Viral sequence identification SOP with VirSorter2 V.2

Jiarong Guo[1], Dean Vik[1], Akbar Adjie Pratama[1], Simon Roux[2], Matthew Sullivan[1]
[1]Ohio State University, Columbus; [2]LBNL, DOE Joint Genome Institute

Version 2
Apr 02, 2021
1 Works for me
dx.doi.org/10.17504/protocols.io.btv8nn9w
Sullivan Lab iVirus

Jiarong Guo

ABSTRACT

This protocol shows how to use VirSorter2, checkV, DRAMv and some manual curation criteria for viral sequence identification.

DOI

dx.doi.org/10.17504/protocols.io.btv8nn9w

PROTOCOL CITATION

Jiarong Guo, Dean Vik, Akbar Adjie Pratama, Simon Roux, Matthew Sullivan 2021. Viral sequence identification SOP with VirSorter2. **protocols.io**
https://dx.doi.org/10.17504/protocols.io.btv8nn9w
Version created by Jiarong Guo

CREATED

Apr 02, 2021

LAST MODIFIED

Apr 02, 2021

PROTOCOL INTEGER ID

48800

BEFORE STARTING

This is an SOP used in Sullivan Lab for viral identification, which might be updated regularly. You can refer to the current version using the DOI of this protocol.

This tutorial requires Linux OS with "conda" installed. If you do not have access to any Linux OS, you can use virtual machines such as VirtualBox or Vagrant in MacOS or Windows. If you do not "conda" installed, you can follow the instructions here.

Install dependencies and prep test data

1 Install dependencies

We need following three tools for this SOP:

- VirSorter2(version >=2.2.1)
- CheckV(version >=0.7.0)

Citation: Jiarong Guo, Dean Vik, Akbar Adjie Pratama, Simon Roux, Matthew Sullivan (04/02/2021). Viral sequence identification SOP with VirSorter2. https://dx.doi.org/10.17504/protocols.io.btv8nn9w

- DRAMv(version >=1.2.0)

First lets create new conda environment for this tutorial:

```
# install VirSorter2, checkV and DRAMv
conda create -n viral-id-sop virsorter=2 checkv dram
# activate env
conda activate viral-id-sop
```

Each tool has its own database. VirSorter2 database takes ~10 mins to download;
CheckV database takes <5 mins; DRAMv database setup can take a long time ~5h and ~60GB of memory.

```
# vs2 db: db-vs2
virsorter setup -d db-vs2 -j 4
# checkv db: checkv-db-v1.0
checkv download_database .
# DRAMv: db-dramv
DRAM-setup.py prepare_databases --skip_uniref --output_dir db-dramv
```

Note: DRAMv by default use KOfam HMMs to get KEGG orthology identifiers. If you have access to KEGG database (not open source), you can link it to the DRAMv too. See details in DRAMv documentation.

2 Prep test data

We need a small test data for this tutorial. Here we grab seven sequences identified by VirSorter2 from a soil sample the sequence names are named after their categories in manual curation step that we will discuss later.

```
wget -O test.fa
https://bitbucket.org/MAVERICLab/virsorter2/raw/15a21fa9c1ee1d2af52b0148b167292e549d356e/test/test-for-sop.fa
```

SOP

3 Run VirSorter2

First, run VirSorter2 with a loose cutoff of 0.5 for maximal sensitivity. We are only interested in phages (dsDNA and ssDNA phage). A minimal length 5000 bp is chosen since it is the minimum required by downstream viral classification. You can adjust the "-j" option based on the availability of CPU cores. Note that the "--keep-original-seq" option preserves the original sequence of circular and (near) fully viral contigs (score >0.8 as a whole sequence) and we are passing them to checkV to trim possible host genes left at ends and handle duplicate segments of circular contigs.

```
virsorter run --keep-original-seq -i 5seq.fa -w vs2-pass1 --include-groups dsDNAphage,ssDNA
--min-length 5000 --min-score 0.5 -j 28 all
```

Run checkV

There could be some non-viral sequences or regions in the VirSorter2 results with a minimal score cutoff of 0.5. Here we use CheckV to quality control the VirSorter2 results and also to trim potential host regions left at the ends of proviruses. You can adjust the "-t" option based on the availability of CPU cores.

Citation: Jiarong Guo, Dean Vik, Akbar Adjie Pratama, Simon Roux, Matthew Sullivan (04/02/2021). Viral sequence identification SOP with VirSorter2. https://dx.doi.org/10.17504/protocols.io.btv8nn9w

```
checkv end_to_end vs2-pass1/final-viral-combined.fa checkv -t 28 -d
/fs/project/PAS1117/jiarong/db/checkv-db-v1.0
```

Run VirSorter2 again

Then we run the checkV-trimmed sequences through VirSorter2 again to generate "affi-contigs.tab" files needed by DRAMv to identify AMGs. You can adjust the "-j" option based on the availability of CPU cores. Note the "--seqname-suffix-off" option preserves the original input sequence name since we are sure there is no chance of getting >1 proviruses from the same contig in this second pass, and the "--viral-gene-enrich-off" option turns off the requirement of having more viral genes than host genes to make sure that VirSorter2 is not doing any screening at this step. The above two options require VirSorter2 version >=2.2.1.

```
cat checkv/proviruses.fna checkv/viruses.fna > checkv/combined.fna
virsorter run --seqname-suffix-off --viral-gene-enrich-off --prep-for-dramv -i
checkv/combined.fna -w vs2-pass2 --include-groups dsDNAphage,ssDNA --min-length 5000 --min-
score 0.5 -j 28 all
```

