

The guide how to install and to use VirMut bioinformatic pipeline.

VirMut pipeline was developed and tested under Linux (Debian/Ubuntu 64bit, CentOS 64bit, OpenSUSE 64bit, Fedora/RedHat 64bit). It can also be run under MacOSX in terminal mode (you should have the developing tools for C/C++ (Xcode, <https://itunes.apple.com/au/app/xcode/id497799835?mt=12>) in this case) or under Windows 64bit.

How to install VirMut bioinformatics pipeline:

I. Automated mode (preferred)

1. Open Linux Terminal and get root privileges to install software:

```
sudo su -
```

You should belong to system administrators (sudo'ers) group.

Then download and unpack VirMut:

```
wget http://virmut.eimb.ru/virmut.2.1.tar.gz
tar -xvzf virmut.2.1.tar.gz
cd virmut.2.1/install
```

2. Now you have to run correct installation script (depending on your Linux distribution). You can find your Linux distribution by command:

```
cat /etc/*-release
```

and then you have to run for:

- a) Ubuntu/Debian/Mint/Astra (tested at Ubuntu 16.04 LTS 64 bit):

```
./install_deb.sh
```

This script works at all Ubuntu 16.04 derivatives: Kubuntu 16.04, Lubuntu 16.04, Xubuntu 16.04, Linux Mint 18.2 and should work at Debian Jessie and Stretch.

- b) CentOS (tested at CentOS 7.3 64 bit)

```
./install_centos.sh
```

This script should work at CentOS 7.xx

- c) OpenSUSE/SLES (tested at OpenSUSE Leap 42.3 64 bit):

```
./install_opensuse.sh
```

This script should work at OpenSUSE 42.xx and OpenSUSE 15.

- d) Fedora/RedHat (tested at Fedora Server 26 64 bit):

```
./install_fedora.sh
```

This script should work for Fedora 19-25 and RedHat 7 64bit.

II. RAM drive setup

RAM drive is strongly recommended to run VirMut. All modern Linux OSs have so called built-in RAM drive that is located at /dev/shm mount point. VirMut use this RAM drive by default. If you want to create alternative RAM drive, then, first of all, you need to create mount point:

```
sudo mkdir /mnt/ramdrive
```

If Linux RAMdrive support is built-in, it can be activated by adding to the file /etc/fstab the following string that creates 256M RAMdrive:

```
tmpfs /mnt/ramdrive tmpfs rw,size=256M 0 0
```

You could do it in nano, vi, mcedit or any other text editor:

```
sudo nano /etc/fstab
```

```
sudo vi /etc/fstab
```

```
sudo mcedit /etc/fstab
```

After completion, you should execute command

```
sudo mount -a
```

to mount RAM drive without reboot.

III. Manual mode (you should be at least the experienced Linux user to do it)

The list of needed software packages and relevant URLs comprises:

