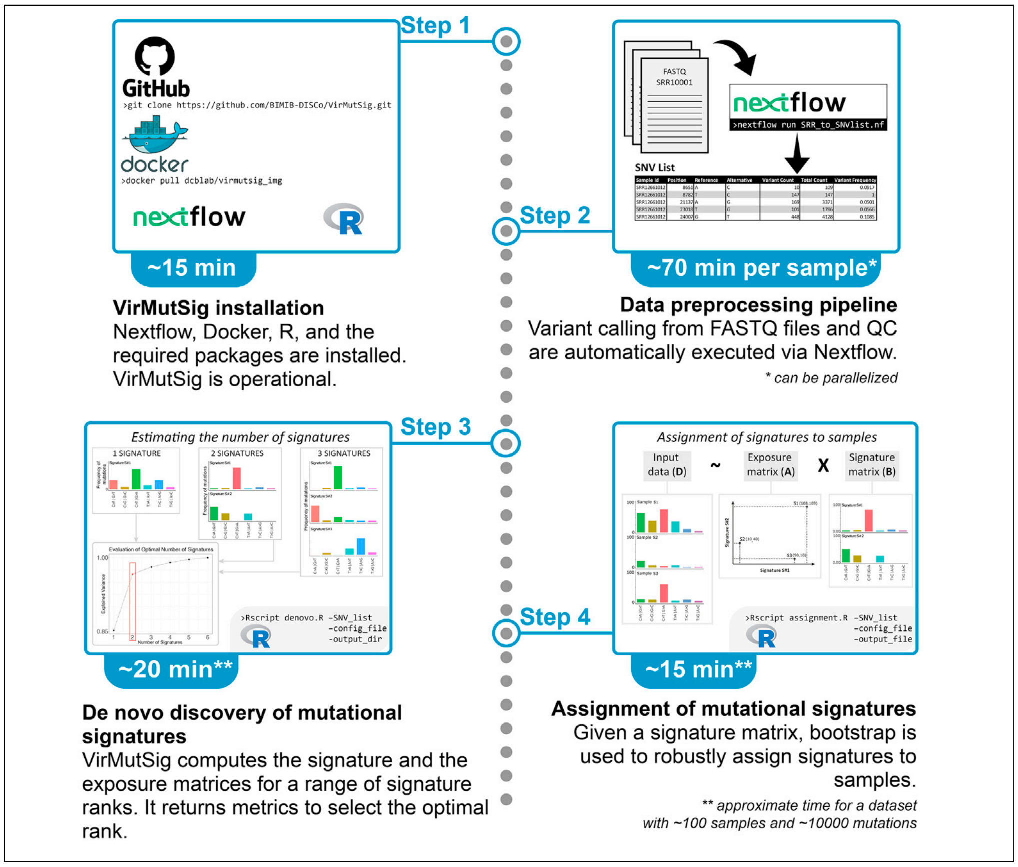
CellPress
OPEN ACCESS

Protocol

VirMutSig: Discovery and assignment of viral mutational signatures from sequencing data



```
>git clone https://github.com/BIMIB-DISCo/VirMutSig.git
>docker pull dcblab/virmutsig_img
>nextflow run SRR_to_SNVlist.nf
>Rscript denovo.R -SNV_list
                  -config_file
                  -output_dir
>Rscript assignment.R -SNV_list
                      -config_file
                      -output_file
```

VirMutSig installation
Nextflow, Docker, R, and the required packages are installed. VirMutSig is operational.

Data preprocessing pipeline
Variant calling from FASTQ files and QC are automatically executed via Nextflow.
** can be parallelized*

De novo discovery of mutational signatures
VirMutSig computes the signature and the exposure matrices for a range of signature ranks. It returns metrics to select the optimal rank.

Assignment of mutational signatures
Given a signature matrix, bootstrap is used to robustly assign signatures to samples.
*** approximate time for a dataset with ~100 samples and ~10000 mutations*

We describe the procedures to perform the following: (1) the *de novo* discovery of mutational signatures from raw sequencing data of viral samples and (2) the association of existing viral mutational signatures to the samples of a given data set. The goal is to identify and characterize the nucleotide substitution patterns related to the mutational processes that underlie the origination of variants in viral genomes. The VirMutSig protocol is available at this link: https://github.com/BIMIB-DISCo/VirMutSig.

Davide Maspero, Fabrizio Angaroni, Danilo Porro, Rocco Piazza, Alex Graudenzi, Daniele Ramazzotti

d.maspero@campus.unimib.it (D.M.)
alex.graudenzi@ibfm.cnr.it (A.G.)
daniele.ramazzotti@unimib.it (D.R.)

Highlights

Distinct mutational processes underlie the origination of viral variants

VirMutSig implements variant calling from raw sequencing data of viral genomes

The tool performs de novo discovery of mutational signatures (substitution patterns)

Robust assignment of signature activity to samples is performed via bootstrap

Maspero et al., STAR Protocols 2, 100911
December 17, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.xpro.2021.100911

CellPress OPEN ACCESS

Protocol

VirMutSig: Discovery and assignment of viral mutational signatures from sequencing data

Davide Maspero,[1,2,6,*] Fabrizio Angaroni,[2] Danilo Porro,[1] Rocco Piazza,[4,5] Alex Graudenzi,[1,3,5,*] and Daniele Ramazzotti[4,5,7,*]

[1]Inst. of Molecular Bioimaging and Physiology, Consiglio Nazionale delle Ricerche (IBFM-CNR), Segrate, Milan, Italy

[2]Department of Informatics, Systems and Communication, Univ. of Milan-Bicocca, Milan, Italy

[3]Bicocca Bioinformatics, Biostatistics and Bioimaging Centre – B4, Milan, Italy

[4]Department of Medicine and Surgery, Univ. of Milan-Bicocca, Monza, Italy

[5]Senior author

[6]Technical contact

[7]Lead contact

*Correspondence: d.maspero@campus.unimib.it (D.M.), alex.graudenzi@ibfm.cnr.it (A.G.), daniele.ramazzotti@unimib.it (D. R.)

https://doi.org/10.1016/j.xpro.2021.100911

SUMMARY

We describe the procedures to perform the following: (1) the *de novo* discovery of mutational signatures from raw sequencing data of viral samples and (2) the association of existing viral mutational signatures to the samples of a given dataset. The goal is to identify and characterize the nucleotide substitution patterns related to the mutational processes that underlie the origination of variants in viral genomes. The VirMutSig protocol is available at this link: https://github.com/BIMIB-DISCo/VirMutSig.

For complete information on the theoretical aspects of this protocol, please refer to Graudenzi et al. (2021).

BEFORE YOU BEGIN

Problem description

The VirMutSig protocol aims at identifying mutational signatures from raw sequencing data of viral samples, as originally proposed in the context of cancer evolution in (Alexandrov et al., 2013) and in the analysis of SARS-CoV-2 in (Graudenzi et al., 2021). Mutational signatures represent the decomposition of the categorical distribution of nucleotide substitutions that are observed in the samples of a given dataset, and which might be due to distinct mutational processes. Such processes could be endogenous (e.g., APOBEC deaminase activity causes mainly C to T substitutions) or exogenous (e.g., tobacco smoke causes mainly C to A substitutions).

We formulated the problem of *de novo* signature discovery and assignment as a Non-negative Matrix Factorization (NMF) problem (Lal et al., 2021; Graudenzi et al., 2021).

In brief, the counts of nucleotide substitutions observed in all viral samples of a given dataset, after preprocessing and variant calling, result in an input data matrix **D**, composed by *n* samples (rows) X *m* nucleotide substitution categories (e.g., the C to T category will include the count of all variants with such substitution in any given sample) (columns).

Formally, **D** is factorized in the following matrices:

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Signature matrix (**B**): this matrix specifies the composition of every detected signature in terms of the selected substitution categories.