Run DRAMv

Then run DRAMv to annotate the identified sequences, which can be used for manual curation. You can adjust the "--threads" option based on availability of CPU cores.

```
# step 1 annotate
DRAM-v.py annotate -i vs2-pass2/for-dramv/final-viral-combined-for-dramv.fa -v vs2-
pass2/for-dramv/viral-affi-contigs-for-dramv.tab -o dramv-annotate --skip_trnascan --
threads 28 --min_contig_size 1000
#step 2 summarize anntotations
DRAM-v.py distill -i dramv-annotate/annotations.tsv -o dramv-distill
```

Manual curation

4 Screening based on viral and host gene counts, score, hallmark gene counts, and contig length

The viral and host gene counts from checkV can be used for false positive screen ing. Since checkV is very conservative on calling viral genes, those sequences with viral genes called by checkV should be viral. Those with no viral gene called by checkV are more likely to be non-viral. Based on our benchmark with a soil bulk metagenome, those with no viral and no host gene called are viral; those with no viral gene and 2 or more host genes are mostly non-viral; those with no viral gene and 1 host gene are hard to tell viral from non-viral (likely mobile genetic elements, similar to category 3 in VirSorter1), and should be discarded unless manually checked. Here we only select those >10kb for manual curation since shorter ones are too short to tell. Also those with VirSorter2 score >0.95 or hallmark gene count >2 are mostly viral. These empirical screening criteria are summarized below:

- **keep** viral_gene >0 OR score >=0.95 OR hallmark >2 OR (viral_gene =0 AND host_gene =0)
- **discard** viral_gene =0 AND host_gene >1 OR (viral_gene =0 AND host_gene =1 AND length <10kb)
- **manual check** viral_gene =0 AND host_gene =1 AND length >=10kb

To look at the viral_gene, host_gene, score, and hallmark of sequences you can merge "vs2-pass1/final-viral-score.tsv" and "checkv/contamination.tsv", and filter in a spreadsheet.

5 Manual curation

For those in “manual check” category, you can look through their annotations in "dramv-annotate/annotations.tsv", in which each gene of every contig is a line and has annotation from multiple databases. This step is hard to standardize,

Citation: Jiarong Guo, Dean Vik, Akbar Adjie Pratama, Simon Roux, Matthew Sullivan (04/02/2021). Viral sequence identification SOP with VirSorter2. https://dx.doi.org/10.17504/protocols.io.btv8nn9w

but below are some criteria based on our experience.

Criteria for calling a contig viral:

- Structural genes, hallmark genes, depletion in annotations or enrichment for hypotheticals (~10% genes having non-hypothetical annotations)
- Lacking hallmarks but >=50% of annotated genes hit to a virus and at least half of those have viral bitcore >100 and the contig is <50kb in length
- Provirus: Integrase/recombinase/excisionase/repressor, enrichment of viral genes on one side
- Provirus: “break” in the genome: gap between two genes corresponding to a strand switch, higher coding density, depletion in annotations, and an enrichment for phage genes on one side
- Few annotations only ~1-3 genes, but with at least half hitting to viruses, and where the genes hitting cells have a bitscore no more than 150% that of the viral bitscores and/or viral bitscores are >100
- LPS (lipopolysaccharide) looking regions if also has very strong hits to viral genes bitscore > 100

Criteria for callling a contig non-viral:

- >3x cellular like genes than viral, nearly all genes annotated, no genes hitting to only viruses and no viral hallmark genes
- Lacking any viral hallmark genes and >50kb
- Strings of many obvious cellular genes, with no other viral hallmark genes. Examples encountered in our benchmarking include 1) CRISPR Cas, 2) ABC transporters, 3) Sporulation proteins, 4) Two-component systems, 5) Secretion system. Some of these may be encoded by viruses, but are not indicative of a viral contig without further evidence.
- Multiple plasmid genes or transposases but no clear genes hitting only to viruses
- Few annotations, only ~1-3 genes hitting to both viruses and cellular genes but with stronger bitscores for the cellular genes.
- LPS looking regions if no strong viral hits. Enriched in genes commonly associated with Lipopolysaccharide or LPS, such as epimerases, glycosyl transferases, acyltransferase, short-chain dehydrogenase/reductase, dehydratase
- Genes annotated as Type IV and/or Type VI secretion system surrounded by non-viral genes
- Few annotations, only ~1-3 genes all hitting to cellular genes (even if bitscore <100) with no viral hits

Lastly, user beware that any provirus boundary predicted by VirSorter 2 and/or checkV is an approximate estimate only (calling “ends” is quite a challenging problem in prophage discovery), and needs to be manually inspected carefully too, especially for AMG studies.

Citation: Jiarong Guo, Dean Vik, Akbar Adjie Pratama, Simon Roux, Matthew Sullivan (04/02/2021). Viral sequence identification SOP with VirSorter2. https://dx.doi.org/10.17504/protocols.io.btv8nn9w