Computer Lab4b – Peptide Monte Carlo Simulation

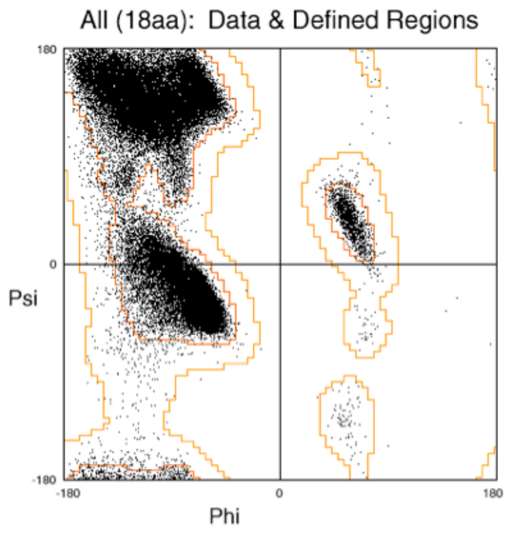
In this experiment, you will carry out a Monte Carlo simulation of a peptide in a vacuum. The force field will be a simple hydrogen bond score. One take-home message for today is that you can create an ensemble of peptide structures distributed according to the energy of the controlling force field; you choose the force field and therefore the distribution of structures. (But, you already knew that!) Lab report answers to the TA (achin14@jhu.edu).

I. Description of LINUS algorithm.

1. LINUS is a suite of Python modules for exploring conformational ensembles of peptides and proteins. You have already used a utility module called **ribosome.py** for building short peptides.

2. The core of LINUS is a torsion angle Monte Carlo simulation. Instead of displacing atoms along a random distance vector as with the Argon simulations, random bonds are rotated a random amount. In our lab experiment today the backbone polypeptide torsion angles ϕ and ψ are rotated to explore peptide conformational space.

   The Monte Carlo torsion angle moves are checked for van der Waals clash between all atoms in the peptide (cf. last lab). Only when there is no VDW overlap is the move tested for the Metropolis criterion, otherwise a different torsion angle move is attempted. The result is that all successful moves are restrained to ϕ,ψ angles allowed in the Ramachandran map. The Ramachandran plot shown in the previous lab was an idealized version made from hard shell molecular models. Actual proteins have "soft shell" atomic radii and some bond flexibility. Therefore, a Ramachandran plot of ϕ, ψ angles found in real proteins looks a little different.



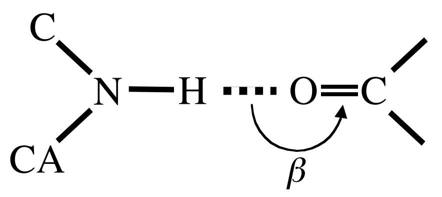
   The black dots in the figure at right show ϕ,ψ values for a data set of residues in the Protein Data Bank of X-ray crystal structures. The outer contour line delimits the allowed region of conformational space for non-proline, non-glycine residues. These allowed regions generally follow the allowed regions shown on the idealized map from the last laboratory.

   You should be familiar with the major regions of a Ramachandran map and

be able to answer questions like the following:

- Where is β-strand region in the figure?
- Where is right handed α-helix region in the figure?
- Where is the left handed α-helix region in the figure?

3. The force field today is a simple hydrogen bond score. This energy function is called a **scoring function** rather than an energy **potential function** because it is not a true analytical potential function such as the LJ function, but rather a simple linear relationship with distance. The hydrogen bond score is both distance and orientation dependent (see the figure below). The score = 1.0 at the optimal hydrogen bond distance (3.5 Å) between the N and the O atoms and decreases linearly to zero at 5.0 Å (We only measure the distance between the "heavy" atoms and ignore the hydrogen). The angle (β) must be 180 +/- 90 degrees to be counted as a hydrogen bond. This scoring function estimates the potential energy of the hydrogen bond, *but in arbitrary units*. We don't assign kcal/mol to this scoring function.



In our experiments today, this function makes up the entire "**force field**" for the simulations; we do not include a Lennard-Jones potential. The conformational population obtained is distributed according to the hydrogen bond score *only*. However, all conformations are sterically feasible because we eliminate those with steric clash before applying the Metropolis criterion.

II. Equilibrate a polyalanine peptide within the "force field".
You will first build two polyalanine peptides, one in strand conformation and one in helical conformation. Both starting conformations will be equilibrated using hydrogen bonding scoring and the minimum energy conformation sampled during the equilibration will be saved (automatically by the program). The point of the experiment is to see whether or not the starting conformation has an effect on the lowest energy conformation found in the ensemble.

Note that you will not be looking at "energy minimized" structures but rather "typical" structures of low energy from an ensemble. There will be variation in your individual results.

1. Login to the cluster and change to your permanent directory (*/home/compbio2/JHEDID/*), initialize the tcsh shell (`tcsh`), make a new directory called *lab4b*, **cd** there and copy the tar archive */home/compbio2/Shared/linus_mc.tar* to your *lab4b/* directory and unpack it.