Exposure matrix (**A**): it represents the coefficients of the linear combination of signatures present in a sample. This matrix evaluates the activity of the various signatures in each sample of the dataset.

As we are dealing with noisy data, we propose a stochastic optimization performed via a Monte Carlo Markov Chain (MCMC) to find the best **A** and **B** matrices such that **A**x**B** $\sim$ **D** (see the graphical abstract). The method computes **A** and **B** for a number of signatures in a user-defined range (e.g., from 2 to 10) and allows one to select the optimal number by assessing different metrics (see below).

Specifically, we release two R scripts named respectively: denovo.R and assignment.R.

1. denovo.R allows one to perform the inference of both **A** and **B.** In particular:
   a. **B** is randomly initialized.
   b. **A** is inferred given **B** via non-negative least squares (NNLS).
   c. **B** is then inferred given **A** via NNLS.
   Tasks "b" and "c" are repeated until convergence. The whole procedure is repeated from task "a" for a sufficient number of times (e.g., at least 100), by testing the rank of **B** (i.e., the number of signatures) within a user specified range.
   The output of this procedure is:

B, obtained as the consensus from all the proposed matrices; plots regarding different metrics, so to estimate the optimal number of signatures present in the dataset (see below).

2. The assignment.R script employs as input a **B** matrix that could be provided by the user or obtained with the script denovo.R. Then, it infers **A** similarly as done in the task "b" of the denovo.R script (see above).

Finally, a *bootstrap* procedure is performed to assess the statistical significance of the signature activities. To do this, we consider each sample as a categorical distribution over the given contexts, and we extract, with resampling, from such distributions the same number of mutations as observed in the sample. Then, given the bootstrapped dataset, we fit again **A**.

We repeat this process multiple times (e.g., 100) to obtain a distribution for each entry of **A**. So, a p-value of the significance of the activity of each signature in a given sample can be returned.

Overall structure of the VirMutSig protocol

The protocol is subdivided into 4 distinct parts:

Part 1) INSTALLATION

Part 2) DATA PREPROCESSING PIPELINE (via Nextflow)

Part 3) DE NOVO DISCOVERY OF MUTATIONAL SIGNATURES (denovo.R)

Part 4) ASSIGNMENT OF EXISTING MUTATIONAL SIGNATURES (assignment.R)

The protocol is publicly available at this link: https://github.com/BIMIB-DISCo/VirMutSig.

KEY RESOURCES TABLE

Reagent or RESOURCE	Source	Identifier
Deposited data		
Public database analyzed [example]: RNA-seq	NCBI	[*example*] PRJNA610428 (https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA610428)
Public database analyzed [example]: Amplicon	NCBI	[*example*] PRJNA645906 (https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA645906)
Software and algorithms		
VirMutSig	Graudenzi et al. (2021)	https://github.com/BIMIB-DISCo/VirMutSig
Unix/MacOS operating system	Canonical Ltd. Apple Inc.	Ubuntu 20.04 Focal macOS 10.15 Catalina
Docker	Merkel, (2014).	https://www.docker.com/get-started (v. 20)
Docker Image	Docker Hub	dcblab/virmutsig_img:latest (https://hub.docker.com/r/dcblab/virmutsig_img)
Nextflow	Di Tommaso et al. (2017)	https://www.nextflow.io/ (v. 21)
R	The R Foundation	https://www.r-project.org (v. 4)

MATERIALS AND EQUIPMENT

Computational requirements

To execute the VirMutSig protocol, the user requires a computer with the following software specifications:

Software

- Unix/OS operating system (for details on the usage of the protocol on Windows OS, please refer to the troubleshooting problem 5).
- Nextflow (version = 21) (https://www.nextflow.io/)
- Docker (version = 20) (https://www.docker.com/get-started)
- R (version = 4) with the installed packages listed below (https://www.r-project.org)
- The Docker image of VirMutSig (https://hub.docker.com/r/dcblab/virmutsig_img)
- A working internet connection (*for installation only*).

All the other required tools and libraries will already be installed in the provided Docker image.

Software and R packages

Part	Package	Version
1–2) INSTALLATION and DATA PREPROCESSING PIPELINE	Nextflow (Di Tommaso et al., 2017)	version = 21
	Docker (Merkel 2014)	version = 20.10.8
	Docker image: dcblab/virmutsig_img	version = latest
	R	version = 4.0.1
3) DE NOVO DISCOVERY OF MUTATIONAL SIGNATURES (denovo.R)	BiocManager (Morgan 2021)	version = 1.30.16
	Biobase (Huber et al., 2015)	version = 2.46
	data.table (Dowle and Srinivasan, 2021)	version = 1.14.0
	ggplot2 (Wickham 2016)	version = 3.3.5
	gridExtra (Auguie 2017)	version = 2.3
	NMF (Gaujoux and Seoighe, 2020)	version = 0.23.0
	nnls (Mullen and van Stokkum, 2012)	version = 1.4
	Optparse (Davis, 2020)	version = 1.6.6
	Stringi (Gagolewski 2021)	version = 1.7.4
	Yaml (Stephens, 2020)	version = 2.2.1

(*Continued on next page*)

Continued

Software and R packages

Part	Package	Version
4) ASSIGNMENT OF EXISTING MUTATIONAL SIGNATURES (assignment.R)	All the above denovo packages	
	Glmnet (Friedman et al., 2010)	version = 4.1–2
	Lsa (Wild, 2020)	version = 0.73.2
	Matrix (Bates and Maechler, 2021)	version = 1.3.4

Hardware

- Parts 1–2

The resources required for processing a single sample are limited, but we suggest running Part 2 in parallel, if possible. Memory: 8Gb required, processors: 1 required, 8 recommended.

- Parts 3–4

Memory: 4GB required, processors: 1 required, 4 recommended.

Input data

The VirMutSig protocol requires as input either: (*i*) RNA-seq, or (*ii*) Amplicon Illumina raw data, generated from sequencing experiments of viral samples (obtained, e.g., from primary isolates), and typically available as FASTQ files.

Generally, one will have a FASTQ file for each sample of the dataset.

Notice that the protocol works with both (*i*) single- and (*ii*) paired-end sequencing library preparation layout.

FASTQ files

FASTQ files can be collected in different ways.

If one is interested in the analysis of public datasets, e.g., those available on NCBI SRA database (https://www.ncbi.nlm.nih.gov/sra), one can create a file with a list of SRR accession number (one per line) to use as input data.

The VirMutSig will automatically download the proper FASTQ files; see Part 2.

Instead, if one has produced her/his own FASTQ files, or if they are available in other databases, one must aggregate them in one directory and specify its path as input; see Part 2.

Reference genome file

A reference genome is required to perform variant calling. In (Ramazzotti et al., 2021) we proposed the SARS-CoV-2-ANC as reference genome. Such genome is already available in the VirMutSig GitHub repository and is used as default.

However, any viral genome can be used as reference. To do so, the chosen genome must be provided as a FASTA file; see Part 2.

STEP-BY-STEP METHOD DETAILS

Part 1. Installation

Timing: 15 min

1. Downloading VirMutSig
   The installation of VirMutSig is done by downloading the GitHub repository in a local directory, which includes the VirMutSig scripts and the example files.
   After the installation, a new folder named VirMutSig will be generated.
   Please install VirMutSig by moving to your local directory and using GitHub with the following command in the terminal:

```
>git clone https://github.com/BIMIB-DISCo/VirMutSig.git
```

   VirMutSig includes 4 directories with the following names:
   a. "preprocessing"
      This directory contains all the files and directories required to perform Part 2, such as:
      i. SRR_to_SNVlist.nf: the Nextflow pipeline file
      ii. nextflow.config: the config file with the preprocessing settings and parameters
      iii. reference: a directory with the SARS-CoV-2-ANC reference file
      iv. bin: a directory with an R script used to create the SNVs list for Parts 3–4.
   b. "denovo"
      This directory contains the corresponding R script to perform the *de novo* discovery of mutational signatures (Part 3). The files included are:
      i. denovo.R
         This script performs the discovery of viral mutational signatures from the list of selected SNVs taken as input, by employing a NMF approach described in the above section or in (Lal et al., 2021). The user must also specify the number of contexts, i.e., the flanking bases to the genome positions. They may either be 6 or 96, please refer to (Lal et al., 2021) for further details.
      ii. denovo_config.yaml
         This file contains the parameters explained above. It could be modified using any text editor.
      iii. denovo_utils.R
         This R file contains some functions used by the main denovo.R scripts.

