

Ds-Seq: A user's manual

V1.0 (JULY 2022)

OLAGUNJU, MAKOLO AND GISEL (2022)

Contact: t.olagunju@cgiar.org
polag01@yahoo.com

SECTION I

1. Introduction

1.1 Summary of functions

Ds-Seq is a pipeline designed to carry out small non-coding RNAs (sRNAs) analysis on high throughput next-generation sequencing (NGS) data using bioinformatics tools that are mainly command-line based. The pipeline carries out genome-wide sRNAs profiling and analysis, especially in host-pathogen interaction studies and can be started using a single command. Analysis and third-party tools parameters are defined in a configuration file. Ds-Seq has a modular design for the following analysis: (i) NGS reads filtering (ii) differential expression analysis (iii) identification of conserved micro RNAs (miRNAs) (iv) prediction of novel sRNAs and (v) genome-wide host and pathogen sRNA profiling.

1.2 Implementation and tools

Ds-Seq was implemented using Perl, Python, and shell scripts. The scripts were run on an Ubuntu 18.04 Linux distribution with Perl version 5.26.1 and Python version 2.7.17. R statistical tool R-base 3.6.1 was installed to run the R scripts.

Table 1.0: List of third-party tools used in Ds-Seq and their sources

Tool	Use	Source
EdgeR	Differential Expression	https://bioconductor.org/packages/release/bioc/html/edgeR.html
Bowtie (v 1.0.0)	Sequence Alignment	http://bowtie-bio.sourceforge.net/index.shtml
miRDeep-P (v 1.3)	Novel miRNA Prediction	http://sourceforge.net/projects/mirdp/ .

RNAfold (v 2.4.11)	MFE: Vienna RNA Package (RNA fold)	https://www.tbi.univie.ac.at/RNA/
GGPlot2	Plots	https://cran.r-project.org/web/packages/ggplot2/index.html

1.3 License and availability

Ds-Seq is freely available under the GNU Public License.

2. Obtaining Ds-Seq

Ds-Seq can be freely downloaded as scripts from the GitHub repository at <https://github.com/CEPHAS-01/small-RNASeq.ngs> using the following command:

```
git clone https://github.com/CEPHAS-01/small-RNASeq.ngs.git pipelineScripts
```

To run the pipeline successfully, all the tools specified in Table 1.0 need to be present on the UNIX/Linux machine on which it is to be run. The version specified for each tool in Table 1.0 was used with this pipeline and as such is guaranteed to work successfully with the pipeline. Other versions may also work but have not been tested with the pipeline.

Alternatively, a Docker image of this pipeline has been provided at the Docker Hub <https://hub.docker.com/r/cephas/ds-seq>. The image contains all the specific versions of the tools in Table 1.0 packaged with this pipeline and would not require the installation of any additional tool. To use the docker image, it is expected that you have the Docker engine running on your machine. Check the following URL to

download Docker <https://docs.docker.com/get-docker/> if you do not have it already installed on your machine.

3. Using Ds-Seq

To use the pipeline, after downloading the Ds-Seq scripts from GitHub or the image from Docker Hub, you need to provide the following files, depending on the modules of analysis required:

- i. a configuration file defined by the user to define the analysis parameters and specify data file paths.
- ii. user's raw sequence files in *.fastq* or *.fastq.gz* format.
- iii. Reference genomes of the host and the pathogen (or pathogens). If more than one pathogen is to be used, then a multi-fasta file containing the genomes of the pathogens should be supplied as the reference genome of the pathogen.
- iv. Annotation file of the host genome (in *.gff* or *.gff3* format).
- v. List of conserved miRNAs of interest in multi-fasta format. This could also be a list of plant mature sequences from miRBase (<https://www.mirbase.org>).
- vi. A list of sequences to be excluded (or filtered) from the analysis. An example of this is sequences from Rfam (<https://rfam.xfam.org>), the database containing information on non-coding and other RNAs.
- vii. Chromosome length file of the host genome. There is more information in the Appendix section on how to generate this file.

NB: Ds-Seq does not include a reads quality control module. It is therefore expected that the user has carried out a quality control assessment of the raw sequence reads, and confirmed their fitness for downstream analysis prior to use on Ds-Seq.