Change to the new, unpacked *linus_mc/* directory. The command to unpack a tar archive after you copy it to the pwd is,

```
tar xvf linus_mc.tar
```

Or you could unpack it directly without copying it first. For example, go to your new *lab4b/* directory and enter,

```
tar xvf /home/compbio2/Shared/linus_mc.tar
```

2. Take a look at the files in the new unpacked subdirectory, *linus_mc/*. The two files, *Linus_equil.py* and *Linus_prod.py*, you see are command run files (equivalent to *mc_nvt.py* in the Ar simulation). There is no parameter file this time; we will set the important parameters in the command file itself. The actual LINUS module files (equivalent to *modules.py* in the Ar simulations) are installed as part of the Python libraries. They will be imported at the appropriate times and you don't need to worry about them. But since you will be using several of these files today it would be useful to have the following environmental variable set,

```
LINUS = /home/apps/lib/python2.7/site-packages/pylinus_1_0
```

You should have this already set if you launched the tcsh shell. Type `env` to ensure that you have this environmental variable set.

3. Now use the **less** command to view the contents of the file *ala.fasta*. This is an amino acid sequence file in the FASTA format. It consists of a title line and then the sequence using one letter codes for amino acid types. Here we designate the sequence to be 13 alanine residues. This file format is common in bioinformatics and if you download a sequence from a web server it may be in this format. Linus has a utility to convert a FASTA sequence file to a **.rib* file for input to **ribosome.py**. The utility is called **fasta2rib.py** and usage syntax is,

```
python2 $LINUS/utils/fasta2rib.py ala.fasta
```

Enter the above command while in your *linus_mc/* directory on the cluster. (Note: You do not need the redirect symbol “>” here). This command will create a file called *ala.rib*. Inspect this file using the **less** command. 1. What conformation is defined by the default ϕ, ψ values: helix or strand?

4. Create a PDB file called *ala_strand.pdb* in the above conformation using the **ribosome.py** program as described in the last lab.

```
python2 $LINUS/utils/ribosome.py < ala.rib > ala_strand.pdb
```

Inspect the new *ala_strand.pdb* file. Does it look like a PDB file?

5. Inspect the contents of the file *Linus_equil.py*. There are two important parameters for this experiment, the **simulation_parameters** and the **hbond_parameters**. Find the segment of code where these are defined. You should see that we will do 50000 cycles of Monte Carlo moves (where a cycle is an attempt to move every residue in the peptide but not at the same time) and we will save a conformation every 50000 cycles so only the last conformation of the simulation will be saved, i.e., the "ensemble" will have only one structure – since this is only equilibration we are not interested in generating a large ensemble. The minimum energy conformation is also saved by default but this is not apparent from the information so far.

   You should also see that the hydrogen bond score is 0.35. This value is in arbitrary units and has been chosen from previous experiments, i.e., it has been *empirically* obtained. Don't worry about the other information in this command file for now.

6. Edit the *normal-python2* file in the *linus_mc/* directory. Change the name of the job (yourJHEDID) to your actual JHED_ID and make sure the command to be executed is,

   ```
   $PYTHONBIN Linus_equil.py ala_strand.pdb
   ```

   Run the simulation with the following command

   ```
   ./normal-python2
   ```

   The simulation will take 5 to 10 minutes depending on the success of your random moves (use `qstat –u \*` to check if your job is running). When your job is finished move the *sim_equil/* directory to a new one as follows (make sure your simulation is finished before you do this),,

   ```
   mv sim_equil/ sim_strand_equil/
   ```

   Then skip to section 7 below, get your second equilibration going and come back here to continue.

   When back here change to the newly created subdirectory called *sim_strand_equil/* in your *linus_mc/* directory. The important file in this directory for now is the minimum energy file called *ala_strand.min* (it is actually the peptide conformation in PDB format but we put a different extension on the filename). The other file here is *ala_strand.pdb* which is a copy of the starting structure (equivalent to *init.pdb* in your Ar simulations).

   Make a new directory in your home directory (or flashdrive) on the Mac called *lab4b* and bring both *ala_strand.min* and the starting structure (*ala_strand.pdb*) back to the Mac using **sftp**. View the two structures of the peptide using PyMOL.

2. Is the minimum energy structure mostly a strand or mostly a helix? (Hint: ala_strand.min S I cartoon).
3. How does this compare with the starting structure (*ala_strand.pdb*)?
This next question is key to understanding this lab: 4. **Why would the structure seen in *ala_strand.min* have a favorable (minimum) energy?**

7. On the cluster, edit the *ala.rib* file so the default ϕ, ψ values define a helix (The appropriate values for these are in the *.rib* file from last lab) and repeat steps 4-6 above to start this helix simulation (use *ala_helix.pdb* as the name of your PDB file and don't forget to change the command in *normal-python2* so that it uses the helix structure as the starting structure). While it is running go back and deal with the strand simulation.

   After running the helix simulation (step 6 above) move the output directory to a new one (make sure your simulation is finished before you do this),

   ```
   mv sim_equil/ sim_helix_equil/
   ```

   Skip to part III, get your production simulation running then come back here to answer questions 5-8 while your production job is running.

