BIO508: Lab Session 1

Announcements

- Welcome!
- Administrative business:
  1. Our emails: Yu `yuj219@mail.harvard.edu` Siyuan `sim653@mail.harvard.edu` (Please cc both TAs in your emails)
  2. The hours during which we usually respond to emails: 8am-5pm M-F
  3. Office hours: 3:30-4:30 on Thursdays for Siyuan, TBD for Yu
  4. Siyuan's office: SPH building 1, 412
- We are happy to answer questions that you have, but unless your question involves specific code please use the discussion board!
- Homework 1 is due at 11:55pm on Monday.

Setting up Python

Installing Python

In order for the operating system to understand what the command `python` means, it must know how to locate that program from the command line. Here are the instructions to get everything up and running.

1. Go to `http://www.python.org/download/`, download the appropriate version of Python 2 for your operating system and install it. As of this writing, the following caveats hold:
   (a) Many 64-bit extensions don't work very well, so even if you're running 64-bit Windows, install 32-bit Python.
   (b) These notes were written for 2.7. Python 3 has been released, but it is not yet compatible with many common add-ons and modules, so we will not be using it just yet (at least not this year!), and you should only install it if you want to get to know it separately on your own.
2. After Python has fully installed, add it to your path:
   (a) On Windows:
      i. Go to the Control Panel System, Advanced System Settings. This should bring you to the Advanced tab of a small dialog box.
      ii. Click the Environment Variables button in the lower right.
      iii. In the bottom half of this window, labeled System Variables, find the variable named `PATH` or `Path`.
      iv. Click to highlight it, then click the Edit button.
      v. Make sure your cursor is at the rightmost edge of the Variable Value textbox, then type `;C:\Python27`.
      vi. Click Ok, then Ok, then Ok again. Log out and back in, or restart your computer.
   (b) On MacOS:
      i. Congratulations! The Python installer should automatically include `python` in your `PATH` so it's accessible from the default Terminal application.

