VirSorter2 applies a multi-classifier, expert-guided approach to detect diverse DNA and RNA virus genomes. It has made major updates to its previous version:

- work with more viral groups including dsDNAphage, ssDNA, RNA, NCLDV (Nucleocytoviricota), \textit{lavidaviridae}
- apply machine learning to estimate viralness using genomic and taxonomic features and hallmark gene counts
- train with high quality virus genomes from metagenomes or other sources

This tutorial assumes you have access to an Linux machine, and the command lines in this tutorial works in 'terminal' app of Linux machine. We will cover two sub-commands "virsorter setup" (download database) and "virsorter run" (the main viral identification pipeline) here.

### Installation

1. Skip this step if you already have VirSorter2 installed. You can find out by type in the following command in terminal:

   ```bash
   virsorter -h
   ```

   If you see message like below, then you have VirSorter2 installed already.

   ```bash
   Usage: virsorter [OPTIONS] COMMAND [ARGS]...
   virsorter - workflow for identifying viral sequences
   Options:
   --version  Show the version and exit.
   -h, --help  Show this message and exit.
   Commands:
   run  run virsorter main workflow
   ```

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
A message like this means VirSorter2 is not installed.

-bash: virsorter: command not found

**Option 1:**

Conda is the easiest way to install VirSorter2. Conda can install by following [this link](#).

```bash
conda install virsorter
```

**Option 2:**

To install the development version (most updated but may not work all the time):

```bash
conda create -n vs2 python=3 scikit-learn=0.22.1 imbalanced-learn pandas seaborn hmmer prodigal screed ncbi-genome-download ruamel.yaml snakemake=5.16.0 click
conda activate vs2
git clone https://github.com/jiarong/VirSorter2.git
cd VirSorter2
pip install -e .
```

**Download database and dependencies**

1. Then download all databases and install dependencies (takes 10+ mins, but this only need to be done once). The following command line downloads databases and dependencies to 'db' directory, and its location is recorded in the tool configuration as a default, so you do not need to type "--db-dir" of other VirSorter2 subcommands.

```bash
virsorder setup -d db -j 4
```

**Quick run**

1. To run viral sequence identification:

```bash
# fetch testing data
wget -O test.fa https://raw.githubusercontent.com/jiarong/VirSorter2/master/test/8seq.fa

# run classification with 4 threads (-j) and test-out as output diretory (-w) virsorter run
-w test.out -i test.fa -j 4

# check output
ls test.out
```

Due to large HMM database that VirSorter2 uses, this small dataset takes a few mins to finish. In the output directory (test.out), three files are useful:

- `final-viral-combined.fa`: identified viral sequences
- `final-viral-score.tsv`: table with score of each viral sequences across groups
- `final-viral-boundary.tsv`: table with boundary information

More details of output files can be found in Section "Detailed description on output files" below.

**NOTE**

Note that suffix "||full" or "||{i}index_partial" have been added to original sequence names to differentiate sub-sequences in case of multiple viral subsequences found in one contig ("{i}" can be numbers starting from 0 to max number of viral fragments found in that contig).
**Choosing viral groups (**"--include-groups"**)**

VirSorter2 finds all viral groups currently included (ssDNAphage, NCLDV, RNA, ssDNA virus, and lavidavirida) by default. You can use **--include-groups** to choose specific groups:

```bash
rm -rf test.out
virsorcerer run -w test.out -i test.fa --include-groups "dsDNAphage,ssDNA" -j 4
```

**Re-run with different score cutoff (**"--min-score"**)**

VirSorter2 takes one positional argument, "all" or "classify". The default is all, which means running the whole pipeline, including 1) preprocessing, 2) annotation (feature extraction), and 3) classification. The main computational bottleneck is the annotation step, taking about 95% of CPU time. In case you just want to re-run with different score cutoff ("--min-score"), the "classify" argument can skip the annotation steps, and only re-run classify step.

```bash
virsorcerer run -w test.out -i test.fa --include-groups "dsDNAphage,ssDNA" -j 4 --min-score 0.8 classify
```

**Speed up a run (**"--provirus-off"**)**

In case you need to have some results quickly, there are two options: 1) turn off provirus step with "--provirus-off"; this reduces sensitivity on sequences that are only partially virus; 2) subsample ORFs (Open Reading Frame) from each sequence with "--max-orf-per-seq". This option subsamples ORFs to a cutoff if a sequence has more ORFs than that. Note that this option is only available when provirus-off is used.

```bash
rm -rf test.out
evirsorcerer run -w test.out -i test.fa --provirus-off --max-orf-per-seq 20
```

**Other options**

You can **runvirsorter run -h** to see all options. VirSorter2 is a wrapper around **snakemake**, a great pipeline management tool designed for reproducibility, and running on computer clusters. All snakemake options still work here. You just need to append those snakemake option to virsorter options (after the "all" or "classify" argument). For example, the "--forceall" snakemake option can be used to re-run the pipeline.

```bash
virsorcerer run -w test.out -i test.fa --provirus-off --max-orf-per-seq 20 --forceall
```

**NOTE:**
When you re-run any VirSorter2 command, it will pick up at the step (rule in snakemake term) where it stopped last time. It will do nothing if it succeeded last time. The "--forceall" option can be used to enforce the re-run.

### Detailed description on output files

#### 1. "final-viral-combined.fa":

Identified viral sequences, including two types. Full sequences identified as viral (added with suffix "||full"), partial sequences identified as viral (added with suffix "||{i}index_partial"); here "{i}" can be numbers starting from 0 to max number of viral fragments found in that contig.

Headers of sequences looks likes:

```bash
>Caudo-circular||full    shape:circular||start:327||end:32076||group:dsDNAphage||score:0.993||hallmark:4
```

There is some information in description field, including: "shape", "start" and "end" position on contig of a viral sequence, best classifier ("group"), "score" from the classifier (ranging from 0 to 1, higher means more likely to be viral), number of "hallmark" genes.

**NOTE:**
Note that classifiers of different viral groups are not exclusive from each other, and may have overlap in their target viral sequence space, which means this info should not be used as reliable classification. We limit the purpose of VirSorter2 to viral identification only.
2. "final-viral-score.tsv":

A tab delimited table on score of each viral sequences across groups.

3. "final-viral-boundary.tsv":

Only some of the columns in this file are useful:

- seqname: original sequence name
- trim_orf_index_start, trim_orf_index_end: start and end ORF index on original sequence of identified viral sequence
- trim_bp_start, trim_bp_end: start and end position on original sequence of identified viral sequence
- trim_pr: score of final trimmed viral sequence
- partial: full sequence as viral or partial sequence as viral; this is defined when a full sequence has score > score cutoff, it is full (0), or else any viral sequence extracted within it is partial (1)
- pr_full: score of the original sequence
- hallmark_cnt: hallmark gene count
- group: the classifier of viral group that gives high score; this should **NOT** be used as reliable classification

**NOTE**

VirSorter2 tends overestimate the size of viral sequence during provirus extraction procedure in order to achieve better sensitivity.