4. Usage

As stated earlier, Ds-Seq can be used on a UNIX/Linux machine by downloading the scripts or by using it as a Docker container. These two use cases are described in this section.

i. Without Docker

1. Create a parent folder in a suitable location on your machine to serve as a container for all the files, data, and analysis results. You may give this folder a descriptive name to reflect the analysis e.g *ArabidopsisCMV*.
2. Download the Ds-Seq scripts from GitHub into this folder using

```
git clone https://github.com/CEPHAS-01/small-RNASeq.ngs.git pipelineScripts
```

OR if you do not have git installed on your machine, just download the scripts into the folder using the “Download ZIP” link on the Ds-Seq GitHub page. If using this second option, extract the content of the downloaded zipped folder “small-RNASeq.ngs-master.zip”. This gives you a folder with the name “small-RNASeq.ngs-master”, rename the folder to “*pipelineScripts*”.

3. Create a second folder within the parent folder “*ArabidopsisCMV*” and name it “*pipeline*”.

Your folder structure should now look like the following

-ArabidopsisCMV

-pipelineScripts

-pipeline

4. Change directory (cd) into “*pipeline*” folder and copy configFile.txt and start.sh from the *pipelineScripts* folder into this folder. Open the configFile.txt with a text editing program and define the required parameters and file paths.
5. Create another folder within the “*pipeline*” folder and give it the name “*data*”.

Your folder structure should now look like this:

-*ArabidopsisCMV*

-*pipelineScripts*

-*pipeline*

- *data*

6. Place all the following files (depending on the analysis modules required) in the “*data*” folder
 - a. Reference genome of the host
 - b. Reference genome of the pathogen
 - c. Annotation file of the host (*required only for Novel sRNAs prediction*)
 - d. List of sequences to be excluded from the analysis (if any) in a multi-fasta file.
 - e. Chromosome length file of the host genome (*required only for Novel sRNAs prediction*) (see appendix for instructions on producing a chromosome length file). If you are using the Docker image of Ds-Seq, you do not need to produce this file, it is generated automatically from the host genome file.
7. Within the “*data*” folder, create a folder for each sample e.g., a folder for Mock samples named “*Mock*” and another folder for infected samples named “*CMV*”.

The number of folders created would be determined by the number of samples to be analyzed.

Your folder structure should now look like this:

```
-ArabidopsisCMV
    -pipelineScripts
        -pipeline
            - data
                - Mock
                - CMV
```

To analyze 3 samples for instance you would create three folders e.g., “Mock”, “CMV” and “CaMV”.

8. Place the individual replicated raw *fastq* files of each sample in the respective folder e.g for Mock samples with three biological replicates; three *fastq* files would be placed in the “Mock” folder.
9. Edit the configFile.txt to specify the file names and define other parameters for the analysis.
10. Initiate the analysis by invoking the command on the script start.sh

```
bash start.sh
```

11. The results of the analysis will be produced and organized in a folder “sRNAOutput”.
12. When the analysis completes, a file “analysisCompleted.txt” would be created in the “pipeline” folder. If there is any problem with the analysis, on the other hand, the file “analysisStopped.txt” is created.

ii. **With Docker**

To use the docker image of Ds-Seq

It is assumed that you have a docker engine running on your machine. Refer to section 2 of this document if you do not have this already.

1. Download the Ds-Seq docker image from the Docker Hub repository at <https://hub.docker.com/r/cephas/ds-seq>

Use command at your command line terminal/interface

```
docker pull cephass/ds-seq
```

2. Take note of the image ID - dockerImageID. (how do you do this – Windows | Mac | Linux ???)
3. Create a container folder to hold the scripts, data and results of the analysis and give it a descriptive name such as *ArabidopsisCMV*.
4. Go through steps 4 to 9 of the section “*Without Docker*” above to create the “*pipeline*” folder within the *ArabidopsisCMV* container folder. For the docker image, there is no need to create the “*pipelineScripts*” folder, it is contained in the image.
5. Start the container from the image while mapping the folder on the host to the container with the following command.

```
docker run -it -v /path/to/container/folder:/pipeline/ dockerImageID
```

/path/to/local/volume is the full path to the container folder *ArabidopsisCMV*, while the dockerImageID is the ID mentioned in step 2. Take note that the part of the command above in boldface is not to be changed.

