**BIOS 10603 Lab 6**

**Your Protein Production Run and Further Protein Analysis**

**Name:**

Please insert all answers and figures into this Word document, and then save the completed lab as “Lastname\_Lab6.docx”. When you have completed the lab, submit it as an attachment to the Lab-6 assignment link on Canvas. The report is due by midnight the day after your lab period.

**Objectives:** Previously you minimized your own protein and submitted it for heat-up and equilibration. In this laboratory, you will submit your own protein **for a production simulation. This is the actual simulation that you will analyze to learn more about the dynamical behavior of your protein.** We will also introduce you to a few more scripts to analyze protein simulations.

**Lab 6 Sections:**

1. Your protein heat-up/equilibration simulation analysis
2. Submission of the production run for your protein
3. Preparing and running your second protein system
4. Further ubiquitin analysis and TcL scripting

Be very careful **in analyzing and submitting your protein simulation**, as we will continue these simulations for the next couple of weeks. Any careless errors here will affect all your simulations moving forward. Let Dr. Haddadian or your TA know if you see anything that you don’t expect (**for example unusual bond stretching between the protein atoms**). Errors at this stage will make the rest of your simulation (and final project) unsuccessful! **One common error occurs when your job runs out of time and the simulation does not complete; if this happens, please refer to the procedures you outlined in question 12 on Lab 5 for next steps and if needed, reach out to your TA for assistance.**

**Simulation setup steps**

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | | **DESCRIPTION** | **Lab** |
| **0** | Build PSF | Adding hydrogens to the protein structure | Lab 3 |
| Solvation | Putting the protein in a water box |
| Ionization | Adding salt at a specified concentration |
| Restraining | Adding restraints to the protein backbone |
| **1** | Energy Minimization | Using NAMD to move the protein into an energetically favorable conformation before kinetic energy is introduced to the system. |
| **2** | Heating up | Gradually warm up the system and apply atmospheric pressure. | Lab 3 & 4 |
| Equilibration | Gradually remove the restraints on the protein backbone to allow it to gradually adopts to the pressure and temperature applied without being deformed. Make the system as close as possible to physiological conditions. |
| **3** | Production Runs | Allowing the system to run undisturbed long enough to observe the wanted behavior. Usually broken up into smaller jobs to check for errors. | Lab 6 |
| **4** | Simulation Analysis | Energy, hydrogen bonds, pressure, temperature, volume NAMD Plot | Lab 3 & 5 |
| Density, RMSD, Rg, RMSF, Hydrogen bonds | Lab 4 & 5 |
| Secondary structures | Lab 6 |
| Timeline | Lab 7 |
| Other analyses | Lab 8 |

**Part 1 – Analysis of Your Protein’s heating-up and equilibration run**

Navigate to your protein’s directory on Midway. First, we want to check that the job is actually completed instead of hitting the wall time or crashing because of an error. To do this, run the command tail XXXX-equil.log.

If the output of the message contains “End of program” at the bottom, then your simulation has been completed successfully! Otherwise, you will need to contact your TA about how to restart your equilibration simulation.

Just as you observed in the last labs if your simulation is completed successfully, there will be files ending with .old and .BAK (these are backup files written by the NAMD program). You may remove these files, as they are unnecessary and make take up extraneous disk space in your scratch space. **Carefully** enter the following line (making a mistake could delete all of your work!!!).There are NO SPACES between the \* and . character in the command. The star is a wildcard indicating that you want to remove all files ending with the extension .BAK and all files ending with the extension .old. **Putting the star in the wrong place could delete all of your files, so please double-check before entering this line.**

**rm \*.BAK \*.old**

Download the following files from Midway to your laptop:

1. Your protein’s heating up and equilibrium .dcd (the trajectory).
2. Your protein’s heating up and equilibrium .log.
3. Your protein’s heating up and equilibrium solvate-ionized.psf.

Open VMD and load your PSF file into the program. After you have loaded the .psf , load the .dcd file. If your VMD crashes while attempting to load the .dcd file, try again, but change the **“Stride” setting from 1 to 5, and if it still crashes, try 10 or another higher number.** Play the trajectory a few times to get a visual understanding of what the simulation looks like.

