Conda for bioinformatics

This document accompanies the DIYtranscriptomics course and is intended to provide basic guidance in the installation of various bioinformatics softwares using Conda. If you have problems...don't worry, we're here to help.

What is Conda and why should you install it?

Install Miniconda

- Mac OS
- Windows OS

Configuring your Conda installation

Create your first Conda environment

Rinse and repeat

Install other software we'll use for the course

Useful Conda tips

Generally useful Conda commands

Don't get carried away with your 'base' conda environment

Backup plan if Conda doesn't work for you

Plan B for Mac OS

Plan B for Windows OS

Troubleshooting tips

What is Conda and why should you install it?

Taken directly from the **Conda manual:**

> *Conda is an open-source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and their dependencies. Conda easily creates, saves, loads, and switches between environments on your local computer. It was created for Python programs but it can package and distribute software for any language.*

Note: when you read 'package' in the text above, just think 'software'. An environment, on the other hand, is the software plus everything else that this software needs to run properly. This point is key to understanding why Conda has become a preferred way for installing a wide range of bioinformatics software – because it does a pretty good job (not perfect) of avoiding **Dependency Hell.**

Install Miniconda

Conda comes in two flavors: Anaconda and Miniconda. We want to install Miniconda, because it's much more lightweight while still meeting all of our needs. Importantly, when we install Miniconda, we'll be getting the Python programming language as part of that

installation.

installation.

Mac OS

Download the Miniconda install script from here

Move this shell script (.sh) file to your home folder on your Mac, and enter the following line into your terminal application

```
bash ~/Miniconda3-latest-MacOSX-x86_64.sh -b -p $HOME/mini
```

Now 'source' conda so that it is available to you from the command line regardless of which directory you're in

```
source $HOME/miniconda/bin/activate
```

This next step may only be necessary if you're running a newer MacOS that uses the zsh shell.

```
conda init zsh
```

Windows OS

We will first install Miniconda and then add three new locations to your system environment path for conda to be recognized as a command in your Command Prompt.

1. Download the Miniconda executable (.exe) from here and double click the .exe to run the setup guide
2. Click "Next >" to continue
3. Click "I Agree"

3. Click "I Agree"
4. Verify that the installation type "Just Me (recommended)" is selected and then click "Next >"
5. Use the default destination folder which should resemble C:\Users\yourname\Miniconda3. We will need the path to this destination folder soon so copy it to your clipboard and then click "Next >"
6. Check "Register Miniconda3 as my default Python 3.9" and then click "Install"
7. Using the search box in the toolbar at the bottom of your screen, search for "environment variables" and then click on "Edit the system environment variables"
8. Click "Environment Variables..."
9. Under "System variables" click on "Path" so that row is highlighted in blue and click "Edit..."
10. Click "New"
11. In the box that appears, paste the file path that you copied in step 5. It should look like **C:\Users\yourname\Miniconda3**
12. Click "New"
13. Paste the file path that you copied in step 5 but modify it so that it looks like **C:\Users\yourname\Miniconda3\Scripts**
14. Click "New"
15. Paste the file path that you copied in step 5 but modify it so that it looks like **C:\Users\yourname\Miniconda3\Library\bin**
16. Click "OK" to close the "Edit environment variable" window
17. Click "OK" to close the "Environment Variables" window
18. Click "OK" to close the "System Properties" window

Configuring your Conda installation

Now make sure Conda works and explore a bit using the lines below

```
conda info #to view all the details about your conda set-u
conda info --envs #to view all the environments available
```

One of the things that makes Conda so great for software installation is that it has access to various channels where many pre-packaged bioinformatics programs can be downloaded with all their dependencies. Let's configure our Conda installation now so that it knows which channels to look for.

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
conda config --set offline false
```

Create your first Conda environment

Some of the most basic pieces of command-line software we discuss and use at the beginning of course aren't available in R/bioconductor. Instead, we'll install these into a single 'environment' using Conda, which makes managing dependencies much less frustrating. We'll be using Conda to install **Kallisto**, **fastqc**, and **MultiQC**.

Begin by creating an empty environment called 'rnaseq'...or name it whatever you'd like

```
conda create --name rnaseq
```

Now activate your newly created environment

```
conda activate rnaseq
```

Notice that your terminal should now show that you have now entered the 'rnaseq' environment (example image below).

Now let's install some commonly used RNA-seq software inside this environment. Begin with Kallisto, which is our go-to tool for read mapping.

Note: if you get a **y/n** question during installation, respond yes by typing 'y' and enter.

Note: the most important piece of software here is Kallisto. If you encounter issues installing FastQC and/or MultiQC, just move on...it will not impact your ability to participate in the course.

