LACE 2.0: an interactive R tool for the inference and visualization of longitudinal cancer evolution models

Gianluca Ascolani [1], ***Fabrizio Angaroni*** [1], ***Davide Maspero*** [1], ***Narra Lakshmi Sai Bhavesh*** [2], ***Rocco Piazza*** [3,4], ***Chiara Damiani*** [5,6], ***Daniele Ramazzotti*** [3], ***Marco Antoniotti*** [1,4], ***and Graudenzi Alex*** [1,4,7]

[1]Dept. of Informatics, Systems and Communication, Universitá degli Studi di Milano-Bicocca, Milan, Italy.
[2]Department of Biological Sciences, Birla Institute of Technology and Science (BITS), Pilani, Rajasthan, India.
[3]Dept.of Medicine and Surgery, Universitá degli Studi di Milano-Bicocca, Milan, Italy.
[4]Bicocca Bioinformatics Biostatistics and Bioimaging Centre – B4, Milan, Italy.
[5]Dept. of Biotechnology and Biosciences, Universitá degli Studi di Milano-Bicocca, Milan, Italy.
[6]ISBE/SYSBIO Centre of Systems Biology, Milan, Italy.
[7]Inst. of Molecular Bioimaging and Physiology, National Research Council (IBFM-CNR), Segrate, Italy.

November 8, 2022

Overview.
LACE 2.0 includes: (a) an easy-to-use module for data processing and organization, which allows one to test and adjust several quality metrics and relevance filters, and to annotate gene variants; (b) a module for the interactive visualization of the inferred cancer evolution models, which returns both the *longitudinal clonal tree* and the *fishplot*, and allows one to examine the different parts of the model (i.e., clones and temporal relations), also by querying external databases such as Ensembl.org. Importantly, LACE 2.0 requires a minimum set of parameters, which are sufficient to mitigate the aforementioned noise sources and infer reliable models of cancer evolution.

In this vignette, we give an overview of the Shiny App by presenting its main functions.

Contents

LACE 2.0 is a new release of the LACE R package. LACE 2.0 is capable of performing clonal evolution analyses for single-cell sequencing data including longitudinal experiments. LACE 2.0 allows to annotate variants and retrieve the relevant mutations interactively based on user-defined filtering criteria; it infers the maximum likelihood clonal tree, cell matrix attachments and false positive/negative rates using boolean matrix factorization. Furthermore, LACE 2.0 allows to investigate cancer clonal evolution under different experimental conditions and the occurrence of single mutations which can be queried via *ensembl* database.

1 Installation of LACE 2.0 R package

The package is available on GitHub and Bioconductor. LACE 2.0 requires R $>$ 4.1.0 and Bioconductor. To install Bioconductor run:

```
if (!require("BiocManager", quietly = TRUE))
```

install.packages("BiocManager")

To install LACE 2.0 run:

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

LACE 2.0 uses *Annovar* and *Samtools suite* as back-ends for variant calling annotation and depth computation, respectively. Please refer to the next section to install them.

2 Installation of other required softwares

Annovar is a widely used variant calling software freely available upon registration to their website at https://annovar.openbioinformatics.org/en/latest/. The package contains *Perl* scripts and variant calling annotation reference databases for the human species. For other databases, please refer to their website. If the scripts are installed in binary search path, then LACE 2.0 will detect them automatically.

Perl (https://www.perl.org/) is required to run *Annovar*.

Samtools suite is a standard set of tools and libraries to handle SAM/BAM/BED file format and perform a variety of common operations on sequencing data. It is freely available at http://www.htslib.org/ and https://github.com/samtools/htslib. To install *Samtools* follow the instructions in their website.

3 Running LACE 2.0

To start LACE 2.0 user interface run:

```
library(LACE)
```

LACEview()

4 Using LACE 2.0

LACE 2.0 has been thought to be used on single cell sequencing data for which it is available variant calling data in standard VCF format and binary aligned data in standard BAM format.

The user is provided with an interface to initiate a project and to set filter thresholds after which annotation of variants, filtering of data and depth at variant sites are retrieved. Both annotation and depth derivation are computationally expensive steps. LACE 2.0 reduces possible re-computation by detecting parameter variations of the user interface and by comparison of the timestamps of interface state, inputs and outputs. Intermediary and final data are stored in the designated folders.

The operation is followed by the possibility for the user to select variants which are drivers of the understudied biological problem.

At this point, the inferential step is executed so that the most likelihood longitudinal clonal tree and clonal prevalences are retrieved together with the best set of false positive/negative rates among those provided by the user.

The results are displayed via an interactive interface.

5 Interface

The interface is divided in two parts interleaved by variant selection and inferential computation parts: the processing interface and the results interface.

5.1 Project creation

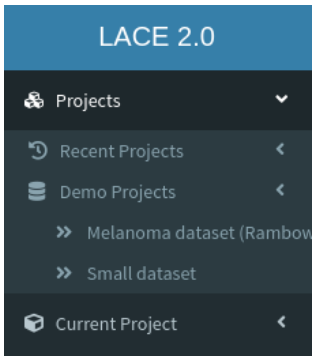
To begin the clonal analyses the user needs to create a project by choosing a folder and a meaningful name for the project.