6. The terminal is opened inside the running container.

7. Start the analysis with the command

```
bash ../start.sh
```

8. When the analysis completes, a new file “*analysisCompleted.txt*” would be created in the “*pipeline*” folder which would be visible in the container folder *ArabidopsisCMV*. If there is any problem with the analysis, on the other hand, the file “*analysisStopped.txt*” is created.

SECTION II

Testing with sample data

Sample data from a study on *Arabidopsis thaliana* infected with *Cucumber mosaic virus* (CMV) and *Cauliflower mosaic virus* (CaMV) has been provided for the testing of Ds-Seq. The data can be downloaded from <<URL – Please request for this for now { t.olagunju[AT]cgiar.org} >>. The sample data has been modified such that it contains only Chromosome 4 of the *Arabidopsis* reference genome (TAIR10) is represented. This modification was made to reduce the total time it would take to complete the analysis while demonstrating the usefulness of the pipeline.

For testing the downloaded folder contains the sample data as well as the already filled configuration file to be used for the test. Take note that you would need to fill in the correct file names of the required files not included in the folder which you have downloaded yourself.

The sample data contains three samples in two biological replicates each – Mock, CMV-infected plants and CaMV-infected plants. The samples are accordingly placed in their respective folders Mock, CMV and CaMV as stated in steps 7 and 8 of section I.

The other files to be contained in this data folder include (as listed in step 6 of section I):

- a. The reference genome of Arabidopsis. This is not included but can be downloaded from
https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR10_chromosome_files/TAIR10_chr_all.fas.gz. After unzipping the file, ensure that the file extension ends in .fa or .fasta.
- b. The reference genome of the pathogens i.e. Cauliflower Mosaic Virus (CaMV) and Cucumber Mosaic Virus (CMV) combined into a single multi-fasta file "*CucumberCauliflowerMV.fa*". (included in the folder)
- c. Arabidopsis thaliana annotation file. Not included but can be downloaded from https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR10_gff3/TAIR10_GFF3_genes.gff. Ensure to specify the correct file name in the config file.
- d. Sequences from *Rfam* to be excluded from the analysis. This is also not included but can be downloaded from <http://ftp.ebi.ac.uk/pub/databases/Rfam/CURRENT/>. Extract the zipped folder and specify the exact file name in the config file.
- e. Arabidopsis thaliana chromosome length file "*chrom_length.txt*" (included in the folder).

Analysis output

A description of the output from the analysis is presented as follows

sRNAOutput

- **miRBase:** conserved sequences mapped to miRBase.

- **Plots:** Different plots including the length distribution, nucleotide position distribution etc.
- **NovelPrediction:** newly predicted sRNAs
- **HostMap:** tab-delimited information from the SAM files of the sequences mapped to the host.
- **Reports:** Contains different reports in tab-delimited files such as the reads mapping statistics, sequence length distribution information of all the samples etc.
- **ExpressionProfile:** Contains expression profile information for different categories of the sequence data
 - **miRBase:** conserved sequences mapped to miRBase
 - **commonSequences:** sequences common to all the samples
 - **NovelPrediction:** Newly predicted sRNAs
 - **Mock:** sequences in the Mock sample
 - **CaMV:** sequences in the CaMV-infected samples
 - **CMV:** sequences in the CMV-infected samples
- **AlignmentMaps:** when selected in the config file will contain the SAM files of the sequence maps to the respective reference sequences.

APPENDIX

1. Generate chromosome length file for the host genome

Samtools (<http://www.htslib.org/>) is required to generate this file. With *Samtools* installed on the machine use the following commands:

```
samtools faidx host-genome.fasta
```

This will produce a file named “*host-genome.fasta.fai*”

```
cut -f 1,2 host-genome.fasta.fai > chromosome_length.txt
```

copy the “chromosome_length.txt” file to the “data” folder.

