Day 3: Exercises with Micromamba

Table of Contents

Scope

The supervisor has tasked you with running ResFinder on a few samples in your collection, to detect whether a novel tigecycline marker gene is present in the isolate collection of your institution. This is properly not the last time you will be asked to do bioinformatic analysis, so you have set yourself on the task of learning to install and run ResFinder locally. In your studies, you have learned that different software can be difficult to install and use manually, and thus you have come across the Anaconda platform for managing these challenges for you. You have looked through your options and have selected *Micromamba* as the program you wish to use for creating software environments and installing bioinformatic tools.

Setup

The exercise of this day assumes that you have successfully completed all exercises from day 1 and day 2. To prevent issues errors be sure to download and unzip the final checkpoint:

```
cd ~/
wget https://github.com/KasperThystrup/Unix4Beginners/raw/Unix4Pros/Final.tar.gz
tar -xvf Final.tar.gz
```

By now, you should have the following folder structure within home (~) called Final/BacterialData.

Conda handout

These exercises rely heavily on the Conda handout document, which is a strong helper while performing exercises. The exercise is designed in a way where you ONLY need to read the highlighted sections, and NOT the entire document in one go.

Difficulties

Some exercises in this document contain tasks, which are designed with varying levels of difficulties. Tasks with multiple levels of difficulties all lead to the same outcome, yet the approach to solving these differs. Therefore, whenever you face a task with multiple levels of difficulties, you should only select a single level of difficulty and solve that corresponding task. Feel free to shift levels of difficulty between exercises, and once you become comfortable with the assignments, you will get a better learning outcome if you start out with higher levels of difficulty and reduce the level if you are facing issues.

The tasks are always phrased in a separate bullet point, followed by sub bullets on the approach to solve the task. It is the approach to solving the tasks which presents various levels of difficulties. Difficulty levels in the assignment are designed with the following principles in mind:

- **Beginner**: Tasks at this level of difficulty should not be challenging to complete. The required information can be found entirely in the Conda handout document.
- **Advanced**: Tasks at this level of difficulty require a bit more individual thinking, but most information required can be found in the Conda handout document. This level is intended for participants with limited command line experiences, or fast learners.
- **Expert**: Tasks at this level of difficulty require some degree of experimentation in the terminal. When solving expert tasks, attempt to avoid using the Conda handout document until you get stuck for a while. This level is intended for participants who are comfortable learning from trial and error.

An example of a task in varying levels:

- Change the language of Ubuntu to Swahili (Seriously - do not do this!!!)
    - **Beginner**: Go to this page and follow the instructions: https://help.ubuntu.com/stable/ubuntu-help/session-language.html.en
    - **Advanced**: Open the settings menu in Ubuntu and figure out the rest.

Exercise 0: Installation

Recommended reading

Before continuing it is recommended to familiarize yourself with **section 1.1 of the Conda handout**.

Checking for pre-existing installation

Uncertain whether Micromamba is already installed or not, you open them terminal:

- Execute the command following: `micromamba --version`

If the version of Micromamba is shown, you can skip directly to Exercise 1.

If you on the other hand get an “command not found” error, this means that *Micromamba* is not installed. To install *Micromamba*, you must first install the program “curl”. In the command-line execute the following command:

```
sudo apt-get install curl
```

Provide your password to proceed with the installation.

Go to the official homepage on *Micromamba* and inspect the installation instructions for your Operating System (OS).

- Go to https://mamba.readthedocs.io/en/latest/installation/micromamba-installation.html
- Follow the **Automatic Install** instructions for Ubuntu or for MacOS (regarding on the OS on your laptop)
- After the installation has finished, close the current terminal window and open a new terminal
    - Execute the following command: `micromamba --version`

Now *Micromamba* should have been successfully installed.

Exercise 1: First environment

Recommended reading

It is recommended to familiarize yourself with **sections 1.3.1**, **1.3.1.a**, and **1.3.2 of the Conda handout** before starting this exercise. Further background information can be found in **sections 1.1 and 1.2**

Creating an empty environment

A good starting point is to try and create an empty environment, just to get the hang of it.

a. Create an empty environment named "empty_env":
   - **Expert**: Inspect the `micromamba create` help page to determine the command.
   - **Beginner**: Reinspect **section 1.3.1.a** to determine the command.

