Tutorial on how to run MD simulation of Lysine in a waterbox using Charmm-Gui

File generation via Charmm-Gui

01. Go to www.charmm-gui.org
02. On the left side of the website, click on the *"Input Generator"*.
03. On the left side of the website, click on the *"Quick MD Simulator"*.
04. Click on *"Browse…"* to select file. Select the file of interest, most likely a PDB extension file. In this case we will use the "Lys.pdb" file. Make sure to check the appropriate *"PDB Format"*, in most cases it is "**PDB**". Click *"Next Step: Select Model/Chain"* on the right side of the window.
05. Make sure that the correct protein segment is selected. In this case we have just one residue for our protein and it should automatically check the box for it. (Note: this is where if you were using a protein that was crystallized with ligands that you could select to keep the ligands or remove them.) Click the *"Next Step: Manipulate PDB"* on the right side of the window.
06. Here we can modify various parts of a protein file. This includes making a mutation in the protein, insuring protonation of a residue, creating disulfide bonds, among other options. For our system, we do not need to do anything special so just click *"Next Step: Generate PDB"*.
07. Now here we can specify our waterbox as well as solvation of ions into the system. First we must pick our waterbox size. The *"Specify Waterbox Size"* allows us to specifically choose the x, y, and z coordinates of our waterbox. The *"Fit Waterbox Size to Protein Size"* allows us to pick a specific waterbox type then chose its length. For our simple system, we will pick the *"Fit Waterbox Size to Protein Size"* with a **waterbox type of rectangular** and an **edge distance of 10.0**.

08. For the ions, we want to include them and have a **150 mM KCl (0.15 M) solution**. (Note: If you change the concentration of the system, you need to make sure you push the *"Calculate number of ions"* button. Also, the *"Add neutralizing ions"* button only generates enough ions to make the system electrically neutral and does not deal with the concentration of the ions in the system.) For the *"Ion Placing Method"* leave it as **Monte-Carlo**. This will insure a good randomization of ion starting points. Click *"Next Step: Solvate Molecule"*.

09. Periodic boundary conditions now must be chosen. These can be explicitly given, or automatically generated. For most systems, you can just chose "*Generate grid information for PME FFT automatically*".

10. We now need to pick our *Force Field*, *Input Generation files*, *Equilibration input*, and *Dynamics input*. For the "*Force Field option*", chose "**CHARMM36**". For the "*Input Generation Option*", this is where you can decide what program you want to run your simulation with. NAMD and AMBER are the two most popular. For our case, select only **NAMD** and deselect other options. For "*Equilibration Input*", keep the **NVT** ensemble. For the Dynamics input, select **NPT** ensemble and have a temperature of **310.15 K (37.15 ℃)**. (Note: for the NPT, NVT, etc. these describe what is held constant for the system. Think of PV=nRT and what you chose is held constant. For example, NPT means constant particle number, constant volume, and constant temperature.) Click *"Next Step: Generate Equilibration and Dynamics Inputs"*.

11. At this point, the needed files should be made for your system. Simply look for the *"download .tgz"* button in the towards the top right to download the need files. A message will pop up about the file. Click *"Save File"* to get the files.

12. To get to the files, look in your downloads. You should see a "<u>charmm-gui.tgz</u>" file. Open it to then see a "<u>charmm-gui</u>" folder. Hit extract and choose where you wish to place the folder to be extracted

to for later use (for instance, your desktop or an appropriately labeled folder).

Running a Simulation

01. First thing to do is to insure your system came out correctly from the file generator. Go into the charmm-gui created directory. Using a visualizing software (such as VMD) on the last generated "<u>.pdb</u>" file (in our case it will be "<u>step3_pbcsetup.pdb</u>"). This is to insure that the system is made how we want it to be (waterbox right size, no missing protein, lipid bilayer correct, etc.).
02. Once you decide the system looks right, we are ready to prepare our simulation. Go to the *"namd"* directory. Inside here, you should see equilibration and production input files ("<u>.inp</u>") as well as other directories that will be used in the simulation that hold force field parameters (restraints, toppar).
03. Now we need to insure our "<u>bash</u>" file is correct. A bash file ends in extension "<u>.bash</u>". It is used to put individual jobs into the queuing system to be run at a later time. This allows for an individual to call for many processors, but let the job wait until that many processors are available. Inside the bash file, there are many small pieces going on (use "*vi holly_slurm_namd2.bash*" to open file). At the top, all the items that start with "<u>SBATCH</u>" are to tell the computer information. This includes how many nodes you wish to use, how many processors per node, maximum amount of time to run the job, and name you wish to give the job to appear in queuing system. Generally the only ones you will change is "ntasks-per-node" and "job-name". For this simulation, set the "<u>ntasks-per-node</u>" to "**1**" and "<u>job-name</u>" to "**Lys-simulation**".
04. Now we need to submit the "<u>.inp</u>" files to actually run the simulation. The bash file is setup to be able to run minimization, equilibration, and production runs. For our system, we need to run

one equilibration run and one production run. While still in the "<u>.bash</u>" file, change the "<u>equil1</u>" to "**1**" and the "<u>prod1</u>" to "**1**".

05. Exit out of the file (":w" to write, ":q" quits file, ":wq" writes and quits file), and put in the command line "sbatch holly_slurm_namd2.bash" to run the job. If the job is executing correctly you should see a line appear saying "*Submitted batch job XXXX*" (the XXXX being the job id).

06. To see if your job is running, type "*squeue*". This gives all people that are in the queue and lists them. You can look for your job by looking at the "*NAME*" column for the name of the job (this case was "**Lys-simulation**") or by looking at the "*USER*" column for your username. To quickly find a job you are running, you can search by username or by jobid. For user name, simply put *"squeue --user=[YOUR_USER_NAME]"*. For example, if someone had the username of *"csrocks447"*, you would simply put *"squeue --user=csrocks447"* and all jobs being run by that user name will be printed in the terminal window. To search for a job by the jobid, simply put *"squeue [YOUR_JOB_ID]"* For example, if your jobid was *"7444"*, you would simply put *"squeue 7444"* and the job with that jobid will be printed to the terminal window.

07. In some cases you realize only after you submitted a job that you submitted it incorrectly. Instead of waiting for the job to make its way through the queue just to crash, you can cancel it before hand. This is done by doing the command *"scancel [YOUR_JOB_ID]"*. Be warned, canceling a job means you forfeit your place in the queue and have to go back to the end of the line.

08. For this simulation, you will get some "<u>.out</u>" files once your job has finished running on the supercomputer. This will be in lines of "<u>step4_equilibration.out</u>" and "<u>step5_production.out</u>". You can insure your job is done by looking for it using the *"squeue"* command explained in step 6. You will also get out various other files that are useful. Most of them will have a "<u>.restart</u>" tag (meant for restarting a simulation should it crash mid way through for some reason), "<u>.old</u>"

(older made file in the simulation meant for a backup), and ".BAK" (also a backup file). The most important files are the ".dcd" files. The ".dcd" files are the frames or "snapshots" of how the protein system looks at any given moment. By playing the frames in order, the dynamics of the system can be observed. The ".dcd" files will be used by loading them into a visualizer software (i.e. VMD).