

Package ‘LACE’

October 14, 2021

Version 1.4.0

Date 2021-05-05

Title Longitudinal Analysis of Cancer Evolution (LACE)

Depends R (>= 4.0.0)

Imports data.tree, graphics, grDevices, igraph, parallel,
RColorBrewer, Rfast, stats, SummarizedExperiment, utils

Suggests BiocGenerics, BiocStyle, testthat, knitr

Name LACE: an R package for the inference of longitudinal cancer
evolution models

Description LACE is an algorithmic framework that processes single-cell somatic mutation profiles from cancer samples collected at different time points and in distinct experimental settings, to produce longitudinal models of cancer evolution. The approach solves a Boolean Matrix Factorization problem with phylogenetic constraints, by maximizing a weighed likelihood function computed on multiple time points.

Encoding UTF-8

LazyData TRUE

License file LICENSE

URL <https://github.com/BIMIB-DISCO/LACE>

BugReports <https://github.com/BIMIB-DISCO/LACE>

biocViews BiomedicalInformatics, SingleCell, SomaticMutation

RoxygenNote 7.1.1

VignetteBuilder knitr

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

git_branch RELEASE_3_13

git_last_commit 9293f28

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

Author Daniele Ramazzotti [aut] (<<https://orcid.org/0000-0002-6087-2666>>),
 Fabrizio Angaroni [aut],
 Davide Maspero [cre, aut],
 Alex Graudenzi [aut],
 Luca De Sano [ctb]

Maintainer Davide Maspero <d.maspero@campus.unimib.it>

R topics documented:

compute.mutation.distance	2
compute.variants.error.rates	3
inference	3
LACE	4
longitudinal.tree.plot	6
longitudinal_sc_variants	8

Index **9**

compute.mutation.distance
compute.mutation.distance

Description

Compute mutation distance among variants from LACE corrected genotype and use it to perform hierarchical clustering.

Usage

```
compute.mutation.distance(inference)
```

Arguments

inference Results of the inference by LACE.

Value

A matrix `mutation_distance` with the mutation distance among variants computed from LACE corrected genotype and related hierarchical clustering.

Examples

```
data(inference)
mutation_distance <- compute.mutation.distance(inference)
```

```
compute.variants.error.rates
      compute.variants.error.rates
```

Description

Compute error rates for the considered variants comparing observed data to LACE corrected genotype.

Usage

```
compute.variants.error.rates(D, inference)
```

Arguments

D	Mutation data from multiple experiments for a list of driver genes provided as a data matrix per time point.
inference	Results of the inference by LACE.

Value

A matrix `variants_error_rates` with the estimated error rates for the considered variants.

Examples

```
data(longitudinal_sc_variants)
data(inference)
variants_error_rates <- compute.variants.error.rates(longitudinal_sc_variants, inference)
```

inference	<i>results obtained with the function LACE on the provided input data from Rambow, Florian, et al. "Toward minimal residual disease-directed therapy in melanoma." Cell 174.4 (2018): 843-855.</i>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Description

results obtained with the function LACE on the provided input data from Rambow, Florian, et al. "Toward minimal residual disease-directed therapy in melanoma." Cell 174.4 (2018): 843-855.

Usage

```
data(inference)
```

Format

results obtained with the function LACE on the provided input data

Value

results obtained with the function LACE on the provided input data

LACE

LACE

Description

Perform inference of the maximum likelihood clonal tree from longitudinal data.

Usage

```
LACE(
  D,
  lik_w = NULL,
  alpha = NULL,
  beta = NULL,
  initialization = NULL,
  random_tree = FALSE,
  keep_equivalent = TRUE,
  check_indistinguishable = TRUE,
  num_rs = 50,
  num_iter = 10000,
  n_try_bs = 500,
  learning_rate = 1,
  marginalize = FALSE,
  error_move = FALSE,
  num_processes = Inf,
  seed = NULL,
  verbose = TRUE,
  log_file = ""
)
```