2. PROCEDURE ON MACBOOK WITH DOCKER

1. Open the Terminal on your Macbook
2. Get the Ds-Seq image from Docker Hub (It is expected that you already have the Docker engine installed and running on your machine).

```
docker pull cephase/ds-seq
```

```
lagunju@Temitayos-MacBook-Air ~ % docker pull cephase/ds-seq
Using default tag: latest
latest: Pulling from cephase/ds-seq
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
10656968028e: Pull complete
c7b0babf8006: Pull complete
e815c257bd61: Pull complete
8d32a38f20e9: Downloading [=====>] 26.42MB/172.5MB
1acc985748e7: Downloading [=====>] 16.86MB/30.96MB
a4e6e47fd044: Downloading [=====>] 12.4MB/57.1MB
b50d0b786bcf: Waiting
102d326420f1: Waiting
205d871fa631: Waiting
bc316ba29d5d: Waiting
01c82f4c770b: Waiting
1d84e85d6df1: Waiting
4d18ba1c02a0: Waiting
00187e3105ce: Waiting
ec91f2a23727: Waiting
ea7159a7fb63: Pulling fs layer
```

3. Get the image ID after the download is complete

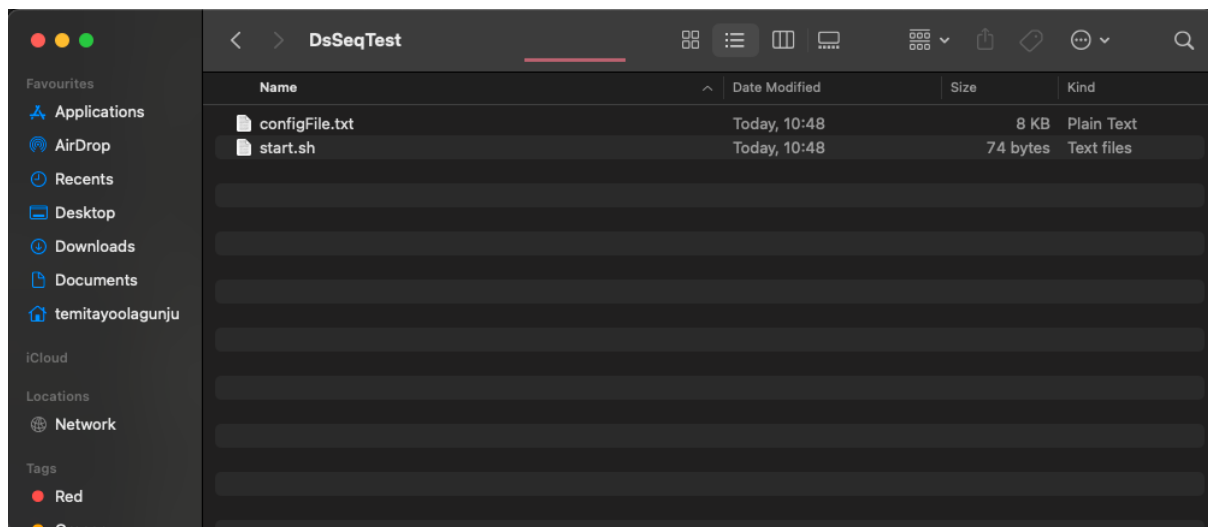
```
docker image ls
```

```
10656968028e: Pull complete
c7b0babf8006: Pull complete
e815c257bd61: Pull complete
8d32a38f20e9: Pull complete
1acc985748e7: Pull complete
a4e6e47fd044: Pull complete
b50d0b786bcf: Pull complete
102d326420f1: Pull complete
205d871fa631: Pull complete
bc316ba29d5d: Pull complete
01c82f4c770b: Pull complete
1d84e85d6df1: Pull complete
4d18ba1c02a0: Pull complete
00187e3105ce: Pull complete
ec91f2a23727: Pull complete
ea7159a7fb63: Pull complete
Digest: sha256:866e4d9341cdc4f6f6d54fe3178435281812508984ddfbce7ac8191c2f979adb
Status: Downloaded newer image for cephase/ds-seq:latest
docker.io/cephase/ds-seq:latest
temitayoolagunju@Temitayos-MacBook-Air ~ % docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
cephase/ds-seq      latest     6fb834097cab  26 minutes ago  1.78GB
docker/getting-started latest     720f449e5af2  8 months ago  27.2MB
temitayoolagunju@Temitayos-MacBook-Air ~ %
```