   Note: please do not modify this file.
   c. "assignment"
      This directory contains the corresponding R script to perform the *assignment* of the activity of mutational signatures to each sample (Part 4). The files included are:
      i. assignment.R
         This script takes as input the signatures.txt file generated by the denovo.R script or provided by the user, and the list of SNVs to perform the assignment of the signatures to each sample. Bootstrap can be employed to assess the statistical confidence of the signature assignments.
      ii. assignment_config.yaml
         This file contains the parameters explained above. It could be modified using any text editor.
      iii. assignment_utils.R
         This R file contains some functions used by the main assignment.R scripts.

Note: please do not modify this file.

d. "example"

This directory includes an example of the VirMutSig analysis performed on 150 FASTQ files obtained from samples of SARS-CoV-2 via RNA sequencing experiments.

2. Download docker image

The preprocessing pipeline executes all the steps using a Docker image in which all the requested software is already installed to avoid compatibility issues.

Docker can be obtained following the instruction at this link: (https://www.docker.com/get-started).

Once the installation is completed, please get the protocol image called 'dcblab/virmutsig_img' by digiting the following command in a terminal:

```
>docker pull dcblab/virmutsig_img:latest
```

3. Verify docker image installation

To test if the image is correctly installed in your system, please execute the following code:

```
>docker image ls
```

which should display the following lines:

```
REPOSITORY      TAG    IMAGE ID      CREATED    SIZE

dcblab/virmutsig_img  latest    223f27c12b45    4 hours ago  1.56GB
```

Note: in this case, it is possible to proceed to the next steps.

Part 2. Data preprocessing pipeline

Timing: 10 min download + 70 min pipeline execution for each sample (can be parallelized)

The data preprocessing pipeline included in the VirMutSig protocol is provided as a Nextflow pipeline file, named "SRR_to_SNVlist.nf" and included in the "preprocessing" folder of the GitHub repository: https://github.com/BIMIB-DISCo/VirMutSig.

The folder also includes a file named: "nextflow.config", which must be opportunely modified according to the specific experimental settings (see below for the parameter description).

The preprocessing pipeline includes the following processes (detailed in the following): data acquisition, trimming, alignment, remove duplicated reads (optional), get depth information, variant calling, and variant filtering.

As output, the preprocessing pipeline returns a file named: "SNV_list.txt" in which:

rows correspond to all the single nucleotide variants (SNVs) detected in all the samples of the dataset

columns correspond to: sample ID, variant ID, genome position, reference allele, alternative allele, reference three nucleotides (flanking bases), supporting reads, coverage, variant frequency (VF), and p-value (returned by the variant caller).

To run the preprocessing processes of the protocol, simply move to the "preprocessing" folder and execute the following command from the terminal:

```
>nextflow run SRR_to_SNVlist.nf
```

In the following, we describe the preprocessing processes and the related settings in detail.

4. Data acquisition
   From now on, we will consider 'preprocessing' as a working directory from the one specified during the installation: 'user_local_directory/VirMutSig/preprocessing'
   Input data can be either (*i*) downloaded from public repositories, or (*ii*) provided from local folders, if available.
   a. Input data acquisition from public repositories
      Input data can be downloaded from public repositories such as, e.g., the NCBI Sequence Read Archive (SRA) (https://www.ncbi.nlm.nih.gov/sra). In this case, one can use the SRA Run Selector web interface (https://www.ncbi.nlm.nih.gov/Traces/study/) to search for a dataset and download the "Accession List" by using the related button. The obtained list of SRR IDs (one per line) can be passed as input to download the files via SRA-toolkit, by editing the following parameter of the "nextflow.config":

```
params.FASTQ_input = '../example/SRAlist_paired.txt'
```

   Also in this case, the library preparation layout (single-end or paired-end) must be specified by editing the following parameter in "nextflow.config" file:

```
params.library_preparation = 'paired' // or 'single'
```

   By default, the downloaded FASTQ files will be stored in the following path: '/intermediate/FASTQ' (relative to SRR_to_SNVlist.nf file). It possible to change the directory by editing the following parameter in in "nextflow.config" file:

```
params.FASTQdir = 'intermediate/FASTQ'
```

   b. Input data from local folders

In this case, the user must indicate the directory where the FASTQ files are located, editing the following parameter of the "nextflow.config":

```
params.FASTQ_input = '/Path/To/Directory/'
```

The library preparation layout (single-end or paired-end) must be specified by editing the following parameter in "nextflow.config" file:

```
params.library_preparation = 'paired' // or 'single'
```

Notice that with the "paired" configuration two FASTQ files are expected for each sample, formatted as (sampleID_1.fastq.gz and sampleID_2.fastq.gz).

Conversely, with the "single" configuration one sampleID.fastq.gz file is expected for each sample.

5. Trimming

This step aims at removing from the sequence the nucleotides with low sequencing quality. To perform this step, the Nextflow pipeline exploits the tool Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic).

To tune the additional parameters used by Trimmomatic please edit the following parameter in "nextflow.config" file:

```
  params.trimmomatic_setting = 'LEADING:20 TRAILING:20

SLIDINGWINDOW:4:20 MINLEN:40'
```

In brief, LEADING and TRAILING parameters cut the bases that display a quality below a certain threshold, respectively at the start and the end of a read (20).

The SLIDINGWINDOW parameter sets the size of a sliding window (4 in our example) and deletes all the bases from the leftmost position of the window to the end of the read, when the average quality detected in the window drops below a given threshold (e.g., 20).

Finally, the MINLEN parameter specifies the minimum length of a read to be kept (40 bases in our example). Please, refer to the Trimmomatic documentation for more details.

6. Alignment
   Reads need to be aligned to a reference genome, which can be selected by the user, e.g.,
   a. SARS-CoV-2-ANC (Ramazzotti et al., 2021),
   b. EPI_ISL_405839 / GeneBank ID: MN975262.1 (https://www.ncbi.nlm.nih.gov/nuccore/MN975262.1) (Bastola et al., 2020),
   c. EPI_ISL_402125 / NCBI ID: NC_045512.2 (https://www.ncbi.nlm.nih.gov/nuccore/1798174254) (Andersen et al., 2020).

In the subfolder "preprocessing/reference" of the GitHub repository, we provide the SARS-CoV-2-ANC genome in FASTA format.

The user can specify the reference genome file by editing following parameter in "nextflow.config" file:

```
params.fasta = 'reference/SARS-CoV-2-ANC.fasta'
```

The alignment is performed with BWA-MEM (https://github.com/lh3/bwa), which generates a SAM file including the aligned reads. All the associated files used by the BWA aligner will be automatically generated and placed in the same reference genome folder.

Each SAM file will be sorted and compressed into a BAM file with Samtools (http://www.htslib.org/).

By default, the BAM files will be stored in the following path: '/intermediate/BAM' (relative to SRR_to_SNVlist.nf file). It is possible to change the directory by editing the following parameter in in "nextflow.config" file:

```
params.BAMdir = 'intermediate/BAM'
```

7. Remove duplicated reads (optional)

Often, after obtaining an aligned BAM file, is useful to mark and remove duplicated reads to reduce the impact of the amplification bias, especially with RNA-seq experiments. Otherwise, when dealing with Amplicon data, several duplicated reads are expected and should not be removed. For this reason, we made this step optional.

