Loading Bioinformatics Pipelines on the Command Line

Before using the Linux command line to perform bioinformatics pipelines, you will need to configure your computer to be able to run these commands. The information below links specifically to a bacterial metagenomics pipeline that can be used for assembly of *Vibrio cholerae* genomes, but many of the same principles can be applied to other pipelines hosted on github.

Important notes for following this tutorial:

- Text with a gray background in `monospace font` represents commands to type in. Generally commands are one line, however, in this document, commands might wrap to the next line visually. We will add a blank line between commands to indicate when multiple commands are present.
- `Monospace blue text with a white background` represents the input and/or output of the terminal. Due to different terminal programs, the appearance of your terminal may look different.
- Bold text surrounded by **< >** is something you will have to replace with your own folder, path, or sample name.

This tutorial will take you through the two main steps of loading a bioinformatics pipeline on your computer:

1. Downloading a desired bioinformatics pipeline from the internet.
2. Building a compute environment (i.e., downloading software) needed to run the pipeline.

STEP 0: Navigate to your HOME Directory

Unless otherwise stated, all commands should be run in your home directory. Generally this is where the command line is located if you open a new command prompt (i.e., Terminal window).

1. If you are not in your home directory (or are unsure), move to your home directory with this command:

```
cd ~/
```

STEP 1: Download the Bioinformatics Pipeline

Bioinformaticians often make their genome assembly and analysis pipelines publicly available on websites such as www.github.com (commonly referred to as "GitHub"), which is specifically set up for hosting code and other types of files. There are two ways to download files and pipelines from GitHub: via your web browser or via the command line. To perform bacterial genomics assembly using CholGen pipelines, use one of the methods below:

Option 1: Download from GitHub using the command line:

1. Install the git command line program using the following command:

```
sudo apt install git-all
```

Note: This command will prompt you for your password. When entering your password on the command line, no characters will appear. However, the command line will still be receiving input so be sure to type your password correctly before pressing Enter.

2. After git is installed, run the following command in your HOME directory to download the folder from GitHub:

```
git clone https://github.com/CholGen/bacpage.git
```

3. Check the download was correct by viewing the contents of your home directory:

```
ls
```

You should see a new directory in your home directory with the name `bacpage/`.

This procedure can be used to download any other pipeline hosted on the GitHub website. If you are interested in loading a different pipeline than the one used here, simply replace the URL in Step 2 with the link found on the other pipeline's GitHub page.

4. Confirm that the genome assembly and analysis folder (`bacpage/`) contains 5 sub-folders: `config/`, `example/`, `resources/`, `test/`, and `workflow/`.

Option 2: Download from GitHub using a web browser:

1. Paste the following link in your web browser. A file named `bacpage.zip` will be automatically downloaded.

   Link: https://github.com/CholGen/bacpage/releases/latest/download/pipeline.zip

2. Unzip this file on your computer and move the resulting folder (named `bacpage/`) into the HOME directory on your computer. You can do this by dragging the `bacpage/` folder from your Downloads folder into your HOME directory.

3. Confirm that the genome assembly and analysis folder (`bacpage/`) contains 5 sub-folders: `config/`, `example/`, `resources/`, `test/`, and `workflow/`.

STEP 2: Build the Compute Environment

In contrast to simple command line programs such as `cd` and `ls`, bioinformatic analysis requires many specialized programs which need to be installed from a wide range of developers, with a wide range of requirements and dependencies. To make it easy for others to repeat their exact processes, bioinformaticians can set up computing "environments" that contain all software and other setup items needed to follow a particular analysis. They can then share the instructions for replicating their compute "environment" on a site such as GitHub, so other people can quickly set up their computers in the exact same way.

The files you downloaded from GitHub above contain instructions on how to build the environment needed to run all of the scripts you also downloaded. These instructions will automatically install all of the software you need to run the bioinformatics scripts. In order to follow these instructions (and load someone else's compute environment on your computer), you'll need to first install a piece of software that can read these instructions.

There are a few different softwares that can be used to build environments. "Mamba" and "Conda" are very common options. We will use "Mamba" because it tends to work faster than "Conda", but if you are ever working with another set of instructions that uses "Conda" instead, know that you are doing something very similar!

Follow the instructions below to install Mamba and then use the Mamba software to build the bacterial assembly environment on your computer. Please note that you will need a network connection to complete these steps, as you'll be downloading Mamba from the internet, and then using the Mamba software to download other programs and software from the web.

1. Navigate to your home directory with the following command.

```
cd ~/
```

2. Install mamba using the following command:

```
curl -L -O
"https://github.com/conda-forge/miniforge/releases/latest/download/Ma
mbaforge-$(uname)-$(uname -m).sh"

bash Mambaforge-$(uname)-$(uname -m).sh
```

You should have noticed that your command line prompt has changed from:

```
[seqlaptop@linuxbox seqlaptop]$
```

to:

```
(base) [seqlaptop@linuxbox seqlaptop]$
```

This indicates that you are now in the `base` mamba environment, and that all of the software you loaded into that environment (called "base") is accessible.

Note: to get out of the base mamba environment (for example, if you're unable to access previously installed software, run `mamba deactivate`). When you leave the mamba environment, you will no longer be able to access software that you installed when you were inside the mamba environment. To get back into the base mamba environment, run `mamba activate`).

Rather than installing everything to the base environment, it's recommended to create task-specific environments, containing only software that is needed for a given task. We are now going to create a new environment with a different name that will specifically have the assembly pipeline software in it.

3. Navigate to the directory where you downloaded the bioinformatics pipeline above. This should be a directory titled `bacpage/` in your home directory.

```
cd ~/bacpage
```

You will need to navigate to this directory any time you want to use the pipeline you downloaded from GitHub above. Run `pwd` and note the path of this directory for future reference.

4. Inside this directory you will find a file called `environment.yaml`. Feel free to open it in a text editor and look at its structure. The file describes all the software that is required for the pipeline. To tell the computer to install all of this software, run the following command:

```
mamba env create -f environment.yaml
```

Note: this installation may take a few minutes depending on your internet speed.

This command will create a new environment called “bacpage,” which is distinct from the default “base” environment discussed above. It can be activated and deactivated in an identical manner to the “base” environment.

5. To use a mamba environment, and all the software we just installed, we have to activate it. Do this with the following command:

```
mamba activate bacpage
```

Your prompt should have changed again to the following:

```
(bacpage) [seqlaptop@linuxbox bacpage]$
```

This indicates that your terminal is now using the `bacpage` environment and can use all the required software. Use `mamba deactivate` to return to the default “base” environment.

6. We will explain this command in more depth later, but run the following to test whether the installation was successful:

```
snakemake --configfile test/test.yaml --all-temp --cores 8
```

This command should take a few minutes to run and then complete without an issue.