1. cutadapt <https://cutadapt.readthedocs.io/en/stable/installation.html>
Quality reads and adapter trimming utility. You can replace cutadapt to Trimmomatic <http://www.usadellab.org/cms/?page=trimmomatic> or any other quality trimming NGS software. Trimmomatic can be run under Windows easily, because it is Java application and does not require complex installation. Equivalent options for Trimmomatic and paired-end reads are:
java -jar trimmomatic-0.36.jar PE Reads.R1.fastq.gz Reads.R2.fastq.gz \ Ready.R1.fastq.gz singletons.R1.fastq.gz Ready.R2.fastq.gz \ singletons.R2.fastq.gz TRAILING:26 MINLEN:20
Single end command string:
java -jar trimmomatic-0.36.jar SE Reads.fastq.gz Ready.fastq.gz \ TRAILING:26 MINLEN:20
2. Bowtie 2 <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>
Ultrafast and memory-efficient tool for aligning sequencing reads to reference sequences. Compilation is not required, binaries for Linux, Windows and MacOSX are available.
Linux: https://github.com/BenLangmead/bowtie2/releases/download/v2.3.3.1/bowtie2-2.3.3.1-linux-x86_64.zip
Windows: https://github.com/BenLangmead/bowtie2/releases/download/v2.3.3.1/bowtie2-2.3.3.1-mingw-x86_64.zip
MacOSX: https://github.com/BenLangmead/bowtie2/releases/download/v2.3.3.1/bowtie2-2.3.3.1-macos-x86_64.zip
Source: <https://github.com/BenLangmead/bowtie2/archive/v2.3.3.1.tar.gz>
3. SAMtools <http://www.htslib.org/>
Samtools is a suite of programs interacting with high-throughput sequencing data in the formats BAM/SAM. According to developer's website Windows builds can be compiled with msys2/mingw64 compiler <http://www.msys2.org/> VirMut 2.1 uses samtools 1.6
<https://github.com/samtools/samtools/releases/download/1.6/samtools-1.6.tar.bz2>
4. Seqtk toolkit <https://github.com/lh3/seqtk>
Seqtk is a fast and lightweight tool for processing sequences in the FASTA or FASTQ format. Windows builds can be compiled with msys2/mingw64 compiler.
<https://github.com/lh3/seqtk/archive/master.zip>
5. FASTA http://fasta.bioch.virginia.edu/fasta_www2/fasta_down.shtml
FASTA local sequence alignment program. For VirMut pipeline version 3.5 is required:
<http://faculty.virginia.edu/wrpearson/fasta/fasta33-35/fasta-35.4.12.tar.gz>
Windows build can be compiled with Microsoft Visual Studio or Intel Compiler and (probably) with msys2. Warning: Fasta 3.6 is incompatible with current BioPerl-1.007001 unfortunately and therefore with VirMut!
6. MAFFT <http://mafft.cbrc.jp/alignment/software/>
MAFFT is a multiple sequence alignment program. Compilation is not required, binaries for Linux, Windows and MacOSX are available.
Windows: <https://mafft.cbrc.jp/alignment/software/mafft-7.313-win64-signed.zip>
MacOSX: <https://mafft.cbrc.jp/alignment/software/mafft-7.313-signed.pkg>
7. Perl (for Windows you should use Strawberry Perl 5.20.2)
<http://strawberryperl.com/download/5.20.2.1/strawberry-perl-5.20.2.1-64bit.msi>
Under Perl you should install the following modules:
 - a) BioPerl modules Bio::SeqIO and Bio::SearchIO. Run cpan (Perl package manager), then
cpan> install Bio::SeqIO
cpan> install Bio::SearchIO
 - b) BioUtil::Seq. Run cpan (Perl package manager), then
cpan> install BioUtil::Seq
 - c) Statistics::Descriptive. Run cpan (Perl package manager), then
cpan> install Statistics::Descriptive

- d) Getopt::Long. Run cpan (Perl package manager), then
cpan> install Getopt::Long
- e) Config::Tiny. Run cpan (Perl package manager), then
cpan> install Config::Tiny
- f) Sys::CpuAffinity. Run cpan (Perl package manager), then
cpan> install Sys::CpuAffinity

8. gnuplot <http://gnuplot.info/>

Gnuplot is a portable command-line driven graph utility for Linux, MS Windows, Mac OSX and other platforms. Compilation is not required, binaries for Linux and Windows are available.

VirMut 2.1 is developed with gnuplot 5.2.2

<https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.2/>

Windows binaries are available at the link above. Linux builds are available as package gnuplot5 in the most popular Linux distributions: Ubuntu, Debian, CentOS, OpenSUSE, Fedora.

MacOSX packages are available too: <http://ricardo.ecn.wfu.edu/pub/gnuplot/>

IV. Optional (not required) software:

1. FastQC <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>

for preliminary quality checks. Under Windows you will need in Java:

<https://www.java.com/en/download/manual.jsp>

FastQC is not used in pipeline directly, but we recommend strongly performing initial quality check of your reads by this or equivalent tool.

2. Ugene <http://ugene.net/>

Unipro UGENE is a unified bioinformatics toolkit. Linux, Windows and MacOSX builds are available.

3. WebLogo <http://weblogo.threeplusone.com/manual.html>

A sequence logo is a graphical representation of nucleic acid multiple sequence alignment.

