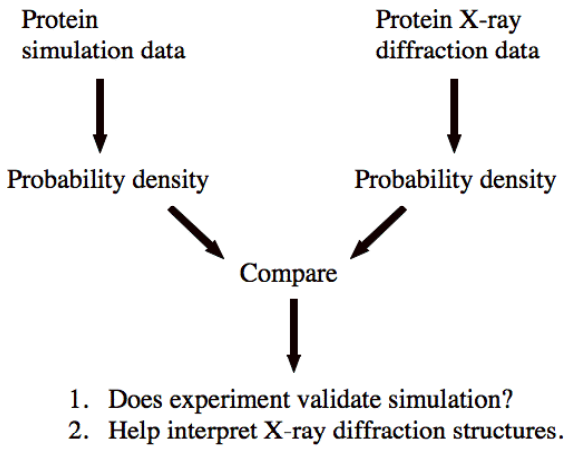
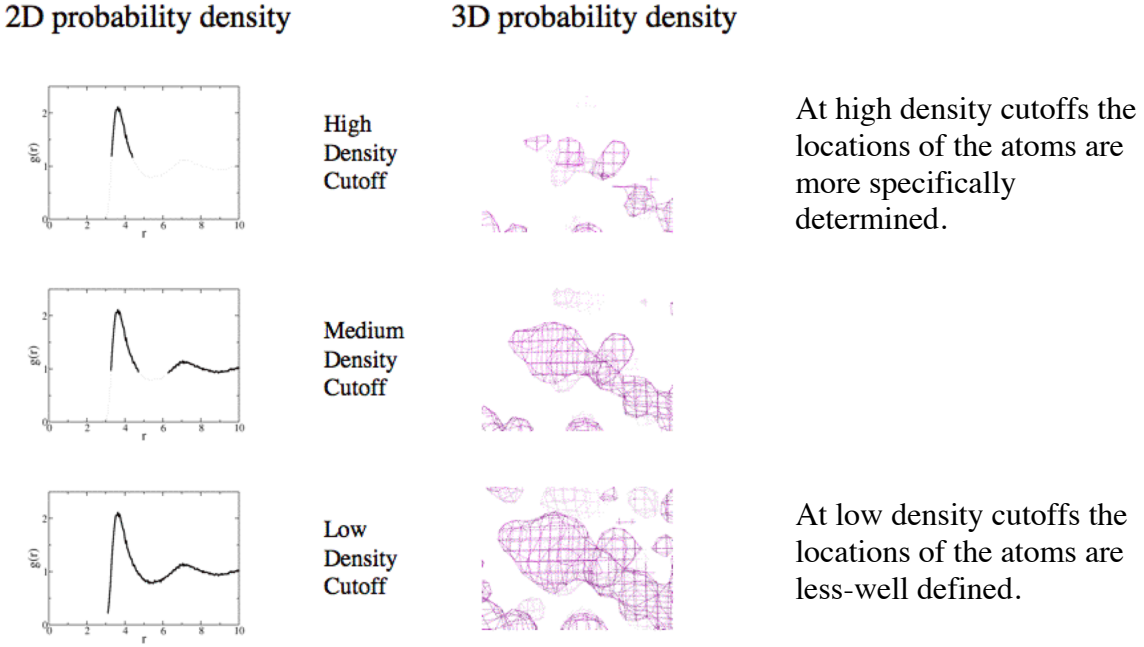
Computer Lab 5 - Protein Simulation in Explicit Water

Part A: Introduction, Solvation, Equilibration

I. Introduction. The goal of this lab project is to understand the conformational diversity which may be represented by a static protein model. We will obtain a Boltzmann energy distributed ensemble of protein conformations; calculate a probability distribution model of the protein from the ensemble; and compare the ensemble-derived model to the X-ray diffraction model of the protein. The comparison is done at the level of probability density maps. The lab is described in schematic form in the following figure,



You have already converted simulation data into probability densities using the results from the argon simulations. A conceptual comparison of the argon probability density plots with X-ray crystallographic probability densities is shown in the figure below.



