**Setting Up, Performing, and Analyzing Dynamics Simulations**

This lab will primarily function as walk through of a dynamics simulation. The objectives for this lab are as follows:

1. Building a system, including building a water box, neutralization, and configuration of the ligand.
2. Heating a system to a desired temperature in preparation for a dynamics simulation.
3. Performing a 10 nanosecond dynamic simulation
4. Analyze the simulations with root mean square deviation

This lab will draw upon what you have learned in previous labs regarding PSF file analysis, energy minimization, atom selections, and constraints.

**Luciferase**

The molecular dynamics simulation you will be performing is an analysis of the effect of an electric field on the protein configuration of the Luciferase enzyme. Luciferase is an enzyme that produces bioluminescence in fireflies through converting luciferin and ATP into oxyluciferin and AMP, releasing a photon of light as oxyluciferin returns to ground state.

The luciferase that we’ll be identifying is found on PDB.org under the identifier 2D1S. Its crystal structure contains a luciferyl-AMP intermediate within its active site. This is the ligand that we’ll be analyzing in our dynamics simulations.

Dr. Brian Mazzeo at BYU conducted experiments that demonstrated that when he applied a 1 megavolt electric field to a solution of luciferase enzymes, he could turn the bioluminescence of the enzyme on and off. This is a surprising result that could have several implications for science. The simulation you will perform will analyze the configuration changes that could explain Brian Mazzeo’s results.

**Build the System**

A crucial step in performing dynamics simulations is to properly set your system. This includes *solvation* (surrounding the protein in generated water molecules), *neutralization* (adding ions to give the system a net charge of 0), and performing energy minimizations. The complexity of this process depends on the complexity of the dynamics simulation you wish to perform.

1. Download the protein of interest in a CHARMM compatible format.
   1. Go to www.charmm-gui.org
   2. On the left sidebar, click “Input Generator”
   3. Then click PDB Reader
   4. In the space “Download PDB File:” type in 2d1s and click “Next Step”
      1. This is the PDB identifier for luciferase.
   5. Select only the SEGID “PROA” and click “Next Step”
   6. Leave all the options unchecked, and click “Next Step”
   7. Download the PSF and CRD files and save them to your lab directory,
   8. Log onto your account in WinSCP, and transfer your downloaded files to the lab5/1\_put\_ligand\_in\_protein folder
2. Analyze the ligand PSF file.
   1. The 1\_put\_ligand\_in\_protein folder contains the ligand psf and crd files. During this first step, we will be placing the ligand into the protein just as it is found in the original PDB.
      1. Downloading and using it directly from PDB.org causes errors, so in this case it is necessary to combine the protein and ligand in CHARMM.
   2. Go to the lab5 home directory, and open the topology and parameter files folder (toppar).
   3. Open the file “lig.str”
      1. This is the topology file generated for the ligand we’ll be using. For each molecule in a topology file, certain criteria must be met. We’re interested in the atom designations found at the start of the file.
   4. To meet the established criteria, these atoms must be separated into “groups”. A group should consist of several atoms that are no more than 12 angstroms apart from each other, and their net charge is equal to zero.
   5. This ligand is comprised of 58 separate atoms which are divided into 6 groups.
   6. The atoms you have are in the correct sequence, but have not been grouped. You can group them by copying the charges into an excel file, and adding up charges until they are equal to 0.
      1. Your final file should be formatted as follows, but with varying numbers of atoms in each group:

GROUP

ATOM C# CG2RC0 0.28 ! 0.000

ATOM N# NG2R50 -0.71 ! 0.000

ATOM C# CG2R53 0.34 ! 0.000

ATOM H# HGR52 0.12 ! 0.000

ATOM N# NG2R51 -0.05 ! 0.000

GROUP

ATOM N# NG2R62 -0.74 ! 0.000

ATOM C# CG2R64 0.50 ! 0.000

ATOM H# HGR62 0.13 ! 0.000

* 1. Save your new RTF file as lig\_new.str

1. Combine the protein and ligand
   1. combine\_protein\_lig.inp is a very simple input script that combines the protein and the ligand in one coordinate file, and performs some minimization calculations on the system.
   2. Open the input script and answer the following question:
      1. ***Describe this script. What are its inputs? What is being manipulated and why? What is the output?***
   3. Edit the script so the input files match the names of your files.
   4. Execute your submission script by typing **sbatch batch.charmmparallelm6** in putty. Make sure the submission script’s input and output files are designated correctly. The script should take under a minute to go to completion.
   5. The output is a CRD and PSF that now contain the protein with the protein and the ligand in the active site.
   6. You can view the output in VMD by uploading your CRD file and selecting “CHARMM Coordinates” from the dropdown menu.
2. Solvation
   1. This is a very simplified solvation script. The 100x100 angstrom water box has already been generated by [www.charmm-gui.org](http://www.charmm-gui.org). You can generate your own water box in charm itself, but using the charmm-gui can speed things up a bit.
   2. Make sure the solvation folder contains the psf and crd you generated previously.
   3. Open solvator.inp and take a look at the script.
   4. You’ll notice a deletion command that is written as follows:
      1. delete atom sort -

select .byres. (resn TIP3 .AND. type oh2 .and. -

((.not. (resn TIP3 .OR. hydrogen)) .around. 2.8)) end

* + 1. ***What is this deletion doing? Why is this important?***
  1. Submit the script by typing ./submit , make sure all your input files are designated correctly.
  2. Open the new system.crd. ***How many water molecules did you generate?***

