

Microbiota amplicOn  
CHaracterization Implement

**MOCHI**

**User Guide**

# TABLE OF CONTENTS

## **Installation of local service**

MacOS.....	3
(A) Install Docker .....	3
(B) Start MOCHI service.....	4
(C) Stop MOCHI service.....	6
Window 10.....	7
(A) Install Docker .....	7
(B) Start MOCHI service.....	8
(C) Stop MOCHI service .....	10
Linux.....	11
(A) Install Docker .....	11
(B) Start MOCHI service .....	11
(C) Stop MOCHI service .....	11
Additional Information .....	12

# INSTALLATION OF LOCAL SERVICE

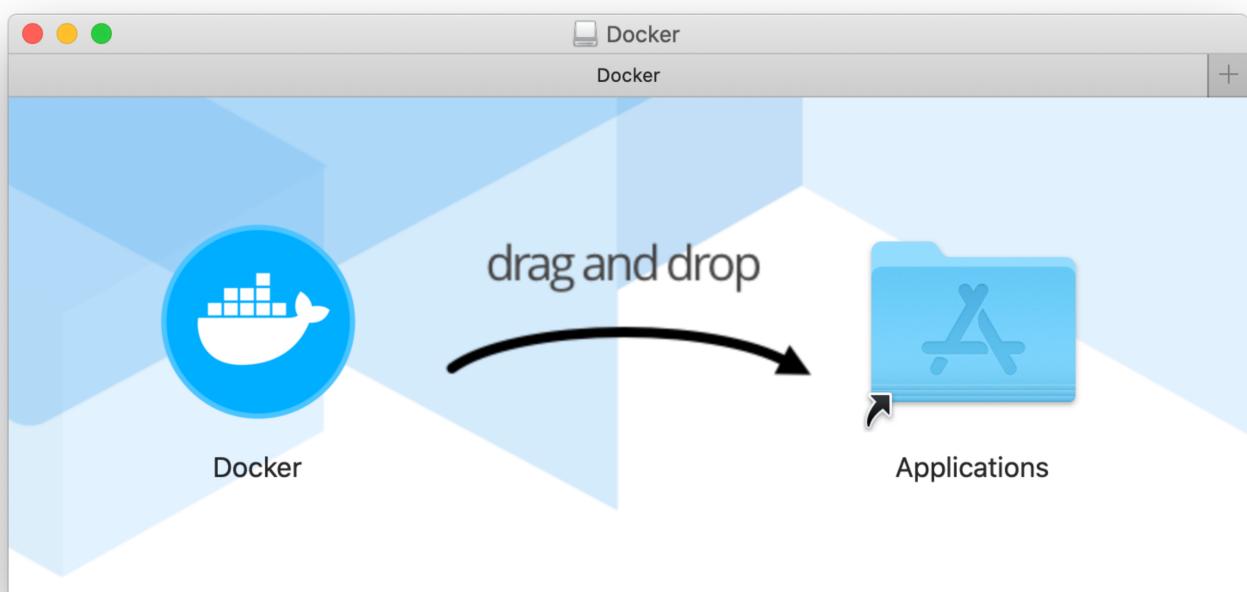
# MacOS

## (A) Install Docker

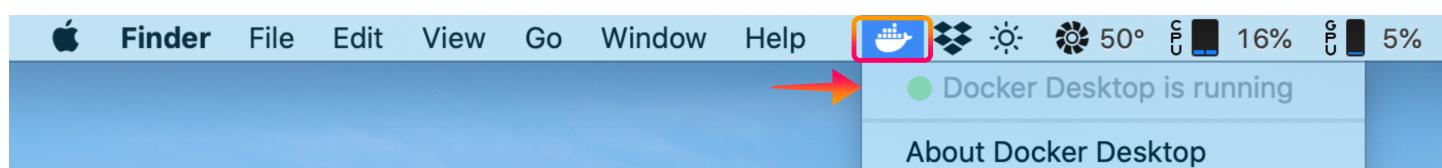
1. Download [Docker Desktop](#).



2. Open “**Docker.dmg**” file. Drag **Docker Desktop** app to the Applications folder.



3. Start the Docker service by double clicking the Docker app. Wait for a few seconds to load, the docker icon should appear in the status bar.