Once successful, *Micromamba* should print a message containing "Empty envrionment created at prefix".

As a sanity check, ensure that the environment exists, and verify that it is indeed empty:

b. Execute `micromamba env list` and ensure that "empty_env" is listed.

c. Activate the "empty_env" and ensure that it is empty
   - **Advanced**: Execute `micromamba --help` and inspect the help page for the subcommand `activate`.
   - **Beginner**: Inspect **section 1.3.2** and replace the environment name with `empty_env`

d. Make a sanity check on whether the environment is truly empty by executing `micromamba list`

In the terminal session, you should now see the activated environment enlisted in parenthesis at the far left side of the command line interface. In this case it should be "(empty_env)". Also, if there is nothing listed under the first line "List of packages in environment", the environment is empty!

Exercise 2: Our first bioinformatic analysis

Recommended reading

It is recommended to familiarize yourself with **section 1.3 of the Conda handout** before starting this exercise. In addition, consider revisiting **sections 1.2**, **1.3.1**, **1.3.1.a**, and **1.3.2 of the Conda handout**

Multi locus sequence type

Within the field of characterizing bacteria, a common characteristic to analyze is the **m**ulti **l**ocus **s**equence **t**ype (MLST) of a given isolate. In the world of whole genome sequencing, MLST can be analyzed from genome assemblies (.fasta files) through the software "mlst" by Torsten Seeman (tseemann). Let us check out whether it exists on the Anaconda platform.

a. Go to Anaconda.org and search for the package called "mlst".
b. Figure out which channel that hosts the "mlst" package
   - **Advanced**: Click on the "mlst" package link. This leads you to the installation instructions which convey information as such: `conda install Channel::Package` (Note that the instructions assume you use the Conda program, don't mind this!)
   - **Beginner**: In the search results, inspect the column "Artifact (owner / artifact)" which denotes information like this Channel / Package

If we were to install this package without specifying a channel, chances are that the installation would fail, as *Micromamba* is not set up to screen the correct channels. To make sure that *Micromamba* knows which channels it should install packages from, you should always specify which channels to install from.

c. Create a new environment called "my_mlst" and make sure "mlst" is installed upon creation.
   - **Expert**: Inspect the `--help` page on Micromamba **create** to figure out how to name your environment and how to point to a channel.
   - **Advanced**: Inspect the options listed in **section 1.3.1** to figure out how to point to a specific channel. Then, build the **create** command, inspired by the command from Exercise 1.
   - **Beginner**: Revisit **section 1.3.1.a** and use the command listed there.
d. Activate and inspect whether it is empty (see exercise 1).

By now you have successfully created your first bioinformatic environment and are now ready for performing your very first bioinformatic analysis.

e. Execute the `mlst --quiet [~/your/file.fasta]` command and figure the sequence type of the selected isolate by inspecting the third column. (The columns are tab-separated)

   - **Expert**: Locate the path of another assembly and use it to replace `[~/your/file.fasta]`. Finally, Add `> [my_sample]_mlst.tsv` to the end of the command replace `[my_sample]` with the sample name (e.g. `> E_coli_CPO096_mlst.tsv`). What does this result in?
   - **Advanced**: Locate the path of another assembly and use it to replace `[~/your/file.fasta]`.
   - **Beginner**: Replace `[~/your/file.fasta]` with `~/Final/BacterialData/E_coli/assemblies/E_coli_CPO096.fasta`.

Well done, you have just completed your very first bioinformatic analysis! Now we are ready to move on to our actual assignment which is to determine whether we can detect the *TOprJ3* gene in our samples.

Exercise 3: Installing ResFinder

Recommended reading

It is recommended to familiarize yourself with **section 1.3.1.b of the Conda handout** before starting this exercise. In addition, consider revisiting **sections 1.2**, **1.3**, **1.3.1**, **1.3.1.a**, and **1.3.2 of the Conda handout**.