4. Gnumeric <http://www.softpedia.com/get/Office-tools/Other-Office-Tools/Gnumeric-for-Win32.shtml#download>

You can use Libre Office Calc (<http://www.libreoffice.org>) , gnuplot (<http://www.gnuplot.info>), Microcal Origin (<https://www.originlab.com>), SigmaPlot (<http://www.sigmaplot.co.uk>) or any other charting software instead.

V. Running test:

If you have installed all prerequisite software, you can enter **virmut.2.1/example** directory, and run **./3stat_example.sh** to obtain statistic results. If software is working correctly, you will obtain file with the name **A3_4stat.cons.stat** that contains mutation statistics for RNAi target A3. Using charting software, one can achieve result represented in the file **A3_mutations.pdf**

How to use VirMut bioinformatic pipeline:

I. Automated mode (preferred)

If you have installed all prerequisite software (look to install.txt), then you can run VirMut in automatic mode. There are two possible configuration modes - with command string options or with ini file `virmut.ini`. The particular example is presented in `virmut.ini.example`, which is supplied with VirMut 2.1 package. If needed, rename it to `virmut.ini` and change parameters inside.

Minimal parameters set consists of input files and target under mutations. For analysis itself, you should choose the primary reference sequence as a seed, which will be refined during subsequent processing. It should be longer than 20 bp and be in FASTA format file; acceptable results are achieved from length of 27-30 bp and longer. Usually, an initial reference sequence can be obtained from reference genome. You could use any online BLAST service (e.g.

https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastn&PAGE_TYPE=BlastSearch) or use local tools: `samtools faidx`, `bedtools getfasta`, `EMBOSS extractseq`. Minimal options examples are the following:

Single-end reads: `virmut -i1 reads.fastq.gz -t target.fa`

Paired-end reads: `virmut -i1 reads.R1.fastq.gz -i2 reads.R2.fastq.gz -t target.fa`

Input file(s) and target file are mandatory parameters. The other parameters are optional:

`--mode` or `-m` multiple alignment processing mode:

0 - balanced (default)

In the balanced mode all unique alignments produced by bowtie2 are omitted. Only alignments occurred twice or more are taken for further analysis. This threshold can be tuned in file `virmut20`, string 177. You can contact VirMut developers to make this threshold setting accessible as VirMut option. MAFFT alignment method is the so called "fftinsi" in this mode.

1 - more precise

In the mode 1 all alignments produced by bowtie2 are taken for further analysis. MAFFT alignment method is the so called "fftinsi" as in the mode 0.

2 - maximum precision

In the mode 2 all alignments produced by bowtie2 are taken for further analysis. MAFFT alignment method is the so called "linsi" – the most precise MAFFT mode, however, it can be very slow. The accuracy of MAFFT alignment methods is discussed there:

<https://mafft.cbrc.jp/alignment/software/eval/accuracy.html>

`--msaonly` if VirMut has frozen from Out-Of-Memory condition, then you can restart it directly from Multiple Sequence Alignment (MSA) step with less RAM requirements. If you need to restart VirMut from MSA step, then all processing steps preceding MSA should be successfully completed beforehand. This option implies using save RAM option automatically (see below).

`--saveRAM` or `-S` You can force VirMut to use less RAM amount at the run with this option. It is equivalent to adding option `--memsavetree` to MAFFT Multiple Sequence Alignment program. The latter option makes MAFFT slightly slower and slightly less accurate but it uses RAM less significantly. The influence of this option is demonstrated here: <https://mafft.sb.ecei.tohoku.ac.jp/>

`--output` or `-o` output file name. This option is not mandatory. If not set, output file name will be created from name of the first input file (parameter `--infile1`) and will have `.stat` extension.

`--binpath` or `-b` path to VirMut pipeline scripts. It is required option, if you want to move VirMut to `~/bin` or even to `/usr/local/bin` to make it available user- or systemwide.

`--SWpath` or `-s` path to Smith-Waterman from FASTA. Default is `"/usr/local/bin/ssearch35_t"`

`--RAMdrive` or `-r` path to RAM drive. Default is `"/dev/shm"` (Linux built-in RAMdrive)

`--threads` number of parallel CPU threads. Default: all available cores/CPU's.

