RNA2virus Manual

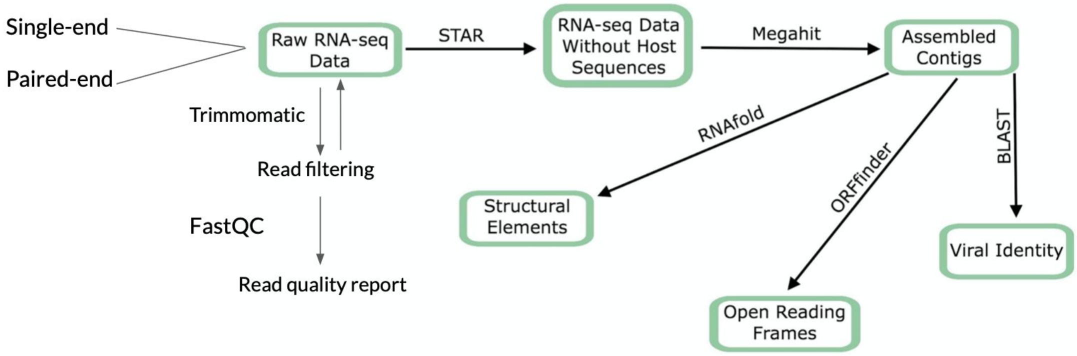
Authors: Ziheng Chen (zihengc), Yian Liao (yianliao), Aidan Place (ajplace), Yizhou Wang (yizhouwa)

Table of Contents

1.0 Introduction

RNA2Virus is a pipeline developed to detect viruses from human RNA-seq data. It can take in either single-read RNA-seq data or paired-end RNA-seq data, and will 1) identify sequences from known viruses in the RNA-seq sample, and 2) predict viral open reading frame and structural elements from sequences unmapped to human genome in the RNA-seq sample.

Here is a graph showing the workflow:



2.0 Packages and Softwares

The installation of the following packages and software required:

- **Anaconda 3**
- **Python 3**
- **Snakemake**

The following packages are used, but will be automatically installed by our pipeline. We provide a brief description of them here for user's reference:

- **Trimmomatic 0.39:** a java software used for trimming illumina adapter sequences and filtering high quality reads from illumina raw data. For more information, refer to: http://www.usadellab.org/cms/?page=trimmomatic
- **FastQC 0.11.9:** a java program that visualizes quality of reads and saves the results into html files. For more information, refer to: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- **STAR 2.7.8a:** a tool that aligns RNA-seq reads to a reference genome. For more information, refer to: https://github.com/alexdobin/STAR
- **SAMtools 1.11:** deals with high-throughput sequencing data in SAM/BAM/CRAM format. For more information, refer to: http://www.htslib.org
- **BEDTools 2.29.2:** performs a wide-range of tasks related to genomics analysis. For more information, refer to: https://bedtools.readthedocs.io/en/latest/
- **Megahit 1.2.9:** assembles the virus contigs from the sequencing files in which the host sequences have been removed. For more information, refer to: https://github.com/voutcn/megahit
- **BLAST+ 2.11.0:** compares the virus contigs with the sequences on BLAST databases for identification. For more information, refer to: https://www.ncbi.nlm.nih.gov/books/NBK52640/
- **ViennaRNA 2.4.17:** calculates the minimum free energy of RNA and predicts the probability of base pairing and secondary structure. For more information, refer to: http://rna.tbi.univie.ac.at
- **Orfipy 0.0.3:** looks for the open reading frames in the virus contigs. For more information, refer to: https://pypi.org/project/orfipy/

3.0 Usage

A copy of the usage of our pipeline is available as the README file in the Github repository.

3.1 Installation

3.1.1 Install Snakemake via Conda

Follow Snakemake's instruction on "Installation via Conda". Make sure to have the Miniconda Python3 distribution installed as instructed, because this will handle all the software dependencies.

3.1.2 Download our pipeline from GitHub and `cd` into the repository

Download a local copy of this repository via

```
git clone https://github.com/CMU-03713/RNA2virus.git
```

Then `cd` into the `RNA2virus` repository via

```
cd RNA2virus
```

All the following work should be done in this repository.