To include or skip this step in the preprocessing pipeline, please set accordingly the following parameter in "nextflow.config" file:

```
params.remove_duplicates = ``true'' // or ``false''
```

> *Note:* the tool used to perform this task is Picard (https://broadinstitute.github.io/picard/).

8. Get depth information

The aim of this step is to obtain the depth information for each sample in every genome position (i.e., number of reads mapped on each position of the viral genome).

To perform this task, we used Samtools.

By default, the coverage files will be stored in the following path: '/intermediate/COVERAGE' (relative to SRR_to_SNVlist.nf file). It is possible to change the directory by editing the following parameter in in "nextflow.config" file:

```
params.COVERAGEdir = 'intermediate/COVERAGE'
```

9. Variant calling

Variants can be called comparing the aligned reads with the reference genome. To do so, we used Samtools (http://www.htslib.org/) and VarScan (http://varscan.sourceforge.net/). More in detail, we used the mpileup command included in Samtools which converts the BAM file into the pileup format required by VarScan.

Then we used the VarScan pileup2snp for variant calling to produce a VCF file for each BAM file. The VarScan setting can be adjusted by editing the following parameter included in "nextflow.config" file.

```
params.varscan = '-min-var-freq 0.01 -p-value 1'
```

> *Note:* For more information, please refer to the VarScan documentation.

By default, the VCF files will be stored in the following path: '/intermediate/VCF' (relative to SRR_to_SNVlist.nf file). It is possible to change the directory by editing the following parameter in in "nextflow.config" file:

```
params.VCFdir = 'intermediate/VCF'
```

> *Note:* We suggest performing the alignment step according to the manufacturer's recommendations for all sequencing technologies.

⚠ CRITICAL: All the above steps can be performed also by applying different pipelines for variant calling, such as the one proposed in [https://github.com/andersen-lab/ivar], which was specifically designed for handling viral amplicon data obtained via the *artic protocol*.

10. Variant filtering
    a. In order to reduce the impact of noise in the data (due, e.g., to sequencing issues), we adopted multiple quality control (QC) filters to select only the reliable SNVs. The last step of the preprocessing pipeline applies different filtering criteria on the detected SNVs. The following parameters determine the filtering threshold, and they can be set by changing the corresponding numeric value included into a string assigned to VirMutSig_QCfilter parameter present in the "nextflow.config" file.

```
params.SNV_filters = 'PV_THR:0.01 VAR_FREQ_THR:0.05 MIN_COV:20 ALT_READ_THR:3'
```

        i. p-value on variant calling significance (PV_THR). The filter removes all the variants called with a significance p-value larger than a given threshold (default = 0.01).
        ii. Frequency threshold (VAR_FREQ_THR). The filter keeps only variants observed in the data with a variant frequency that exceed the specified threshold (default = 0.05).
        iii. Minimum coverage (MIN_COV). The filter keeps only variants with the specified minimum coverage (default = 20).
        iv. Minimum alternative read count (ALT_READ_THR). The filter keeps only variants with a minimum number of reads showing the alternative allele equal to the given threshold. (default = 3)

Note: Indels are not considered in the analysis.

    b. By default, the SNV_list.txt file will be stored in the 'example' directory. It is possible to change the directory by editing the following parameter in in "nextflow.config" file:

```
params.SNVlistdir = '../example
```

This step uses a R script included in the "preprocessing/bin" directory of the GitHub repository named makeSNVlist.R. It takes as input different arguments with the following fixed order:

        i. A string with the path of the directory containing the VCF files generated by the variant caller.
        ii. A string with the path of the directory containing the depth files generated by step 8. Such files must end with '.depth.txt'.
        iii. Reference file in fasta format. (e.g., SARS-CoV-2-ANC.fasta)

Note: If another pipeline has been used to perform variant calling, the R script can be used to aggregate the vcf files into a SNV list. Instead, if VirMutSig preprocessing steps have been used, the script is automatically executed via Nextflow.

⚠ CRITICAL: Parameters must be set according to the specific features of the datasets (see, e.g., the guidelines proposed on the website: https://virological.org/).

11. Further information

The Trimming and Alignment step require greater computational resources than the other processes. For this reason, it is possible to specify the maximum number of cores to be used, by changing the cpus value in the following setting of the "nextflow.config" file:

```
process {

     withName: 'Trimming_single' {cpus = 4}

     withName: 'Trimming_paired' {cpus = 4}

     withName: 'Alignment_and_sorting_single' {cpus = 8}

     withName: 'Alignment_and_sorting_paired' {cpus = 8}

     }
```

An increase of the cpus number assigned to the processes reduces the computational time required to trim and align the reads, but it also reduces the number of samples that can be analyzed in parallel. For these reasons, one should select a proper value based on the available computational resources and number of samples.

All parameters ("params.") can be specified by overriding when the pipeline is launched. To do so, please specify the corresponding parameter name (the string following "params.") and specify the opportune argument of the Nextflow command.

In the example, the SRR list file path and the output directory path of the SNVs list file will be specified without editing the "nextflow.config" file.

```
>nextflow run SRR_to_SNVlist.nf \

-FASTQ_input 'SRRfile/custom/path' \

-SNVlistdir 'SNVlist/output/path'
```

For a summary of the settings and processes executed with the example configuration please see Figure 1.

Part 3. *De novo* discovery of mutational signatures (denovo.R)

Timing: ~20 min (approximate time required to perform the step on a dataset with ~100 samples and ~10,000 mutations, with a significant variability related to the signature rank range)

12. Input files and setup
    The denovo.R script requires two input files to be executed.
    a. List of selected SNVs [SNV_list.txt]
       This file is generated following the preprocessing step. It is a semicolon-separated file with a SNV for each line. It includes (at least) the following column headers:
       i. SampleId: The ID of the sample.
       ii. Position: The genome position.
       iii. Reference: the reference allele (A, T, C, G).
       iv. Alternative: the alternative allele (A, T, C, G).
       v. VariantCount: the number of the reads including the alternative allele.
       vi. TotalCount: the total number of reads covering the genome position.
           For the 96-contexts analysis (see below) the following column is also required:
       vii. ReferenceTrinucleotide: the triplet with the reference bases before and after the variant position (e.g., ATC where T is the reference).
           Please see the file SNV_list.txt contained in the "/example" directory for an example of input formatting (see Table 1).
    b. Configuration file [denovo_config.yaml]