Proceeding with the installation

We have selected ResFinder for performing screening of our samples, and are ready to start the installation process:

a. Determine which channel the "ResFinder" package is hosted on. See **section 1.3.1.**
   - **Advanced**: Use a search engine such as Google, to navigate directly to the installation page for the ResFinder. In the search engine, enter *"resfinder conda"*.
   - **Beginner**: Go to Anaconda.org and search for the "ResFinder" package.

b. Create an environment called "resfinder" and ensure that the "ResFinder" package is installed during creation.
   - **Expert**: Read **section 1.3.1.b** and create an environment file in home called "resfinder.yaml" using nano. Copy-paste the contents from the "my_mlst.yaml" example file into the resfinder.yaml file, and replace all dependencies with only " - resfinder". Next, make sure the correct channels are included, and finally, tweak the command listed in **section 1.3.1.b** to create the environment.
   - **Beginner**: Revisit **section 1.3.1.a** and tweak the listed command to make the correct environment name and to install ResFinder rather than MLST.

c. Activate the "resfinder" environment and execute `run_resfinder.py --help` to validate whether the package is correctly installed within the environment.

Exercise 4: Setting up and running ResFinder

Recommended reading

It is recommended to familiarize yourself with **section 1.3.3** and **the whole chapter 2 of the Conda handout** before starting this exercise. In addition, consider revisiting **sections 1.2**, **1.3**, **1.3.1**, **1.3.1.a**, and **1.3.1.b of the Conda handout**.

Database setup

Before we can start the analysis, we must set up a database of resistance genes.

a. Download "resfinder_db" and make sure it is in a folder called ~/Databases. In the end, the database should be located here ~/Databases/resfinder_db
   - **Expert**: Visit https://bitbucket.org/genomicepidemiology/resfinder_db/src and read the description for how to download the database using the git command. Once downloaded, make sure that resfinder_db/ is moved into ~/Databases/. If ~/Databases/ do not exist, you must first make it manually.
   - **Beginner**: Execute the command listed in **section 2.2**

b. Check whether ResFinder detects the *TOprJ3* gene, by running ResFinder on ~/Final/BacterialData/Resfinder/P_aeruginosa_TOprJ3-positive.fasta
   - **Expert**: Inspect section 1.3.3 and execute the ResFinder software without activating the "resfinder" environment. Tweak the command listed in **section 2.2** swapping out the E_coli sample with the P_aeruginosa sample.
   - **Beginner**: Activate the "resfinder" environment, then tweak the command listed in **section 2.2** by first replacing the *E_coli* fasta file with the *P_aeruginosa* FASTA file, and then update the output directory to match the P_aeruginosa sample name.

c. Inspect the results file located in ~/Final/BacterialData/Resfinder/P_aeruginosa_TOprJ3-positive/ResFinder_results_tab.txt
   - Beginner: Print its contents using the cat ~/Final/BacterialData/Resfinder/P_aeruginosa_TOprJ3-positive/ResFinder_results_tab.txt command
   - Expert: Use the less -S command to list the file contents in single line format.

We have now validated that we can detect our target gene in a bacterial genome file, it is time to screen our own collection

d. Run ResFinder yet again, but this time screen the ~/Final/BacterialData/Resfinder/our_first_nanopore.fasta file ,do we find the *TOprJ3* gene in our own collection?

- **Advanced**: Reuse the command from before to screen for resistance genes in the our_first_nanopore sample. Once done, rerun ResFinder on other samples in the ~/Final/BacterialData/Resfinder/ folder
- **Beginner**: Reuse the command from before, replacing the sample with our_first_nanopore (make sure to replace the .fasta file as well as the output folder)

Congratulations, you have now completed your first bioinformatic assignment!

Bonus exercises

PlasmidFinder and VirulenceFinder has been developed by the same team which provides ResFinder. Therefore, the syntax on these programs is similar to ResFinder.

a. Make an environment containing PlasmidFinder and VirulenceFinder
   - **Expert**: Read **section 1.3.1.b** and create an environment file in home called "finders.yaml" using nano. Copy-paste the contents from the "my_mlst.yaml" example file into the finders.yaml file, and replace all dependencies with the two lines:
     - plasmidfinder
     - virulencefinder
   - Next, make sure the correct channels are included, and finally, tweak the command listed in **section 1.3.1.b** to create the environment.

b. Set up databases for both tools in the ~/Databases folder.
   - **Expert**: Read the instructions for both tools and remember to move the databases into the ~/Databases folder.
     - Plasmidfinder_db
     - Virulencefinder_db

c. Finally, Execute PlasmidFinder and VirulenceFinder individually on any given sample
   - **Expert**: Inspect **section 1.3.3** and execute each of the finders individually without activating the "finders" environment on any assembly file. Do note that each tool must be executed in two separate commands (and not in a single command)