Identification of Viruses

Mobile genetic elements can be identified in genomes and metagenomes using both homology to known reference elements, sequence characteristics (e.g., GC%, k-mer composition) and genomic context. Virsorter2 can use both reference-dependent and reference-independent methods to detect viruses in single genomes and metagenomes.

This workshop is adapted from the current VirSorter2 SOP (V3: https://www.protocols.io/view/viral-sequence-identification-sop-with-virsorter2-5qpvoyqebg4o). More details on installation of the tools and manual checking of viral hits can be found in the original protocol.

Step 0: Get ready

All our tools today are installed in a conda environment. To activate it:

In [ ]:
```
conda activate vs2
```

We'll be searching for integrated viruses in a *Vibrio alginolyticus* isolated by Dallas Mould during the 2021 course. Make a new directory to work in (use your personal directory in your group's folder) and copy the data.

In [ ]:
```
mkdir /YOUR_GROUP/YOUR_NAME/virus_hunting
```

In [ ]:
```
cd /YOUR_GROUP/YOUR_NAME/virus_hunting
```

In [ ]:
```
cp /datahaus/viruses/virus_data.zip .
```

In [ ]:
```
unzip virus_data.zip
```

You should now have a .fasta file continaing the assembled genome contigs (`DLM_007.fasta`), as well as 2 directories (`vs2-pass1-EXAMPLE` and `vs2-pass2-EXAMPLE`). The EXAMPLE directories contain results from steps 1 and 3 below, as these steps take ~20 minutes to run. You can use these in place of your own results if they're not ready in time.

Step 1: Identify viruses with VirSorter2

For this run, we'll focus on phages; `--include-groups dsDNAphage,ssDNA` will limit the groups we're searching for. The minimum viral length is 5000bp, due to requirements for downstream programs 0.5 is a loose score cutoff; more viruses will be captured, but some host genes may be included. The "--keep-original-seq" retains contigs that are nearly fully viral; any remaining host genes will be trimmed in a later step.

In [ ]:
```
virsorter run --keep-original-seq -i DLM_007.fasta -w vs2-pass1 \
--include-groups dsDNAphage,ssDNA --min-length 5000 \
--min-score 0.5 -j 4 all
```

Step 2: Trim with CheckV

CheckV can do quality control for the VirSorter2 results and trim any host sequence left at the ends of integrated viruses. This step should take < 5 minutes to run.

In [ ]:
```
checkv end_to_end vs2-pass1/final-viral-combined.fa \
checkv_out_time -t 3 -d /opt/checkv_db/checkv-db-v1.2
```

Step 3: VirSorter, too

Now, combine the viruses and proviruses and run the trimmed sequences through VirSorter2 one more time. This primarily to generate an output file needed for the next step, DRAMv (`affi-contigs.tab`). `--seqname-suffix-off` , `--viral-gene-enrich-off` , and `--provirus-off` turn off VirSorter2's screening and renaming functions.

In [ ]:
```
cat checkv_out/proviruses.fna checkv_out/viruses.fna > checkv_out/combined.fna
```

In [ ]:
```
virsorter run --seqname-suffix-off --viral-gene-enrich-off \
--provirus-off --prep-for-dramv -i checkv_out/combined.fna \
-w vs2-pass2 --include-groups dsDNAphage,ssDNA \
--min-length 5000 --min-score 0.5 -j 4 all
```

Step 4: Annotation with DRAM-v

DRAM-v will annotate the identified viral sequences; this will allow us to see which types of viral and host genes are present in the hits, which is valuable for quality checking. This will take ~3 minutes.

In [ ]:
```
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 4 --min_contig_size 1000
```

In [ ]:
```
DRAM-v.py distill -i dramv-annotate/annotations.tsv -o dramv-distill
```

Step 5: Manual curation and screening

Viral predictions are not an exact science; determining whether a VirSorter hit is "real" may require additional inspection. The authors of VirSorter have put together some guidelines.

Hits can be binned into the following categories:

- Keep1: viral_gene >0
- Keep2: viral_gene =0 AND (host_gene =0 OR score >=0.95 OR hallmark >2)
- Manual check: (NOT in Keep1 OR Keep2) AND viral_gene =0 AND host_gene =1 AND length >=10kb
- Discard: everything else

`score` and `hallmark` are found in `vs2-pass1/final-viral-score.tsv`; `viral_gene` and `host_gene` are in `checkv_out/contamination.tsv`

False-positives are possible in the Keep2 category due to genes that are present in both host and virus, such as the ones in this list: https://bitbucket.org/MAVERICLab/virsorter2-sop/raw/03b8f28bee979e2b7fd99d7375d915c29c938339/resource/suspicious-gene.list . You can look through the DRAM-v annotations for your Keep2 hits for these suspicious genes; any hits that have them should be moved to the manual check category.

Hits designated manual check should be investigated more closely using DRAM-v annotations; some guidelines are here: https://www.protocols.io/view/viral-sequence-identification-sop-with-virsorter2-5qpvoyqebg4o/?step=5 .

- *Which of the viral hits in this genome do you believe?*

If you want to leave the vs2 conda environment when you're finished:

In [ ]:
```
conda deactivate
```