

Computer Lab 1a - UNIX and Data Plotting

The computer labs will be in Room G98 UTL, Mudd Hall. Access to this room is restricted by JHU card and as a member of the class your card will be validated.

In this computer lab we will interact with the computers using a terminal window and the command line rather than through interaction with a graphical user interface (GUI). We will "type raw text on a naked command line" and avoid the whole skeuomorphic debate.

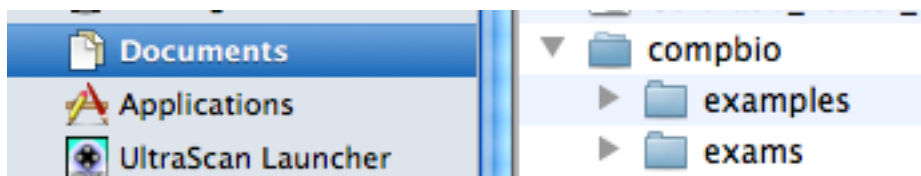
We will use two flavors of a UNIX-like operating system: LINUX and Mac OSX. Most simulations will be calculated on a remote computer cluster running LINUX; data will be transferred back to the Mac's for analysis, plotting and display.

Access to the cluster will be through the campus network using a terminal window on one of the Mac computers in the UTL (or your own laptop). The Mac's will also be used to display molecular models and plot simulation results.

Warning: If you are new to UNIX and a networked environment this lab will be somewhat bewildering. Don't worry - learning UNIX is like learning to ride a bicycle. Once you learn UNIX it is second nature and you never forget it.

The most difficult thing for many people is to think in terms of a directory tree rather than folders as displayed in the GUI. Consider the following terminology.

One **hierarchy** of folders in my Mac laptop home directory is:



Think of this as a directory tree that looks like this in UNIX:

```
Documents/compbio/examples/  
Documents/compbio/exams/
```

We say that *compbio/* is a **subdirectory** of *Documents/* and that both *examples/* and *exams/* are subdirectories of *compbio/*. If we are in *compbio/* we go **down** one level to *examples/* and if we are in *examples/* we go **up** one level to *compbio/* and **up** two levels to *Documents/*.


Also, if we are in *exams/* the **full path** is */Users/fleming/Documents/compbio/exams/* where */Users/* is a **top level** directory (notice the leading slash).

The other problem many people have is remembering where they are in the directory tree: Which computer am I on and where am I on the directory tree on that computer? Sounds trivial but it isn't. You will learn how to keep track of where you are in the exercises below.

I. Mac Account

You may sit at any one of the computers in room G98 to carry out the computer labs. To login to the Mac you should know your official JHED login name and password.

Throughout this course we will use different fonts in the lab guides (like this document) to indicate the following:

- **Courier-bold**, text that you should type on the computer.
 - `Courier-normal`, text displayed in the terminal window.
 - **Helvetica**, text seen on widgets and GUI menus that you click.
 - `Times-normal`, text that is descriptive.
 - **Times-normal**, highlighted questions with answers emailed to the TA (achin14@jhu.edu).
 - *Times-italics*, names of directories and files on the computer.
 - `[filename]`, means the actual filename **without the brackets**. The brackets indicate a variable name.
1. To login on a Mac enter your JHED ID and password.
 2. Click the **Terminal** icon in the Dock to launch a terminal window. (If you don't have a terminal app icon in the Dock, go to the **Applications/Utilities** folder, find the terminal application icon  and click it). Launching a terminal window is the first action you should perform whenever you log in to the Mac. Most programs used in this course will be started by typing commands in the terminal window. Enter the following command in the terminal window,

```
echo $0
```

3. Go to the course website on the **Documents | Lecture Notes and Lab Guides** page and click on the [tcshrc file](#) link next to the PDF link for this document. Save this file to your iMac home directory. Then in the terminal window enter the following command

```
mv tcshrc.txt .tcshrc  
(Note the dot before the last file name.)
```

Now click on the **Terminal | Preferences | General** menu, click the button labeled **Command (complete path)**: and enter the following in the space,

```
/bin/tcsh
```

Close the terminal application and relaunch it. Then enter the following command again,

```
echo $0
```

[Pause here while we go over UNIX shell environments, and setting up your terminal app defaults]

4. More terminal windows may be opened at any time by pressing the **⌘** and **n** keys simultaneously (the cursor must be active in a terminal window to do this), or by using the **File** pull down menu and clicking **New Shell** item.
5. If you have a flashdrive insert your flash drive in the back of the Mac. In the Terminal window enter this command,

```
cd /Volumes/[your_FlashDriveName]
```

You are now in your “home directory on the Macs”. You should start here every time you log in. All files saved on the Macs should be under */Volumes/[your_FlashDriveName]* or they may be lost when you log off! If you don’t want to use a flashdrive you can just work in the default home directory given to you when you open a terminal window. But don’t forget to save your work at the end of class on JHBox, Dropbox, Google Drive, etc.