The RDF plot on the left above is a 2D mapping of the *relative* position probabilities of atoms. The magenta electron density map above is a 3D mapping of the *absolute* position probabilities of atoms. The latter is from a cubic lattice where the value at each vertex is the probability of finding an atom at that 3D position. The contour lines (magenta) are drawn to connect vertices of similar probabilities.

You will make a 3D probability density map similar to the electron density map above from your protein simulation. Imagine that you take each configuration from a protein simulation and place it, in turn, in a 3D cubic lattice. You sum the number of times an atom was close to each lattice vertex and then calculate the probabilities of finding an atom at each vertex. By connecting vertices of similar probabilities you can create contours that define the 3D probability map.

An ensemble of protein conformations will be obtained from a molecular dynamics simulation of the protein in a bath of water; this is called an all-atom simulation in contrast to the case where solvent is modeled with a continuum mathematical description. Rigorous potential energy calculations will be performed at each time step of the simulations using a complete force field.

The project is scheduled to take three lab sessions. During the first session we will setup our system and start the equilibrium and production runs. During the second session we will analyze the production trajectory. For the third session we will calculate a probability density map from the ensemble of protein structures and compare the calculated density map to the X-ray crystal electron density map.

The molecular graphics programs VMD and PyMOL will be used during this project. The CHARMM force field will be used. We will use the program NAMD to carry out the simulations of a protein in water. NAMD has become one of the standard software tools in molecular simulation research. It is free for academic users and more information can be found on the web site, www.ks.uiuc.edu/Research/namd/.

Answer the yellow highlighted questions and send the answers to achin14@jhu.edu.

II. Protein Simulation.

Note: Some of your simulations for the final project will take days to complete. You may want to check on them and restart jobs when you are away from campus. In order to access kirin when not on the JHU network you will need to install a VPN client on your laptop. There is a link to do this from the left menu bar on your myJHU page.

A. Construction and equilibration. The protein we will study is a small 58 residue protein that functions as a protease inhibitor. It binds tightly to the enzyme trypsin and the crystal structure we start with is actually a complex of the inhibitor and trypsin isolated from bovine pancreas. The structure of the complex is available from the PDB under the code 2PTC (2-pancreatic trypsin complex). The complete project involves the following steps:

1) View the starting structure using molecular graphics.
2) Put the inhibitor protein (not the enzyme protein) in a periodic boundary primitive cell and fill with water and ions.
3) Equilibrate the system to the desired pressure and temperature using MD.
4) Run a production MD simulation and save a trajectory of conformations.
5) View a movie of the protein undergoing conformational fluctuations.
6) Compare atomic fluctuations in the simulation with calculated fluctuations of the crystal structure.
7) Calculate a probability density map of atomic positions for the ensemble of conformations in the simulation.
8) Compare the simulation ensemble probability map to the crystallographic electron density map.

Now for the specific steps.

1. Login to the cluster and start the tcsh shell (**`tcsh`**). Unpack */home/compbio2/Shared/protsim.tar* in your JHEDID directory under the *compbio2/* directory and change to the new subdirectory called *protsim/*.

```
cd [JHEDID]
tar xvf /home/compbio2/Shared/protsim.tar
cd protsim/
```

2. In the new subdirectory *protsim/*, you should have the following files:
2PTC.pdb - PDB file of the protein complex (enzyme and inhibitor)
top_all27_prot_lipid.inp – CHARMM force field topology file
par_all27_prot_lipid.inp – CHARMM force field parameter file
inh.pdb – PDB file of the inhibitor only
min_equil.conf – simulation command file for NAMD
production.conf – simulation command file for NAMD
run.namd - cluster job queue submission script for NAMD
write_frames.tcl – A VMD script
phi_psi_mean_SD_frames.py – A python script

3. View (**`view`**) the *top_all27_prot_lipid.inp* file; search for the regular expression 'RESI ALA' (**`/RESI ALA`**) and inspect this block of information. It describes the topology and partial charges of alanine. 1. What is a name for the block of lines that start with `"IC"`?
[Note for final project. You may also want to look at the topologies for histidine. In CHARMM there are three different topologies with residue names, HSD, HSE, HSP (there is no HIS). You will have to choose one of these forms of histidines for the final project.]

