED43966, #FD744688, #197EC0FF

Source

Lancet, NEJM and The Simpsons palettes; inspired by the plots in Lancet journals, the plots in the New England Journal of Medicine and the colors used in the TV show The Simpsons, respectively.

`Phylotree-class` *S4 class to represent phylogenetic trees.*

Description

S4 class to represent phylogenetic trees.

Slots

`B` A data.frame containing the square matrix that represents the ancestral relations among the clones of the phylogenetic tree.

`clones` A vector representing the equivalence table of the clones in the phylogenetic tree.

`genes` A vector representing the equivalence table of the genes in the phylogenetic tree.

`parents` A vector representing the parents of the clones in the phylogenetic tree.

`tree` A `Node` class object representing the phylogenetic tree.

`labels` A vector representing the tags of the genes in the phylogenetic tree.

`phylotree_to_B` *Get B matrix from* `Phylotree` *object.*

Description

This function retrieves the B matrix of a `Phylotree` object.

Usage

```
phylotree_to_B(phylotree)
```

Arguments

`phylotree`	A `Phylotree` class object.

Value

A `data.frame` representing the B matrix of the phylogenetic tree.

Examples

```
# Get the B matrix of a tumor instance
# composed by 10 subpopulations of
# clones
B <- create_instance(
       n = 10,
       m = 4,
       k = 1,
       selection = "neutral")$B

# Create a new 'Phylotree' object
# on the basis of the B matrix
phylotree <- B_to_phylotree(B)

# Get the B matrix of the phyotree
b1 <- phylotree_to_B(phylotree)
```

`phylotree_to_tree` *Extract tree from* `Phylotree` *object.*

Description

This function extracts the tree structure from a given Phylotree object.

Usage

```
phylotree_to_tree(phylotree)
```

Arguments

`phylotree`	An object of the `Phylotree` class.

Value

The tree structure of the input `Phylotree` object.

`place_clones_space` *Create a model for the spatial distribution of the clones in a tumor*

Description

This function creates a Gaussian mixture model for the spatial distribution of the clones in a tumor in a 1D space. In the model, each component represents a clone, and the mean of the component represents the position of the clone in the space. The standard deviation of the components is fixed to 1, while the mean values are random variables.

Usage

```
place_clones_space(B)
```

Arguments

`B`	A matrix representing the mutation relationships between the nodes in the phylogenetic tree (B matrix).

Value

A list containing two elements: 'spatial_coords', a data frame where each column represents the density of a clone at different spatial coordinates, and 'x', a numeric vector representing the spatial coordinates.

Examples

```
# Create a model for the spatial distribution of the clones in a tumor represented by the
# B matrix B_mat

# Create a mutation matrix for a phylogenetic tree with 10 nodes and k = 2
B_mat <- create_B(10, 2)

clone_placement <- place_clones_space(B_mat)
```

`plot` *Plot a Phylotree object.*

Description

Plot a Phylotree object.

Usage

```
plot(object, labels = FALSE)

## S4 method for signature 'Phylotree'
plot(object, labels = FALSE)
```

Arguments

`object`	A `Phylotree` object.
`labels`	A label vector.

`plot_p` *Plot a phylogenetic tree with proportional node sizes and colors.*

Description

This function plots a phylogenetic tree with nodes sized and colored according to the proportions of each clone.

Usage

```
plot_p(phylotree, proportions)
```

Arguments

`phylotree`	An object of the `Phylotree` class representing the phylogenetic tree to be plotted.
`proportions`	A numeric vector representing the proportions of each clone in the phylogenetic tree. The length of this vector should be equal to the number of clones in the tree.

Value

A graph representing the phylogenetic tree, with node sizes and colors reflecting clone proportions.

`plot_phylotree` *Plot a phylogenetic tree*

Description

This function generates a plot of a phylogenetic tree. If the ‘labels‘ parameter is set to TRUE, the nodes of the tree will be labeled with the labels stored in the `Phylotree` object.

Usage

```
plot_phylotree(phylotree, labels = FALSE)
```

Arguments

`phylotree`	An object of the `Phylotree` class representing the phylogenetic tree to be plotted.
`labels`	A logical value indicating whether to label the nodes with the labels stored in the `Phylotree` object. Default is FALSE.

Value

A plot of the phylogenetic tree.

`plot_proportions` *Plot a phylogenetic tree with proportional node sizes and colors*

Description

This function plots a phylogenetic tree with nodes sized and colored according to the proportions of each clone. If a matrix of proportions is provided, multiple phylogenetic trees will be plotted, each corresponding to a row of proportions.

Usage

```
plot_proportions(phylotree, proportions, labels = FALSE)
```

Arguments

`phylotree`	A `Phylotree` class object representing the phylogenetic tree to be plotted.
`proportions`	A numeric vector or matrix representing the proportions of each clone in the phylogenetic tree. If a matrix is provided, each row should represent the proportions for a separate tree.
`labels`	A logical value indicating whether to label the nodes with gene tags (if `TRUE`) or gene indices (if `FALSE`). Default is `FALSE`.

Value

A graph representing the phylogenetic tree, with node sizes and colors reflecting clone proportions.

Examples

```
# Create an instance
# composed by 5 subpopulations of clones
# and 4 samples
instance <- create_instance(
       n = 5,
       m = 4,
       k = 1,
       selection = "neutral")

# Extract its associated B matrix
B <- instance$B

# Create a new 'Phylotree' object
# on the basis of the B matrix
phylotree <- B_to_phylotree(B = B)

# Generate the tags for the genes of
# the phyogenetic tree
tags <- LETTERS[1:nrow(B)]

# Plot the phylogenetic tree taking
# into account the proportions of the
# previously generated instance
plot_proportions(phylotree, instance$U, labels=TRUE)
```

Index