Software and Using Terminal

BIOMEDIN/BIOPHYS/CME/CS 279

Fall 2020

1   Policies

In this class, there will be exercises which involve some coding that use existing software packages. We will make sure that all required software is available on all the Learning and Technologies Spaces (LTS) machines so that you need not install any software on your local machines. While you are free to install the programs on your own machine and develop locally, in general, if you encounter issues with installation or development, we encourage you to work on the LTS machines. Feel free to post issues to Piazza or come to office hours if you have any issues.

2   Installing Anaconda

If you plan on working on your local computer, we **highly** recommend installing Anaconda for Python3.8 (https://www.anaconda.com/products/individual). MacOS and Linux users may install Anaconda following the instructions on the Anaconda website.

If you are a Windows user and planning on working locally for the first assignment, you will need to perform an extra step during install by selecting the "Add Anaconda to my PATH environment variable" option when prompted. (see below)
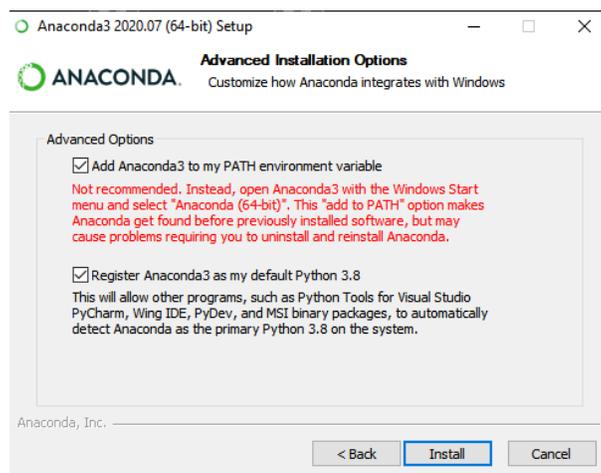
Windows-specific installation step

3 Installing PyMOL

If you are working locally for the first assignment, you will need to download and install PyMOL2.4 (https://www.pymol.org/) on your computer. PyMOL is also available on all the LTS Macs, which can be accessed remotely (see Assignment 1 for details). If you cannot get PyMOL to run on your computer, please use any of the LTS machines or contact the TAs.

4 Basic Shell Commands in Terminal

For those of you who've never accessed a computer through a terminal, there are a few commands that will be helpful to know. At a high level, you are navigating the computer's files and invoking programs by typing commands rather than using a mouse with a graphical display.

Here is a list of commands that may be helpful:

- `pwd` – prints the current working directory

- `cd <new directory>` – changes the working directory to the new directory
  Note that . refers to the working directory and .. refers to the parent directory

- `ls` – lists the files in the current directory

- `cp <original> <copy>` – copies the original file to the new copy

- `mv <original> <new>` – moves the original file to the new file.
  (This is also how you rename files.)

If you want additional information about any command or program, you can use `man <command>` which brings up the command's manual page.

To navigate to a specific folder through terminal, you will use the **cd** command. You can think of this like navigating through finder. For example, if I want to navigate to my assignment 1 folder (called "assn1") in my Downloads folder, I would type the following:

```
cd Downloads/assn1
```

If you would like to see what contents exist in the current directory, you would use the **ls** command. For example, if I wanted to see the contents of my Downloads folder, I would type:

```
ls Downloads
```

This will result in a list of all the files and folders within the Downloads folder in terminal.

Once you have navigated to your desired directory and would like to execute a program that lives in the current directory, you invoke

```
./<program name> <program arguments>
```

In our case, almost all programs that you will run will be Python scripts, which you invoke using `python`.

```
python <program name> <program arguments>
```

If you want to experiment in an interactive Python window, you can just invoke `python`, which will start a Python interpreter.

```
python
```

Note that **LTS** users will need to make a specific call to the 3.8 version of Python. You would invoke the commands:

```
python3 <program name> <program arguments>
```

and

```
python3
```

5   Editing Files

If you are editing files directly on Terminal, you will have to open them with a text editor. Two popular editors are `emacs` and `vim`. If you prefer editing files outside of terminal, Sublime Text editor is a popular choice.