## (B) Start MOCHI service

1. Download “**docker-compose.yml**” from [NCTU website](#), and place it under a folder named “**MOCHI**”.



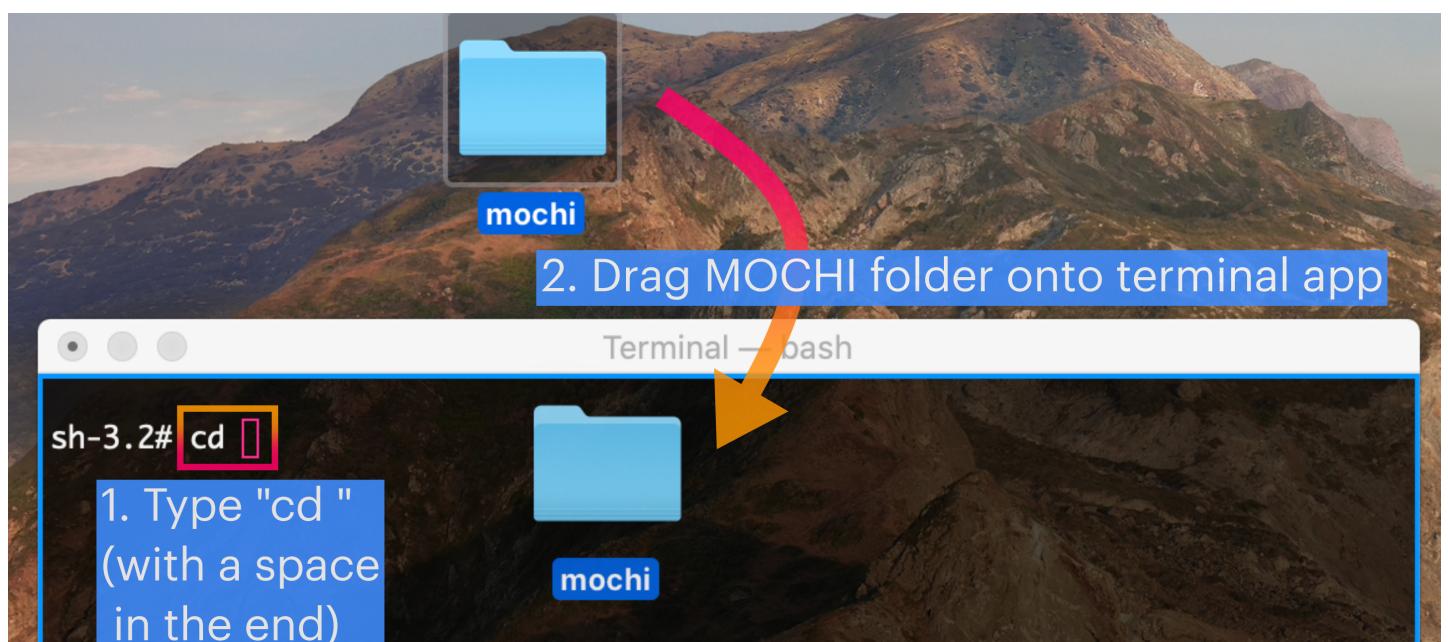
2. Open **Terminal** app. (Press “Command + Space” to open spotlight and type “terminal”.)