**Q1.** Describe briefly what you see happening in your system (focus mainly on the protein by not showing the water, and **check for unusual bond stretching between the protein atoms or large protein deformations**). Any existence of bond stretching will be extremely evident using the ‘Lines’ or ‘Licorice’ drawing methods, so make sure to use these drawing methods to check for bond stretching.

**Q2.** Using “NAMD Plot,” attach plots and **explain any trends you observe** for the following quantities:

1. The volume of your system - Attach a figure and describe why you see either a constant or non-constant value.
2. The total energy of your system - Attach a figure and describe any patterns that you see. Do the values you observe agree with what you expect?

**Q3.** Find the RMSD of your protein (backbone atoms only) at all time points and plot this using the “RMSD Trajectory Tool” in VMD. **Make sure that the protein is aligned to the first frame of the trajectory (or the last frame of minimization)**. Attach a figure. Describe any patterns that you see and explain why they occur.

**Q4.** Adjust your RMSF script so that it performs the time average fluctuation calculation over **only** the stable part of your protein equilibration trajectory. The stable part of a trajectory is the frame range where the RMSD values are more or less constant (after the initial large jump).

1. Determine the stable part of the trajectory. A rough estimate of the frames that this occurs will suffice
2. **Paste the changes you make to your RMSF code below.**
3. Plot the RMSF of your protein during **the stable part of the trajectory**. **Make sure that the protein is aligned with the first frame of the stable trajectory.** The plot should include the RMSF with labeled axes.
4. **Where of the protein has the peak RMSF values you see in the plot above? Explain why. Hint: view the motion of those segments in VMD.**
5. Why do we only want to measure the RMSF based on the stable part? Why do we align the protein to the first frame of the *stable* trajectory? Why don’t we align it to the first frame of the overall trajectory?

**Part 2 – Submission of the production run for your protein**

**The goal for the production run is to produce data that closely approximates the behavior of the protein in the same conditions in real life.** Biological phenomena happen in a timescale much longer than the one for equilibration and minimization. **For the purposes of this class, we want to obtain 40ns of simulation time for your final project** (if you work with an anti-freeze protein, please talk to Dr. Haddadian).

**Q5.** We will be doing an NPT production run. What does NPT stand for?

In general, it is not recommended to submit a very long continuous MD job. The supercomputer could behave unexpectedly and kill your job, or you may overlook a mistake in your configuration file that does not raise an error at submission time, but that causes undesired behavior of your protein during the simulation. The best method is to submit your production run in several steps and check each of them when they are finished before continuing the simulations. When all these jobs are finished correctly, we will concatenate all of them in a single DCD file.

A good way to obtain 40 ns of the simulation time is to **submit two jobs that have 20 ns of *timesteps* each.** Therefore, before we run the first production simulation, we want to find out how much time the protein equilibration took to complete so we can estimate how long it would take to run a 20 ns job. Run the tail command on your protein equilibration log file. A sample output looks like this:

FINISHED WRITING RESTART VELOCITIES

WRITING EXTENDED SYSTEM TO OUTPUT FILE AT STEP 10000000

CLOSING EXTENDED SYSTEM TRAJECTORY FILE

WRITING VELOCITIES TO OUTPUT FILE AT STEP 10000000

WRITING COORDINATES TO OUTPUT FILE AT STEP 10000000

CLOSING COORDINATE DCD FILE

====================================================

**WallClock: 60731.304688 CPUTime: 60731.304688 Memory: 3179.367188 MB**

**End of program**

**Q5.** We are interested here in the WallClock time, which is reported in seconds. **Show your work in doing the following calculations:**

1. How many hours did your equilibration simulation take to complete? (Remember, WallClock time is reported in seconds)
2. How many *timesteps* were completed in your equilibration simulation?
3. How many *timesteps* per hour did your equilibration simulation achieve?
4. Calculate how many hours it takes to run a 20 ns simulation job, multiply this number by 1.2, and then round down to a nice, even number. This is the time that you need to ask for in your job submit script (remember we specified how many femtoseconds pass with each timestep in the .conf file).