If the project path does not contain former LACE projects, it asks the user to create a new one ("Create project") and all intermediate and final results will be stored in structured sub-folders of the chosen project path.

LACE 2.0 automatically recognizes if the selected path is a LACE project and proposes the user to load it (Load project)

5.2 Sidebar and Demos

The sidebar is divided in Projects and Current Project. Projects contain a set of recently opened projects which can be reloaded by selecting the respective project name.



Under Projects, there is the sub-menu Demos. LACE 2.0 provides the user with two demos:

- Melanoma dataset [Rambow, Florian, et al. "Toward minimal residual disease-directed therapy in melanoma." Cell 174.4 (2018): 843-855.] containing single cell variant annotations and depths for a total of 674 cells sampled at 4 different time points:
    1. sample before treatment
        - BRAF inhibitor treatment
    2. sample after 4 days
    3. sample after 28 days
    4. sample after 57 days
- Small dataset containing only 3 cells per 4 sampling time points.

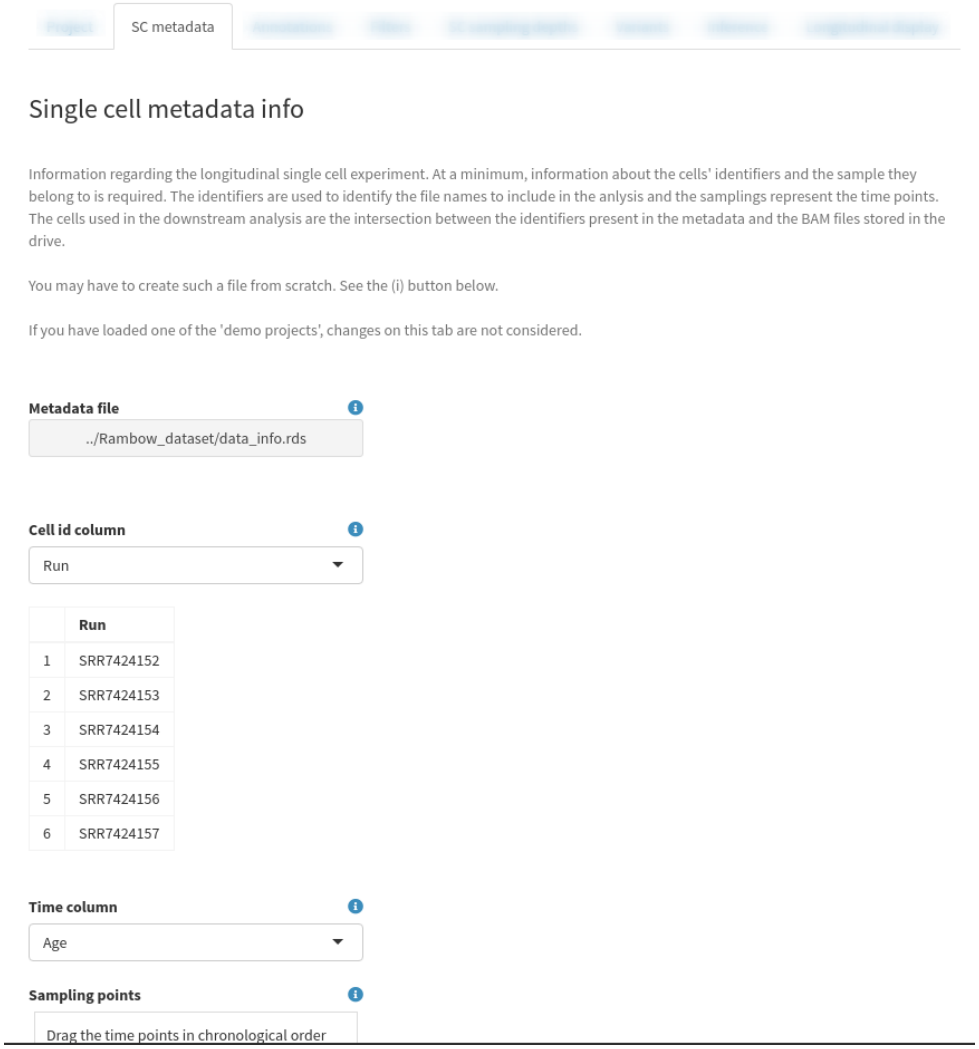
Current Project allows the user to save and load the parameters of the whole project or of the active tab.

5.3 Processing interface

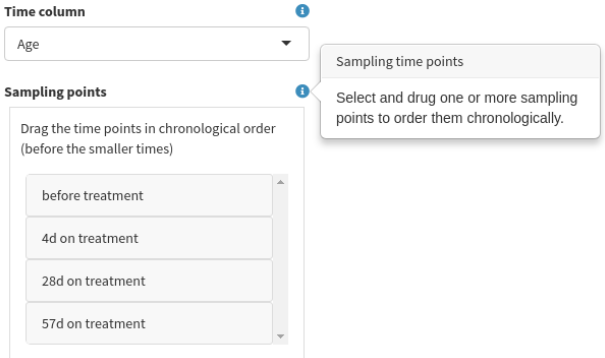
5.3.1 Single Cell Metadata

In order to perform clonal analyses it is required to provide some information about the experiment such as the cell IDs which are used to retrieve the VCF/BAM files and the sample name each cell belongs to. These metadata are generally included with the experimental

sequencing data as part of the library preparation. LACE 2.0 accepts tsv/csv tabular formats with headers and rds standard data R format. Metadata file can include more information relative to the experimental setup and the platform used.