II. Cluster account

From the Mac we also will want to connect to the remote computer cluster in the Department of Biophysics. While on the cluster we will learn some common UNIX commands.

1. We use the secure shell (**ssh**) protocol to connect to a remote computer. In a terminal window login to the cluster by entering the following two commands (those in courier bold font),

```
ssh compbio2@kirin.kit.jhu.edu (Answer yes to any questions.)  
compbio2@kirin.kit.jhu.edu's password: pfleming
```

then immediately enter,

```
tcsh (This command sets up your environment on the cluster.)
```

Q1: What is the default shell for terminal? Can you name some other types of shells that UNIX can run with?

After connecting to the cluster using the **ssh** command you may navigate the directory tree (i.e. organization of folders), create and edit files, move files, remove files, start programs, etc.; but you will do all these things only with the command line in a terminal window. Programs on the cluster with graphics displays will not send the display back to the Mac.

Change to your home directory on the cluster (this should be there under the *compbio/* directory),

```
cd [yourJHED_ID]
```

You should start here every time you log in to the cluster. You can make new directories within (under) your home directory as shown below.

Note that on the cluster you have to enter *tcsh* as the first command every time you log on. This is not necessary on the Mac's because you changed the terminal preferences and that should be good until IT wipes your home directory.

2. Practice the following UNIX commands after you are on the cluster.

```
pwd
```

- print current working directory

```
ls
```

- list contents of this directory

```
mkdir topic1
```

- make new directory called *topic1*

```
cd topic1
```

- change to directory *topic1*

```
pwd
```

- make sure you changed to the correct directory

```
cp ../../Shared/helix.pdb .
```

- copy file called *helix.pdb* from the *Shared/* directory here. The *Shared/* directory is two levels up and so you needed to specify the **relative path** with *../../*

```
ls
```

- list contents of this directory

```
less helix.pdb
```

- display contents of file, *helix.pdb* [**less** is a file viewing program]

```
[press return]
```

- scroll down one line

```
d
```

- scroll down one page

```
b
```

- scroll up one page

```
q
```

- quit

```
man less
```

- read documentation on the command **less**. (Scroll down and up with **d** and **b** as above).

```
q
```

- quit the man page viewer

```
cd ..
```

- change up one directory (what directory are you now in?)

```
exit (enter twice, once to exit tcsh and once to exit kirin)
```

- log off the cluster (you should now be back to the Mac)

A list of UNIX commands and utility programs may be found at <http://www.oreillynet.com/linux/cmd/>. (Note: This site is linked from the course web site also). Although this looks like a daunting list, if you work within a UNIX environment a familiarity with these commands will make your life much easier.

Note: You can also read man pages on the Mac (it is a UNIX-like machine). For example, just type

```
man mkdir
```

while in a terminal window on the mac or *any* UNIX-like computer.

Note 2: You may use either **more** or **less** to view file contents– your preference (see the man page for **more**).

III. Fetch files to the Mac using secure FTP

Now we will learn how to fetch files from the cluster to a directory on the local Mac. The “fetch” program is called **sftp** (secure file transfer protocol). This protocol opens a connection from one computer to another (usually from the Mac to the cluster in this course but it could be to any computer in the world). You use this connection to “get” files from the remote computer and deposit them on the local computer or “put” files from the local computer and deposit them on the remote computer. You can use sftp ONLY to “get” and “put” files, not to manipulate or edit files on the remote computer.

On the Mac terminal window type the following:

```
cd /Volumes/[your_FlashDriveName]
```

or just

```
cd (if you are not using a flash drive)
```

-returns to your home directory. You should be there already but this is insurance.

Q2. How do you print to the screen the name of your current working directory to find out where you are?

```
mkdir lab1
```

- make new directory called *lab1* (If the directory already exists you will be told of this, just continue with the next command).

```
cd lab1
```

- change to that directory

```
sftp compbio2@kirin.kit.jhu.edu (and enter the password  
pfleming)
```

- open file transfer protocol session with your account on the cluster

ls

- view what is in the directory you just arrived at (you should see everyone's home directory)

cd [yourJHED-ID]