3. Run `cd /path/to/MOCHI` to navigate under MOCHI directory.

```
sh-3.2# cd /Users/Mac/Desktop/mochi
```

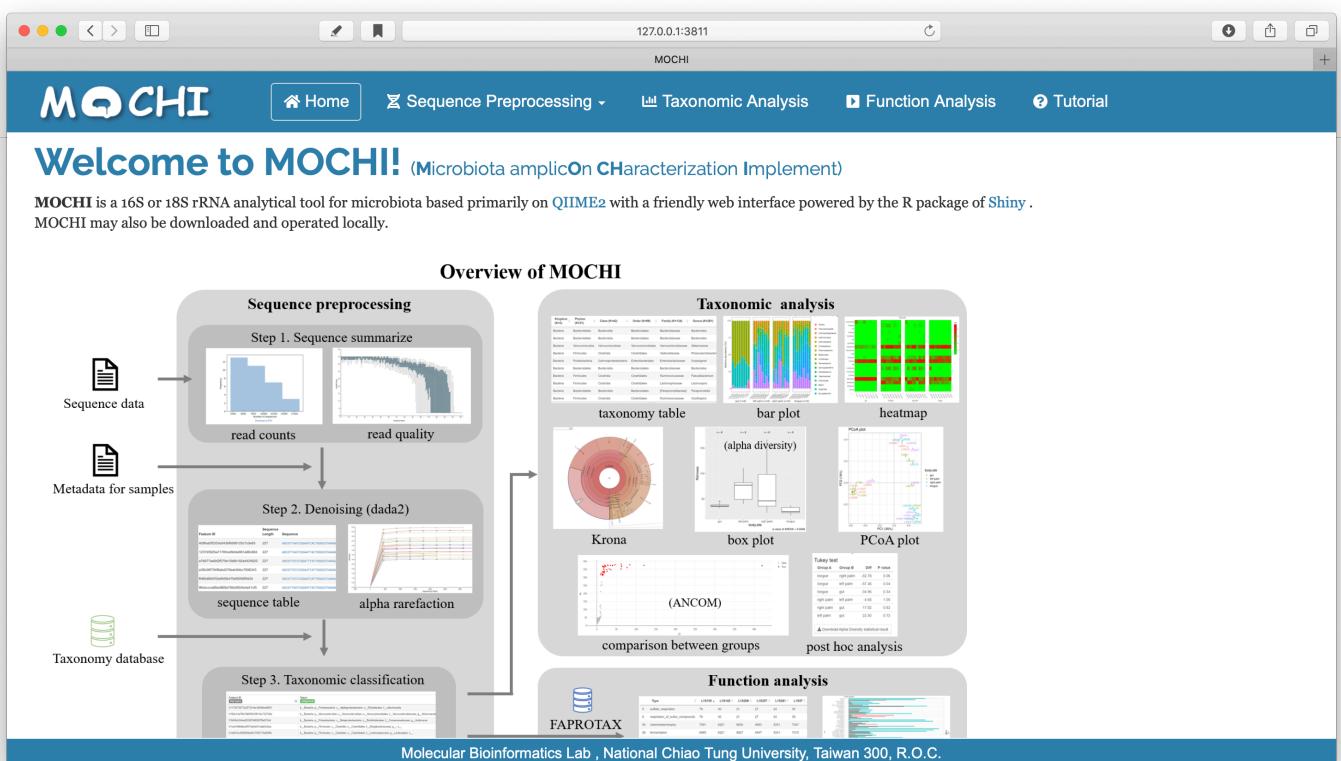
Hint: If you do not know the folder path of MOCHI, type `cd` with a space, and drag MOCHI folder onto the terminal window. The folder path should appear on the terminal automatically.



4. Run `docker-compose up -d` to download and start the MOCHI image from Docker Hub.  
 (The download process will only run during the first setup. The size of MOCHI image is around 8 GB, and the running time depends on the download speed.)

```
[powang@MacBook-Pro--PoWang mochi % docker-compose up -d
WARNING: Some services (mochi_server) use the 'deploy' key, which will be ignored. Compose does not support 'deploy' configuration - use `docker stack deploy` to deploy to a swarm.
Creating network "mochi_default" with the default driver
Pulling mochi_server (dockerjjz/mochi_local:...)...
latest: Pulling from dockerjjz/mochi_local
f15005b0235f: Pull complete
1901fd813023: Pull complete
a92940affedf: Pull complete
dbebda29cb22: Pull complete
3c63b26b92fd: Pull complete
e0c15c0b4e0b: Pull complete
Digest: sha256:1501a145eb826f9f799239964eb064170fdc5be8abcd7b04fd9a61c888b9dee
Status: Downloaded newer image for dockerjjz/mochi_local:latest
Creating mochi_server ... done]
```

5. Open the browser, and type **127.0.0.1:3811** on the address bar. A MOCHI interactive webpage is now available to begin your analysis.



## (C) Stop MOCHI service

1. Run **docker-compose down** to stop MOCHI.

```
[powang@MacBook-Pro--PoWang mochi % docker-compose down]  
[  
WARNING: Some services (mochi_server) use the 'deploy' key, which will be ignored. Compose does not support 'deploy' configuration - use `docker stack deploy` to deploy to a swarm.  
Stopping mochi_server ... done  
Removing mochi_server ... done  
Removing network mochi_default
```

⚠ Stopping MOCHI service or restarting the computer will remove all the data without saving!  
Remember to export the analyzed data in advance.