Use the XXXX-mini-heatup-equil-npt1.conf file from Canvas to prepare your own system for the production run. You need to modify the following parts:

* 1. The file name: change it to XXXX-mini-heatup-equil-npt1.conf (Where XXXX is your protein’s PDB code)
  2. Coordinates: this should be the **last frame .pdb** file **after minimization** (same as the equilibration coordinates)
  3. Structure: the **solvated** **and ionized** PSF file (same as the equilibration structure)
  4. Parameter file names (same as minimization/equilibration parameters)
  5. Input name: the “common” file names after the equilibration simulation (output name from equilibration)
  6. CellBasisVector1 (same as your protein’s equilibration values)
  7. CellBasisVector2 (same as your protein’s equilibration values)
  8. CellBasisVector3 (same as your protein’s equilibration values)

**Q6.** Look at the config file and answer the following questions.

1. The constraints are set to off. What is the reason for this?
2. We use the binvelocities parameter instead of temperature, as we used in the heat-up/equilibration config file. What is the reason for this?
3. We used the same periodic conditions as in the heat-up/equilibration config file. Will this cause a problem? Explain.

When you have made all of these edits in your configuration file, edit your job submit script. Ask for **4 nodes** and **the time you calculated**. Change the input *config* and *log* file names and submit the job.

**After your NPT1 is completed, double-check that there were no errors before next week’s lab, and then you can submit NPT2. You will want to visualize the protein behavior in the NPT1 trajectory before submitting NPT2 to ensure the protein behaves as you expect it. In particular, check that the periodic images are as you would expect and that no bond stretching occurs.**

**In -NPT2.conf,**

1. **change the inputname to the output name from NPT1**
2. **change the suffix for outputname and restartname to “-NPT2”.**

**NPT2 will be the second 20 ns of the full simulation. Please make sure that at least one of the two simulations have completed successfully before next week’s lab!**

**Part-3 Preparing and running your second protein system**

**By now all of you should have a clear idea about your second protein system. You need to prepare and run with similar steps as your wild-type (native) system.**

**Q7.**

1. What will you run for your second system? Why did you choose this as the second system?
2. Describe the steps you are taking to prepare and run this system.

**Part 4 – Further Ubiquitin Analysis with TcL scripting**

Measuring the fluctuations in distance between different parts of the protein is a very useful way to gain insight into the protein’s conformational changes. This can be achieved by a simple TcL script as described below:

**Q8.** Instead of modifying a given script as before, we are asking you to write the necessary code with some general instructions:

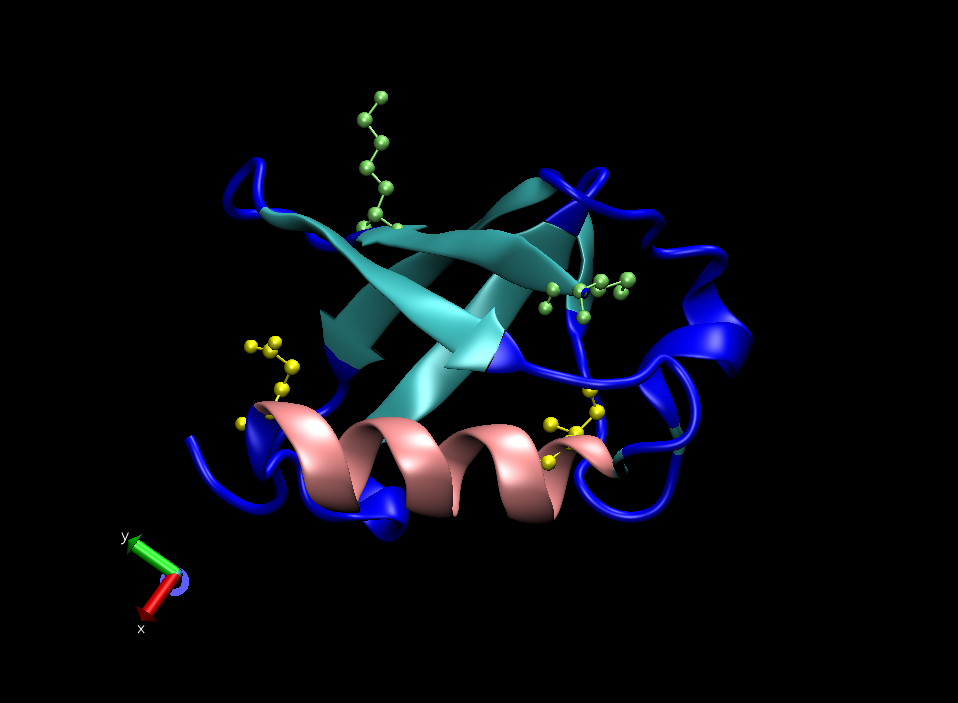
1. Create a new file called **distance.tcl**.
2. Add a new process using the proc command called distance. Distance should take in 3 arguments, seltext1, seltextx2, and outfilename. You can refer to density-script.tcl in Lab 5 for an example of declaring a process.
3. Create two variables, sel1 and sel2. These variables will refer to the variables passed in through seltext1 and seltext2.
4. Open the output file and define another variable containing the number of frames in the trajectory.
5. Using a for-loop, looping through the total number of frames, first update each selection to the current frame and then create two variables and set them to the center of mass of each selection respectively. See below for a hint on measuring the center of mass.
6. Find the distance between these two selections for each frame and write this value to the output file. See below for a hint on measuring distance.
7. Close your output file.