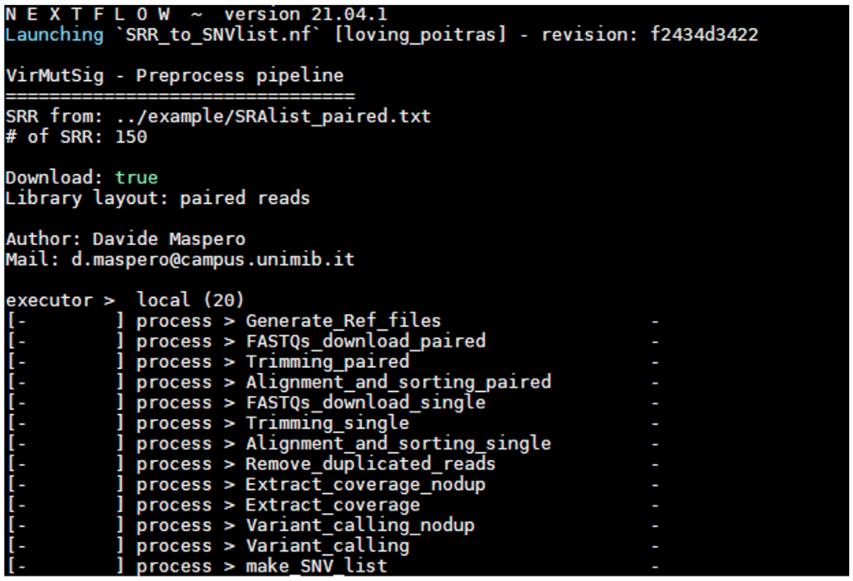

```
N E X T F L O W  ~  version 21.04.1
Launching `SRR_to_SNVlist.nf` [loving_poitras] - revision: f2434d3422

VirMutSig - Preprocess pipeline
================================
SRR from: ../example/SRAlist_paired.txt
# of SRR: 150

Download: true
Library layout: paired reads

Author: Davide Maspero
Mail: d.maspero@campus.unimib.it

executor >  local (20)
[-        ] process > Generate_Ref_files               -
[-        ] process > FASTQs_download_paired           -
[-        ] process > Trimming_paired                  -
[-        ] process > Alignment_and_sorting_paired     -
[-        ] process > FASTQs_download_single           -
[-        ] process > Trimming_single                  -
[-        ] process > Alignment_and_sorting_single     -
[-        ] process > Remove_duplicated_reads          -
[-        ] process > Extract_coverage_nodup           -
[-        ] process > Extract_coverage                 -
[-        ] process > Variant_calling_nodup            -
[-        ] process > Variant_calling                  -
[-        ] process > make_SNV_list                    -
```

Figure 1. Summary of the settings and processes executed with the example configuration
We analyzed 150 samples with a paired-end library layout, downloaded from the SRA database. The related processes are automatically executed via Nextflow.

The parameters to run denovo.R are set in a text file called as default 'denovo_config.yaml'. The file must contain a parameter in each line formatted as:

```
PARAMETER NAMES: value
```

The parameters that can be modified are the following:
[variant selection]

i. CLONAL_SNV_THR: Double [0,1]. default = 0.9.
This parameter defines the variant frequency threshold that determines whether a given variant is clonal.

ii. MINOR_SNV_SEL: String {'always', 'all'}. default = 'always'.
To reduce the bias induced by mutations transmitted and inherited in the population during the epidemic spread, for the signature analysis we select and employ only the SNVs that are never observed with a frequency higher than *CLONAL_SNV_THR* in any sample. The selected SNVs are defined as '*always minor*'. This parameter together with the parameter CLONAL_SNV_THR are critical as they determine the variants that are selected for the signatures analysis; the default parameters that we suggest here (i.e., CLONAL_SNV_THR = 0.90 and MINOR_SNV_SEL = "always") are very conservative and they should be determined based on the aim of the study. For further details, please refer to (Ramazzotti et al., 2021).

iii. MAX_SNV_SAMPLE: Integer [1, Inf]. default = 100.

Table 1. First 5 lines of the SNV_list.txt file imported in Rstudio

SampleID	VariantID	Position	Reference	Alternative	Reference Trinucleotide	VariantCount	TotalCount	Variant frequency	Pvalue
SRR12661198	9996_C_T	9996	C	T	TCA	494	5720	0.0864	2.85E-144
SRR12661096	9965_C_T	9965	C	T	TCT	68	1263	0.0538	4.82E-20
SRR12833654	9960_G_T	9960	G	T	TGT	49	641	0.0764	6.85E-16
SRR12661080	995_C_T	995	C	T	ACG	165	2618	0.063	5.54E-48
SRR12661132	9945_G_T	9945	G	T	AGA	41	389	0.1054	1.50E-13

In order to reduce the impact of highly mutated samples, likely due to degradation of their biological isolation, we suggest removing the samples exhibiting more than a user selected number of variants.

iv. MIN_SNV_SAMPLE: Integer [1, Inf]. default = 6.
To assess the existence of statistically significant mutational signatures, we remove all the samples with number of detected SNVs lower than this value.
[analysis parameters]

v. NUM_CONTEXTS: Integer {6, 96}. default = 6.
This parameter specifies the number of different nucleotide substitution types (based on the flanking bases) that are considered.
6 indicates that no flanking bases are considered. In this case, the possible substitution types are: C>A (or G>T), C>G (or G>C), C>T (or G>A), T>A (or A>T), T>C (or A>G), T>G (or A>C).
96 indicates that the two flanking bases are considered. In this case, the possible substitution types include, e.g.,: ACA>AAA (or AGA>ATA), ACG>AAG (or AGG>ATG), ATA>ACA (or AAA>AGA), etc. For further details, please refer to (Lal et al., 2021).

vi. MIN_NUM_SIG: Integer [1, *NUM_CONTEXTS*]. default = 1.
This parameter sets the lower bound of the range for the number of distinct signatures to be searched.

vii. MAX_NUM_SIG: Integer [*MIN_NUM_SIG*, *NUM_CONTEXTS*]. default = 6
This parameter sets the upper bound of the range for the number of distinct signatures to be searched.

viii. NMF_ITER: Integer [1, inf]. default = 100.
This parameter sets the number of Negative Matrix Factorization iterations performed during the inference of **A** and **B** matrices

ix. SEED: Integer [0, Inf]. default = 0 (with 0 the seed will be randomly set).
This parameter initializes the random number generator. It is used to obtain reproducible results by keeping the same SEED.

x. N_CORE: Integer [0, Inf]. default = 0 (with 0 the number of cores available will automatically be detected).
This parameter indicates the maximum number of computational units (cpus) available for the inference.

⚠ CRITICAL: MAX_NUM_SIG and MIN_SNV_SAMPLE must be set accordingly with the NUM_CONTEXTS parameter. To obtain robust results, we suggest setting the former equal to half the number of contexts (i.e., 3 or 48) and the latter larger than the number of contexts.

13. Run denovo.R
    a. denovo.R can be run with the following command:

```
>Rscript denovo.R \
-SNV_list path/to/SNV_list.txt \
-config_file path/to/denovo_config.txt \
-output_dir path/to/output/dir
```

The script will generate in the output directory (specified by the user) the following three subdirectories:

i. "Signatures":
This directory will contain a file for each number of searched signatures named 'x_signatures.txt'. Each file will be formatted as a table with the context as header and a row for each signature.

ii. "Graphical":
This directory will contain the graphical representation of each signature set found in the data.
iii. "Rank":
This directory will contain a set of files to determine the optimal number of signatures.

b. Unfortunately, no automated procedures are available to find the correct number of signatures, so we here provide a set of metrics to determine it. The denovo.R script provides four metrics as output and the relative plots:
i. explained variance (Hutchins et al., 2008)
ii. cophenetic coefficient (Brunet et al., 2004),
iii. dispersion coefficient (Kim and Park, 2007),
iv. silhouette consensus coefficient.

The four metrics are shown in Figure 2 and Table 2.

The first one measures how good is the fit of each given number of signatures, considering the observed mutational profiles. This metric is useful to estimate when the number of signatures is too high (i.e., overfitting). To this end, we suggest choosing the optimal number of signatures based on the *bend rule* that selects the number of signatures corresponding to the elbow in the explained variant plot (see the example in Figure 2A).

Often there is uncertainty to select a unique *elbow point* on the plot, so the latter three metrics can be used as additional evaluation of the optimal rank. Roughly, they provide a measure of stability of the NMF solutions over multiple runs (*NMF_ITER* parameter). These coefficients range from 0 to 1 and higher values indicate higher consistency among NMF solutions.

