User Manual: Docker with Natrix2

Dustin Finke, Department of Biodiversity
University of Duisburg-Essen, Essen, Germany

Repository:

`https://github.com/dbeisser/Natrix2`

Each Natrix2 `main` release includes a new Docker image.

Contents

1 Preparations

1.1 Docker Installation

Ensure that a Docker version is installed before attempting installation.

```
$ docker --version # Check which Docker version is installed
```

If Docker is not installed, visit the official site, select your operating system, and follow the installation instructions. Once installed, proceed to the next step.

Install Docker: **https://docs.docker.com/engine/install/**

1.2 Pull Natrix2 Image

If you have installed Docker, you can download the Docker image. This takes some time until the image is downloaded. As soon as the download is finished, you can continue with the next step.

```
$ docker pull dbeisser/natrix2:latest # Download Natrix2 image
```

Once the Docker image has been downloaded completely, you can check it.

```
$ docker images # Check whether the image has been loaded
```

1.3 Set up the Environment

Before you can use Docker, you need to create the following folders for your Docker container. These folders are important for the corresponding input and output files.

```
# Folder structure for the Docker container (to be created locally)
./natrix2/                  # Main project folder
    input/                  # Contains files needed for analysis
        samples/            # Input data files for analysis
        config.yaml         # Configuration file
        primer.csv          # Primer table
    output/                 # Saves analysis results
        results/            # Folder for the results
    database/               # Saves reference databases
```

1. Copy the `config.yaml` file and the `primer.csv` file into the `input` folder. You can create a subfolder for your samples, but **the configuration file and the primer table must remain in the** `input` **folder**.

2. Open the `config.yaml` file with any editor and **adjust the parameters for your samples**. Make sure to set the parameters for your CPU cores and your memory (RAM) before starting the analysis.

3. **Specify the folder path for your data** in the `config.yaml` so it can be found. If you use a subfolder, extend the path: `input/samples`. The same applies to your primer table.

```
# Example configuration from the config.yaml file
general:
    filename: input/samples          # Samples folder
    output_dir: output/results       # Results folder
    primertable: input/primer.csv    # Primer table path
    cores: 20                        # Number of CPU cores
    memory: 10000                    # RAM in Megabytes
    ...
```

2 Launch Docker

2.1 Create Docker Container

The Docker container includes all necessary tools pre-installed, so you won't need to download anything initially. **Replace `</your/local/>` with the path to your `natrix2` folder.** After connecting to the container, you can start your analysis. Refer to the section on **Run Natrix2** for details, and you can test your Docker container with our sample data in the section **Test Execution**.

```
# Replace </your/local/> with your paths to the natrix2 folder
# Example: /your/local/natrix2/input => /path/to/natrix2/input

$ docker run -it --label natrix2_container -v /your/local/natrix2/input:/
   app/input -v /your/local/natrix2/output:/app/output -v /your/local/
   natrix2/database:/app/database dbeisser/natrix2:latest bash
```

2.2 Run Natrix2

To run the preparation script and Snakemake manually, activate the Snakemake environment.

```
$ conda activate natrix # Activate the Snakemake environment
```

Execute the preparation script `create_dataframe.py` to create the dataframe for your samples. It is important to specify the name of your configuration file.

```
$ python3 create_dataframe.py input/config.yaml # Create dataframe
```

Now you can start your pipeline so that your samples can be analyzed. If the pipeline stops (due to an error or intentional interruption), rerunning the command will restart it from that point.

```
# Set your configuration file and specify the number of CPU cores
$ snakemake --use-conda --configfile input/config.yaml --cores <cores>

# Example: using <config.yaml> with 10 CPU cores:
$ snakemake --use-conda --configfile input/config.yaml --cores 10
```

If you want Natrix2 to start automatically, use the script `docker_pipeline.sh`. This script will handle the setup and start Natrix2 for you.

```
# Replace <config> with your config file name
$ ./docker_pipeline.sh config
```

After your analysis, type `exit` to return to the Docker container or leave the container.

2.3 Test Execution

To test your Docker container before using your own data, you can use our test dataset, which contains Nanopore data. Simply run the container with the `test_docker.yaml` configuration file.

```
# Start a test run using our sample data
$ ./docker_pipeline.sh test_docker
```