Arguments

- | | |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| D | Mutation data from multiple experiments for a list of driver genes. It can be either a list with a data matrix per time point or a SummarizedExperiment object. In this latter, the object must contain two fields: assays and colData. Assays stores one unique data matrix pooling all single cells observed at each time point and colData stores a vector of labels reporting the time point when each single cell was sequenced. Ordering of cells in assays field and colData field must be the same. |
| lik_w | Weight for each data point. If not provided, weights to correct for sample sizes are used. |
| alpha | False positive error rate provided as list of elements; if a vector of alpha (and beta) is provided, the inference is performed for multiple values and the solution at maximum-likelihood is returned. |

beta	False negative error rate provided as list of elements; if a vector of beta (and alpha) is provided, the inference is performed for multiple values and the solution at maximum-likelihood is returned.
initialization	Binary matrix representing a perfect phylogeny clonal tree; clones are rows and mutations are columns. This parameter overrides "random_tree".
random_tree	Boolean. Shall I start MCMC search from a random tree? If FALSE (default) and initialization is NULL, search is started from a TRaIT tree (BMC Bioinformatics . 2019 Apr 25;20(1):210. doi: 10.1186/s12859-019-2795-4).
keep_equivalent	Boolean. Shall I return results (B and C) at equivalent likelihood with the best returned solution?
check_indistinguishable	Boolean. Shall I remove any indistinguishable event from input data prior inference?
num_rs	Number of restarts during mcmc inference.
num_iter	Maximum number of mcmc steps to be performed during the inference.
n_try_bs	Number of steps without change in likelihood of best solution after which to stop the mcmc.
learning_rate	Parameter to tune the probability of accepting solutions at lower values during mcmc. Value of learning_rate = 1 (default), set a probability proportional to the difference in likelihood; values of learning_rate greater than 1 increase the chance of accepting solutions at lower likelihood during mcmc while values lower than 1 decrease such probability.
marginalize	Boolean. Shall I marginalize C when computing likelihood?
error_move	Boolean. Shall I include estimation of error rates in the MCMC moves?
num_processes	Number of processes to be used during parallel execution. To execute in single process mode, this parameter needs to be set to either NA or NULL.
seed	Seed for reproducibility.
verbose	Boolean. Shall I print to screen information messages during the execution?
log_file	log file where to print outputs when using parallel. If parallel execution is disabled, this parameter is ignored.

Value

A list of 9 elements: B, C, clones_prevalence, relative_likelihoods, joint_likelihood, clones_summary and error_rates. Here, B returns the maximum likelihood longitudinal clonal tree, C the attachment of cells to clones, corrected_genotypes the corrected genotypes and clones_prevalence clones' prevalence; relative_likelihoods and joint_likelihood are respectively the likelihood of the solutions at each individual time points and the joint likelihood; clones_summary provide a summary of association of mutations to clones. In equivalent_solutions, solutions (B and C) with likelihood equivalent to the best solution are returned. Finally error_rates provides the best values of alpha and beta among the considered ones.

Examples

```
data(longitudinal_sc_variants)
inference = LACE(D = longitudinal_sc_variants,
  lik_w = c(0.2308772,0.2554386,0.2701754,0.2435088),
  alpha = list(c(0.10,0.05,0.05,0.05)),
  beta = list(c(0.10,0.05,0.05,0.05)),
  keep_equivalent = TRUE,
  num_rs = 5,
  num_iter = 10,
  n_try_bs = 5,
  num_processes = NA,
  seed = 12345,
  verbose = FALSE)
```

`longitudinal.tree.plot`

longitudinal.tree.plot

Description

Plot a longitudinal tree inferred by LACE.