1. Neutralization
   1. Dr. Mazzeo performed his experiments in a solution with a .2M concentration of ammonium sulfate. For the purpose of our experiment, we’ll be using potassium chloride.
   2. Using the charm-gui, it was determined that 108 KCl molecules will result in a .2M concentration. This means 108 K+ and 108 Cl-.
   3. However, in order for the overall net charge of the system to be neutral, we must check the charge of protein. The ligand is neutral, so we don’t have to worry about its charge.
   4. A simple script can check the charge of the solution. check\_charge.inp contains a simple command:
      1. SCALar CHARge STAT

print ?STOT

* 1. Run the script, and check the charge of the system prior to neutralization. The total charge appears as the value following “total =”.
  2. ***Considering the net charge of the system prior to neutralization, how many additional K+ or Cl- must be added to the 108 in order to reach a neutral net charge for the system?***
  3. Open the script neutralize.inp. Does your answer to the previous question correspond with the number of generated ions in this script?
  4. Run this neutralizae.inp script:
     1. *ABOUT THE SCRIPT*: This script streams 3 additional subscripts: watervars.str, ioncoor.str, and watercolor.str. These three scripts identify 217 water molecules to delete through randomly generated numbers, making room for the ions to be added. The water ions are then deleted, and the ions replace the water ions former positions, with 2 exceptions: 2 chloride ions are added within the protein to take the place of chloride ions that were found in the original pdb script.

***Minimize and Heat your System***

The system must be heated to an appropriate temperature prior to simulation. Since we

are simply looking at conformational changes in the protein due to electric fields, our equilibration will also serve as our simulation.

1. Open the folder 4\_minimize\_heat in WinSCP
2. Open the submit file batch.submit
   1. This file submits your script to the Fulton supercomputer.
   2. The line **#SBATCH -N4 -n16 --mem-per-cpu=2G -t10:00:00 -C 'm7'** designates several important settings for your submission. **–N4** refers to the number of nodes that you’ll run your script on. Generally speaking, the more nodes you request the faster your script will run. **–n16** refers to the number of processors requested per node, with a maximum of 16. **–mem-per-cpu** will usually be held at a constant **2G**. **–t10:00:00** designates the amount of time you anticipate your script to run. **-C 'm7'** designates that we’ll be running our scripts on the m7 supercomputer.
   3. Change the number of requested nodes to 16, and the requested time to 5 hours.
3. Open the input file minimize\_heat.inp.
   1. Scroll down to the ! CRYSTAL portion of the script.
      1. This command generates an infinite crystal of our system. This will allow the system to remain confined to a cubic structure, and the outer edges of the cube will factor in the Van der Waals of the molecules that are within 14 angstroms of the outer edge.
      2. ***What do you think “crystal defi cubi 100.0 100.0 100.0 90.0 90.0 90.0” is designating?***
   2. Now view the ! EWALD portion of the script.
      1. This scripts allows us to more accurately calculate the energy of the system by calculating of the average energy of each molecule across an infinite number of identical crystal images.
   3. View the ! CONSTRAINTS portion of the script.
      1. Three constraints will be applied in this script. We’ll initially apply a harmonic constraint on the protein, to ease the impact of the electric field. PULL EFIELD 1E9 XDIR 1.0 SELE ALL END creates our 1E9-volt electric field constraint in the X-direction, which is applied to the whole system. Finally, we must apply a center-of-mass constraint to the protein, which will prevent the charged luciferase enzyme from migrating through the water box as a result of the electric field.
   4. View the ! MINIMIZATION portion of the script.
      1. ***How many minimization steps and what kind of minimizations are being performed?***
      2. Prior to the second round of minimization, ***what harmonic constraint is being removed?*** ***Why do you think that is?***
   5. View the ! HEAT SYSTEM: HEAT TO 300segment of the script.
      * 1. You can see that two files are opened for writing: 0.rst and 0.dcd.
        2. After these files are opened, our heating script follows:
        3. dynamics cpt leap verl strt nstep 40000 time 0.0015 -

iunrea -1 iunwri 31 iuncrd 32 -

isvfrq 2000 nsavc 200 nsavv 0 -

inbfrq -1 nprint 250 iprfrq 0 ntrfrq 100 -

pcon pgam 25 pmass 500 pref 1.0 surface tension 0.0 -

ihtfrq 2 teminc 0.06 -

imgfrq 50 ixtfrq 1000 cutim 14 -

iasors 0 iasvel 1 iscvel 1 iseed 1 -

firstt 0.0 finalt 300.0

* + - 1. There are several values we need to understand when running this heating script:
         1. **dynamics cpt leap verl strt** – this begins our dynamics simulation as a constant pressure and temperature (cpt) leap verlet simulation.
         2. **nstep 40000** – this designates the number of steps to be performed.
         3. **time 0.0015** – This designates the length in time (in picoseconds) of a single step, or how long a simulation is allowed to move before the velocities of the system are recalculated and adjusted. .0015 picoseconds, or 1.5 femtoseconds. ***How many picoseconds of simulation would be carried out with the current script?***

