

# Package ‘LACE’

November 26, 2020

**Version** 1.2.0

**Date** 2020-10-20

**Title** Longitudinal Analysis of Cancer Evolution (LACE)

**Depends** R (>= 4.0.0)

**Imports** 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\_12

**git\_last\_commit** d6cbf0e

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2020-11-25

**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](mailto:d.maspero@campus.unimib.it)>

**R topics documented:**

compute.mutation.distance . . . . .	2
compute.variants.error.rates . . . . .	2
inference . . . . .	3
LACE . . . . .	4
longitudinal.tree.plot . . . . .	6
longitudinal_sc_variants . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

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,
  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**

<code>D</code>	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: <code>assays</code> and <code>colData</code> . <code>assays</code> stores one unique data matrix pooling all single cells observed at each time point and <code>colData</code> stores a vector of labels reporting the time point when each single cell was sequenced. Ordering of cells in <code>assays</code> field and <code>colData</code> field must be the same.
<code>lik_w</code>	Weight for each data point. If not provided, weights to correct for sample sizes are used.
<code>alpha</code>	False positive error rate provided as list of elements; if a vector of <code>alpha</code> (and <code>beta</code> ) is provided, the inference is performed for multiple values and the solution at maximum-likelihood is returned.
<code>beta</code>	False negative error rate provided as list of elements; if a vector of <code>beta</code> (and <code>alpha</code> ) is provided, the inference is performed for multiple values and the solution at maximum-likelihood is returned.
<code>initialization</code>	Starting point of the mcmc; if not provided, a random starting point is used.
<code>keep_equivalent</code>	Boolean. Shall I return results (B and C) at equivalent likelihood with the best returned solution?

<code>check_indistinguishable</code>	Boolean. Shall I remove any indistinguishable event from input data prior inference?
<code>num_rs</code>	Number of restarts during mcmc inference.
<code>num_iter</code>	Maximum number of mcmc steps to be performed during the inference.
<code>n_try_bs</code>	Number of steps without change in likelihood of best solution after which to stop the mcmc.
<code>learning_rate</code>	Parameter to tune the probability of accepting solutions at lower values during mcmc. Value of <code>learning_rate</code> = 1 (default), set a probability proportional to the difference in likelihood; values of <code>learning_rate</code> greater than 1 increase the chance of accepting solutions at lower likelihood during mcmc while values lower than 1 decrease such probability.
<code>marginalize</code>	Boolean. Shall I marginalize C when computing likelihood?
<code>error_move</code>	Boolean. Shall I include estimation of error rates in the MCMC moves?
<code>num_processes</code>	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.
<code>seed</code>	Seed for reproducibility.
<code>verbose</code>	Boolean. Shall I print to screen information messages during the execution?
<code>log_file</code>	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 = FALSE,
  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,
  show_plot = TRUE,
  filename = "lg_output.xml",
  labels_show = "mutations",
  clone_labels = NULL,
  show_prev = TRUE,
  label.cex = 1,
  iter_max = 100,
  size = 500,
  size2 = NULL,
  tk_plot = FALSE,
  tp_lines = TRUE,
  tp_mark = TRUE,
  tp_mark_alpha = 0.5,
  legend = TRUE,
  legend_position = "topright",
  legend_cex = 0.8
)
```

### Arguments

<code>inference</code>	Results of the inference by LACE.
<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>iter_max</code>	Maximum number of iteration to be used to remove intersecting edges.
<code>size</code>	Specify size of the nodes. The final area is proportional with the node prevalence.

size2	Specify the size of the second dimension of the nodes. If NULL (default), it is set equal to "size".
tk_plot	If TRUE, uses tkplot function from igraph library to plot an interactive tree. Default is FALSE.
tp_lines	If TRUE (default) the function draws lines between timepoints.
tp_mark	If TRUE (default) the function draws different colored area under the nodes in different time points.
tp_mark_alpha	Specify the alpha value of the area drawn when tp_mark = TRUE.
legend	If TRUE (default) a legend will be displayed on the plot.
legend_position	Specify the legend position.
legend_cex	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  $N \times M$  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; column 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`, [2](#)

`inference`, [3](#)

LACE, [4](#)

`longitudinal.tree.plot`, [6](#)

`longitudinal_sc_variants`, [7](#)