You should now be able to open a terminal and run `python` and see something like:

```
Python 2.7.9 (v2.7.9:648dcafa7e5f, Dec 10 2014, 10:10:46)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This allows you to input python commands one at a time. You can quit by typing `quit()`. But because if we have something complicated to do this kind of programming is very tedious and inefficient, we generate text files with the appropriate Python commands and then ask Python to execute them. Next, we will install the editors which we will be using. For this class, you may use jEdit, Eclipse with PyDev, or another editor of your choice. We will go through installing only jEdit and Eclipse, although you need only install your editor of choice.

Installing jEdit

jEdit is essentially a fancy text editor like Notepad. It allows syntax highlighting, line numbering, bracket matching, automatic indenting and other features that make it really useful for programming. It is also pretty intuitive, really easy to install on a Mac and relatively easy to install on Windows. To install jEdit:

1. Download jEdit from `http://www.jedit.org`
2. (a) On Windows:
   i. Start up a command prompt
   ii. Run the following command (including quotes):
   ```
   ftype Python.Source="C:\Windows\System32\javaw.exe" -jar "C:\Program Files
   \jEdit\jedit.jar" -reuseview "%1"
   ```
   iii. The command should silently complete. If you don't get any weird errors, type the following two lines:
   ```
   assoc .py=Python.Source
   assoc .pm=Python.Source
   ```
   iv. To make jEdit look less ugly:
      A. Go to the Utilities menu and select Global Options.
      B. Select Text Area, change Anti Alias Smooth Text to "subpixel", and check the Fractional Font Metrics box.

   (b) On Mac:
   i. Open the .dmg file
   ii. Copy the jEdit file into the Applications folder
3. Let's try creating and running a test_jedit.py file.
   (a) Open jEdit; you'll see an untitled window.
   (b) Save the file (CTRL-S or CMD-S) as `test_jedit.py` somewhere on your machine, but remember the path to the file (the top dropdown bar).
   (c) We have now generated the `test_jedit.py` file. Add the following line to the bottom of the file:
   ```
   print("Hello, world!")
   ```
   (d) Open up a terminal, and type `python` followed by the path to your file. If all goes as planned, the string "Hello, world!" should appear as output. Congratulations!

Some python practice

The very simplest way to interact with Python is through the commandline interpreter. Open python by typing the `python` command in the terminal and then run the following commands:

1. `print "Hello, world!"`
2.
```
name = "My Name"
print "Hello, " + name
```
3. Try changing "My Name" to your own name in the above commands and see what happens.
4. `24*23 - 68 + 2*2*2*3`

Now that we have our editors installed, we can do some simple commands to see how everything works. Open up the `test_jedit.py` file that you created in jEdit. Enter the above three commands into the file, and then run the file. You should get the same output as when you entered in the commands through the interpreter.

Introduction to the Command Line

NOTE: Some very useful material related to this topic can be found in Haddock and Dunn, Chapter 4.

The first thing we want to do is start up the terminal so we can do some practice. Table 1 shows how to do this for Windows or Mac.

Windows	MacIntosh
Either select Run from the Start menu and type `cmd`, or select Start/Programs/Accessories/Command Prompt	Navigate to Applications/Utilities and launch Terminal

Table 1: How to start the terminal for Windows or Mac.

Commands for doing various common and useful tasks such as opening or running files, changing directories, etc. are on the attached sheets at the back of the lab. The commands for performing these actions in Windows are in table 2, while the commands for performing these actions on a MacIntosh are in table 3. Some useful shortcuts are shown in table 4.

Practicing with the command line

To familiarize yourself with these commands, do the following:

1. Make an empty document of your choice (for example, a Word document or a text document) and save it in your home directory.
2. Open the commandline.
3. From the commandline, navigate to your home directory. Confirm you are at the right place by listing the files in that directory and finding the document you just created.
4. Make a new directory called `bio508_practice` in your current directory.

5. Move your file into `bio508_practice`.
6. List all the files in `bio508_practice`.
7. Move into `bio508_practice`.
8. Remove your file.

iPython

Installing iPython

1. Download Anaconda from `http://continuum.io/downloads`.
2. Restart your computer
3. To update to the correct version:
```
conda update conda
conda update ipython
```
   (a) When prompted, enter `y` for "yes".

Running the iPython interpreter

To run iPython, use the command `ipython` in the terminal. You should see something like this:

```
Python 2.7.9 |Anaconda 2.1.0 (x86_64)| (default, Dec 15 2014, 10:37:34)
Type "copyright", "credits" or "license" for more information.

IPython 2.3.1 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

We can use this as we used the python interpreter earlier, but it has some nicer properties. You can try running the commands from the python practice to see how it works. Also try the following *new* commands—some of the output should look familiar!

1. On Mac: `!ls` On Windows: `!dir`
2. `!mkdir Test/`

As before, we quit the interpreter using `quit()`. Some other useful features about iPython that come in handy later:

1. A `?` after a variable tells you all about that variable. For example:
```
a = 5
a?
```
2. `!` before a command tells iPython to run it in the terminal. For example, `!mkdir <dir>` creates a new directory called `<dir>`

Running iPython Notebook

To run iPython Notebook, use the command `ipython notebook`. This will open up a new tab in your browser at the *Dashboard*, which is essentially a list of all your notebooks. If you would like, you can make a new notebook using the "New Notebook" button in the upper right. Here, you can experiment with the capabilities of iPython. In lieu of that for this lab, we will continue working with the example we went over in class, but this time you can work on your own machine!

To access the example notebook from class:

1. Keep the iPython Notebook open on your browser.
2. Go to the course website (`http://huttenhower.sph.harvard.edu/bio508`) and download "Activity 01: Multiple sample hypothesis test UPDATE" saving it somewhere on your computer.
3. In iPython Notebook, drag the file you saved to the list of notebooks.
4. Hit the blue upload button on the right of the new notebook.
5. Now click on the new link.

