# Snaq Documentation

## Contents



## 1 Background

This software is an excellent snakemake tool to run the analysis efficiently and productively.

# 2 Installation

## 2.1 Snaq installation

### 2.1.1 Clone github repository

```
$ git clone https://attayeb/snaq.git
```

or download the zipped release file, [this part will be explained when the repository is made public]

### 2.1.2 Download the latest release

the latest release of this snakemake pipeline is in github repository: `https://github.com/attayeb/snaq/releases`

```
$ wget https://github.com/attayeb/snaq/releases/tag/untagged-77b9a20481d70db1aff2
```

## 2.2 Snakemake installation

### 2.2.1 Linux, Mac, and windows using Windows Linux Subsystem

The easiest way to install snakemake is by using Conda; however, other ways are mentioned in detail in snakemake website `https://snakemake.readthedocs.io/en/stable/getting_started/installation.html`

Mamba is a lighter conda interface that is recommended by Snakemake developers to be installed.

```
# install mamba using conda-forge
$ conda install -n base -c conda-forge mamba
$ conda activate base
$ mamba create -c conda-forge -c bioconda -n snakemake snakemake
$ conda activate snakemake
```

### 2.2.2 Using docker

Docker can be used in Windows, Linux, and MAC. You need to have the docker client installed in your system and pull the snakemake image from dockerhub. The snakemake image is maintained by the same guys who created the snakemake. For more details, you can check this link `https://hub.docker.com/r/snakemake/snakemake`

```
$ docker run -it -v C:\snaq:/work -w /work \
     snakemake/snakemake {command}
```

`docker run`: docker main command

`-it`: This means interactive and allow sending command at the end.

`-v`: will map [Host directory]:[inside container directory]

`-w`: sets the working directory inside the container, and in that directory snakemake commands will be executed.

`snakemake/snakemake`: The docker image that is going to run.

`{command}`: Whatever command you write here will be executed inside the container after it is built.

Docker will run an ubuntu system inside a container. It will map the user folder in which he has snaq installed to the work folder inside the container system, will run the analysis using the container system, and save the output, including the installation of Qiime2, other tools, and databases in the host directory. The user can use the results in the host without any issue.

# 3 Preparing data for analysis

## 3.1 Input data structure

Snaq can analyze paired-end fastq files from Illumina sequencers. They should be saved in a new folder inside `results/` folder in the main snaq folder

## 3.2 Data set concept

Because this pipeline allows automatic installation of applications and downloading of databases, the size of the whole pipeline is relatively large. Therefore, we decided that it is better to give the pipeline the ability to analyze multiple data sets, which should be stored in separate folders in the `/results` folder.

# 4 Snakemake commandline

All commands are executed by snakemake Snakemake requires two important parameters `--cores {Number of cores}` and `--use-conda` then the target file name to be produced. For more details of snakemake command-line tools, please check: `https://snakemake.readthedocs.io/en/stable/`

If the user already has all the requirements for the pipelines installed, he can omit `--use-conda` and run the pipeline in the environment that he prepared. however, the number of cores

```
$ snakemake --use-conda --cores 10 results/AB/AB.qza
```

`--use-conda`: To tell snakemake to use the rule specific conda environment. ]

`--cores`: number of allowed parallel jobs by the user.]

in the case of docker

```
$ docker run -it -v C:\snaq:/work -w /work snakemake/snakemake \
      snakemake --use-conda --cores 10  results/AB/AB.qza
```

Finally, the submitted command should be like this:

# 5 Stages

## 5.1 Import data

Importing data takes two steps. First step is creating a manifest file to read all the file names in the specified folder and list them in a manifest file following the QIIME2 requirement. This manifest file is used in the second step to identify the files that need to be imported to form QIIME zipped artifact (QZA) of the data.

fastq files have to be saved in `data/AB/` folder.