The user should identify and select the ID and sampling columns. The user can also reorder the sampling names



so to recreate the chronological order of longitudinal events of the experiment, or to explore different orders.

5.3.2 Annotations

Annotations of variant calling data is performed using *Annovar* as back-end. Each mutation of each cell is annotated based on the annotation database used. *Annovar* provides a database for the human species useful to tag cancer mutations and whenever possible their functional effect.

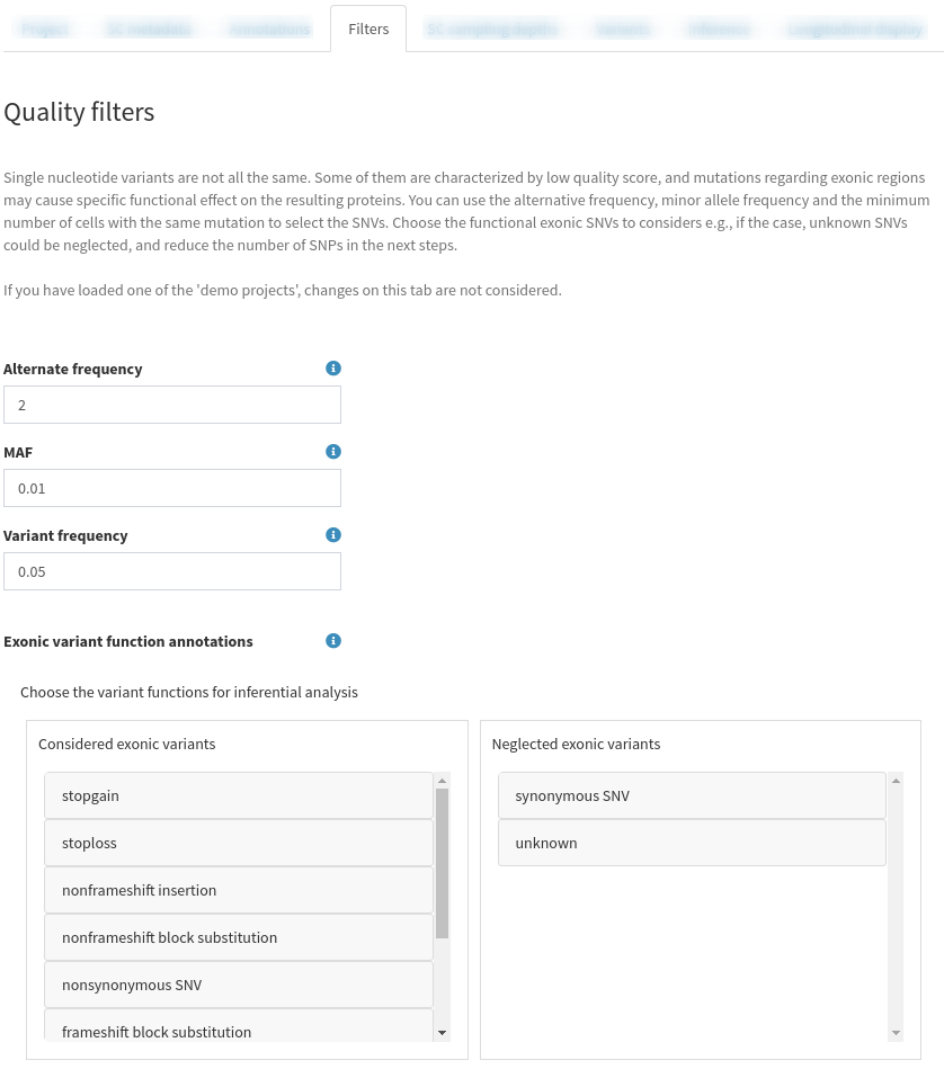
If *Annovar*'s *Perl* scripts are available in the standard binary paths, they are detected and the *Annovar* folder field is autocompleted; otherwise, the user should provide the folder containing the *Perl* scripts. The user should also provide the database to use for the annotation and the folder containing the variant calling files (VCF).

5.3.3 Variant filtering

All types of single cell sequencing data is characterized by various sources of noise which depends on the technology used. Detected mutation might be characterized by low quality score or low statistical power. Some mutations can be neglected, while others might cause relevant effects, especially on exonic regions where variations can result in changes of the translational process and modify their functional form.

In order to avoid small and possibly spurious fluctuations on sequencing data, variants can be filtered if they have low supporting evidences. The user can set the minimum values for:

1. the number of reads supporting the alternative alleles in a cell,
2. the frequency of the minor allele for each referenced SNP included in a default global population,
3. the cell frequency per sample showing the mutation at the same site.