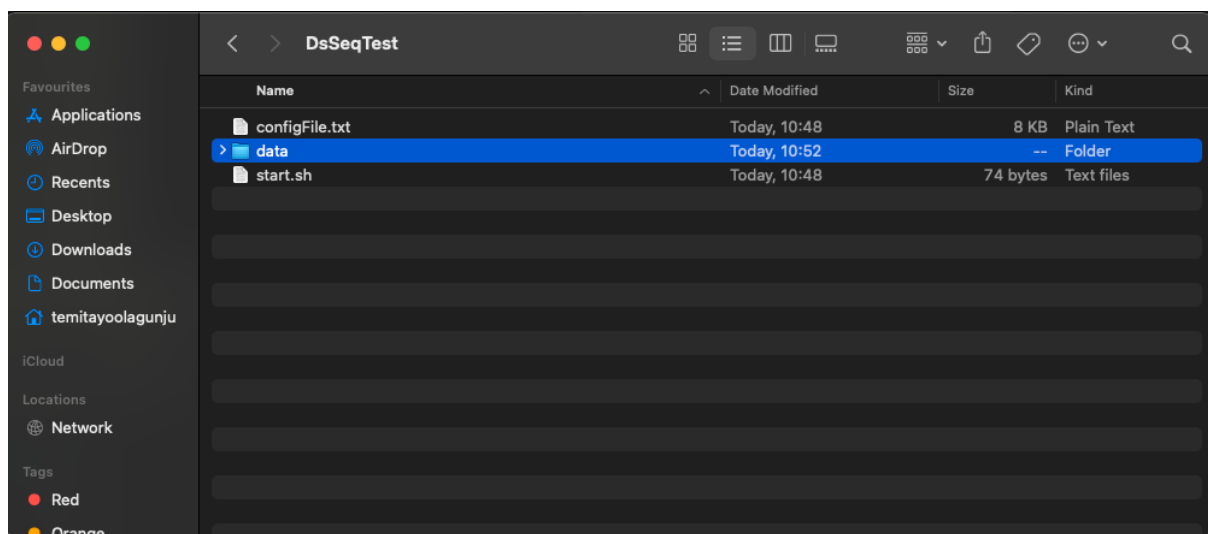
4. On Finder, create a folder where you would want your analysis done. "DsSeqTest".

Applications	> DsSeqTest	Today, 10:42	-- Folder
AirDrop	> Temitayo	Today, 08:05	-- Folder

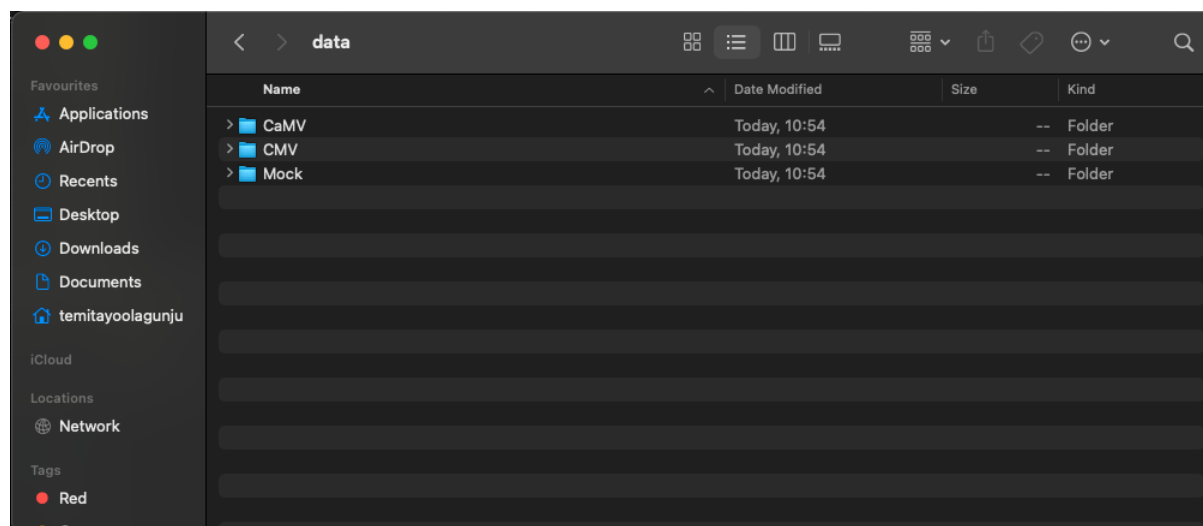
- Open the folder and place the two required files configFile.txt and start.sh there



