Package 'VERSO'

November 12, 2026

Version 1.22.0

Date 2026-03-09

Title Viral Evolution ReconStructiOn (VERSO)

Depends R (>= 4.1.0)

Imports utils, data.tree, ape, parallel, Rfast, stats

Suggests BiocGenerics, BiocStyle, testthat, knitr

Name VERSO: an R package for the inference of viral evolution models

Description Mutations that rapidly accumulate in viral genomes during a pandemic can be used to track the evolution of the virus and, accordingly, unravel the viral infection network. To this extent, sequencing samples of the virus can be employed to estimate models from genomic epidemiology and may serve, for instance, to estimate the proportion of undetected infected people by uncovering cryptic transmissions, as well as to predict likely trends in the number of infected, hospitalized, dead and recovered people. VERSO is an algorithmic framework that processes variants profiles from viral samples to produce phylogenetic models of viral evolution. The approach solves a Boolean Matrix Factorization problem with phylogenetic constraints, by maximizing a log-likelihood function. VERSO includes two separate and subsequent steps; in this package we provide an R implementation of VERSO STEP 1.

Encoding UTF-8

License file LICENSE

URL https://github.com/BIMIB-DISCo/VERSO

BugReports https://github.com/BIMIB-DISCo/VERSO

biocViews BiomedicalInformatics, Sequencing, SomaticMutation

RoxygenNote 7.3.3

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/VERSO

git_branch RELEASE_3_23

git_last_commit 68c974c

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Daniele Ramazzotti [aut] (ORCID:
<https://orcid.org/0000-0002-6087-2666>),
Fabrizio Angaroni [aut],
Davide Maspero [cre, aut],
Alex Graudenzi [aut],
Luca De Sano [aut] (ORCID: <https://orcid.org/0000-0002-9618-3774>)

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

Contents

inference 2
variants 3
VERSO 3
write.newick.tree 5

Index **6**

`inference` *Results obtained running VERSO on the provided input dataset.*

Description

Results obtained running VERSO on the provided input dataset.

Usage

```
data(inference)
```

Format

results obtained running VERSO on the provided input dataset

Value

results obtained running VERSO on the provided input dataset

variants *Mutation data obtained by variant calling from raw data of a selected set of SARS-CoV-2 samples available from NCBI BioProject PRJNA610428.*

Description

The dataset includes variants for a selected set of 15 SARS-CoV-2 samples obtained by variant calling from raw data available from NCBI BioProject PRJNA610428.

Usage

```
data(variants)
```

Format

SARS-CoV-2 variants

Value

SARS-CoV-2 variants

Source

NCBI BioProject PRJNA610428

VERSO *VERSO*

Description

Perform the inference of the maximum log-likelihood VERSO phylogenetic tree.

Usage

```
VERSO(
  D,
  alpha = NULL,
  beta = NULL,
  initialization = NULL,
  random_tree = FALSE,
  keep_equivalent = TRUE,
  check_indistinguishable = TRUE,
  marginalize = FALSE,
  num_rs = 10,
  num_iter = 10000,
```

```
  n_try_bs = 1000,
  num_processes = Inf,
  verbose = TRUE,
  log_file = ""
)
```

Arguments

D	Input data for the inference reporting presence (as 1), absense (as 0) or missing information (as NA) for a set of variants.
alpha	False positive error rate provided as a verctor; 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 a verctor; 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 tree; genotypes 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 variant from input data prior inference?
marginalize	Boolean. Shall I marginalize C when computing likelihood?
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 changes in likelihood of best solution after which to stop the MCMC.
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 1, NA or NULL.
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, phylogenetic_tree, corrected_genotypes, genotypes_prevalence, genotypes_summary, log_likelihood and error_rates. Here, B returns the maximum likelihood variants tree (inner nodes of the phylogenetic tree), C the attachment of patients to genotypes and phylogenetic_tree VERSO phylogenetic tree, including both variants tree and patients attachments to variants; corrected_genotypes is the corrected genotypes, which corrects D given VERSO phylogenetic tree, genotypes_prevalence the number of patients and observed prevalence of each genotype

and genotypes_summary provide a summary of association of mutations to genotypes. In equivalent_solutions, solutions (B and C) with likelihood equivalent to the best solution are returned. Finally log_likelihood and error_rates return the likelihood of the inferred phylogenetic moldel and best values of alpha and beta as estimated by VERSO.

Examples

```
data(variants)
set.seed(12345)
inference = VERSO(D = variants,
                  alpha = c(0.01,0.05),
                  beta = c(0.01,0.05),
                  check_indistinguishable = TRUE,
                  num_rs = 5,
                  num_iter = 100,
                  n_try_bs = 50,
                  num_processes = 1,
                  verbose = FALSE)
```

`write.newick.tree` *write.newick.tree*

Description

Write a phylogenetic tree as inferred by VERSO to a newick format file.

Usage

```
write.newick.tree(phylogenetic_tree, phylogeny_file = "phylogenetic_tree.new")
```

Arguments

`phylogenetic_tree`	Inference results by VERSO.
`phylogeny_file`	File where to save the phylogenetic tree in newick format.

Value

A phylogenetic tree as inferred by VERSO in newick format.

Examples

```
data(inference)
write.newick.tree(phylogenetic_tree = inference,
                  phylogeny_file = "inference_tree.new")
```

Index

inference, 2

variants, 3
VERSO, 3

write.newick.tree, 5