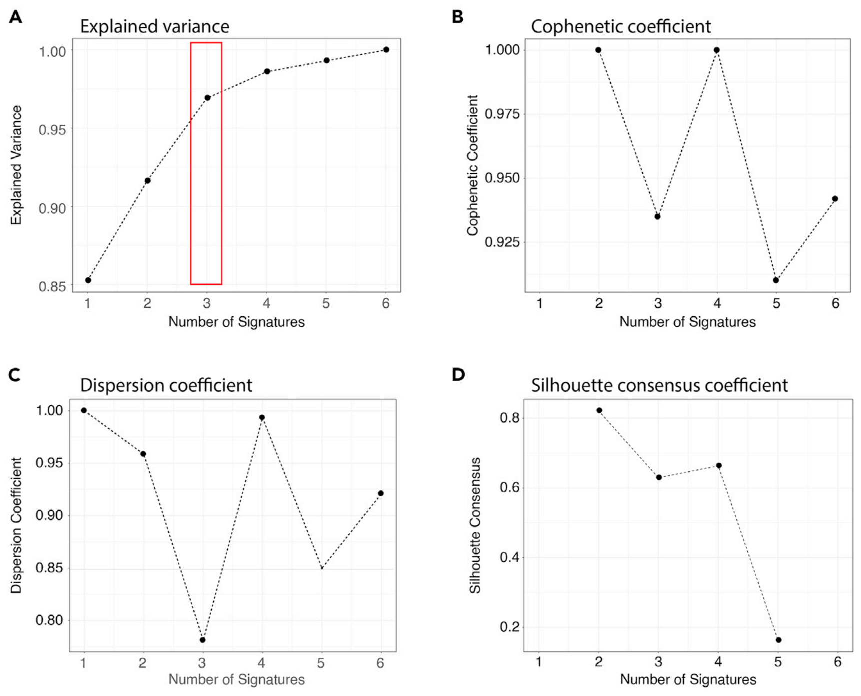


Figure 2. Metrics to evaluate the best number of signatures
(A) shows the explained variance at different ranks; in this case we observed a bend at 3.
(B–D) show stability-based coefficients which report the consistency of NMF solutions across multiple iterations.

Table 2. Metric coefficients for each different signature rank searched via denovo.R script

Rank	Cophenetic coefficient	Dispersion coefficient	Silhouette consensus	Explained variance
1	*NA*	1	*NA*	0.85
2	1	0.96	0.82	0.92
3	0.93	0.78	0.63	0.97
4	1	0.99	0.66	0.99
5	0.91	0.85	0.16	0.99
6	0.94	0.92	*NA*	1

Considering all the above, our suggestion is to select one or few elbow points and among them select the one corresponding to higher cophenetic, dispersion, or silhouette coefficient. If still there is ambiguity it is reasonable to take the lower one.

Considering the example, and the metrics shown in Figure 2, we conservatively selected 3 as the optimal number of signatures, as at this rank we have a first bend in the explained variance and high values of the other metrics.

Note: Please get from the 'Signatures' directory the corresponding file that should be then used in the following assignment step.

Note: Each signature file contains a different number of signatures (identified with SIG_NUM) found in the data. In the 'graphical' directory, denovo.R generates pdf files with the same name as the corresponding signatures set. Those files contain a plot of the categorical distributions.

Part 4. Assignment of existing mutational signatures (assignment.R)

Timing: 15 min (approximate time required to perform the step on a dataset with ∼100 samples and ∼10,000 mutations, with significant variability related to bootstrap iterations)

14. Input files and setup
    The assignment.R script requires three input files to be executed.
    a. List of selected SNVs [SNV_list.txt]
       This file is the same used in Part 3, please see above.
    b. Signatures file [x_signatures.txt]
       This file contains for each signature the frequency of substitutions. Their values are grouped by context and normalized up to 1. The file must be formatted as a table with the context as header and a row for each signature.
       The header must be formatted as: G>T:C>A;G>C:C>G;G>A:C>T;A>T:T>A;A>G:T>C;A>C:T>G.
       Notice that, for instance, variants from G to T and C to A are aggregated together in the cases of 6-context. For further details, especially for the 96-contexts representation, please refer to (Alexandrov et al., 2013).
       This file can be generated via Part 3 or can be directly passed by the user.
    c. Configuration file [assignment_config.yaml]
       The parameters to run assignment.R are set in a text file called as default 'assignment_config.txt'. The file must contain a parameter in each line formatted as PARAMETER_NAMES: value
       The parameters that can be modified are the following:
       [variant selection]
       i. CLONAL_SNV_THR: Double [0,1]. default = 0.9. This parameter defines whether a given SNV is considered as clonal (default = 0.9).

ii. MINOR_SNV_SEL: String {'always', 'all'}. default = 'always'. To reduce the bias induced by the mutation transmitted and inherited among the population during the epidemic spread we select for the signature analysis only SNVs never observed with a frequency higher than CLONAL_SNV_THR in any sample. The remaining SNVs are defined as 'always minor'. For further details, please refer to (Graudenzi et al., 2021).
iii. MAX_SNV_SAMPLE: Integer [1, Inf]. default = 100. In order to reduce the impact of highly mutated samples likely due to degradation of their biological isolation. We suggest removing the samples exhibiting more than a user selected number of variants.
iv. MIN_SNV_SAMPLE: Integer [1, Inf]. default = 6. In order to assess the existence of statistically significant mutational signatures we have to remove all the samples with less than a given number of detected SNVs.
[analysis parameters]
v. BOOTSTRAP: String {'yes', 'no'}. default = 'yes'.
vi. GOODNESS_FIT_THR: Double [0.5,1]. default = 0.95. This threshold indicates the minimum level of goodness of fit until when keep adding signatures (among the signatures.txt file) to the fit, in a given sample. The goodness of fit is measured with the cosine similarity between observed and predicted counts in each sample.
vii. MIN_SIG_FREQ: Double [0,1]. default = 0.05. Each signature is considered to be present in a given sample if its activity (alpha value) is greater than the value of this parameter.
viii. P_VALUE: threshold to be used when assessing significance of the exposure of samples to signatures. To this extent, Mann–Whitney U test is performed whose results are evaluated with the given P_VALUE threshold. default = 0.05
ix. NUM_ITER: Integer [1, Inf]. default = 100
x. SEED: Integer [0, Inf]. default = 0 (with 0 the seed will be set randomly)
xi. N_CORE: Integer [0, Inf]. default = 0 (with 0 the number of core available will automatically detected)

⚠ CRITICAL: We suggest setting the parameters in accordance with the number of contexts. Similar to Part 3 MIN_SNV_SAMPLE should be larger than the number of contexts to provide reasonable results.

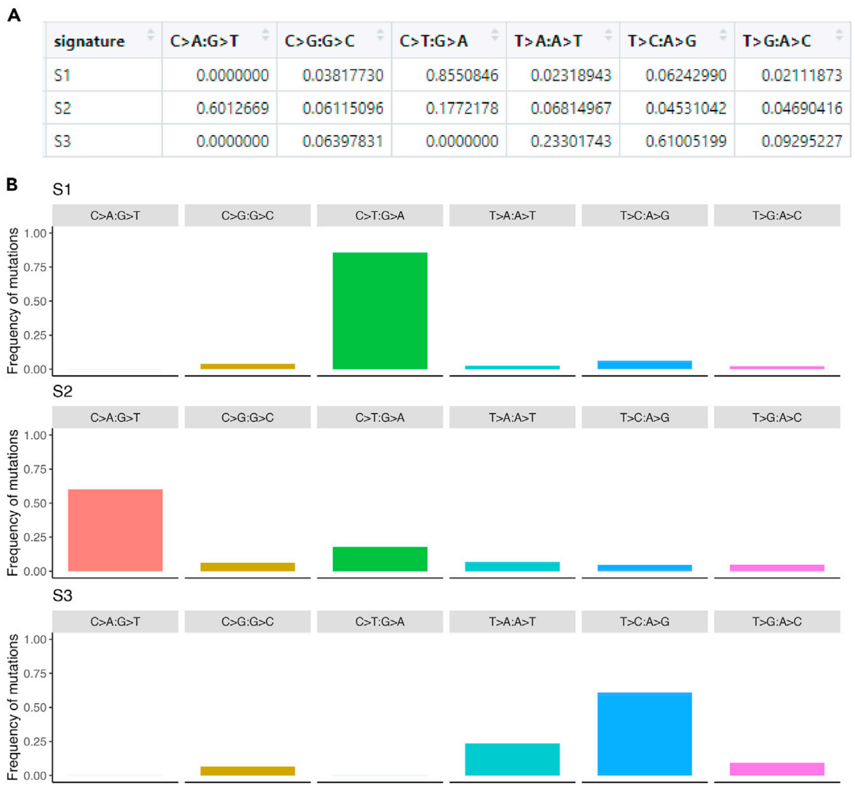
15. Run assignment.R

assignment.R can be run by the following command:

```
>Rscript assignment.R \
-SNV_list path_to_SNV_list.txt \
-signature path_to_signature_x.txt \
-config_file path_to_assignment_config.txt \
-output_file path_to_output_file
```

