FHGAON Tutorial

FHGAON runs on all systems using docker. So, make sure to install docker on your system and while using the FHGAON pipeline keep the docker application open.

Note: Windows users need to install WSL (windows subsystem for Linux) before installing the docker. Instructions to install WSL2 can be found here.

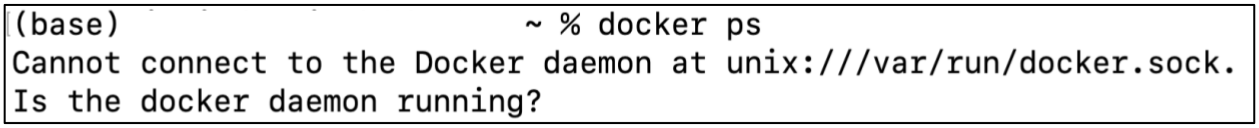
Docker download link: https://www.docker.com/products/docker-desktop/

Docker Desktop

The #1 containerization software for developers and teams

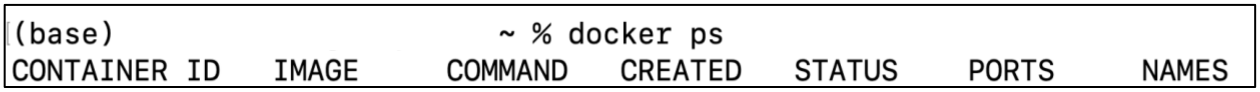
Your command center for innovative container development

After installation of Docker, check if the application is running properly using the following command:



```
(base)                          ~ % docker ps
Cannot connect to the Docker daemon at unix:///var/run/docker.sock.
Is the docker daemon running?
```

If you get the above error, then docker engine is not running. Therefore, start the docker application first and rerun the above command.



```
(base)                          ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Now docker is up and running. You can move to run the FHGAON using following instructions.

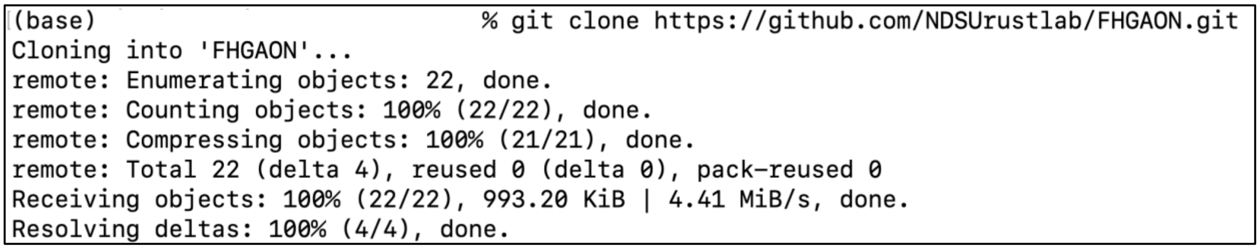
Requirements:

- FHGAON need at least 12 CPUS and 64 GB RAM to run.
- You don’t need to install any software except for Docker and python3.

FHGAON instructions:

Step I: Clone the GitHub repository to your local machine using following commands:

git clone https://github.com/NDSUrustlab/FHGAON.git