```
conda install -c bioconda kallisto
```

Test that it works!

```
kallisto
```

If your Kallisto installation worked, then you should see something in your terminal that resembles the output below (basically, Kallisto is saying "I'm here, now what would you like me to do?!"). If so, take a second to pat yourself on the back – you just installed your first piece of software using Conda! 🎉 🎊

```
kallisto 0.48.0

Usage: kallisto <CMD> [arguments] ..

Where <CMD> can be one of:

    index         Builds a kallisto index
    quant         Runs the quantification algorithm
    quant-tcc     Runs quantification on transcript-compat
    bus           Generate BUS files for single-cell data
    merge         Merges several batch runs
    h5dump        Converts HDF5-formatted results to plair
    inspect       Inspects and gives information about an
    version       Prints version information
    cite          Prints citation information

Running kallisto <CMD> without arguments prints usage info
```

Note: if you are using Windows and the kallisto installation using conda was unsuccessful, follow the instructions in the Plan B for Windows OS section.

Rinse and repeat

Now that you have Kallisto installed, you're going to install additional software into the same 'rnaseq' environment.

Run `conda install -c bioconda fastqc` and `conda install -c bioconda multiqc`

Check that both installed correctly.

Note: if your laptop runs Windows, you may encounter some issues with fastqc. It should install without issue but fastqc may not be recognized as an internal or external command, operable program or

recognized as an internal or external command, operable program or batch file. If so, no worries, it won't affect your ability to participate in the course. However, you may want to try installing a similar program for quality control analysis of raw reads, called **fastp**. You can install fastp using `conda install -c bioconda fastp`. Another alternative is to install FastQC manually and use it in its interactive mode. Instructions for this can be found in the Plan B for Windows OS section

Install other software we'll use for the course

Now that we are done installing software in our rnaseq environment, we can exit this environment by typing `conda deactivate`. Let's create some additional environments for other software you might want to use on your laptop.

Note: for the purposes of this course, the most important piece of software listed below is Kallisto-bustools in python (**kb-python**) for single cell RNA-seq analysis at the end of the course. If you encounter issues installing sourmash or centrifuge, just move on...it will not impact your ability to participate in the course.

Note: run each line below separately, rather than copying/pasting the entire block of code.

```
conda create -y --name kb python=3.8 #create an environmer
conda activate kb #activate that newly created environment
pip install kb-python #install kb-python in the environmer
kb #test that it works!
```

We'll use **sourmash** for creating and analyzing 'sketches' of HTS data later in the course. Let's create a new and separate environment for this

```
conda create --name sourmash #create an empty environment
conda activate sourmash #activate that newly created envi
conda install -c bioconda sourmash #install sourmash in th
sourmash #test that it works!
```

We'll use Centrifuge for rapid and memory-efficient classification of DNA sequences from microbial samples

```
conda create --name centrifuge #create an empty environmer
conda activate centrifuge #activate that newly created env
conda install -c bioconda centrifuge #install centrifuge i
centrifuge #test that it works!
```

Useful Conda tips

Check out this article for a nice breakdown of the between Conda and the package manager, Pip.

Generally useful Conda commands

```
# Displaying useful info related to conda on your machine
conda list #shows you everything installed in your current
conda list -n [ENV NAME] #shows you everything installed i
conda config --show #shows your config file, which you may
conda config --show channels #shows you channels from the
conda remove --name myenv --all #remove any environment (s
conda search myenv #search your channels for a specific pa
conda update --all #update conda
nano $HOME/.condarc #view your list of channels
```

Don't get carried away with your 'base' conda environment

When you install conda, you automatically get a 'base' environment. In fact, you may find that when you open your terminal or shell application, that you are placed in the base env by default. Avoid installing lots of software in base or, eventually, you will run into conflicts.

Backup plan if Conda doesn't work for you

You should only be reading this if the steps above failed. So, what do you do if Conda doesn't install properly or you aren't able to install the software above? No worries, we can probably help in **the lab session devoted to troubleshooting software installation**. In the event that we can't resolve your IT issues, we have a backup plan to help you get the most essential software for the course installed.

Plan B for Mac OS

If you're running a Mac OS, then it's a good idea to install **Homebrew**, which has nothing to do with Conda but is a fantastic package manager for the MacOS. Although this isn't essential for the class, it will make your life **a lot** easier when you try to install software in future. To get Homebrew, enter the following line into your terminal.

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercont
```

Some, but not all, of the software we installed using Conda above is also available for MacOS using Homebrew. Go ahead and install as follows:

```
brew install kallisto
brew install fastqc
```