Furthermore, the user can choose which functional exonic variation should be considered or neglected. For example, there are cases in which unknown and synonymous mutations in exonic regions are disregarded because their effect cannot be explicitly related to the experimental condition. All possible variant functions and their explanations are resumed in the following table:

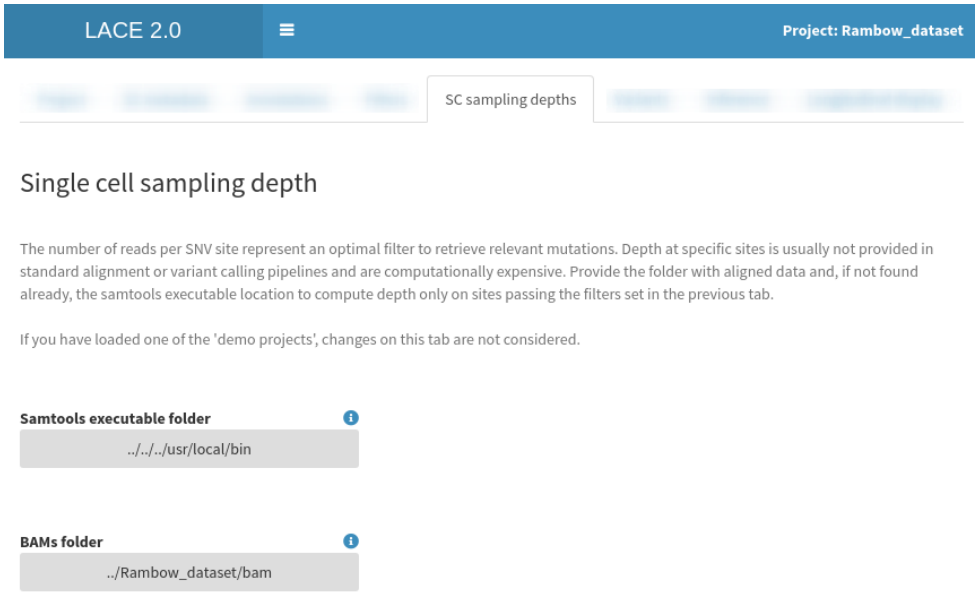
Annotation (+)	Explanation
frameshift insertion	an insertion of one or more nucleotides that cause frameshift changes in protein coding sequence
frameshift deletion	a deletion of one or more nucleotides that cause frameshift changes in protein coding sequence
frameshift block substitution	a block substitution of one or more nucleotides that cause frameshift changes in protein coding sequence

Annotation (+)	Explanation
stopgain	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block substitution that lead to the immediate creation of stop codon at the variant site. For frameshift mutations, the creation of stop codon downstream of the variant will not be counted as "stopgain"!
stoploss	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block substitution that lead to the immediate elimination of stop codon at the variant site
nonframeshift insertion	an insertion of 3 or multiples of 3 nucleotides that do not cause frameshift changes in protein coding sequence
nonframeshift deletion	a deletion of 3 or mutliples of 3 nucleotides that do not cause frameshift changes in protein coding sequence
nonframeshift block substitution	a block substitution of one or more nucleotides that do not cause frameshift changes in protein coding sequence
nonsynonymous SNV	a single nucleotide change that cause an amino acid change
synonymous SNV	a single nucleotide change that does not cause an amino acid change
unknown	unknown function (due to various errors in the gene structure definition in the database file)