```
(base)                              % git clone https://github.com/NDSUrustlab/FHGAON.git
Cloning into 'FHGAON'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 22 (delta 4), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (22/22), 993.20 KiB | 4.41 MiB/s, done.
Resolving deltas: 100% (4/4), done.
```

Move into the FHGAON directory.

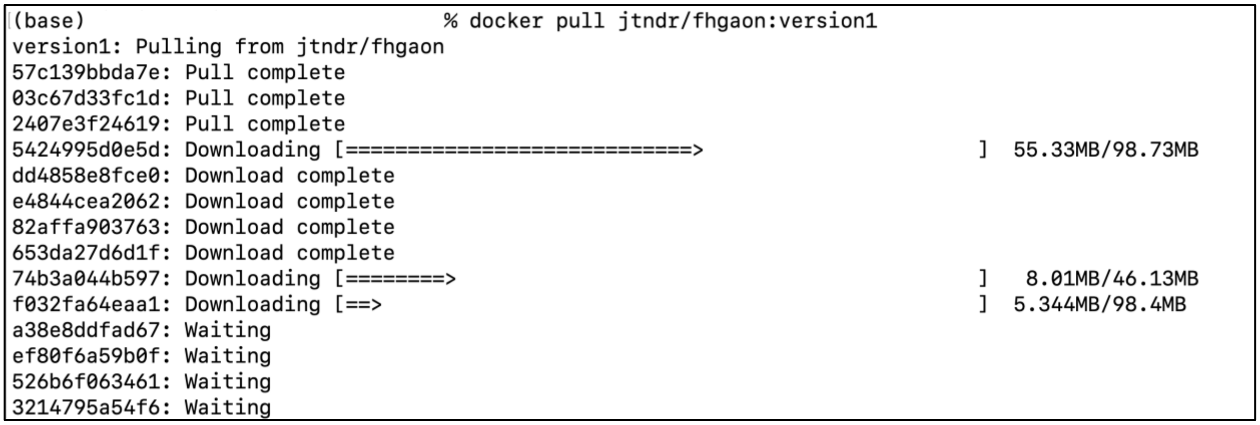
cd FHGAON

Step II: Copy the raw pod5 files folder or basecalled fastq file to FHGAON folder.

Step III: Pulling docker container from docker hub.

docker pull jtndr/fhgaon:version1

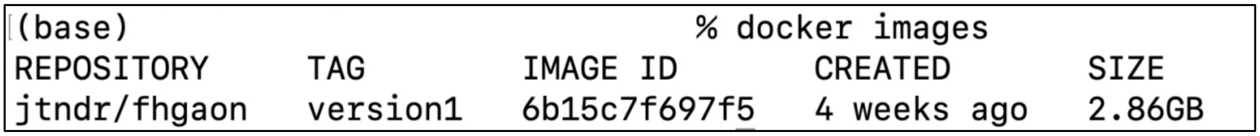
It will start downloading the docker from docker hub.



```
(base)                                  % docker pull jtndr/fhgaon:version1
version1: Pulling from jtndr/fhgaon
57c139bbda7e: Pull complete
03c67d33fc1d: Pull complete
2407e3f24619: Pull complete
5424995d0e5d: Downloading [============================>                      ]  55.33MB/98.73MB
dd4858e8fce0: Download complete
e4844cea2062: Download complete
82affa903763: Download complete
653da27d6d1f: Download complete
74b3a044b597: Downloading [========>                                          ]   8.01MB/46.13MB
f032fa64eaa1: Downloading [==>                                                ]  5.344MB/98.4MB
a38e8ddfad67: Waiting
ef80f6a59b0f: Waiting
526b6f063461: Waiting
3214795a54f6: Waiting
```

When the download is complete, check the docker container using:

docker images



```
(base)                                      % docker images
REPOSITORY      TAG         IMAGE ID       CREATED        SIZE
jtndr/fhgaon    version1    6b15c7f697f5   4 weeks ago    2.86GB
```

Step IV: Running the pipeline for whole genome assembly of nanopore data

Important things:

- Make sure you are inside the FHGAON folder and your pod5 folder or basecalled fastq file is copied in the FHGAON folder.
- Provide the estimated genome size.

Mode 1: If basecalling is already done, run the pipeline as follows:

```
docker run -v $(pwd):/app/ jtndr/fhgaon:version1 -q input_file.fastq -g estimated_genome_size -t number_of_threads -p project_name -o output_folder_name
```

change the green text with your file name and genome size.

Mandatory options (in green color):

-q : provide the name of your input fastq file

-g : provide estimated genome size

Optional:

-t : number of threads (default is 12)

-p : your project name

-o : output folder name you want

Mode 2: If basecalling is required, run the pipeline as follows:

```
python3 ./scripts/fhgaon.py -r pod5/ -m basecalling_model -b basecalling_type | docker run -v $(pwd):/app/ jtndr/fhgaon:version1 -g estimated_genome_size -t number_of_threads -p project_name -o output_folder_name
```

Mandatory option (in green color):

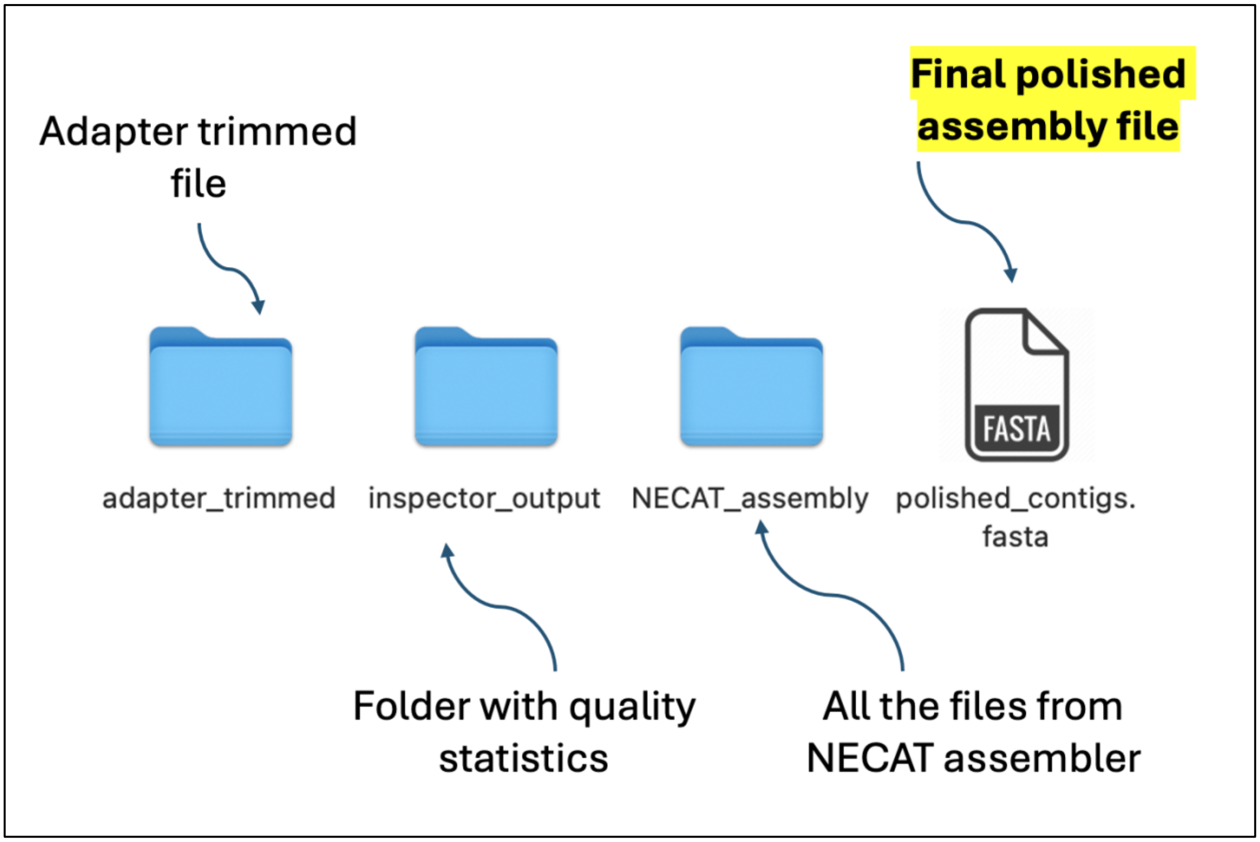
-r : folder which contain pod5 files

-m : provide the basecalling model (check this link for more information on basecalling models)

Optional:

-b : basecalling type (simplex (default) or duplex)

After successful pipeline run, you will find following folders and final assembly file in your output folder (it will be present in the FHGAON folder in which you copied your file in the start)


Final polished assembly file
Adapter trimmed file
FASTA
adapter_trimmed
inspector_output
NECAT_assembly
polished_contigs.fasta
Folder with quality statistics
All the files from NECAT assembler