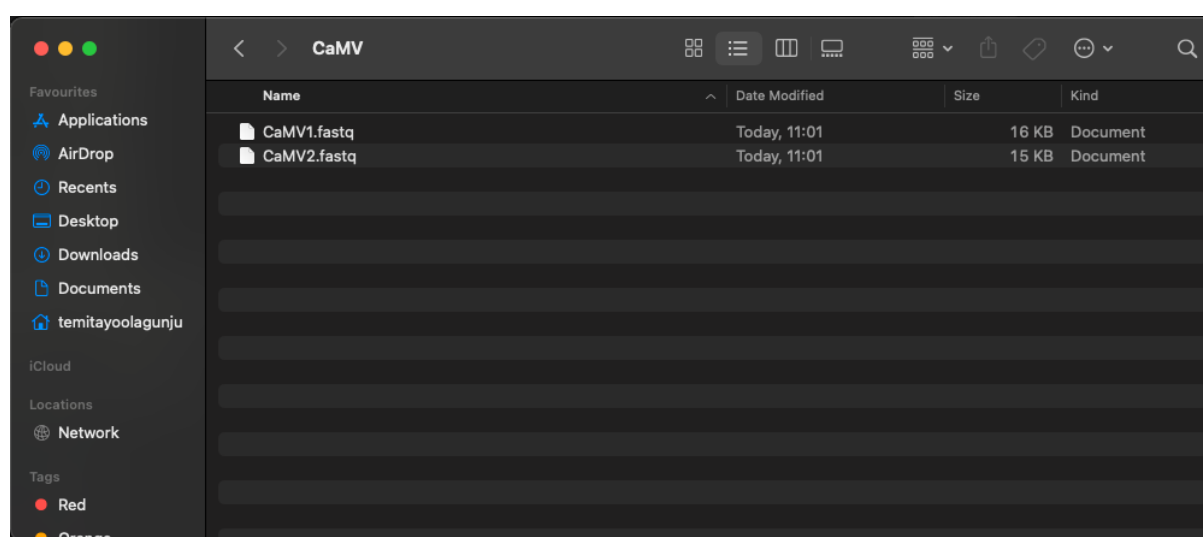
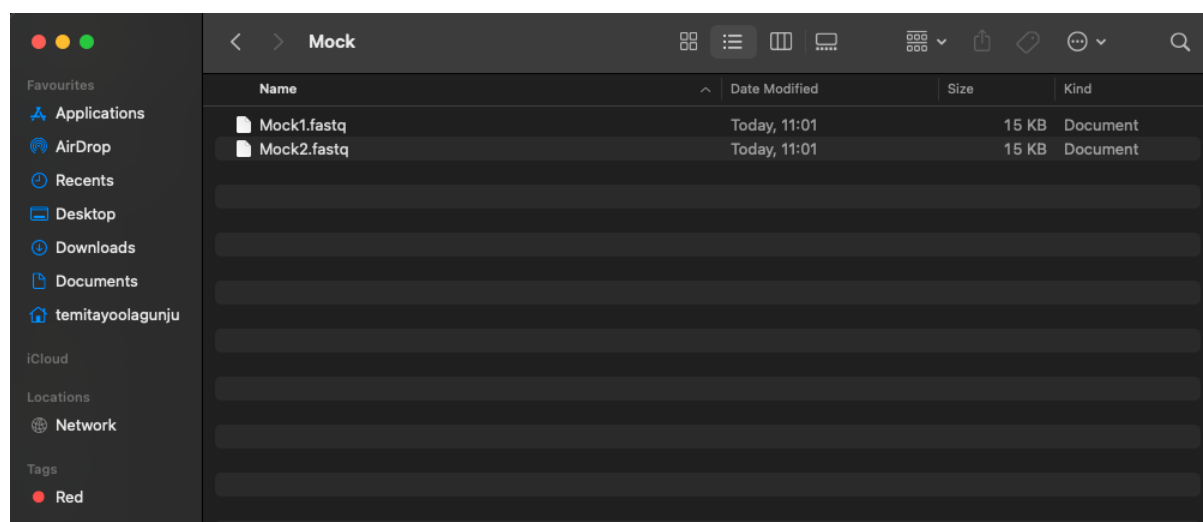
- Create "data" folder to hold all the data required for the analysis, and open it.