Plan B for Windows OS

Work through the following steps to install Kallisto on your Windows OS:

1. Obtain administrative access for your computer
2. You will need to be able to unzip files. If you don't have software that can do this, you can download **WinRAR** and install it in Program Files

2021)

4. Right click the downloaded zip file and choose "Extract here" or "Extract all". A new folder will appear called "kallisto". Move it to your Program Files directory (e.g. C:\Program Files)
5. Kallisto is now installed on your computer but cannot be accessed from any location in the Command Prompt until you add it to your computer's path system variable like we did with Miniconda. Using File Explorer, open the "kallisto" folder you just created, click in the bar showing the file path so it is highlighted in blue and copy this file path
6. Using the search box in the toolbar at the bottom of your screen, search for "environment variables" and then click on "Edit the system environment variables"
7. Click "Environment Variables..."
8. Under "System variables" click on "Path" so that row is highlighted in blue and click "Edit..."
9. Click "New"
10. In the box that appears, paste the file path that you copied in step 5. It should look like **C:\Program Files\kallisto**

5. It should look like **C:\Program Files\kallisto**
11. Click "OK" to close the "Edit environment variable" window
12. Click "OK" to close the "Environment Variables" window
13. Click "OK" to close the "System Properties" window
14. Relaunch Command Prompt (or your RStudio session if you are working in the RStudio Terminal). Check for proper installation by typing 'kallisto' (without quotes) into Command Prompt (or RStudio Terminal)
15. Although not required for running kallisto, you should install **Cygwin** to give your PC some linux functionality (like running shell scripts)

Work through the following steps to install FastQC on your Windows OS:

1. Go to https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ and click "Download Now"
2. Click "FastQC v0.11.9 (Win/Linux zip file)"
3. Click "OK" to open the compressed zip folder
4. Click "Extract all"
5. Click "Browse..." to navigate to C:\Program Files and click "Select Folder"
6. Click "Extract"
7. Although you've now installed FastQC, it relies on Java which also must be installed. Go to https://adoptopenjdk.net/ and verify that "OpenJDK 11 (LTS)" and "HotSpot" are selected and click "Latest release"
8. Click "Save File"
9. Once it's downloaded, double click on "OpenJDK11U-jdk_x64_windows_hotspot_11.0.12_7.msi"
10. In the setup wizard, click "Next"

11. Review the default installation settings and click "Next"
12. Click "Install"
13. Click "Finish"
14. To check that Java was installed correctly, open Command Prompt and run `java --version` . You should see:

```
openjdk 11.0.12 2021-07-20
OpenJDK Runtime Environment Temurin-11.0.12+7 (build
11.0.12+7)
OpenJDK 64-Bit Server VM Temurin-11.0.12+7 (build
11.0.12+7, mixed mode)
```

15. To check that FastQC was installed correctly, navigate to C:\Program Files\FastQC and double click on "run_fastqc.bat". A Command Prompt window will automatically open but you can ignore it and close it once you are done with FastQC. FastQC should also open

Work through the following steps to use FastQC in its interactive mode:

1. Navigate to C:\Program Files\FastQC and double click on "run_fastqc.bat". A Command Prompt window will automatically open but you can ignore it and close it once you are done with FastQC. FastQC should also open
2. Click "File" and "Open..."
3. Navigate to the folder containing the .fastqc.gz files you would like to analyze
4. Select all .fastqc.gz files of interest and click "Open"
5. There will be a tab for each file and each will be analyzed
6. As each file's analysis is done, click "File" and "Save report..."

7. Navigate to the folder that contains your RStudio project and all other related files
8. Click "Save"
9. Steps 5-7 will need to be repeated for each file
10. After manually saving all of the FastQC reports, you can edit the readMapping.sh in Sublime Text 3 to comment out the fastqc command so that it is ignored but the rest of the script will run. MultiQC will be able to find your FastQC reports if you saved them to the correct folder

```
#fastqc *.gz -t 4
```

Troubleshooting tips

If you use conda long enough, it's only a matter of time before you will run into issues where an environment will fail to install (usually with an error like "failed to resolve conflicts", or it might just hang forever on 'solving environment'). If this happens, you *may* be able to fix the issue simply by changing the way your Conda installation handles channels.

- first check your channel priority setting the conda configuration file with

```
conda config --show channel_priority
```

- then change to the opposite (e.g. if set to strict, change to flexible):

```
conda config --set channel_priority flexible
#alternatively, conda config --set channel_priority strict
```

- Now retry installing your conda environment.