1 nanosecond = 1,000 picoseconds = 1,000,000 femtoseconds

* + - * 1. **nsavc 200** – this determines the number of steps before the coordinates are saved to the .dcd file

***How many times will the script save to the .dvd file during this run? (This is how many frames that you’ll view in VMD)***

* + - * 1. **iunwri 31** – this will write our RESTART file to the open unit 31. Restart files are essential for restarting our simulations from where our previous simulation finished off.
        2. **iuncrd 32** – this will write our .dcd file to the open unit 32.
        3. **pcon** – this command and its subsequent values designate that the simulation will run under constant pressure.
        4. **firstt 0.0 finalt 300.0 –** this will heat our script from 0.0 K to 300.0 K.

1. Run the script.
   1. Submit your script to the supercomputer in the UNIX console by typing

sbatch batch.submit

* 1. You can view the progress of your script in several ways:
     1. Type “squeue –u (your fsl username)” for a quick update of your script run time

OR

* + 1. Type “watch squeue –u (your fsl username)” for a second-by-second update of your script run time. You can exit out of this mode by typing CTRL-C
    2. Type “watch tail -50 (output file)” to view a second-by-second update of the last 50 lines of your output script.

1. View the output files in VMD
   1. In VMD, load a New Molecule and select the 0.crd output file. Be sure to select “CHARMM Coordinates” from the dropdown menu.
   2. When loaded, right click your molecule in the VMD Main Menu, and select “Load Data into Molecule”
   3. Load in the 0.psf file, and the 0.dcd file.
   4. ***Using your intuition, what should happen to the ions in an electric field? Do you see this occurring in your short simulation?***

***Perform Dynamics Simulations***

Now we are going to perform our dynamics simulation. Open the folder 5\_simulate, and the input file 1E9\_sim. You’ll notice that this file is very similar to the heating script, without the minimizations. The differences are primarily in the dynamics simulation.

1. Analyze the differences in the dynamics settings:
   1. dynamics cpt leap verl restart nstep 400000 time 0.00125 -

iunrea 30 iunwri 31 iuncrd 32 -

isvfrq 2000 nsavc 800 -

inbfrq -1 nprint 2000 iprfrq 2000 ntrfrq 100 -

PCONst pmass 500.0 pgamma 0.0 tbath 300.0 PREFerence 1.0 -

hoov tmass 1000 tbath 300 tcoup 5.0 tref 300 -

imgfrq 50 ixtfrq 1000 cutim 14 -

ichecw 0 iscvel 1 -

finalt 300.0

*What are the differences between this simulation and the heating script? Ask your TA what these differences mean, or look them up on* [*www.charmm.org*](http://www.charmm.org)*.*

1. Prepare the submission script.
   1. You’ll notice that there are various folders within 5\_simulate. This is necessary to organize the output files of our simulation.. To perform a long simulation it is necessary to pause the simulation and restart it from where we left off. This is to decrease the amount of consecutive wall time on the supercomputer, and to protect us from having to restart the entire simulation should an error occur.\
   2. Your submit.batch file contains a variable, rst, that will be included in your script, it’s necessary to increment this variable by one every time you run the script. This will change the variable in your script, and the consecutive .dcd, .rst, .crd, and .xtl files.
2. Run the script once with the current settings.
   1. This will take approximately 3 days, be sure to check back and submit the script the second time.
3. Repeat with an EFIELD of 1E6 and with no EFIELD.
   1. Remember to increase the rst variable within the submission script before submitting the same script with different settings for the EFIELD.

***ANALYSIS***

The analyses that we’re going to perform on our system are the Root Mean Square Deviation (RMSD), and a graph of the distance between two residues over time. These are simple analyses, and we’re going to be using VMD in this case.

1. RMSD Analysis
   1. In WinSCP, transfer the 0.psf, 0.crd, 1.dcd, 2.dcd, and 3.dcd files onto your local computer.
   2. When complete, open VMD on your computer.
   3. Go to File > New Molecule. Load 0.crd, be sure to specify that the file contains CHARMM coordinates in the drop down menu.
   4. Load 0.psf, 1.dcd, 2.dcd, and 3.dcd. If your computer cannot handle 3000 frames, then you can load thes first dcd, perform an RMSD analysis after each dcd file is loaded, delete the frames, and repeat the process for the next dcd, etc.
   5. Once all the frames are loaded, go to Extensions > Analysis > RMSD Visualizer Tool
   6. In the new window, type “all segid PROA” in the atom selection dialogue box.
   7. Next, click “ALIGN” and then “RMSD”.
   8. Select “Plot Result”
   9. ***Describe the data that you see. What causes the changes in RMSD?***
2. Distance between atoms over time.
   1. !! INCLUDE HERE A BRIEF TUTORIAL ON GRAPHING THE DISTANCE BETWEEN ATOMS OVER TIME!!