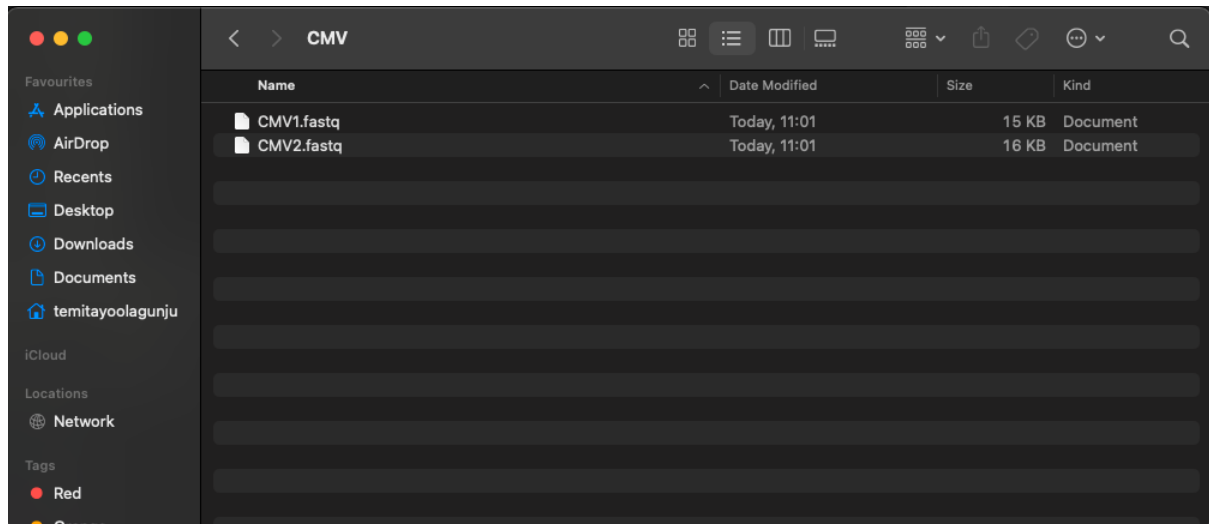


- Create folders for the samples, based on the experiment. In this illustration, three samples are to be analyzed – Mock, CMV, and CaMV.

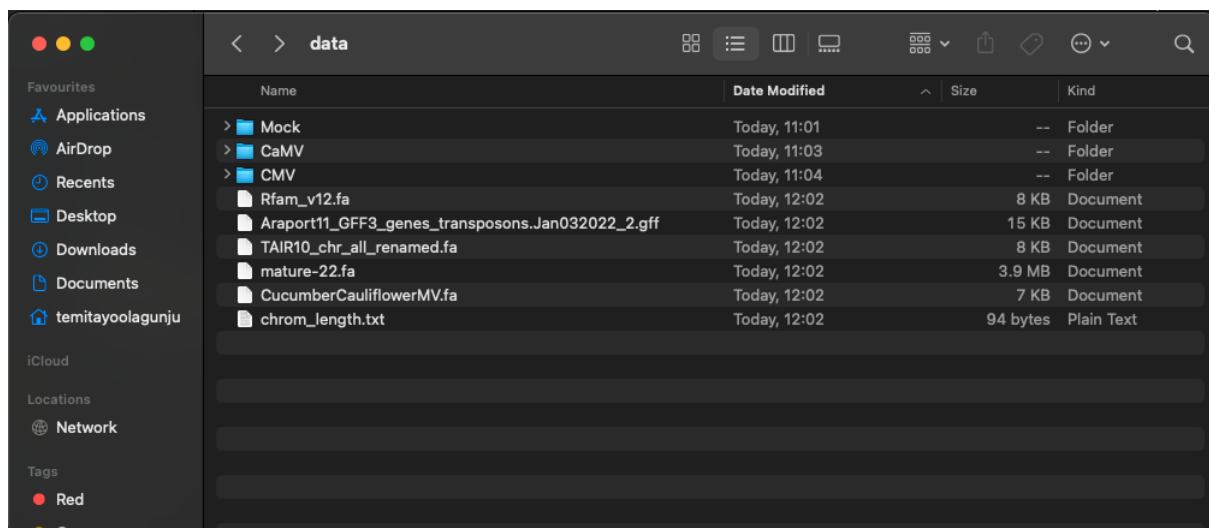


8. Put the respective raw sequence files in the folders





9. Add other files to the “data” folder



10. Go back one step to edit the “configFile.txt” configuration file to reflect the filenames placed in the data folder as well as other analysis parameters.