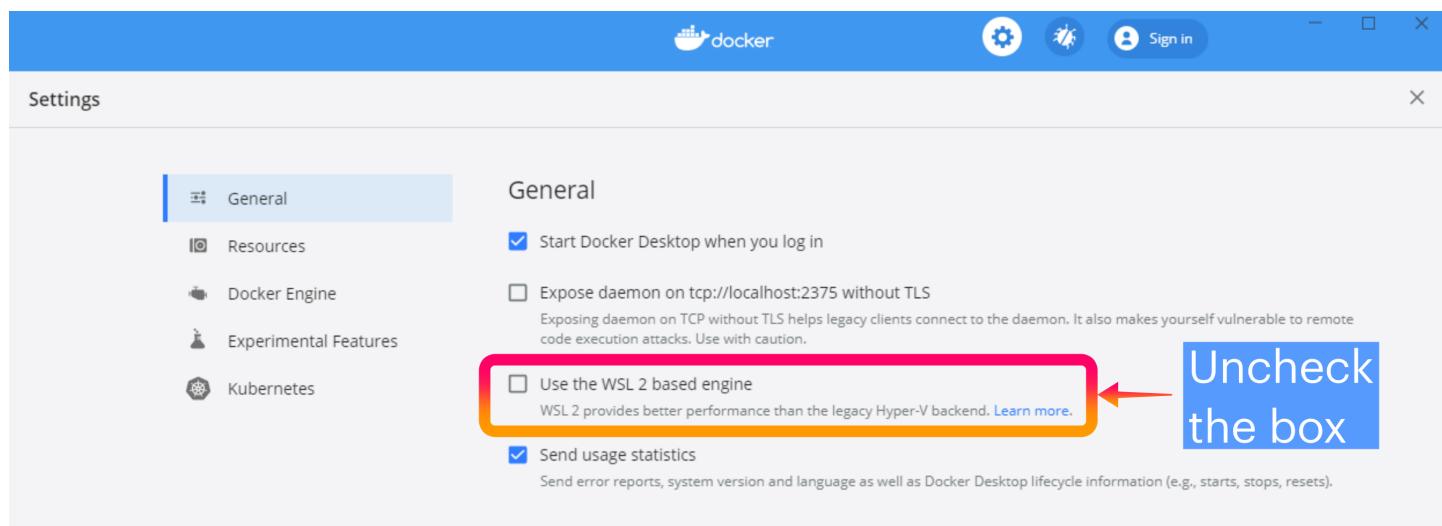
# Window 10

## (A) Install Docker

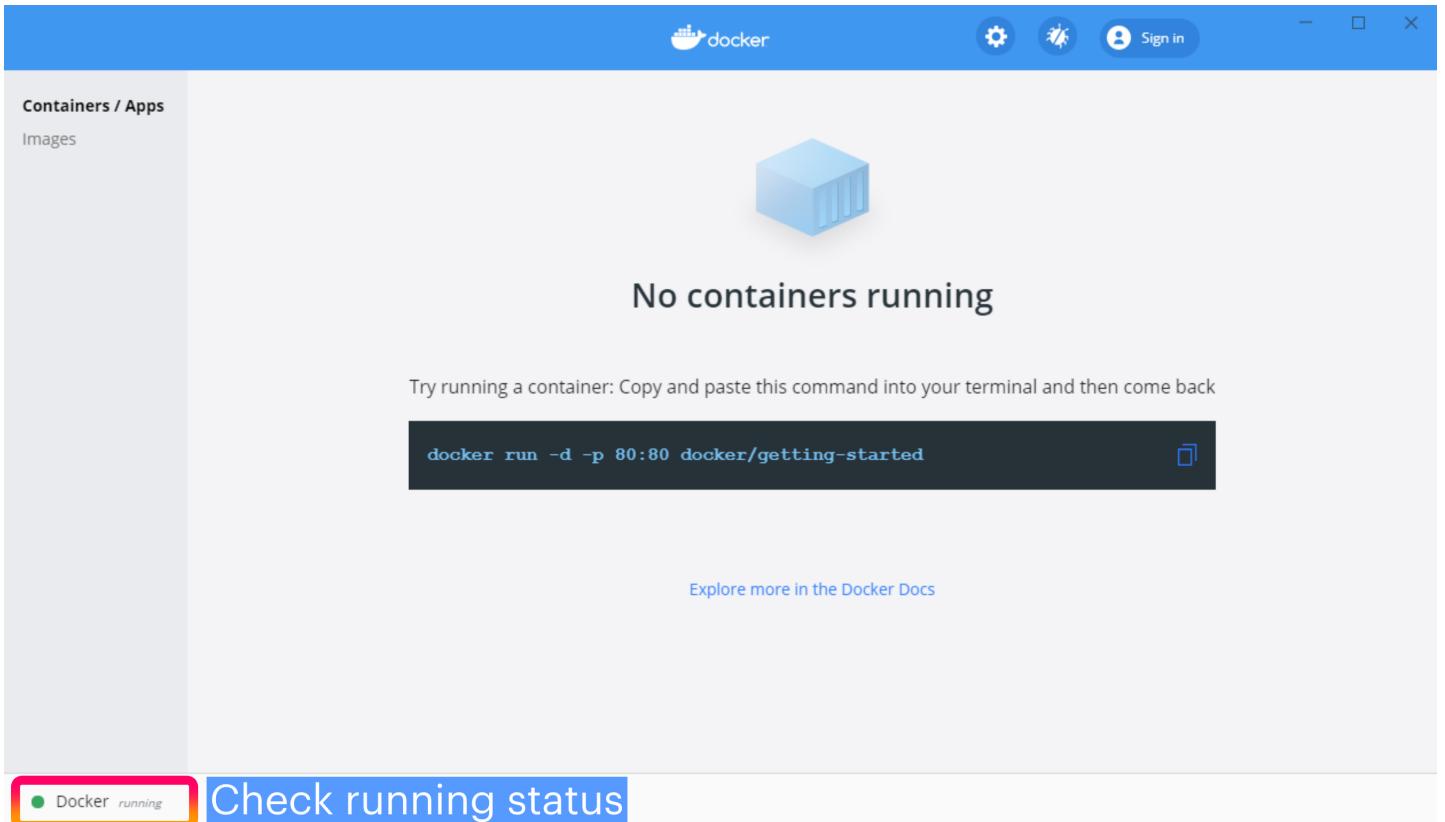
1. Download [Docker Desktop](#).



