Analysis of SARS-CoV-2 viral phylogenies with VERSO

Daniele Ramazzotti [1], ***Fabrizio Angaroni*** [2], ***Davide Maspero*** [2,3], ***Carlo Gambacorti-Passerini*** [1], ***Marco Antoniotti*** [2], ***Alex Graudenzi*** [4], ***and Rocco Piazza*** [1]

[1]Dept. of Medicine and Surgery, Univ. of Milan-Bicocca, Monza, Italy.
[2]Dept. of Informatics, Systems and Communication, Univ. of Milan-Bicocca, Milan, Italy.
[3]Fondazione IRCCS Istituto Nazionale dei Tumori, Milan, Italy.
[4]Inst. of Molecular Bioimaging and Physiology, Consiglio Nazionale delle Ricerche (IBFM-CNR), Segrate, Milan, Italy.

May 19, 2021

Overview. VERSO (Viral Evolution ReconStructiOn) is an algorithmic framework that processes variants profiles from viral samples, to produce phylogenetic models of viral evolution from clonal variants and to subsequently quantify the intra-host genomic diversity of samples. VERSO includes two separate and subsequent steps; in this repository we provide an R implementation of VERSO STEP 1.

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

Contents

1 Using the VERSO R package

We now present an example of phylogenetic analysis by VERSO using mutation data from a set of SARS-CoV-2 samples; 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.

We first load the data. Notice that the input data to VERSO is an array reporting variants either observed (as 1 in the matrix), not observed (as 0) or missing (as NA, i.e., due to low coverage).

```
library("VERSO")
data(variants)
head(variants)
```

```
##             8782_T_C 17747_C_T 17858_A_G 18060_C_T 28144_C_T 29095_T_C
## SRR11241254        1         0         0         0         1         1
## SRR11241255        0         1         1         1         0         1
## SRR11247076        0         1         1         1         0         1
## SRR11247077        0         1         1         1         0         1
## SRR11247078        0         1         1         1         0         1
## SRR11278091        0         1         1         1         0         1
```

We setup the main parameter in oder to perform the inference. The first main parameter to be defined as input is represented by the false positive and false negative error rates, i.e., alpha and beta. When multiple set of rates are provided, VERSO performs a grid search in order to estimate the best set of error rates.

```
alpha = c(0.01,0.05)
beta = c(0.01,0.05)
head(alpha)
```

```
## [1] 0.01 0.05
```

```
head(beta)
```

```
## [1] 0.01 0.05
```

We can now perform the inference as follow. Make sure to set the random seed to ensure reproducibility.

```
set.seed(12345)
inference = VERSO(D = variants,
                  alpha = alpha,
                  beta = beta,
                  check_indistinguishable = TRUE,
                  num_rs = 5,
```

```
                    num_iter = 100,
                    n_try_bs = 50,
                    num_processes = 1,
                    verbose = TRUE)
```

```
## Performing inference for a total of 2 different values of alpha and beta.
```

```
## Performing inference for alpha = 0.01 and beta = 0.01
```

```
##
Current best lik. = -0.60 | Restart # 1/5 | Iter # 51 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -0.60 | Restart # 2/5 | Iter # 84 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -0.60 | Restart # 3/5 | Iter # 71 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -0.60 | Restart # 4/5 | Iter # 61 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -0.60 | Restart # 5/5 | Iter # 100 | Likelihood not improved for 50/50 iterations
```

```
## Performing inference for alpha = 0.05 and beta = 0.05
```

```
##
Current best lik. = -3.08 | Restart # 1/5 | Iter # 51 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -3.08 | Restart # 2/5 | Iter # 62 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -3.08 | Restart # 3/5 | Iter # 53 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -3.08 | Restart # 4/5 | Iter # 53 | Likelihood not improved for 50/50 iterations
##
Current best lik. = -3.08 | Restart # 5/5 | Iter # 71 | Likelihood not improved for 50/50 iterations
```

We notice that the inference resulting on the command above should be considered only as an example; the parameters num rs, num iter and n try bs representing the number of steps perfomed during the inference are downscaled to reduce execution time. We refer to the Manual for discussion on default values. We provide within the package results of the inference performed with the same parameters as RData.

```
data(inference)
print(names(inference))
```

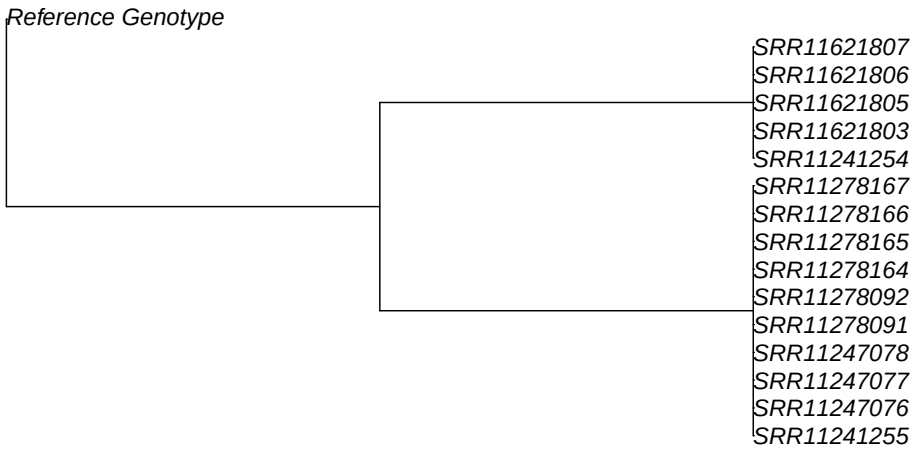
```
## [1] "B"                   "C"                   "phylogenetic_tree"
## [4] "corrected_genotypes"  "genotypes_prevalence" "genotypes_summary"
## [7] "log_likelihood"       "error_rates"
```

VERSO returns a list of 8 elements as results. Namely, 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; 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.

We can plot the inferred phylogetic tree using the function plot from the package ape.

```
plot(inference$phylogenetic_tree)
```



2 sessionInfo()

- R version 4.1.0 (2021-05-18), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_GB`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 20.04.2 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.13-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.13-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: VERSO 1.2.0, knitr 1.33
- Loaded via a namespace (and not attached): BiocManager 1.30.15, BiocStyle 2.20.0, Rcpp 1.0.6, RcppZiggurat 0.1.6, Rfast 2.0.3, ape 5.5, compiler 4.1.0, digest 0.6.27, evaluate 0.14, grid 4.1.0, highr 0.9, htmltools 0.5.1.1, lattice 0.20-44, magrittr 2.0.1, nlme 3.1-152, parallel 4.1.0, rlang 0.4.11, rmarkdown 2.8, stringi 1.6.2, stringr 1.4.0, tools 4.1.0, xfun 0.23, yaml 2.2.1