### 5.2 Trimming

The input of this step is qza of a sequence data, `results/AB/AB.qza`, then we add the steps needed depending on the users' requirement using + sign. for example if the user wants to use fastp to crop 10 nucleotides from R1 and 20 from R2, then the command should be added to the file name +`fp-f10-r20`, the target becomes: `results/AB/AB+fp-f10-r20crop.qza`. The output will be a qza file with sequence data inside. If another step needs to be done, it should follow the same way of giving the command. If bbduk needs to be used for quality trimming, then + and the second command need to be added in the form of `bb-t{threshold}`. If the threshold is 18 then the command should be `bb18t`, which makes the target: `results/AB/AB+fp-f10-r20crop+bb18t.qza`.

These two steps produce QZA files, fastq-sequences.

### 5.3 DADA2

Next step applies DADA2 algorithm using QIIME2 DADA2-plugin, to run this, we need to add a command `dd` after + sign. DADA2 produce 3 output files, `dd_seq.qza`, `dd_table.qza`, `dd_stats.qza`. if this is the last step of the analysis then the target can be any one of them. Example:

`results/AB/AB+fp-f10-r20+bb-t18+dd_stats.qza`.

### 5.4 Taxonomy assignment

Three classifiers are already attached to this pipeline and will be downloaded automatically when they call for the first time. Taxonomy assignment happens after the DADA2 step. To do it `cls-gg` for greengenes, `cls-silva` or `cls-silvaV34`.

### 5.5 Diversity measurements

This stage computed alpha and beta diversity, for alpha diversity, "simpson", "chao1", "shannon_entropy", "observed_features", for Beta diversity: "unifrac", "braycurtis" and "jaccard" distances are measured. No need to run a taxonomy assignment before calculating diversity. However, a rarefaction step is required.

```
$ snakemake --use-conda --cores 10 \
↪  results/AB/AB+bb-t16+fp-f17-r21+dd+rrf-d10000+alphadiversity.tsv
```

### 5.6 Summary

The summary step will produce all the required output.

## 6 Tutorial example

let's assume that we have a data set of Fastq files, and we want to analyze it using Snaq. Snakemake, Snaq, and docker (in the case of windows).

### 6.1 Saving data in data folder

Create a new folder in `data/` with the name AB

### 6.2 Run full analysis

### 6.3 Using docker

```
$ docker run -it -v C:\snaq:/work -w /work snakemake/snakemake \
      snakemake --use-conda --cores 10  \
      results/AB/AB+bb-t18+fp-f17-r21+dd+cls-gg+rrf-d10000.zip
```

### 6.4 Command line

You need to activate the snakemake environment in conda and send the snakemake commands inside the <snaq> folder.

```
$ conda activate snakemake
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-gg+rrf-d10000.zip
```

### 6.5 Run partial analysis

If the user wants to stop in the middle for example after Dada2 application:

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd_table.qza
```

Then if the user wants to run taxonomy assignment:

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-gg_taxonomy.qza
```

Further steps can run subsequently in order if needed.

### 6.6 Trying multiple parameters

#### 6.6.1 eg. Taxonomy classifier

if the user wants to compare the results of using greengenes vs Silva classifiers, he can run the analysis upto taxonomy classification for greengenes "cls-gg":

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-gg_taxonomy.qza
```

and then run the same using "Silva."

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-silva_taxonomy.qza
```

Then The user can check the output and choose the one he likes to continue for the next steps. It is also possible to complete the analysis using Silva and greengens and compare the results after examining alpha and beta diversity.

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-silva+rrf-d10000.zip
```

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-gg+rrf-d10000.zip
```

### 6.6.2 eg. Rarefication depth

Suppose a user did the full analysis using a rarefication depth threshold of 10000. He found out that the threshold was too big and decided to rerun the analysis using 5000 threshold. How can that be done?

simply the user needs to send another command with the new threshold:

```
$ snakemake --use-conda --cores 10 \
    results/AB/AB+bb-t18+fp-f17-r21+dd+cls-silva+rrf-d5000.zip
```

Snakemake will look for the required files to create the final zip file. If they are available, they won't be made again. Quality trimming, primer cropping, dada2, and taxonomy assignment will not run again, and the intermediate file will be used. However, the parts affected by rarefication, such as alpha and beta diversities, biom tables, etc., will be produced using the new rarefication threshold.

Users can easily differentiate between these results depending on the file name (the informative file name system).