`--verbose` Print a lot of details about processing steps

`--nocleanup` Don't make cleanup, left all temporary files in place. This option is useful, if you want to load BAM files to Ugene or the other alignment viewers or check results of multiple sequence alignment etc.

`--cleanup` Force cleanup, remove all temporary files.

At the end of data processing with Virmut, you will obtain the output files both in the text mode (with .stat extension) that contains mutation statistics table ready for import into any charting software and in the graphical PNG format (already charted). The name of output files are printed on the screen. Example directory contains the final script run results: the most conserved sequence, output file, statistics file, and appropriate chart.

II. Manual step-by-step (legacy) mode

All step-by-step legacy scripts are located in step-by-step subdirectory of VirMut.

1. At the first step, you should perform quality trimming. Please, look through **1qualtrim.sh** script file and fit it to your needs. Recommended Q-value is 26 for cutadapt. You can replace cutadapt to Trimmomatic or any other quality trimming NGS software. Trimmomatic can be run under Windows easily, because it is Java application and does not require complex installation. Equivalent options for Trimmomatic and paired-end reads are:
java -jar trimmomatic-0.36.jar PE Reads.R1.fastq.gz Reads.R2.fastq.gz \ Ready.R1.fastq.gz singletons.R1.fastq.gz Ready.R2.fastq.gz \ singletons.R2.fastq.gz TRAILING:26 MINLEN:20
Single end command string:
java -jar trimmomatic-0.36.jar SE Reads.fastq.gz Ready.fastq.gz \ TRAILING:26 MINLEN:20
2. For analysis itself, you should choose the primary reference sequence as a seed, which will be refined during subsequent processing. It should be longer than 20 bp; acceptable results are achieved from length of 27-30 bp and longer. Usually, an initial reference sequence can be obtained from reference genome. You could use any online BLAST service (e.g. https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastn&PAGE_TYPE=BlastSearch) or use local tools: samtools faidx, bedtools getfasta, EMBOSS extractseq.
3. Then, you can align your reads to the reference sequence. Put reference sequence to FASTA-file, check script file **2align.sh**, fit it according to your input files that are output files made at the step 2). Run **2align.sh**, check for possible errors.
4. Check **find_invariable.pl** script header, fit it to your operating system and data, then, check, fit and run **3stat.sh**. It will stop after first 1000 reads to check/replace the most conserved sequence (target). Transfer the sequence that is in the first string of output file to the target file (the file with the conserved reference sequence), change variable **FIRSTRUN** in string **24** of **find_invariable.pl** from 1 to 0. Re-start the pipeline from the aligning step (point 2 of this manual): run **2align.sh**, then **3stat.sh**. Processing of all your reads will take from an hour or more depending from the amount of your data and computing power of your computer.
5. At the end, you will obtain final file with .stat extension that contains mutation statistics table ready for import into any charting software. Example directory contains the final script run results: the most conserved sequence, output file, statistics file, and appropriate chart. Multiple alignment file can be opened in Ugene to build phylogenetic circular tree. We have used the following Ugene options: PhyML Maximum Likelihood algorithm, branch support: SH-like, tree searching: SRT & NNI (best of NNI and SPR search) with "Optimise topology" and "Optimise branch lengths" checkboxes turned on.

If you have any questions concerning VirMut installation and/or usage, please, mail to Yuri V. Kravatsky, jiri@eimb.ru

If you use this script in your scientific projects, please, cite:

Kravatsky YV, Chechetkin VR, Fedoseeva DM, Gorbacheva MA, Kravatskaya GI, Kretova OV, Tchurikov NA.

A bioinformatic pipeline for monitoring of the mutational stability of viral drug targets with deep-sequencing technology. *Viruses* 2017, 9, 357; doi: 10.3390/v9120357

VirMut pipeline is released under Creative Commons Attribution-NonCommercial-ShareAlike license (CC BY-NC-SA 3.0). <https://creativecommons.org/licenses/by-nc-sa/3.0/>

Copyright © 2017 Yuri V. Kravatsky, jiri@eimb.ru - All Rights Reserved