Package ‘LACE’

March 15, 2024

Version 2.6.1
Date 2024-02-22
Title Longitudinal Analysis of Cancer Evolution (LACE)
Depends R (>= 4.2.0)
Imports curl, igraph, foreach, doParallel, sortable, dplyr, forcats, data.tree, graphics, grDevices, parallel, RColorBrewer, Rfast, stats, SummarizedExperiment, utils, purrr, stringi, stringr, Matrix, tidyr, jsonlite, readr, configr, DT, tools, fs, data.table, htmltools, htmlwidgets, bsplus, shinyvalidate, shiny, shinythemes, shinyFiles, shinyjs, shinyBS, shinydashboard, biomaRt, callr, logr, ggplot2, svglite
Suggests BiocGenerics, BiocStyle, testthat, knitr, rmarkdown
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
License file LICENSE
URL https://github.com/BIMIB-DISCO/LACE
BugReports https://github.com/BIMIB-DISCO/LACE
biocViews BiomedicalInformatics, SingleCell, SomaticMutation
RoxygenNote 7.3.1
VignetteBuilder knitr
git_url https://git.bioconductor.org/packages/LACE
git_branch RELEASE_3_18
git_last_commit 32cdacb
git_last_commit_date 2024-02-22
Description

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

Usage

```r
## S3 method for class 'mutation.distance'
compute(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

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

Description

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

Usage

```r
## S3 method for class 'variants.error.rates'
compute(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

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


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

Description

Perform the 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 = "",
show = TRUE
)

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 al-
       pha) is provided, the inference is performed for multiple values and the solution
       at maximum-likelihood is returned.

initialization  Binary matrix representing a perfect philogeny 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 Bioinfor-

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 infer-
                          ence?

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 incease
               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.
show  Boolean. Show the interactive interface to explore the output.

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,
    show = FALSE)

Description

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

lacedata(
  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 philogeny clonal tree; clones are rows and mutations are columns. This parameter overrides "random_tree".


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

   shiny interface

Examples

data(longitudinal_sc_variants)
lacedata(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)
LACEview displays a Shiny user interface to handle the VCF and BAM files processing that is needed to construct the input for the LACE inference algorithms. The function generates also the maximum likelihood longitudinal clonal tree, and shows the output for further explorations of the results.

Usage

LACEview()

Value

The GUI

Installation

The package is available on GitHub and Bioconductor. LACE 2.0 requires R > 4.1.0 and Bioconductor.

To install directly from github run:

```r
remotes::install_github("https://github.com/BIMIB-DISCo/LACE",
                        dependencies = TRUE)
```

Dependencies

LACE 2.0 uses Annovar and Samtools suite as back-ends for variant calling annotation and depth computation, respectively.

Annovar is a variant calling software written in Perl freely available upon registration to their website at [https://annovar.openbioinformatics.org/en/latest/](https://annovar.openbioinformatics.org/en/latest/).

Perl can be found and installed at [https://www.perl.org/](https://www.perl.org/).

Samtools suite is a set of tools to handle SAM/BAM/BED file format. It is freely available at [http://www.htslib.org/](http://www.htslib.org/). To install Samtools follow the instructions in their website.

Note

The function LACE is still available for retrocompatibility.
lace_interface  

**Description**

This function generates a longitudinal clonal tree and a graphic interface to explore the data using as input the clonal tree formatted in the same way as the one produced by LACE during the imputation steps.

**Usage**

```r
lace_interface(
  B_mat,
  clones_prevalence,
  C_mat,
  error_rates,
  width = NULL,
  height = NULL,
  elementId = NULL,
  info = ""
)
```

**Arguments**

- `B_mat` (Required). B is the clonal tree matrix where columns are the clonal mutations and and rows are the clones. The clonal tree matrix should contain a column and a row named "Root" representing the root of the tree and the wild type, respectively. B is a binary matrix where 1 are the mutations associated to the clones. The wild type column has all ones.
- `clones_prevalence` (Required). The clonal prevalence matrix.
- `C_mat` (Required). The corrected clonal attachment.
- `error_rates` (Required). The false positive alpha and false negative beta error rates used to infer the clonal tree.
- `width` (optional). Size of the window interface.
- `height` (optional). Size of the window interface.
- `elementId` (optional). Element id.
- `info` (Optional). HTML formatted text with information regarding the experiments.

**Value**

An implementation of the htmlwidgets.
longitudinal.tree.plot

Description

Plot a longitudinal tree inferred by LACE.

Usage

```r
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

- `inference`: Results of the inference by LACE.
- `rem_unseen_leafs`: If TRUE (default) remove all the leafs that have never been observed (prevalence = 0 in each time point).
- `show_plot`: If TRUE (default) output the longitudinal tree to the current graphical device.
- `filename`: 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).
- `labels_show`: 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.
clone_labels  Character vector that specifies the name of the nodes (genotypes). If it is NULL (default), nodes will be labeled as specified by "label" parameter.

show_prev    If TRUE (default) add to clones label the corresponding prevalence.

label.cex    Specify the size of the labels.

size        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 drewed when tp_mark = TRUE.

legend      If TRUE (default) a legend will be displayed on the plot.

legend_position Specify the legend position.

label_offset Move the mutation labels horizontally (default = 4)

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

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
Index

compute.mutation.distance, 2
compute.variants.error.rates, 3

inference, 4

LACE, 4
lace_interface, 10
lacedata, 6
LACEview, 9
longitudinal.tree.plot, 11
longitudinal_sc_variants, 12