2. Open "**Docker Desktop Installer.exe**" and follow the instruction to setup.
3. Start docker service by **clicking the Docker icon**. (By default, the WSL2 engine is used to speed up the performance, but it is not installed along with the docker package. Please follow the [instructions](#) to install WSL2 or turn off the engine in the settings panel.)



Hint: If the docker service has been successfully turned on, a green indicator will appear on the bottom-left in the docker app.



Check running status

## (B) Start MOCHI service

1. Download “**docker-compose.yml**” from [NCTU website](#), and place it under a folder named “**MOCHI**”.

The screenshot shows the MOCHI installation page. It has a navigation bar with 'Home', 'Sequence Preprocessing', 'Taxonomy Analysis', 'Function Analysis', 'Tutorial', 'MOCHI Installation' (which is highlighted), and 'User Results'. The main content area is titled 'MOCHI Installation' and contains the text 'Download docker-compose.yml and follow the installation tutorial.' Below this is a button labeled 'docker-compose.yml' with a download icon. An orange arrow points from the text 'Check running status' in the previous screenshot to this download button.

2. Open **MOCHI folder**, type “**powershell**” in the address bar, and press enter. This will open a command-line shell under MOCHI directory.

The screenshot shows a Windows file explorer window. The address bar has 'MOCHI' and 'powershell' entered. An orange arrow points from the text 'Check running status' in the previous screenshot to the 'powershell' entry in the address bar. The file list shows a file named 'docker-compose.yml' with a size of 1 KB. The file list table has columns for 'Name', 'Date modified', 'Type', and 'Size'.

Name	Date modified	Type	Size
docker-compose.yml	12/14/2020 5:55 PM	YML File	1 KB

3. In Powershell, run `docker-compose up -d` to download and start the MOCHI image from Docker Hub. (The download process will only run at the first setup. The size of MOCHI image is around 8 GB, and the running time depends on the download speed.)

```
PS C:\Users\dodolab\Desktop\MOCHI> docker-compose up -d
Creating network "mochi_default" with the default driver
Pulling mochi_server (dockerjjz/mochi_local_version:)...latest: Pulling from dockerjjz/mochi_local_version
f15005b0235f: Pull complete
1901fd813023: Pull complete
a92940affedf: Pull complete
dbebda29cb22: Pull complete
3c63b26b92fd: Pull complete
e4191a297544: Pull complete
Digest: sha256:794909d921dfe9cc55edba44a3fe66701e43845c5c5578bf1e194064e071cbe
Status: Downloaded newer image for dockerjjz/mochi_local_version:latest
Creating mochi_server_version ... done
```

4. Open the browser, and type **127.0.0.1:3811** on the address bar. A MOCHI interactive webpage is now available to begin your analysis.