**Paste your completed code below.**

**Here are some hints for setting to the center of mass and measuring distance:**

Measuring the center of mass is done, again, with the “measure” command. You’ve measured centers before. Here, you need to weight each atom by its mass. If you need to look this up, the documentation for the *measure* command is here:

<http://www.ks.uiuc.edu/Research/vmd/vmd-1.9/ug/node135.html>.

As for measuring the distance, remember that you are dealing with vectors (you should choose from vecadd, vecsub, vecdot, and veccross).

**Now, load your ubiquitin protein equilibration trajectory**. First, we will investigate the distances between two pairs of points in ubiquitin. In the figure to the right, we show the sites we will investigate. Residues 1 and 6 are at the ends of a -sheet and are colored green, while residues 23 and 34 are at the ends of the alpha helix and are colored yellow. Run the following commands:

source distance.tcl

distance "protein and resid 1" "protein and resid 6" dist\_beta\_sheet.dat

distance "protein and resid 23" "protein and resid 34" dist\_alpha\_helix.dat

**Q9.** Analysis of secondary structure lengths.

1. Using R, plot the distances between residues 1 and 6 across the frames and describe the trend.
2. Using R, plot the distances between residues 23 and 34 across the frames and describe the trend.
3. Fit a linear regression to the -sheet distance overlaying on your plot. (Hint: Use the lm function. You can type? lm in the R console to see its documentation.) Attach the plot below.
4. Is there a clear trend in the -sheet distance over time? (You do not need to go into any statistical arguments). Explain why this may be the case.
5. Using R, plot the distances between residues 53 and 61 across the frames and describe the trend.
6. Describe the difference between the two -sheets.

VMD has an extensive script library, which can be found here: <http://www.ks.uiuc.edu/Research/vmd/script_library/>.

This library contains many user-submitted scripts to perform various functions in VMD. You may find this useful when performing analysis for your final project.

As a practice, we will use the “trajectory\_path” script from the library. Go to the following link and download the script into your working directory

<http://www.ks.uiuc.edu/Research/vmd/script_library/scripts/trajectory_path/>

**Q10.** Trajectory path analysis. **Make sure that the trajectory has not been aligned to a reference yet.** Following the documentation provided, source the script and draw the path of the center of mass of the protein backbone atoms in blue.

1. Paste the line of command you used to draw the trajectory.
2. Attach a picture below in which the path is clearly visible.
3. Now, align the trajectory to its first frame using the RMSD trajectory tool. Draw in red the trajectory of the center of mass of the protein backbone atoms for the aligned trajectory. Attach a picture that clearly shows the path of the center of mass in both trajectories.
4. How and why does alignment affect the center of mass movement? Briefly explain what you see.

Depending on your protein and the second simulation you choose to run, the trajectory\_path script may be very useful for analysis. You are encouraged to browse through the VMD script library in your spare time and think about possible analyses that you may want to perform on your protein.