yFrom PDB to VMD: Setting up, Simulating, and Analyzing a [membrane] Protein in the Dror Lab

Overview

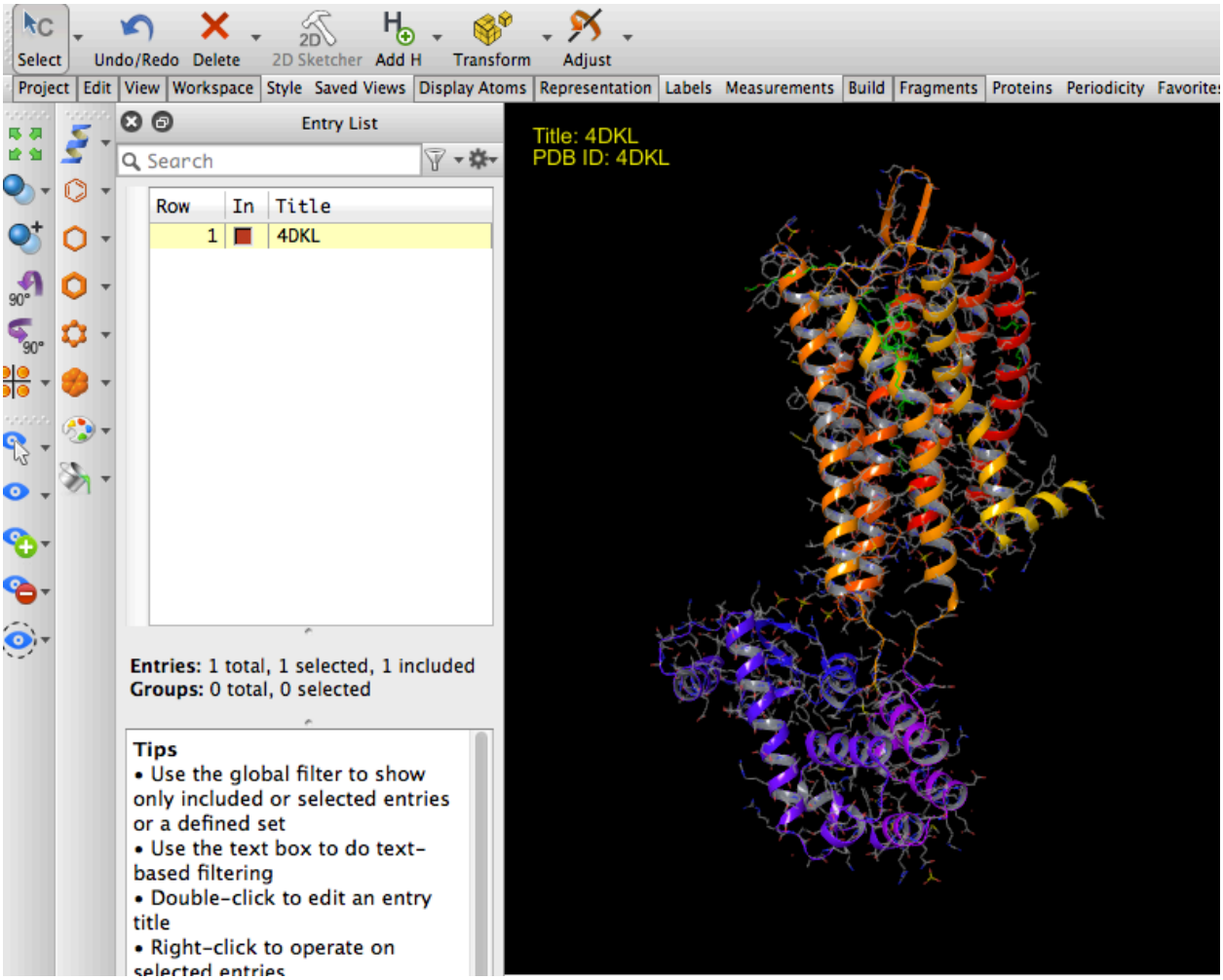
This tutorial will describe how to use the DrorLab infrastructure established mainly by Evan Feinberg (enf@stanford.edu), Robin Betz (rbetz@stanford.edu), AJ Venkatakrishnan (ajvenkat@stanford.edu), among others, to setup, simulate, and analyze membrane proteins. This is a rather involved process that can largely be divided into four distinct, though interrelated, steps:

1. Protein preparation
    a. This includes assigning protonation states; forming disulfide bridges; filling missing side chains; adding capping groups
    b. Outputs: coordinate file (PDB or MAE)
2. System building
    a. This includes embedding the protein into a lipid bilayer; placing the membrane and protein system into a box of solvent with neutralizing ions; setting the bond and parameter information
    b. Outputs: coordinate file (PDB or MAE) and topology file (e.g., PSF)
3. Force field parameter assignment
    a. This involves choosing which biomolecular force field (e.g., CHARMM, AMBER, OPLS) you will use to model your system, and then assigning the parameters of that force field to that system
    b. If you have a non-standard amino acid or a small molecule drug, in this step you will also find or create parameters for the non-standard group, and determine how this will interact with the rest of the system
    c. Outputs: MD-code–specific files
        i. For AMBER: coordinate (.inpcrd) and topology (prmtop)
4. MD Simulation
    a. This involves creating files listing the parameters for the MD simulation as a whole, i.e. temperature, equilibration length, etc. Think of it as the protocol for the MD simulation.
    b. Inputs: for AMBER, coordinate (inpcrd) and topology (prmtop) information and protocol description (.in file)
    c. Outputs: trajectory file for the MD simulation (.nc, .mdcrd, etc)
5. Analysis
    a. This will mostly be conducted in VMD

1. Protein Preparation

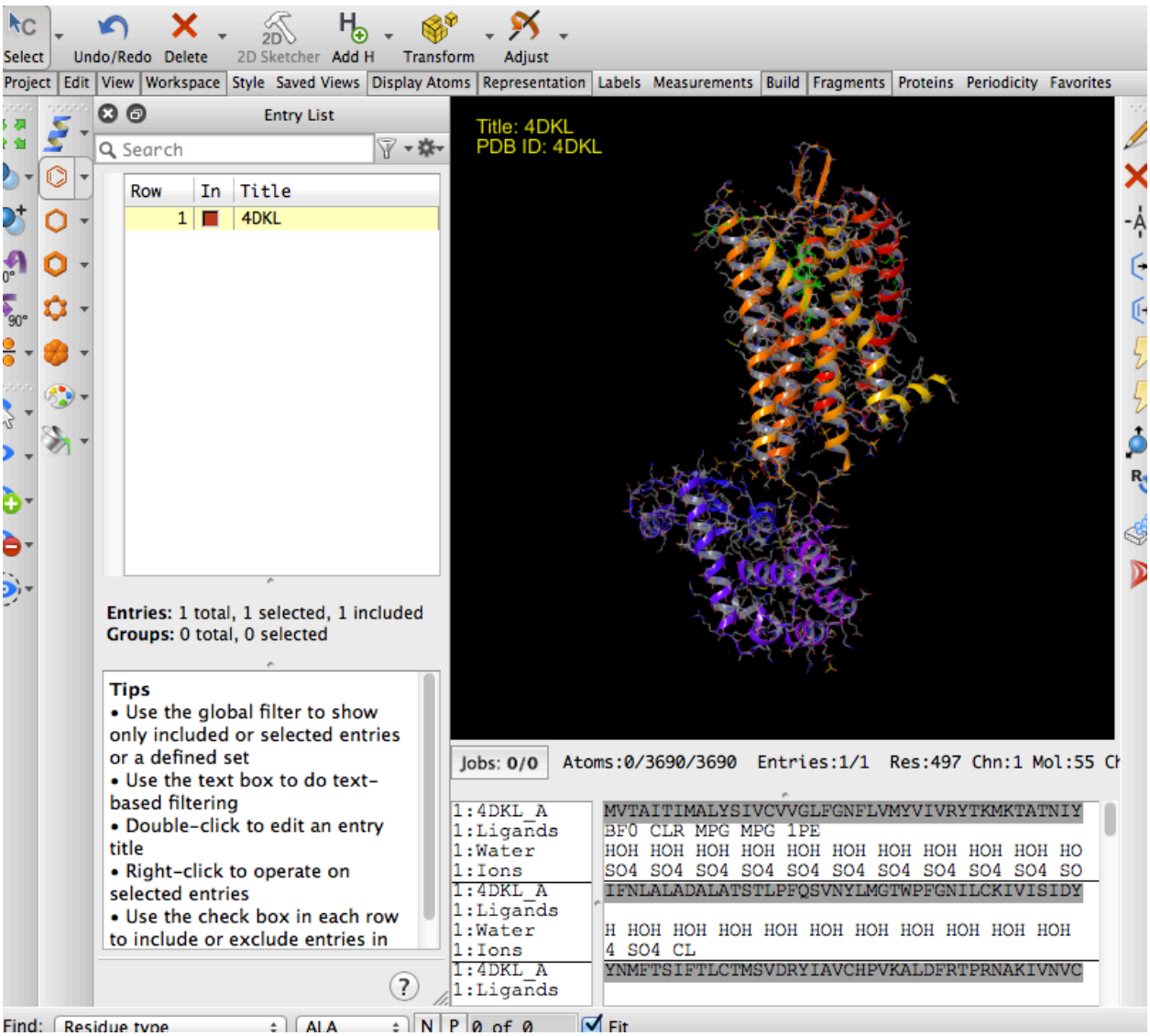
First, download your protein as a PDB file from the PDB's website. Use "Import" to open the file in Maestro. In Maestro, open the toolbars for "Project," "Edit," "View," "Workspace," "Display Atoms," "Representation," "Build," "Fragments." For this example, we will use the inactive conformation of the Mu-Opioid Receptor, PDB code 4DKL, to walk through this process.