The screenshot shows the MOCHI web application running in a browser. The title bar says "MOCHI". The address bar shows "127.0.0.1:3811". The main header has tabs for "Home", "Sequence Preprocessing", "Taxonomic Analysis", "Function Analysis", and "Tutorial". Below the header, a large blue banner says "Welcome to MOCHI! (Microbiota amplicOn CHaracterization Implement)". It describes MOCHI as a 16S or 18S rRNA analytical tool based on QIIME2, powered by Shiny, and available locally. The main content area is titled "Overview of MOCHI". It shows a flowchart of the analysis steps:

- Sequence preprocessing:** Starts with "Sequence data" (file icon). It includes "Step 1. Sequence summarize" (read counts, read quality) and "Step 2. Denoising (dada2)" (sequence table, alpha rarefaction).
- Taxonomic analysis:** Starts with "Metadata for samples" (file icon). It includes "taxonomy table", "bar plot", "heatmap", "Krona", "(alpha diversity)", "box plot", "PCoA plot", "comparison between groups", and "post hoc analysis".
- Function analysis:** Starts with "Taxonomy database" (database icon). It includes a "FAPROTAX" section with a table and a chart.

At the bottom of the page, it says "Molecular Bioinformatics Lab , National Chiao Tung University, Taiwan 300, R.O.C."

## (C) Stop MOCHI service

1. Run **docker-compose down** to stop MOCHI.

```
PS C:\Users\dodolab\Desktop\MOCHI> docker-compose down
Stopping mochi_server_version ... done
Removing mochi_server_version ... done
Removing network mochi_default
```

⚠ Stopping MOCHI service or restarting the computer will remove all the data without saving!  
Remember to export the analyzed data in advance.

# Linux

---

## (A) Install Docker

- ▶ Please follow the [official guidance](#) to setup and start docker engine on Ubuntu.

## (B) Start MOCHI service

1. Download “**docker-compose.yml**” from [NCTU website](#), and place it under a folder named “**MOCHI**”.



2. Navigate to the MOCHI directory.

```
$ cd /path/to/place/MOCHI
```

3. Download and start the MOCHI image from Docker Hub. (The download process will only run during the first setup. The size of MOCHI image is around 8 GB, and the running time depends on the download speed.)

```
$ docker-compose up -d
```

4. Open the browser, and type **127.0.0.1:3811** on the address bar. A MOCHI interactive webpage is now available to begin your analysis.

## (C) Stop MOCHI service

1. Run the following command to stop MOCHI service.

```
$ docker-compose down
```

⚠ Stopping MOCHI service or restarting the computer will remove all the data without saving them! Remember to export the analyzed data in advance.

# Additional Information

---

- \* If you wish to restart your computer and resume working progress later, please run **docker-compose stop**. This will create a paused MOCHI service, and can be restored with command **docker-compose up -d**. Please be cautious that interrupting the MOCHI service when an analysis is running will not save the running process. You will need to re-run the analysis later.
- \* Remember to navigate under the MOCHI folder (where “**docker-compose.yml**” is located) before starting, stopping or restoring MOCHI service.
  - ▶ For Mac, Please see [MacOS / step B-3](#).
  - ▶ For Window 10, Please see [Window 10 / step B-2](#).
  - ▶ For Linux, Please see [Linux / step B-2](#).

- \* After installing MOCHI, two folders are created under the MOCHI folder:

## 1. **seqs\_folder**

- ▶ This folder is used to store user’s sequence data which will be loaded in “Sequence preprocessing - Sequence summary”.
- ▶ The format of sequence data need to be fastq.gz.
- ▶ The filename of sequence data need to satisfy [Casava 1.8 demultiplexed format](#).

## 2. **taxa\_database**

- ▶ This folder is used to store taxonomy database, such as Greengenes, Silva and PR2. Please see “Sequence preprocessing - Taxonomy classification”.

- \* If you wish to change the default path to the sequence data and the taxonomic database, open the “**docker-compose.yml**” file, replace the texts “**./seqs\_folder**” and “**./taxa\_database**” with new paths to the sequence data and the taxonomic database, respectively. Relative path is allowed.