   Bring back to the Mac the two new PDB files (*ala_helix.min* and *ala_helix.pdb*), inspect with PyMOL and answer the following questions:
   5. What conformation is the new low energy structure *ala_helix.min*; mostly helix or mostly strand?
   6. Is it similar to the previous low energy structure from the strand starting conformation in terms of helix content? Should it be different?
   7. Does the starting structure have an effect on the equilibrated minimum energy structure in terms of whether you end up with helix or strand?
   8. Do you think the simulation followed a Markov process?

III. What is the energy distribution of the peptide MC ensemble?
In this experiment, you will do a simulation of the polyalanine peptide using torsion angle Monte Carlo as implemented in the LINUS suite of programs and plot the distribution of energies found in the resulting trajectory of conformations. The point of the experiment is to determine if one can obtain a distribution of energy states using torsion angle moves and a simple hydrogen bond scoring function.

1. On the cluster inspect the command file *Linus_prod.py*. How does it differ from *Linus_equil.py*? Hint: Use the UNIX command **diff** as in

   ```
   diff Linus_prod.py Linus_equil.py
   ```

   You should see that we will save every 5th accepted conformation (with 50,000 cycles of attempts) to create a trajectory of structures and that the output directory will be called *sim_prod*.

2. Run the production command file on one of your equilibrated minimum energy peptide PDB files as follows:
   Edit the *normal-python2* file so that *Linus_prod.py* is the command file.
   The *normal-python2* file expects the starting structure to be in a file with the name *ala.pdb*. To make your simulation starting structure your "equilibrated" energy minimum structure do the following: In your *linus_mc/* directory enter,

   ```
   cp sim_helix_equil/ala_helix.min ala.pdb
   ```

   Then submit your simulation to the queue. After your simulation is finished you should have a newly created directory called *sim_prod/*. The trajectory containing multiple conformations is in the *ala_traj.pdb* file in the *sim_prod/* directory. Look at the contents of this file. Notice that each new conformation starts with the word `MODEL` and ends with the word `ENDMDL`.

   9. How many conformations should be in this file? You can check the number of conformations using the following UNIX commands,

   ```
         grep MODEL ala_traj.pdb | wc -l
   ```

   10. If you don't remember what the commands **grep** and **wc** mean how do you find out?
   Bring the *ala_traj.pdb* file back to the Mac using sftp.

3. Plot a histogram of the distribution of energies for each conformation in the trajectory. Fortunately, the simulation output included the energy for each conformation in the header region of each PDB segment in your *ala_traj.pdb* file. (Remember, these are not meant to be real energies – just arbitrary scores from the scoring function.) You can see the first few lines of the trajectory by using the UNIX command **head**,

   ```
         head ala_traj.pdb
   ```

   From the result of this command you should see that the energy value is in the fifth column of a line starting with the word MODEL. We can extract a list of the energy values using the following commands,

   ```
         grep MODEL ala_traj.pdb | awk '{print $5}' > ener.dat
   ```

   Plot the *ener.dat* data using your favorite plotting program and make a histogram For xmgrace you can use the same commands that you learned in Lab1c_PythonData. (Hint: Use the range –5 to 1 with #bins = 20 for the histogram). You should see a psuedo bell-shaped curve but it will be truncated at energy = 0 because this state represents the absence of hydrogen bonds and it is the state with least favorable energy when the only energy component is hydrogen bonds. The point is to determine if you obtained a reasonable distribution of energies in your trajectory.

Clearly the more hydrogen bonds the lower the potential energy. From a theoretical point of view the system would not run downhill to the lowest energy conformation (maximum number of hydrogen bonds) because the higher energy conformations have more ways to achieve that higher energy. This entropy effect balances the downhill energy tendency to create the Boltzmann distribution (very approximate in this short simulation with a discrete value scoring function as the only force field term).

11. How is this “escape” from the downhill energy tendency achieved in your simulation?

Now we ask the question: What conformations were explored in your simulation? For the ergodic hypothesis to be satisfied we should have sampled all possible conformations. We did not quite do that in the small number of attempts in this simulation but you can get an idea of the conformations sampled by looking at a movie of your trajectory. Try this:

```
pymol ala_traj.pdb
ala_traj   H l everything
ala_traj   S l cartoon
Movie l Frame Rate  l 5 FPS
```

And use the movie controls in the lower right to play the movie. Because LINUS uses torsion angle MC the first residue is always in the same position and it is hard to view the trajectory. Let's superposition all the structures using a least squares algorithm to fit the CA atoms in each frame to the first structure. Stop the movie and in the PyMOL command window enter the following,

```
PyMOL> cmd.intra_fit("name ca", 1)
```

and play the movie again. Pymol won't calculate secondary structure on the fly for the trajectory so you will have to interpret the worm to see where the helices are. 12. Would you say that many different conformations were sampled or not?

For homework assignment #6 you will address the question of conformational sampling in a more quantitative way. Save the directory *sim_prod/* and the *ala_traj.pdb* file on the cluster to use in hmwk6.