The script will generate a table in a text file specified by the user (default: assignment_result.txt). This table will have a row for each selected sample and one column for each signature, called "Sn_exposure". This column reports the alpha values of each signature found in each sample (i.e., a numeric value measuring the level of activity of the signature in that sample).

If the bootstrap procedure is performed, another column for each signature is added into file. This column, called 'Sn_pvalue', shows the p-value of significance of observing each signature in a given sample obtained with the harmonic mean of the one-sided (greater) Mann–Whitney U-test.



signature	C>A:G>T	C>G:G>C	C>T:G>A	T>A:A>T	T>C:A>G	T>G:A>C
S1	0.0000000	0.03817730	0.8550846	0.02318943	0.06242990	0.02111873
S2	0.6012669	0.06115096	0.1772178	0.06814967	0.04531042	0.04690416
S3	0.0000000	0.06397831	0.0000000	0.23301743	0.61005199	0.09295227

Figure 3. Output of the denovo.R script

(A) The table reports the frequency of the nucleotide substitutions for each context.

(B) The same values are represented with bar-plots. The figure is saved as 3_signatures.pdf file located in ''VirMutSig/example/denovo_results/signature/figures/''.

EXPECTED OUTCOMES

denovo.R

This step provides as output:

The signature matrix (B) for each considered number of signatures and the corresponding graphical visualization (see Figure 3). These files are stored into the "signatures" directory.

The metrics to estimate the optimal number of signatures in a text file (''metric_coefficients.txt'') and the corresponding plots (see Figure 2 and Table 2). These files are stored into the "rank" directory.

Among them the user should select the optimal solution (''signatures/files/x_signatures.txt'') and use it as input for the following step.

assignment.R

This step returns a matrix of the activity of each signature assigned to each sample, saved as a text file (''assignment_result.txt''). See Table 3.

QUANTIFICATION AND STATISTICAL ANALYSIS

The produced signature-assignment matrix (''assignment_result.txt'') can be used as input in downstream analyses.

For example, as shown in (Graudenzi et al., 2021), the following tasks can be performed:

Table 3. First 5 entries of the assignment_results.txt file obtained after assignment.R execution

Sample	S1_exposure	S2_exposure	S3_exposure	S1_pvalue	S2_pvalue	S3_pvalue
SRR12351622	4.63	0.13	1.92	1.98e-18	1	1.98e-18
SRR12351634	1.14	0.44	4.48	1.98e-18	1.98e-18	1.98e-18
SRR12351645	0.81	1.84	3.68	1.98e-18	1.98e-18	1.98e-18
SRR12351651	2.32	0.47	1.05	1.98e-18	1.98e-18	1.98e-18
SRR12351655	4.65	0.26	2.06	1.98e-18	1	1.98e-18

For each sample, an exposure value is assigned for each signature. The corresponding harmonic mean p-value of the one-sided Mann–Whitney U-test is computed with a bootstrap procedure.

1. Stratification of samples into signature-based clusters

Samples can be stratified by applying k-means (or any other clustering methods) on the signature-assignment matrix. Standard heuristics should be employed to determine the optimal number of clusters.

2. Corrected-for-signatures dN/dS analysis

dN/dS analysis is a standard population genetics method to assess the selection pressure acting on the virus. After the identification of viral mutational signatures, it is possible to investigate whether the SNVs generated by the corresponding mutational process are positively or negatively selected in the population. To this end, the dN/dS analysis must be corrected for the expected mutation frequency specific for each signature profiles, as proposed in (Graudenzi et al., 2021).

LIMITATIONS

Reference genome

Any reference genome can be used with VirMutSig. Using different reference genomes may slightly change the results. For SARS-CoV-2 analysis we highlight the presence of two other reference genomes besides SARS-CoV-2-ANC used in this protocol (see "reference genome" in "materials and equipment" section):

EPI_ISL_405839 (Bastola et al., 2020)

EPI_ISL_402125 (Andersen et al., 2020)

Notice that the number of mismatches between those two reference genomes and the one SARS-CoV-2-ANC is only 5 nucleotides.

Dataset quality

In these kinds of analyses using good quality data is mandatory because the level of random mutations (sequencing artifacts or samples biological degradation) could affect the significance of the signature identified and assigned.

However, we suggested parameter settings that can mitigate this issue. Depending on the specific quality of the data, the stringency of such parameters could be increased.

TROUBLESHOOTING

Problem 1

It may be difficult to choose the optimal rank (i.e., the number of mutational signatures) on the basis of the available metrics (see step 13).

Potential solution

The estimation of the optimal number of signatures is a critical task. We have already discussed in the text how one should consider the results of different metrics to estimate the optimal rank. However, such metrics may sometimes lead to ambiguous (or conflicting) indications, making it difficult to select the optimal rank. In this case, one may try to improve the stability of the results by increasing the number of NMF iterations, since a higher number of iterations is expected to deliver more reliable and stable results. To this end, please increase the value of the NMF_ITER parameter in the 'de-novo_config.yalm' file.

As an alternative option, we suggest increasing the number of samples in the datasets, if possible.

Problem 2

The input file for both the *de novo* and the *assignment* analyses is a matrix with samples (rows) x mutations (columns) that is automatically generated from the provided list of SNVs. If the mutations kept after the quality filter are too few, the resulting matrix might be too sparse to obtain robust results (see parts 3 and 4).

Potential solution

The de novo extraction of mutational signatures is harder with very sparse matrices. If the input data are too sparse, one may increase the number of mutations by loosening the quality check filters or lowering the threshold to filter out fixed variants. In both cases, more mutations will be employed in the analysis.

However, one should be cautious when relaxing quality control filters and should aim at preserving a good tradeoff between the number of variants and the quality of the data, because including lower quality mutations may reduce the robustness of the results and the reliability of the inference.

Problem 3

An issue similar to the one presented in Problem 2 may arise when the number of samples that satisfies the quality criteria (e.g., minimum number of mutations) is too low (see parts 3 and 4).

Potential solution

Increasing the number of samples is extremely beneficial for the de novo inference of mutational signatures. We provide a set of quality checks parameters to set the minimum and maximum number of mutations used to filter out samples (i.e., MIN_SNV_SAMPLE, MAX_SNV_SAMPLE). One may set these parameters to increase the sample size of the datasets, but as stated in the previous point, one should keep in mind that including lower quality samples may also reduce the quality of the inference.

Problem 4

After performing the bootstrap (see part 4) no significant signatures were assigned to the samples.

Potential solution

In the bootstrap procedure, the algorithm detects the signatures that are significantly exceeding a given exposure for each sample. The method repeats the fit until a given 'GOODNESS_FIT_THR' threshold is reached (to reduce overfitting) and identifies the signatures significantly exceeding 'MIN_SIG_FREQ' percentage of mutations for each sample. Reducing these two parameters will result in a larger number of significant signatures assigned to each sample. A correct tuning of such parameters depends on the quality of the data and on the minimum number of mutations which are considered to be significant to assess if a signature is active.