Double click the ribbon icon on the far left toolbar to show the ribbon representation of your structure. You will notice that the system can be roughly divided into two components: the receptor itself, and a T4 lysozyme covalently bound to the receptor on its cytoplasmic side. To delete it, change “select” on the top left of the screen to “Chain,” and the icon should become a “C”.



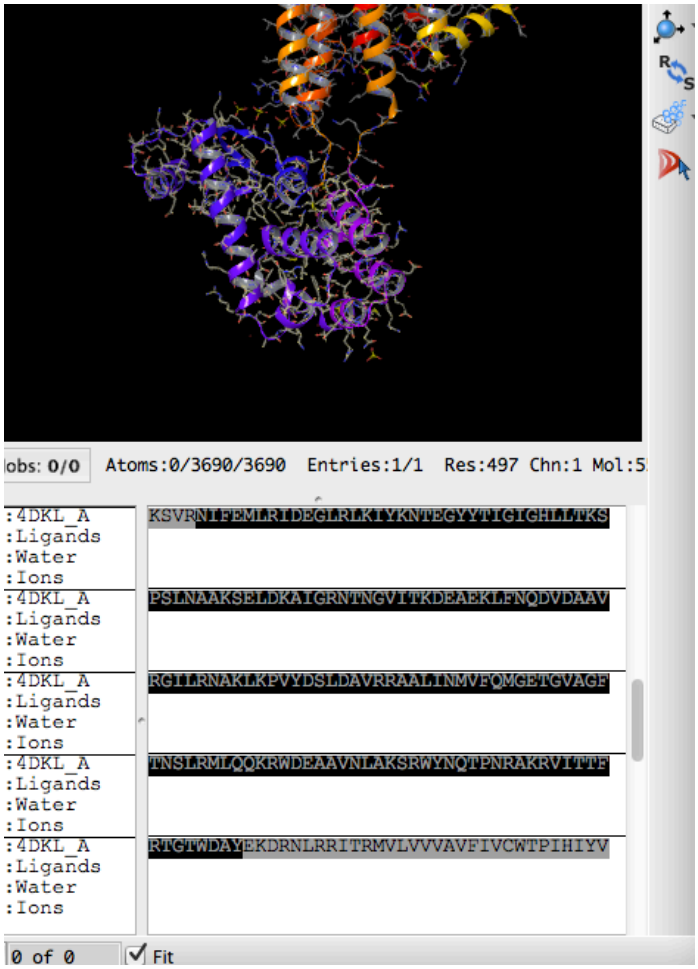
But alas! When you click on the T4 Lysozyme, the whole system is selected. This is because the T4 Lysozyme and M-OR receptor are given the same “Chain” name in the PDB file itself. Now it’s on to Plan B for deleting the T4 Lysozyme: manually selecting its residues.

Under “Window,” click on “Sequence Viewer,” and your screen should look like this:



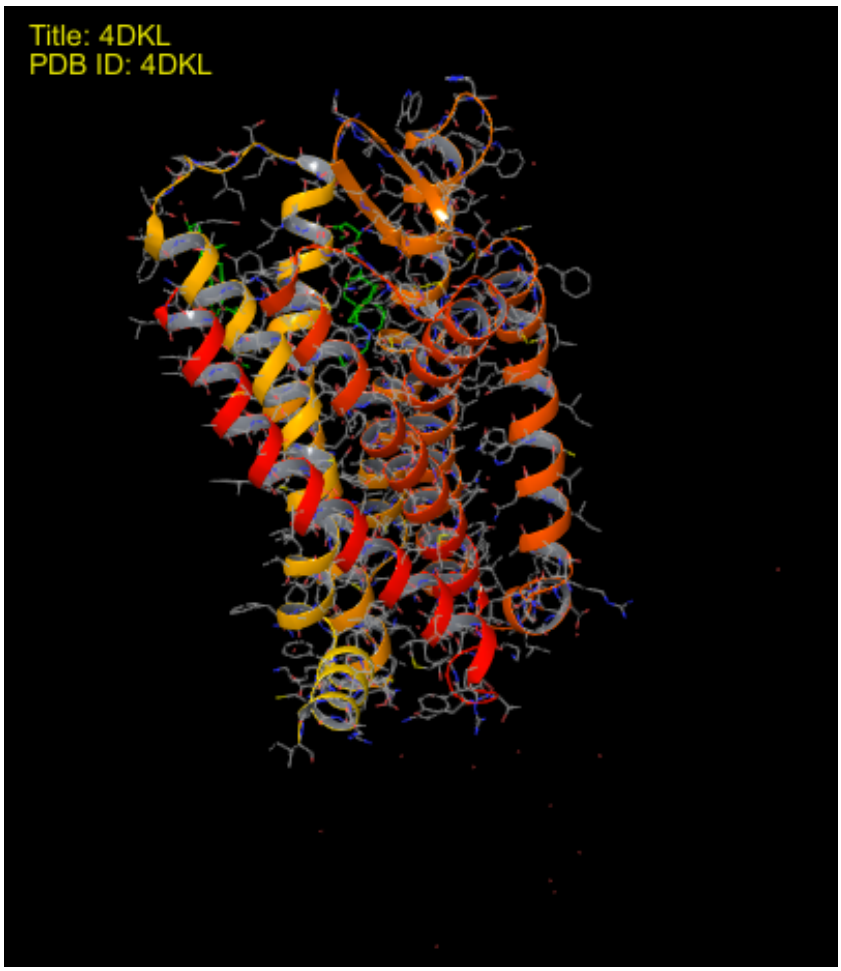
.

Select all non-MOR protein residues. The last protein residue before the T4 lysozyme is 263, and the first protein residue following it is 270. So select all residues in between by clicking and dragging over the Sequence Viewer:



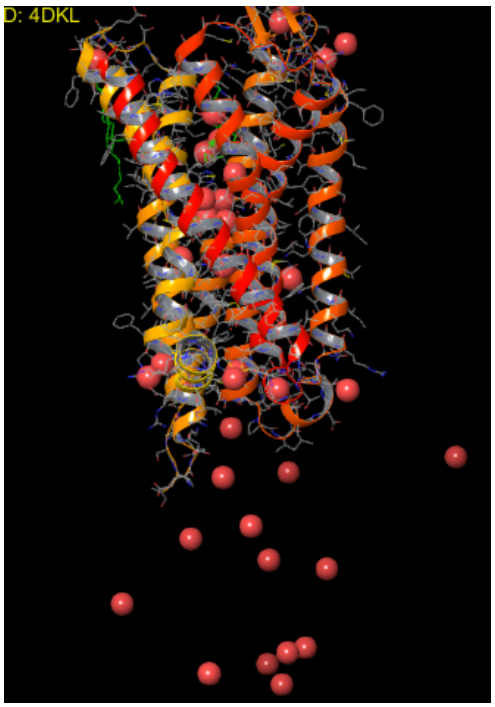
Now right click, and hit “delete.” Now delete all the other molecules you don’t want to simulate. Go to Edit → Select Atoms → Select. Go to the Residue tab, and select Residue Type. Scroll to the bottom to find the non-protein residues. Select all residues except for BF0 (the drug), CLR (cholesterol), and HOH (crystallographic waters). Click “Add”, then OK, then right click in the Workspace and delete those atoms.

Your structure should now look like this:

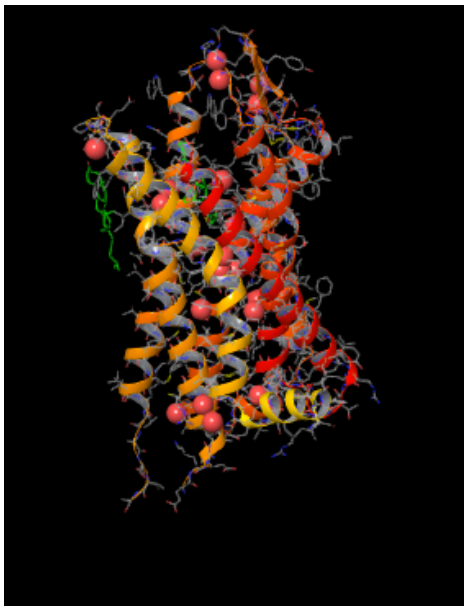


If you will use CHARMM-GUI or another system setup tool for which having too many crystallographic waters will artificially increase the solvent box size, manually delete all waters exterior to the protein. To do this, select all waters in the Sequence Viewer, right click, and choose “CPK” as “Representation.” This should make all waters show up as little red spheres. Now individually select them in the Sequence Viewer one-by-one, and if that water is qualitatively exterior to the protein, delete it. This process will hopefully be replaced by an automated script soon.