Usage

```
longitudinal.tree.plot(
  inference,
  rem_unseen_leafs = TRUE,
  show_plot = TRUE,
  filename = "lg_output.xml",
  labels_show = "mutations",
  clone_labels = NULL,
  show_prev = TRUE,
  label.cex = 1,
  size = 500,
  size2 = NULL,
  tk_plot = FALSE,
  tp_lines = TRUE,
  tp_mark = TRUE,
  tp_mark_alpha = 0.5,
  legend = TRUE,
  legend_position = "topright",
  label_offset = 4,
  legend_cex = 0.8
)
```

Arguments

<code>inference</code>	Results of the inference by LACE.
<code>rem_unseen_leafs</code>	If TRUE (default) remove all the leafs that have never been observed (prevalence = 0 in each time point)
<code>show_plot</code>	If TRUE (default) output the longitudinal tree to the current graphical device.
<code>filename</code>	Specify the name of the file where to save the longitudinal tree. Dot or graphml formats are supported and are chosen based on the extension of the filename (.dot or .xml).
<code>labels_show</code>	Specify which type of label should be placed on the tree; options are, "mutations": parental edges are labeled with the acquired mutation between the two nodes (genotypes); "clones": nodes (genotypes) are labeled with their last acquired mutation; "both": either nodes and edges are labeled as specified above; "none": no labels will show on the longitudinal tree.
<code>clone_labels</code>	Character vector that specifies the name of the nodes (genotypes). If it is NULL (default), nodes will be labeled as specified by "label" parameter.
<code>show_prev</code>	If TRUE (default) add to clones label the corresponding prevalence.
<code>label.cex</code>	Specify the size of the labels.
<code>size</code>	Specify size of the nodes. The final area is proportional with the node prevalence.
<code>size2</code>	Specify the size of the second dimension of the nodes. If NULL (default), it is set equal to "size".
<code>tk_plot</code>	If TRUE, uses tkplot function from igraph library to plot an interactive tree. Default is FALSE.
<code>tp_lines</code>	If TRUE (default) the function draws lines between timepoints.
<code>tp_mark</code>	If TRUE (default) the function draws different colored area under the nodes in different time points.
<code>tp_mark_alpha</code>	Specify the alpha value of the area drawn when <code>tp_mark = TRUE</code> .
<code>legend</code>	If TRUE (default) a legend will be displayed on the plot.
<code>legend_position</code>	Specify the legend position.
<code>label_offset</code>	Move the mutation labels horizontally (default = 4)
<code>legend_cex</code>	Specify size of the legend text.

Value

An igraph object `g` with the longitudinal tree inferred by LACE.

Examples

```
data(inference)
clone_labels = c("ARPC2", "PRAME", "HNRNPC", "COL1A2", "RPL5", "CCT8")
longitudinal.tree.plot(inference = inference,
                      labels = "clones",
```

```
clone_labels = clone_labels,  
legend_position = "topleft")
```

longitudinal_sc_variants

mutation data from Rambow, Florian, et al. "Toward minimal residual disease-directed therapy in melanoma." Cell 174.4 (2018): 843-855.

Description

the dataset includes somatic single nucleotide variants at the single cell resolution. SNVs are called from SMARTseq2 fastq obtained from Gene Expression Omnibus database with the accession number: GSE116237. The dataset includes single cell data from a PDX melanoma model before and on treatment with BRAF and MEK inhibitors. The fastq files are processed to obtain the mutational profile following GATK best practice (<https://gatkforums.broadinstitute.org/gatk/discussion/3891/calling-variants-in-rnaseq>) using the GRCh38 human genome as reference. Mutation data are stored in an N x M binary matrix with N single cells and M somatic single nucleotide variants. Row names report the ID of the fastq file related to a specific single cell; columns names report the SNV that are formatted as GeneName_chromosome_position_referenceAllele_alternateAllele. Each matrix entry can be 1 (mutation detected), 0 (mutation absent) or NA (too low coverage to determine the presence or absence of that mutation). For further details, please refer to the Methods Section and the section 3.1 of supplementary materials of Ramazzotti, Daniele, et al. "Longitudinal cancer evolution from single cells." bioRxiv (2020).

Usage

```
data(longitudinal_sc_variants)
```

Format

list of mutation data for four time points

Value

list of mutational data for a total of 475 single cells

Source

Rambow, Florian, et al. "Toward minimal residual disease-directed therapy in melanoma." Cell 174.4 (2018): 843-855.

Index

`compute.mutation.distance`, [2](#)
`compute.variants.error.rates`, [3](#)

`inference`, [3](#)

LACE, [4](#)
`longitudinal.tree.plot`, [6](#)
`longitudinal_sc_variants`, [8](#)