(+) see *Annovar*

5.3.4 Single cell sampling depth

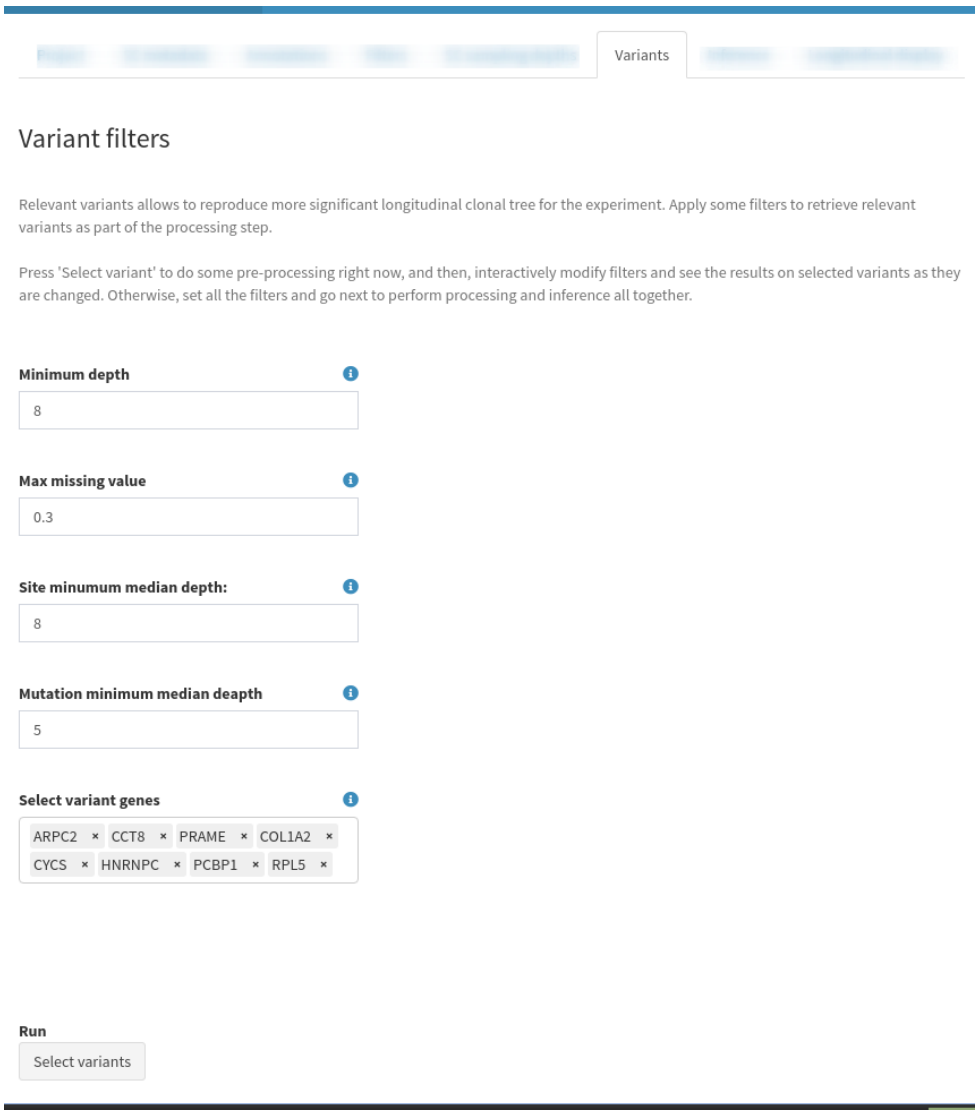
The number of reads per SNV site represents an optimal filter to retrieve relevant mutations. Depth at specific sites is usually not provided in standard alignment or variant calling pipelines and it is computationally expensive to retrieve. The user should provide the folder with aligned data and, if not found already, the *samtools* executable folder location to compute depth only on sites passing the variant filters set in the previous tab.



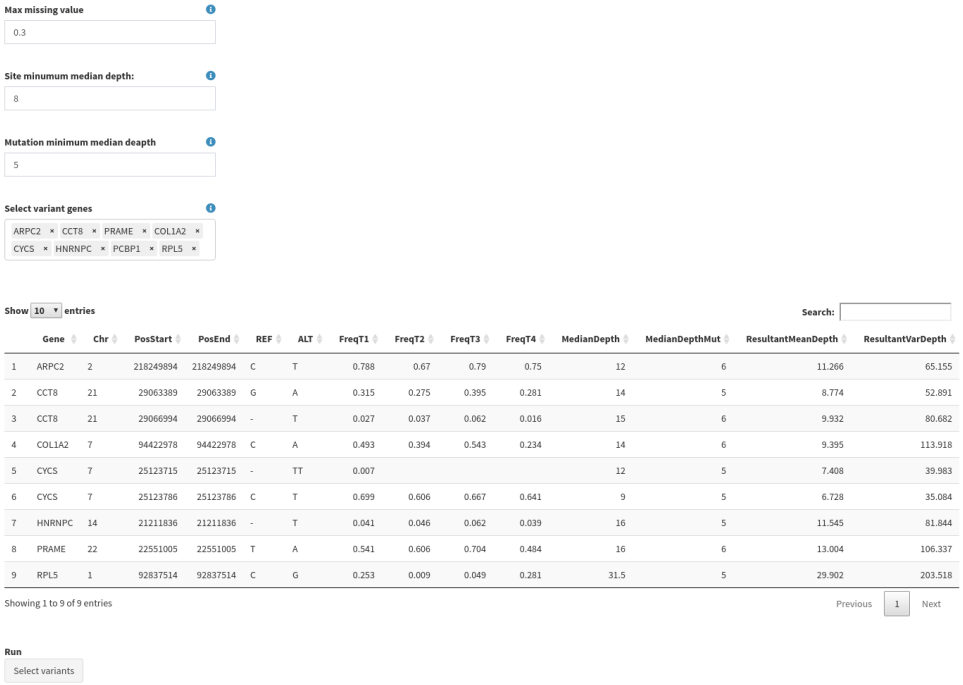
5.4 Selection of relevant variants

Not all mutations are distinctive of the disease or experiment understudied. Identifying relevant and driver variants allows to reproduce a more significant longitudinal clonal tree. The user can select a set of filters based on gold standards and other analyses such as:

1. the minimum number of reads at given mutation site
2. the maximum number of missing data per gene
3. the minimum median depth per locus
4. the minimum median depth supporting the mutation
5. subset of known genes



Finding the parameters to select variants is not an easy task, and the user might not know in advance how to choose the best set of filters. Hence, the user can press "Select variants" (Select variants) to perform the aforementioned computations on VCF and BAM files to derive all the necessary aggregated information on the sampled cells. Afterward, the user is presented with a interactive live preview of variants passing the filters (including relevant parameters which can help in the selection) while values are changed.