4. View (**`view`**) the *par_all27_prot_lipid.inp* file; search for the regular expression 'NONBONDED' (**`/NONBONDED`**) and inspect this block of information. This section gives the Lennard-Jones parameters for the different atom types. 2. What are the ε and VDW radius of the first type of carbon, 'C', in the list?

Make a *protsim* directory in your home directory on the Mac, cd there and use **sftp** to fetch the files *2PTC.pdb, top_all27_prot_lipid.inp* and *inh.pdb* from the cluster back to the Mac. View *2PTC.pdb* using PyMOL. Color the molecule by chain ID (**color chain**).

> 3. Which one (color) do you think is the inhibitor? (Hint: The enzyme has an active site pocket and the inhibitor binds to the pocket inhibiting the enzyme.)

Quit PyMOL.

Open VMD by typing **vmd** in the same terminal window on the Mac.
Load *inh.pdb* by clicking File → New Molecule... menu item in the VMD Main window. In the Molecule File Browser use the Browse... button to find the file *inh.pdb*. Load it by pressing the Load button. (Or just type **vmd inh.pdb** if you are in the **tcsh** shell). This is the inhibitor protein alone.

In a separate terminal window open the *protsim/* directory on the Mac and use **vim** to create a file called *prot.pgn* containing the following,

```
package require psfgen
topology top_all27_prot_lipid.inp
pdbalias residue HIS HSE
pdbalias atom ILE CD1 CD
segment PROT {pdb inh.pdb}
patch DISU PROT:5 PROT:55
patch DISU PROT:14 PROT:38
patch DISU PROT:30 PROT:51
coordpdb inh.pdb PROT
guesscoord
writepdb start.pdb
writepsf start.psf
```

This will read the topology file, take the *inh.pdb* file and change some atom names, designate the cysteine disulfide bonds, guess coordinates for hydrogens and write out a new PDB file (*start.pdb*) and a "structure file" (*start.psf*) with bonding information for the protein. Now run your *prot.pgn* script from within VMD by typing the following at the vmd> prompt,

```
vmd> source prot.pgn
```

Having run *prot.pgn*, two new files will now appear in your directory: *start.pdb* and *start.psf*. Check this by typing **ls** in the terminal window. Quit VMD by clicking File | Quit or by typing **quit** in the terminal window with the vmd> prompt.

Now you will add a box of water around the starting structure. Launch **vmd** again without any arguments, and enter the following commands,

```
vmd > package require solvate
vmd > solvate start.psf start.pdb -t 5 -o solvate
```


The "`-t 5`" argument creates the water box with a layer of water 5 Å in each direction from the atom with the largest coordinate in that direction. the "`-o solvate`" argument creates output files with the basename *solvate*. Quit **vmd** and note the names of the new files created in the *protsim/* directory on the Mac.

Now you will add salt to neutralize the system. Launch **vmd** with the solvated system

`vmd solvate.psf solvate.pdb` (Order is important)
Extensions | Modeling | Add Ions
Note the default Output prefix for file names.
Click the radio button: Neutralize and set NaCl concentration to 0.2M
Click Autoionize

Quit **vmd**

Now let's check everything. Launch **vmd** with the ionized system,

`vmd ionized.psf ionized.pdb`
Graphics | Representations
Selected Atoms | **`protein`** <return>, Drawing Method | New Cartoon
Create Rep
Selected Atoms | **`ion`** <return>, Drawing Method | VDW
Create Rep
Selected Atoms | **`water`** <return>, Drawing Method | Bonds, Material | Transparent

This is the system you will simulate. Zoom and rotate the system to inspect it. Use the following commands to check that the autoionize procedure gave you a system with net charge of zero (or very close to zero).

vmd > **`set all [atomselect top all]`**
vmd > **`set charge [vecsum [$all get charge]]`**

A value close to zero should appear on screen. If not something is wrong.