- change to your home directory on the cluster

ls

- list contents of this directory (on the cluster), you should see the subdirectory *topic1/* that you created in the previous ssh session

cd topic1

- change to that directory (on the cluster)

ls

- list contents of this directory (on the cluster)

get helix.pdb

- fetch the file called *helix.pdb* from the cluster

bye

- end ftp session

The file named *helix.pdb* should now be on the Mac in the directory you were in when you started the FTP session (*lab1/*). Display it using **less** to be sure it contains data that looks like this:

```

COMPND      VAL
SEQRES  1  13  VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL
ATOM    1  N   VAL    1    0.000  0.000  0.000  1.00  0.00
ATOM    2  CA  VAL    1    1.458  0.000  0.000  1.00  0.00
ATOM    3  C   VAL    1    2.009  1.422  0.000  1.00  0.00
ATOM    4  O   VAL    1    1.745  2.201 -0.916  1.00  0.00
ATOM    5  CB  VAL    1    2.022 -0.759 -1.215  1.00  0.00

```

To summarize: **ssh** is used to log in to a remote computer and manipulate and edit files, run programs, etc. It is the same as being on the remote computer. **sftp** is used to connect to a remote computer for transferring files. You can navigate the directory tree but you can NOT edit files or run programs using sftp - you can only **get** or **put** files.

IV. Create and edit a data file using the vim editor



One of the most common tasks in computational studies is the editing of files. There are many file editors one may choose but these editors are not always installed on a computer or may not work over the network to edit files on a remote computer. We will use an editor called **vim**. It is on *every UNIX type computer in the world (!)*, it works remotely, it is very powerful and once you learn it you will never need any other editor. It does take some getting used to if you want to do fancy stuff; but simple editing is fairly easy once you learn a few keystrokes.

On the Mac in your *lab1/* directory type the following:

```
vim test.dat
- This starts a vim editing session and creates a new file named test.dat
i
- this enters "insert mode", starting before the cursor
0.0    0.0
1.0    1.0
2.0    4.0
3.0    8.0
4.0    16
5.0    25.0
- enter two columns of data, use "tab" between the two entries on a line
esc
- leave insert mode
:w
- write the file to disk
:q
- quit vim session (Only works after you escape (esc) the insert mode.
  Thus, the Twitter comment above.)
less test.dat
- take a look at the data and notice that the 8.0 should be 9.0 (the second
column is the square of the first) and that we forgot the decimal after 16. We will
re-open the file with vim and make the corrections.
vim test.dat
- open file for editing and use arrow keys to place the cursor over the 8
r9
- replace the 8 with a 9
- Now use the arrow keys to place the cursor over the 6 of the 16
a
- enter "append mode" after the cursor
.0
- add the necessary characters
esc
- leave insert mode
:wq
- write the file and quit the editing session with one command.
less test.dat
- look at the data again to inspect your corrections
```

A full tutorial on using the **vim** editor may be found at <http://pages.jh.edu/~pfleming/compbio/files/MasteringVI.html>

V. Plot data using **xmgrace**

Most data created by molecular simulations and their analyses consists of files containing columns of numbers. We either want to plot the columns as graphs or display the column data as molecular graphics images. Today we will learn how to plot two columns of data (x, y data) using a version of the program GRACE called **xmgrace**. Again there are many plotting programs available. We will use **xmgrace** because it is free, works on any UNIX type computer, is very powerful and is the default plotting program for VMD.

In a terminal on the Mac go to the directory containing the file, *test.dat*.

Launch the **xmgrace** application with the following command

/opt/local/bin/xmgrace

Now grab the **xmgrace** window border at the lower right corner and expand the window to display the complete axes.

Now import the data file

Click Data | Import | ASCII

Highlight *test.dat* and click OK

Click cancel

Click Plot | Set appearance to open the menu box

Click on Symbol properties | Type | None and choose Circle

Click on the Apply button

Click Close

- label the graph

Click Plot | Axis properties

in Label String box write: "**x Axis**"

Click Apply

Click Edit: X axis box and choose, Y axis

In Label String box write: "**y axis**"

Click Accept

Click Plot | Graph appearance

In Title Box write: "**Graph Title**"

Click Accept

Click File | Save

and give a file name ending in **.agr** (e.g., *test.agr*) at the end of the directory path in the bottom box labeled, **Selection**.

Click OK

Click Exit to end the **xmgrace** session.

Inspect the contents of your present working directory (**ls**).

You should have a file with a **.agr** extension.

Re-display the saved plot by typing,

/opt/local/bin/xmgrace [filename].agr

where [filename] is the name you chose in the steps above.

You will be using Xmgrace for some homework assignments. **All plots handed in should have a title and have the axes labeled!**

VI. Manipulating files

For some homework assignments you will be asked to plot x, y data but the file you work with may have many columns in it not just two. **Xmgrace** wants a file with just two (or three) columns of data so we want to learn how to extract two specific columns from a multiple column data file and put the two columns in a new file. We will use a utility called **awk** that is on all UNIX type computers to accomplish this extraction.