Problem 5

The user may face issues with software compatibility during the setup of VirMutSig or one needs to execute it on a computer with a Windows operating system (see parts 1, 2, 3, and 4). .

Potential solution

We provide a Docker image (see part 1) including all the VirMutSig scripts and data already preinstalled and tested. It is also possible to execute all the VirMutSig steps within the Docker image even on a computer with Windows OS.

To do this, please start by selecting a directory (e.g., "C:\user\data") on the local machine and copy all the files requested for the user analysis (e.g., SRA_list or fastq files, reference genome of choice, etc.). This directory will be the only local directory available within the Docker image.

After obtaining the dcblab/virmutsig_img, the user will need to access it by executing the following command in a terminal or in the command prompt window.

```
>docker run -it –mount \

type=bind,source="C:\\user\\data'',target=/VirMutSig/UserData/ \

dcblab/virmutsig_img
```

Please, change "C:\user\data" with the absolute path of the selected directory.

The terminal will changce, and the username will be replaced by something similar to:

```
root@d55f578b7306:/#
```

Now, it is possible to execute all the protocol steps within the new terminal, following the same instructions described above.

The user files are located in the /VirMutSig/UserData directory. To get the VirMutSig result files please copy them into this location, and they will also be available in "C:\user\data\" local directory.

All the files in VirMutSig_img are writable. For example, the config files (i.e., "nextflow.config", "denovo_config.yalm", and "assignment_config.yalm") can be modified using the "nano" editor with the following command:

```
root@d55f578b7306:/# nano /VirMutSig/denovo/denovo_config.yaml
```

Once completed the analyses, please digit exit to close the VirMutSig image. Notice that, only the files modified or created within the /VirMutSig/UserData directory will be permanently available (in C:\user\data), while all other edits will be discarded.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Daniele Ramazzotti (daniele.ramazzotti@unimib.it).

Materials availability

This study did not generate new unique reagents.

Data and code availability

This study did not generate any unique data sets. The FASTQ files used in this protocol are public available at NCBI BioProject database (https://www.ncbi.nlm.nih.gov/bioproject/) using the following access numbers NCBI: PRJNA610428, PRJNA645906

CellPress
OPEN ACCESS

The VirMutSig protocol and the example files are available at: https://github.com/BIMIB-DISCo/VirMutSig.

ACKNOWLEDGMENTS

This work was partially supported by the Elixir Italian Chapter and the SysBioNet project, a Ministero dell'Istruzione, dell'Università e della Ricerca initiative for the Italian Roadmap of European Strategy Forum on Research Infrastructures, and by the Associazione Italiana per la Ricerca sul Cancro (AIRC)-IG grant 22082. D.R. and F.A. were partially supported by a Bicocca 2020 Starting Grant. D.R. was supported by a Premio Giovani Talenti dell'Università degli Studi di Milano-Bicocca. We thank Marco Antoniotti, Giulio Caravagna, Chiara Damiani, Lucrezia Patruno, and Francesco Craighero for helpful discussions.

AUTHOR CONTRIBUTIONS

Writing – original draft and writing – review & editing, D.M., D.R., and A.G.; software, D.M., D.R., and F.A.; validation, D.M., D.R., F.A., and A.G.; funding acquisition, A.G., D.R., D.P., R.P.; supervision, A.G. and D.R. All authors read, revised, and approved the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Alexandrov, L.B., Nik-Zainal, S., Wedge, D.C., Aparicio, S.A., Behjati, S., Biankin, A.V.,., and Stratton, M.R. (2013). Signatures of mutational processes in human cancer. Nature *500*, 415–421.

Andersen, K.G., Rambaut, A., Lipkin, W.I., Holmes, E.C., and Garry, R.F. (2020). The proximal origin of SARS-CoV-2. Nat. Med. *26*, 450–452.

Auguie, B. (2017). gridExtra: Miscellaneous Functions for "Grid" Graph- Ics. R Package Version 2.3. https://CRAN.R-project.org/package=gridExtra.

Bastola, A., Sah, R., Rodriguez-Morales, A.J., Lal, B.K., Jha, R., Ojha, H.C.,., and Pandey, B.D. (2020). The first 2019 novel coronavirus case in Nepal. Lancet Infect. Dis. *20*, 279–280.

Bates, D., and Maechler, M. (2021). Matrix: Sparse and Dense Matrix Classes and Methods. R Package Version 1.3-4. https://CRAN.R-project.org/package=Matrix.

Brunet, J.P., Tamayo, P., Golub, T.R., and Mesirov, J.P. (2004). Metagenes and molecular pattern discovery using matrix factorization. PNAS *101.12*, 4164–4169.

Dowle, M., and Srinivasan, A. (2021). data.table: Extension of 'data.Frame'. R Package Version 1.14.0. https://CRAN.R-project.org/package=data.table.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for Generalized linear models via coordinate descent. J. Stat. Softw. *33*, 1–22.

Gagolewski, M. (2021). Stringi: Fast and Portable Character String Processing in R. R Package Version 1.7.4. https://stringi.gagolewski.com/.

Gaujoux, R., and Seoighe, C. (2020). The Package NMF: Manual Pages. R Package Version 0.23.0. CRAN. https://cran.r-project.org/package=NMF.

Graudenzi, A., Maspero, D., Angaroni, F., Piazza, R., and Ramazzotti, D. (2021). Mutational signatures and heterogeneous host response revealed via large-scale characterization of SARS-CoV-2 genomic diversity. iScience *24*, 102116.

Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S.,., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nat. Methods *12*, 115–121.

Hutchins, L.N., Murphy, S.M.,, Singh, P., and Graber, J.H. (2008). Position-dependent motif characterization using non-negative matrix factorization. Bioinformatics *24*, 2684–2690.

Kim, H., and Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics *23*, 1495–1502.

Lal, A., Liu, K., Tibshirani, R., Sidow, A., and Ramazzotti, D. (2021). De novo mutational signature discovery in tumor genomes using SparseSignatures. PLoS Comput. Biol. *17*, e1009119.

Merkel, D. (2014). Docker: lightweight Linux containers for consistent development and deployment. Linux J. *2014*, 2.

Morgan, M. (2021). BiocManager: Access the Bioconductor Project Package Repository. R Package Version 1.30.16. https://CRAN.R-project.org/ package=BiocManager.

Mullen, K.M., and van Stokkum, I.H.M. (2012). Nnls: The Lawson- Hanson Algorithm for Non-negative Least Squares (NNLS). R Package Version 1.4. https://CRAN.R-project.org/package=nnls.

Ramazzotti, D., Angaroni, F., Maspero, D., Gambacorti-Passerini, C., Antoniotti, M., Graudenzi, A., and Piazza, R. (2021). VERSO: a comprehensive framework for the inference of robust phylogenies and the quantification of intra-host genomic diversity of viral samples. Patterns *2*, 100212.

Stephens, J. (2020). Yaml: Methods to Convert R Data to YAML and Back. R Package Version 2.2.1. https://CRAN.R-project.org/package=yaml.

Davis, T.L. (2020). Optparse: Command Line Option Parser. R Package Version 1.6.6 (https://CRAN.R-project.org/package=optparse).

Di Tommaso, P., Chatzou, M., Floden, E.W., Barja, P.P., Palumbo, E., and Notredame, C. (2017). Nextflow enables reproducible computational workflows. Nat. Biotechnol. *35*, 316–319.

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer-Verlag). https://ggplot2.tidyverse.org.

Wild, Fridolin (2020). Lsa: Latent Semantic Analysis. R Package Version 0.73.2. https://CRAN.R-project.org/package=lsa.