11. Go back to your Terminal window and navigate to the folder you created in step 4. In this case, the path to the folder is “/Users/temitayoolagunju/Documents/DsSeqTest”

12. Use the command

```
docker run -it -v /path/to/container/folder:/pipeline/ dockerImageID
```

replace “/path/to/container/folder” with the path to your folder created in step 4 and “dockerImageID” with the image ID you obtained in step 3.

The correct command should look like this


```
docker run -it -v /Users/temitayoolagunju/Documents/DsSeqTest:/pipeline/  
6fb834097cab
```

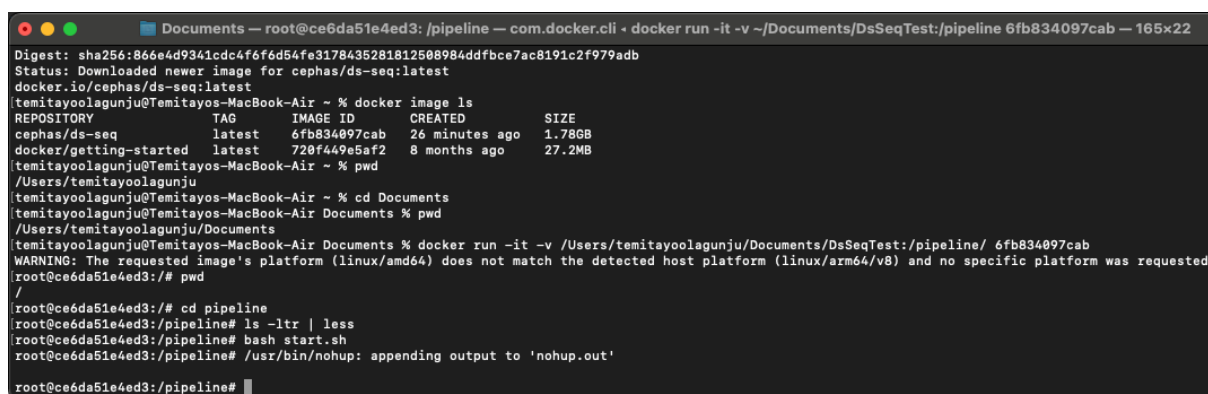
13. If this is successful, you will be taken to a prompt ending with “#”.

Enter the following command:

```
cd pipeline
```

14. Then type the following command to start the pipeline

```
bash start.sh
```



```
Documents — root@ce6da51e4ed3: /pipeline — com.docker.cli - docker run -it -v ~/Documents/DsSeqTest:/pipeline 6fb834097cab — 165x22  
Digest: sha256:866e4d9341cdc4f6f6d54fe3178435281812508984ddfbce7ac8191c2f979adb  
Status: Downloaded newer image for cephas/ds-seq:latest  
docker.io/cephas/ds-seq:latest  
temitayoolagunju@Temitayos-MacBook-Air ~ % docker image ls  
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE  
cephas/ds-seq        latest      6fb834097cab  26 minutes ago  1.78GB  
docker/getting-started latest      720f449e5af2  8 months ago  27.2MB  
temitayoolagunju@Temitayos-MacBook-Air ~ % pwd  
/Users/temitayoolagunju  
temitayoolagunju@Temitayos-MacBook-Air ~ % cd Documents  
temitayoolagunju@Temitayos-MacBook-Air Documents % pwd  
/Users/temitayoolagunju/Documents  
temitayoolagunju@Temitayos-MacBook-Air Documents % docker run -it -v /Users/temitayoolagunju/Documents/DsSeqTest:/pipeline/ 6fb834097cab  
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested  
root@ce6da51e4ed3:/# pwd  
/  
root@ce6da51e4ed3:/# cd pipeline  
root@ce6da51e4ed3:/pipeline# ls -ltr | less  
root@ce6da51e4ed3:/pipeline# bash start.sh  
root@ce6da51e4ed3:/pipeline# /usr/bin/nohup: appending output to 'nohup.out'  
root@ce6da51e4ed3:/pipeline#
```

15. When the analysis commences, you should see the following feedback on the screen

“/usr/bin/nohup: appending output to 'nohup.out’”

The folder holding your analysis will start to be populated with files. If there is an error, the file “analysisStopped.txt” will be created. When the analysis completes successfully, the file “analysisCompleted.txt” would be created. A view of the output folder is shown below.