Max missing value

0.3

Site minumum median depth:

8

Mutation minimum median deapth

5

Select variant genes

ARPC2 × CCT8 × PRAME × COL1A2 × CYCS × HNRNPC × PCBP1 × RPL5 ×

Show 10 entries Search:

	Gene	Chr	PosStart	PosEnd	REF	ALT	FreqT1	FreqT2	FreqT3	FreqT4	MedianDepth	MedianDepthMut	ResultantMeanDepth	ResultantVarDepth
1	ARPC2	2	218249894	218249894	C	T	0.788	0.67	0.79	0.75	12	6	11.266	65.155
2	CCT8	21	29063389	29063389	G	A	0.315	0.275	0.395	0.281	14	5	8.774	52.891
3	CCT8	21	29066994	29066994	-	T	0.027	0.037	0.062	0.016	15	6	9.932	80.682
4	COL1A2	7	94422978	94422978	C	A	0.493	0.394	0.543	0.234	14	6	9.395	113.918
5	CYCS	7	25123715	25123715	-	TT	0.007				12	5	7.408	39.983
6	CYCS	7	25123786	25123786	C	T	0.699	0.606	0.667	0.641	9	5	6.728	35.084
7	HNRNPC	14	21211836	21211836	-	T	0.041	0.046	0.062	0.039	16	5	11.545	81.844
8	PRAME	22	22551005	22551005	T	A	0.541	0.606	0.704	0.484	16	6	13.004	106.337
9	RPL5	1	92837514	92837514	C	G	0.253	0.009	0.049	0.281	31.5	5	29.902	203.518

Showing 1 to 9 of 9 entries Previous 1 Next

Run

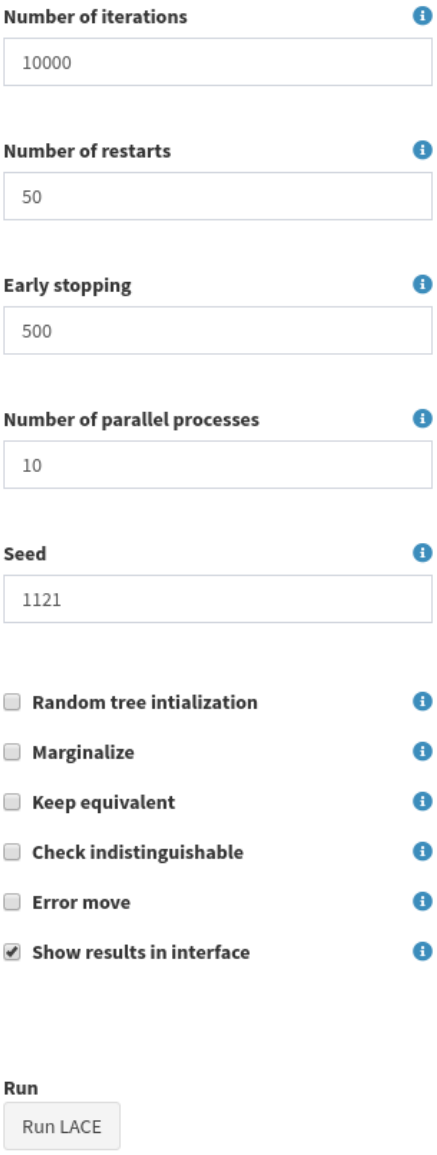
Select variants

5.5 Inference

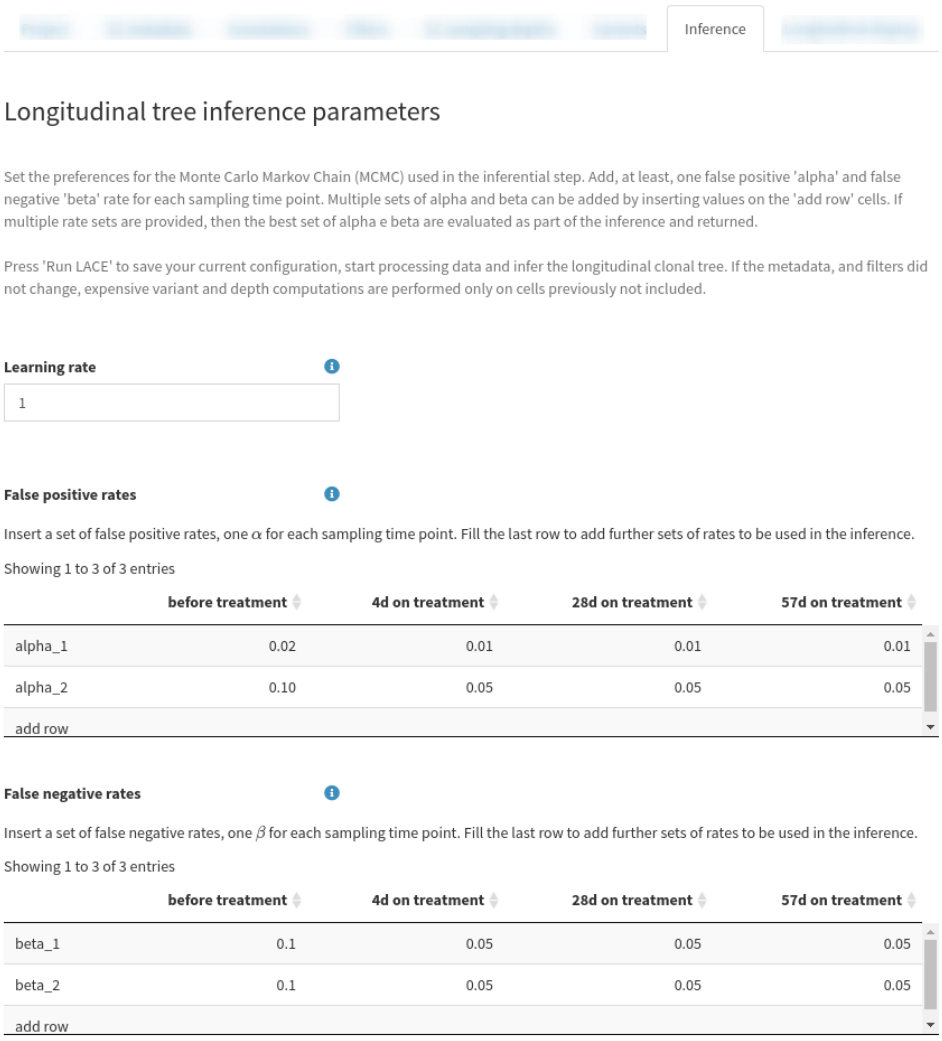
The inferential tab allows the user to set all the parameters to solve the boolean matrix factorization problem and estimate the model parameters of the Bayesian model using a MCMC to maximize the likelihood.