Tables of Utilities

Command Line

Task	Command	Usage	Example	Notes
Change the current directory	`cd`	`cd <path>`	`cd c:\Users\eschwager\Desktop\problems02`	`<path>` can be absolute or relative
See files in the current directory	`dir`	`dir`	`dir \Users \eschwager`	`dir <path>` will show the files in `<path>`
Launch an application or open a file	program or file path	`<file>`	`python problems02.py`	This will execute the file; to edit a file, use the `edit` (or `start` on 64-bit versions) command
Move a file	`move`	`move <file> <path>`	`move problems02.py c:\Users\eschwager\Desktop`	***CAREFUL:*** If `<path>` is a file name, the original file will be replaced!
Delete a file	`del`	`del <file>`	`del problems02.py`	***CAREFUL:*** This is *complete* deletion; it does not go to the recycle bin, it just disappears
Rename a file	`ren`	`ren <old> <new>`	`ren problems02.py ILikeToast.py`	`<old>` and `<new>` are file names
Create a new directory	`mkdir`	`mkdir <path>`	`mkdir ..\problems03`	If `<path>` exists, you will get an error but not hurt anything
Keyboard interrupt	CTRL-C		`grep` CTRL-C	Use to kill a command or process that's stuck

Table 2: Commands for performing some common tasks using the Windows command line.

Task	Command	Usage	Example	Notes
Change the current directory	`cd`	`cd <path>`	`cd ~/Desktop/problems02`	`<path>` can be absolute or relative
See files in the current directory	`ls`	`ls`	`ls ~/Applications`	`ls <path>` will show the files in `<path>`
Launch an application or open a file	program or file path	`<file>`	`python problems02.py`	If your file is not executable, open it using the `open` command
Move a file	`mv`	`mv <file> <path>`	`mv problems02.py ~/Desktop/problems02`	***CAREFUL:*** If `<path>` is a file name, you'll overwrite the file!
Delete a file	`rm`	`rm <file>`	`rm problems02.py`	***CAREFUL:*** This is *complete* deletion; it does not go to the trash, it just disappears
Rename a file	`mv`	`mv <old> <new>`	`mv problems02.py ILikeToast.py`	`<old>` and `<new>` are file names
Create a new directory	`mkdir`	`mkdir <path>`	`../problems03`	if `path` already exists, you will get an error message but not hurt anything.
Keyboard interrupt	CTRL-C		`grep` CTRL-C	Use to kill a command or process that's stuck

Table 3: Commands for performing some common tasks using the MacIntosh command line.

Action	Windows Command	MacIntosh Command
Repeat last command	Up arrow (↑)	Up arrow (↑)
Complete a partial file name	Tab	Tab
Move to the start of the line	Home key	CTRL-A
Move to the end of the line	End key	CTRL-E

Table 4: Miscellaneous useful shortcuts for Windows and MacIntosh command line.

jEdit

Task	**Shortcut**	**Menu**	**Effect**
New	CTRL-N	File	Create a new, empty file
Open	CTRL-O	File	Open an existing file
Close	CTRL-W	File	Close the current file
Save	CTRL-S	File	Save the current file
Save as		File	Save the current file with a new name
Undo	CTRL-Z	Edit	Undo the previous action
Redo	CTRL-E CTRL-Z	Edit	Redo previous undone action
Cut	CTRL-X	Edit	Remove selection, place in clipboard
Copy	CTRL-C	Edit	Copy selection to clipboard
Paste	CTRL-V	Edit	Insert the contents of the clipboard
Select all	CTRL-A	Edit	Select the entire current file
Go to line	CTRL-L	Edit	Move cursor to line number
Find/Replace	CTRL-F	Search	Find/replace the given text
Next Editor	CTRL-Page up	View	Activate the next open file
Previous Editor	CTRL-Page down	View	Active the previous open file
Indent Selection	ALT-Right Arrow	Edit/Indent	Move selected lines one tab to the right
Unindent Selection	ALT-Left Arrow	Edit/Indent	Move selected lines one tab to the left
Select Text	Shift-Left/Right Arrow Shift-Home/End Shift-Page Up/Down		Select text while moveing the cursor
Move by Word	CTRL-Left/Right Arrow		Move cursor by word rather than by character
View Line Number			Watch the lower left corner of your window

Table 5: Shortcuts for common commands in jEdit; on a Mac, the Command key replaces CTRL.