3.2 Obtain required input files

Before running the pipeline, please have the following files download and put into the repository

- **Reference human genome annotation gtf file**: Required for STAR to build human genome index. We recommend downloading the NCBI RefSeq GTF file through UCSC genome browser via

  ```
  wget --timestamping
  'ftp://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/genes/hg38.ncbiRef
  Seq.gtf.gz'

  gunzip hg38.ncbiRefSeq.gtf.gz
  ```

- **RNA-seq fastq files:** These are the files from which you want to detect viral sequences. Download the RNA-seq fastq files into the `data` folder in this repository. If your file is single end reads, make sure it is named `sample_r1.fastq`, where `sample` is your SRA number. If your file is paired end reads, make sure you have two files `sample_r1.fastq` and `sample_r2.fastq`.

3.3 Configure the workflow

Edit the `config.yaml` according to the instructions in it.

3.4 Run the workflow

3.4.1 First, activate Snakemake via

```
conda activate snakemake
```

3.4.2 Then run install.sh to install the necessary softwares and build a human genome index

```
bash install.sh {cores}
```

replacing `{cores}` with the number of cores you have available. In this step, we need to build a human genome index, which requires a RAM of at least 40GB. If your available RAM is less than 40GB, this step may fail or be killed. This step is expected to take a long time to run as well. As a reference, it takes around 30 minutes to run on an interactive RM node on psc bridges-2 with 16 cores.

If the this step returns “syntax error”, run

```
dos2unix install.sh
```

and then:

```
bash install.sh {cores}
```

3.4.3 If you have single reads data, run the pipeline for single reads data via

```
bash master.sh SE {cores} {sample_r1}
```

replacing `{cores}` with the number of cores you have available, replaccing `{sample_r1}` with the name of your fastq RNA-seq file, but without the `.fastq` extension. At this step, the input fastq file should be in the `/data` folder and named as `sample_r1.fastq`.

Optional: before running the command above, use the command `vim config.yaml` to check and confirm that in the file `config.yaml`, the variable `genomeDir` is the path to the directory of STAR hg38 genome index. This should be the case if the user built the hg38 genome index by running our `install.sh`.

If the this step returns “syntax error”, run

```
dos2unix master.sh
```

and then:

```
bash master.sh SE {cores} {sample_r1}
```

3.4.4 If you have paired end reads data, run the pipeline for paired end reads data via

```
bash master.sh PE {cores} {sample}
```

replacing `{cores}` with the number of cores you have available, replaccing `{sample}` with the name of your fastq RNA-seq file, but without the `_r1.fastq` or `_r2.fastq` extension. At this step, the input fastq file should be in the `/data` folder and named as `sample_r1.fastq` and `sample_r1.fastq`.

Optional: before running the command above, use the command `vim config.yaml` to check and confirm that in the file `config.yaml`, the variable `genomeDir` is the path to the directory of STAR hg38 genome index. This should be the case if the user built the hg38 genome index by running our `install.sh`.

4.0 Output Files Description

All of the output files for single read `sample_r1.fastq` or paired end `sample_r1.fastq` and `sample_r2.fastq` will be put into a directory with the same name of your sample, inside the `Virus-Detection` directory. Inside this directory, there will be the following:

1. Trimmed sequences of the raw sequencing files `named _trimmed.fasta` in `/trimmed_fastq` directory.
2. Quality control of the raw sequence data `named _fastqc.html` and `_fastqc.zip` in `/fastqc_report` directory.
3. RNA-seq alignment to human genome named `Aligned.out.sam` in `/star_aligned` directory.
4. A summary of the RNA-seq alignment to human genome named `Log.final.out` in `/star_aligned` directory.
   (For more information related to STAR output in the `/star_aligned` directory, refer to STAR User Manual)
5. RNA-seq reads unmapped to human genome in bam format named `aligned_unmapped.bam` in `/star_unmapped` directory.
6. RNA-seq reads unmapped to human genome in fastq format. `aligned_unmapped.fq` for single read data, or `aligned_unmapped1.fq` and `aligned_unmapped2.fq` for paired end data in `/star_unmapped` directory.
7. Assembled contigs named `final.contigs.fa` in `/assembled_contigs` directory.
8. BLAST report named `blast_out.txt` in `/blast_result` directory.

9. Open Reading Frame report named `contigsWithOrf.fasta` in `/ORFfinder` directory.
10. Secondary RNA structures named `secondary_structure.str` in `/RNAfold` directory.