First, here is how you would extract the first column from a multiple column data file:

```
awk '{print $1}' [data_file]
```

Where the syntax `[data_file]` means whatever the file with data is called. This command would display column 1 ($\$1$) of the data file to the screen. Try the command on the file *test.dat* created previously.

Did column 1 appear on the screen?

Q3. How do you view *test.dat* to check this?

Now use the `awk` commands to display column 2 of the file *test.dat*. We can't do anything with the data on the screen so we want to **redirect** it to a new file. Here is the command to do this:

```
awk '{print $1}' [data_file] > [new_file]
```

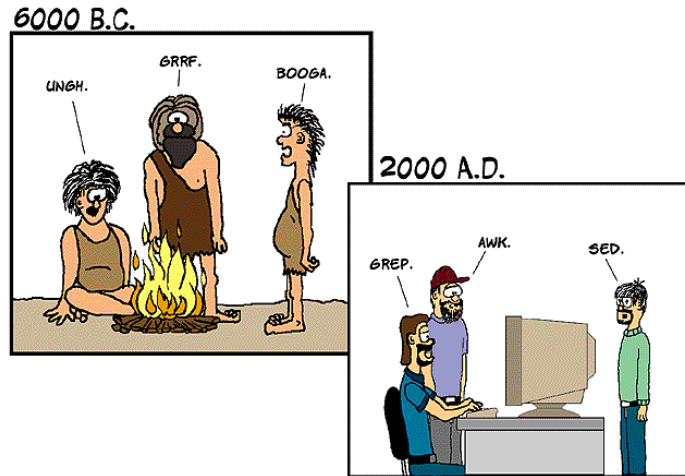
The ">" character will "redirect" the output that normally goes to the screen to the file called whatever you type as the name, *new_file* (you don't use the brackets). Try this command with your file called *test.dat* to create a new file with only the first column of *test.dat* and call the new file *col1.dat*.

Here is how you would extract columns 1 and 4 from a multiple column file and put them in a new file for plotting with `xmgrace`:

```
awk '{print $1, $4}' [four_column_file] > new_file.dat
```

I called the new two column file *new_file.dat* because **xmgrace** likes to see a *.dat* extension (but this is arbitrary and not necessary).

EVOLUTION OF LANGUAGE THROUGH THE AGES.



COPYRIGHT (C) 1999 ILLIAD

[HTTP://WWW.USERFRIENDLY.ORG/](http://www.userfriendly.org/)

Q4. What is wrong with the above cartoon? (Hint: As Justin Trudeau said, “This is 2016.”)

Use **sftp** to the cluster and fetch the file *peptide_SS.dat* back to the directory *lab1* on the Mac. The easiest way to do this is to change to directory *lab1* on the Mac and do the **sftp** session from that directory (do the same as above for the **sftp** session). Once you are logged on to the cluster in the */home/compbio/* directory using **sftp**, change to the *Shared/* directory and get the file,

```
cd Shared/
get peptide_SS.dat
```

Use the **bye** command to exit the **sftp** session on the cluster and come back to the Mac.

The file *peptide_SS.dat* contains 5 columns that represent the secondary structure propensities of each residue in a 21 residue peptide. The column data are arranged as follows:

- Column 1: Residue number in the peptide
- Column 2: Helix propensities
- Column 3: Sheet propensities
- Column 4: Turn propensities
- Column 5: Coil propensities

Inspect the file using the **less** command. Notice that for each residue the values in columns 2-5 add to 1.0.

Use **awk** to make four separate files containing two columns. Each file will contain the residue numbers in column 1 and the propensities for a type of secondary structure in column 2 (e.g. make files called: *helix.dat*, *sheet.dat*, *turn.dat*, *coil.dat*).

- Plot these data using **xmgrace** on a single plot
- Use different line types for each secondary structure (in Plot | Set Appearance | Line Properties)
- Add a legend to label the different data curves (in Plot | Set Appearance)
- Label the axes
- Provide a title for the plot
- Save the plot in a file with an *.agr* extension

Show your plot to the instructor.

Note: You cannot use **xmgrace** when remotely logged on the cluster. It has a graphical interface that will not be displayed back on the Mac. (If you are confused about which machine the terminal is currently active on type, **hostname**. The name of the active computer will be displayed).

VII. Continued tutorials

Open a browser (click on the Safari icon in the Dock and go to pages.jh.edu/pfleming/compbio/links.html and spend as much time during and after the lab as needed with the following

1. Work through the tutorial **Mastering VI** from the above links page.
2. From the above links page (or using the man pages in a terminal window) peruse the following UNIX commands that are available and that we will use in the course: **gzip, tar, less, head, tail**.

VIII. For the experts

Edit your *.tcshrc* file so you can just type **xmgrace** instead of **/opt/local/bin/xmgrace** every time.