Homework Assignment 1

DUE in class, Tuesday, September 19

Purpose:

The goal of the first homework assignment is to help you to become more comfortable with the Python language and a little data analysis. The method for doing this will be to explore the form of the Lennard-Jones potential (1).

$$U(r(i,j)) = 4\varepsilon\left(\left(\frac{\sigma}{r(i,j)}\right)^{12} - \left(\frac{\sigma}{r(i,j)}\right)^{6}\right) \qquad (1)$$

where *r*(*i*,*j*) is the distance between the centers of two atoms (atom *i* and atom *j*), *ε* is a measure of the interaction energy at optimal distance and σ is a measure of the atom diameter. You will become intimately familiar with this formula during the course.

Background Information:

This expression describes the potential energy between two atoms. The equation includes both an attractive term (negative potential energy) and a repulsive term (positive potential energy). The Lennard-Jones potential is an important equation for modeling the behavior of atoms and is the first force field used in the course (the concept of a force field will be explained later).

In order to get a grasp for the shape of the potential described by equation (1), simulate some data using the python program, *LJ.py*, which you can fetch from the *Shared/* directory under the 2 directory on the cluster (kirin).

Insert your flashdrive into the Mac, open a terminal window, cd to your home directory on the flashdrive, and make a subdirectory for the homework assignment, go there and then fetch a file called *LJ.py* from the cluter.

```
cd /Volume/[JHED-ID]/
mkdir hmwk1
cd hmwk1
sftp compbio2@kirin.kit.jhu.edu
```

Enter the password, `pfleming`

```
cd Shared/
get LJ.py
bye
```

Type the following on the command line of your terminal window

```
python3 LJ.py
```

This tells the computer to run the program *LJ.py*
You should see four columns of data printed onto your screen.

Now, before you continue, take a look at the program you just used

```
less LJ.py
```

You can use the **arrow keys** to move the cursor through the text, then hit **q** to exit back to the command line. Take a moment to read through the code, understanding the flow of information. What instruction is at the end of *LJ.py*? As you can see, if we ran *LJ.py* we would expect to have columns reporting the atom separation $r(i,j)$, the attractive term, the repulsive term, and the total Lennard-Jones potential printed out on the screen for each $r(i,j)$.

Now type

```
python3 LJ.py > LJoutput.dat
```

This tells the computer to run the program *LJ.py* again. But rather than the output being printed to the screen as before, the output is written to the file *LJoutput.dat*.

Now take a look at the output file *LJoutput.dat*.

```
less LJoutput.dat
```

You should see 4 columns of data. You can hit **return** to move down one line at a time, hit **space bar** to move one screen at a time, or hit **q** to exit back to the command line.

Review the use of the **awk** command in UNIX. For example

```
awk '{print $1,$3}' LJoutput.dat > LJrepulsive.dat
```

This statement copies columns 1 and 3 from *LJoutput.dat* and writes them to *LJrepulsive.dat*. Use the **less** command to look at *LJrepulsive.dat* and *LJoutput.dat* to confirm this.

Assignment:

Run the *LJ.py* program and output the attractive, repulsive and overall energy terms of the Lennard-Jones potential to a file. For each distance, plot each potential energy term, and the total potential energy on a graph using **xmgrace** (a total of 3 lines per graph). [Label your plots with a title, subtitle giving the epsilon and sigma values, legend and axes with appropriate units (see below).]

One result that should be clear from your plot is that the magnitude of LJ interaction energy between two atoms is small. But imagine we had a box of many atoms (or a protein of many atoms) and we could change the average distance between atoms. What does the **total system energy** look like as the average atom-atom distance changes? Change your code so that the output is for the total of 100 pairwise interactions at each distance in the plot. The easiest way to do this is to multiply the `tot_LJ` variable by 100.0 before the `print` statement. In fact, do this before the cut-off so that your plot will be scaled appropriately. Edit and run the program again, output to a new file, and plot the new data as above with the **total** LJ potential energies for one atom pair and the **total system** LJ potential energy for 100 atoms on the same plot.

Email a Word document containing on one page **2 Plots** from the output of *LJ.py* (the components of the LJ potential with total, and the comparison of total LJ energy for single atom pair and 100 atom pairs). Adjust the scale of the axes to include all the data but the scale should be the same on both plots.

Please include your name on the top of the page. No name, no credit.

Think about the following question, we will discuss it in class: Does atom-atom interaction energy (of the type described by the LJ potential function) stabilize folded protein structure as compared to the unfolded state?