Center the system in the primary cell by entering the following commands at the vmd > prompt (You may have to enter the <return> key to get the prompt.

vmd > **`measure center $all`**
vmd > **`$all moveby [vecinvert [measure center $all]]`**
vmd > **`measure center $all`**

You should now have essentially 0, 0, 0 for the center coordinates.
Write out the centered system,

vmd > **`$all writepdb ionized_center.pdb`**

Now you will find the dimensions of the primary cell; you will need this information later.

vmd > **`measure minmax $all`**
(You will see a line giving minimum x, y and z and maximum x, y and z coordinates for your system. Write these six numbers down – two decimal points is good or just copy and paste the line into a **vim** session.)

Quit **vmd**.

We will want to have a "`1.00`" in the B-factor column of the protein atoms only (see below). To do this a possible VIM command could be (once you have opened an editing session on the ionized_center.pdb file,

Note that your protein atoms start on line 2 of the ionized_center.pdb file
Search for the last line of the protein (or first line of water which is easier)
`/TIP3`
Move the cursor up one line to be at the last line of the protein
`:2,. s/1.00  0.00/1.00  1.00/`
`:wq`

Now use **sftp** to put the following files on the cluster in your *protsim/* directory there,
ionized.psf
ionized_center.pdb

C. Start the equilibration run.
You will now run a short series of equilibration simulations. First, the system will be energy minimized to bring the PDB structure into most favorable conformation according to the force field, then the water and ions will be relaxed around a fixed protein, then the whole system will be equilibrated with unconstrained MD.

In a terminal window open to your *protsim/* directory on the cluster use **vim** to edit the *min_equil.conf* file. You have to add the cell dimensions that you saved from above. Find the "`Periodic Boundary Conditions`" section and change the question marks to the x, y, z cell dimensions. The cell dimensions have to be calculated from the minmax values above. One or two decimal places is fine.

Now scroll down to the "PME" section and change the `PMEGridSize` for x, y and z to be appropriate for your system. Choose values from the list given: Your values should be just larger than the values you entered above for the `cellBasisVectors`.

Edit the *run.namd* file so that "`yourJHEDID`" is your actual JHED ID. Then submit your job to the queue,

`./run.namd min_equil.conf min_equil.log 6`

Check that your job is running using the **`qstat –u \*`** command. You can also check

your *min_equil.log* file to see how your job is going (after a while).

While your min_equil job is running edit the `PMEGridSize` for `x`, y and z sections in *equil2.conf* and *equil3.conf* to be the same as in the *min_equil.conf* file.

When the min_equil job is finished run the equil2 job,

```
./run.namd equil2.conf equil2.log 6
```

Then when this is finished, run the equil3 job,

```
./run.namd equil3.conf equil3.log 6
```

Check all your log files to make sure that all three equilibration runs ended gracefully. The tail end of each log file should look something like the following,

```
WRITING VELOCITIES TO OUTPUT FILE AT STEP 100000
=================================================

WallClock: 2567.721191  CPUTime: 2169.115479  Memory: 11.059464 MB
```

D. Start the production run.
When the minimization and equilibration run is finished, start your production run. Edit the PMEGridSize x, y and z values in *production.conf* to be the same as those you used in the *min_equil.conf* file. Then, to launch your production run,

```
./run.namd production.conf production.log 6
```

Check that your job is running using the **`qstat –u \*`** command. You can also check your *production.log* file to see how your job is going.

4. How many configurations do you expect to be in the trajectory? (Hint: Use `dcdfreq` and number of steps `run`)

5. Why does 2500000 steps equate to 5 ns?

In the next lab you will analyze the trajectory created by this simulation.