Inferential step uses the following set of parameters:

1. Learning rate
2. False positive rates for each sample
3. False negative rates for each sample
4. Number of iterations in each MCMC search
5. Number of restart for the MCMC 5 Early stopping number of iterations with no growing likelihood
6. Number of parallel processes
7. Random seed to recreate simulations
8. Initialize the clonal tree randomly
9. Marginalize the cell attachment matrix
10. Keep equivalent solutions and return all of them
11. Check indistinguishable event and remove them
12. Estimate error rates of MCMC moves
13. Show results



The user must insert at least one false positive rate and one false negative rate value for each sample. By double clicking on any cell belonging to the row "add row", the user can insert a new set of false or positive rates for each sample. During the inferential step, the maximization of the likelihood for each set of rates is performed. The best results are returned.



Press "Run LACE" to perform all computation and estimation steps and visualize the results.

5.6 Longitudinal display and outputs interface

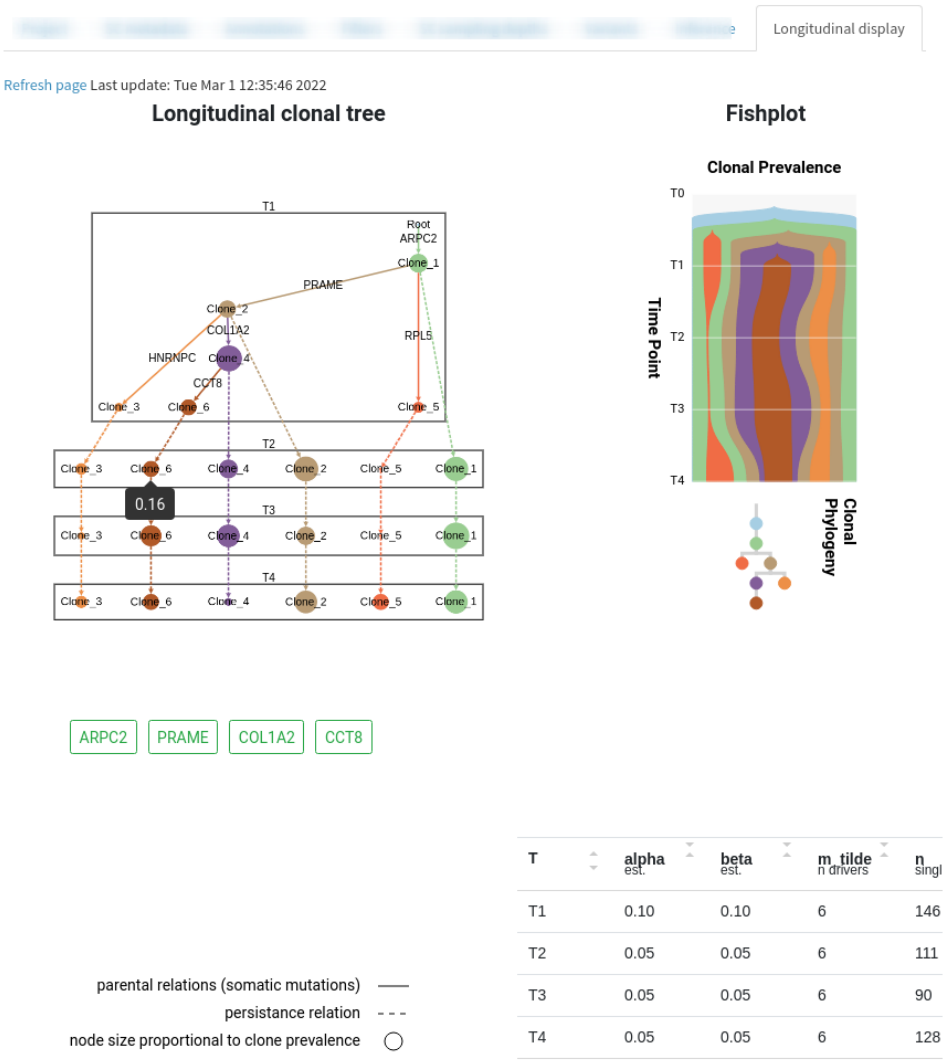
The longitudinal display tab shows the longitudinal clonal tree, the fishplot and the clonal tree, together with the best false positive and negative rate parameters.

The longitudinal clonal tree is placed on the top left side. Continuous edges are used to represent parental relations between clones, while dashed edges show persistence relations. The size of the nodes is proportional to the clonal prevalence. Moving the mouse over a node, it is possible to see the clonal prevalence value. While moving the mouse over the mutation labels, it shows the complete set of mutations for that clone under the longitudinal clonal tree.

The fishplot is on the top right side of the tab. Passing the mouse over the ribbons, the clonal prevalences of the clones at different time points are displayed. Similarly, with the mouse over the time tags, the clonal prevalences of all clones at the specified time point are visualized.

By clicking on a mutation label of the longitudinal clonal tree as well as on a fishplot ribbon or a clonal tree node, the details of the mutation characterizing the clone can be queried and retrieved from the *Ensembl* database.

The bottom part of the tab is used to display the tree legend and the best estimated set false positive/negative rates.



T	alpha est.	beta est.	m_tilde n drivers	n singl
T1	0.10	0.10	6	146
T2	0.05	0.05	6	111
T3	0.05	0.05	6	90
T4	0.05	0.05	6	128

6 sessionInfo()

- R version 4.2.1 (2022-06-23), `aarch64-apple-darwin20`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Running under: `macOS Ventura 13.0`
- Matrix products: default

- BLAS:
  `/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib`
- LAPACK:
  `/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: LACE 2.2.0, SummarizedExperiment 1.28.0, knitr 1.39
- Loaded via a namespace (and not attached): AnnotationDbi 1.60.0, Biobase 2.58.0, BiocFileCache 2.6.0, BiocGenerics 0.44.0, BiocManager 1.30.18, BiocStyle 2.26.0, Biostrings 2.66.0, DBI 1.1.3, DT 0.23, DelayedArray 0.24.0, GenomeInfoDb 1.34.2, GenomeInfoDbData 1.2.8, GenomicRanges 1.50.1, IRanges 2.32.0, KEGGREST 1.38.0, Matrix 1.4-1, MatrixGenerics 1.10.0, R6 2.5.1, RColorBrewer 1.1-3, RCurl 1.98-1.7, RSQLite 2.2.14, Rcpp 1.0.9, RcppTOML 0.1.7, RcppZiggurat 0.1.6, Rfast 2.0.6, S4Vectors 0.36.0, XML 3.99-0.10, XVector 0.38.0, assertthat 0.2.1, biomaRt 2.54.0, bit 4.0.4, bit64 4.0.5, bitops 1.0-7, blob 1.2.3, bsplus 0.1.3, cachem 1.0.6, callr 3.7.1, cli 3.3.0, codetools 0.2-18, compiler 4.2.1, configr 0.3.5, crayon 1.5.1, curl 4.3.2, data.table 1.14.2, data.tree 1.0.0, dbplyr 2.2.1, digest 0.6.29, doParallel 1.0.17, dplyr 1.0.9, ellipsis 0.3.2, evaluate 0.15, fansi 1.0.3, fastmap 1.1.0, filelock 1.0.2, foreach 1.5.2, fs 1.5.2, generics 0.1.3, glue 1.6.2, grid 4.2.1, highr 0.9, hms 1.1.1, htmltools 0.5.2, htmlwidgets 1.5.4, httpuv 1.6.5, httr 1.4.3, igraph 1.3.5, ini 0.3.1, iterators 1.0.14, jsonlite 1.8.0, later 1.3.0, lattice 0.20-45, learnr 0.10.1, lifecycle 1.0.1, logr 1.3.3, lubridate 1.8.0, magrittr 2.0.3, markdown 1.1, matrixStats 0.62.0, memoise 2.0.1, mime 0.12, parallel 4.2.1, pillar 1.7.0, pkgconfig 2.0.3, png 0.1-7, prettyunits 1.1.1, processx 3.7.0, progress 1.2.2, promises 1.2.0.1, ps 1.7.1, purrr 0.3.4, rappdirs 0.3.3, readr 2.1.2, rlang 1.0.4, rmarkdown 2.14, rprojroot 2.0.3, shiny 1.7.1, shinyBS 0.61.1, shinyFiles 0.9.2, shinydashboard 0.7.2, shinyjs 2.1.0, shinythemes 1.2.0, shinyvalidate 0.1.2, sortable 0.4.5, stats4 4.2.1, stringi 1.7.8, stringr 1.4.0, this.path 1.0.1, tibble 3.1.7, tidyr 1.2.0, tidyselect 1.1.2, tools 4.2.1, tzdb 0.3.0, utf8 1.2.2, vctrs 0.4.1, withr 2.5.0, xfun 0.31, xml2 1.3.3, xtable 1.8-4, yaml 2.3.5, zlibbioc 1.44.0