So we start with this:

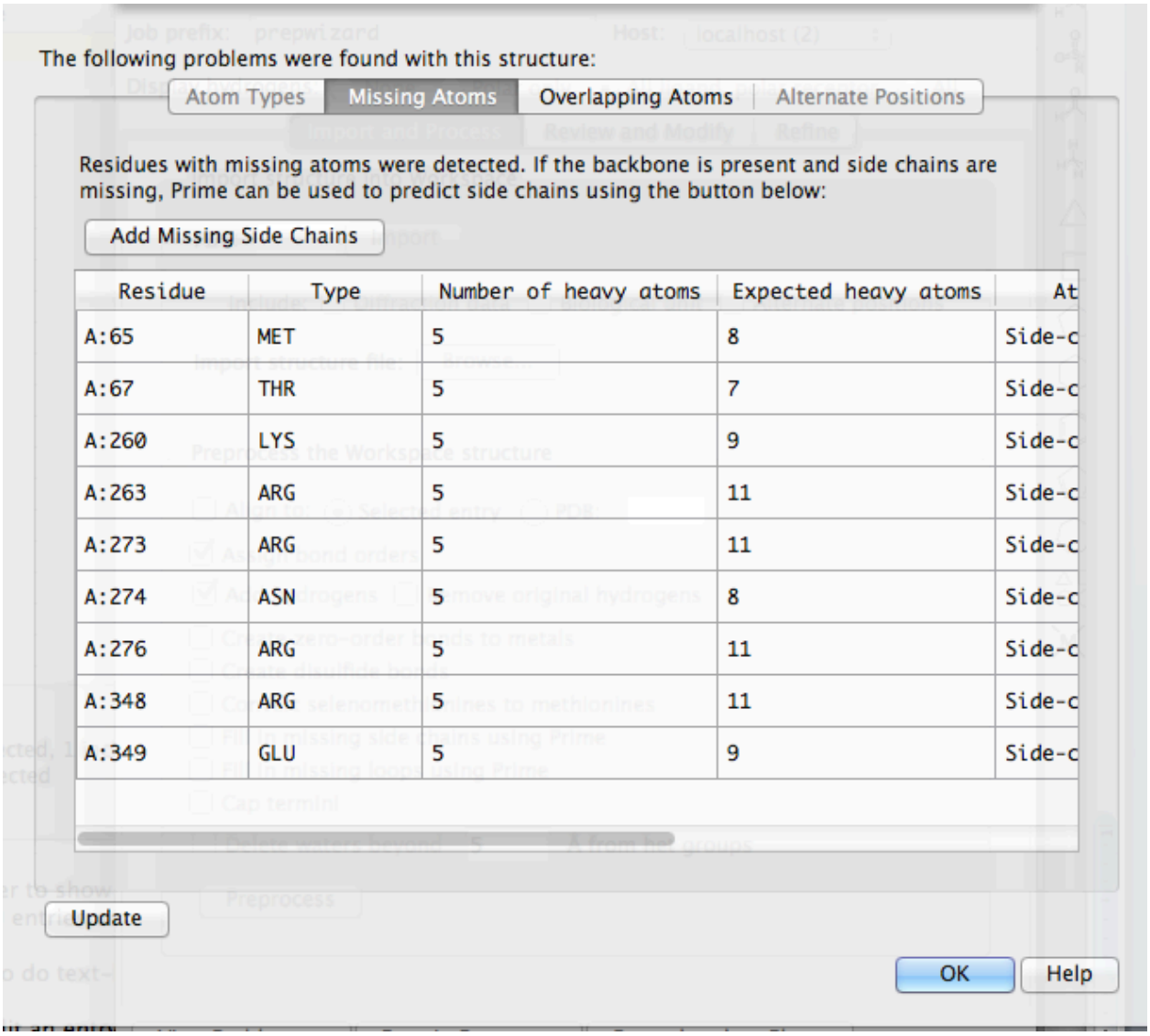


and end up with this:



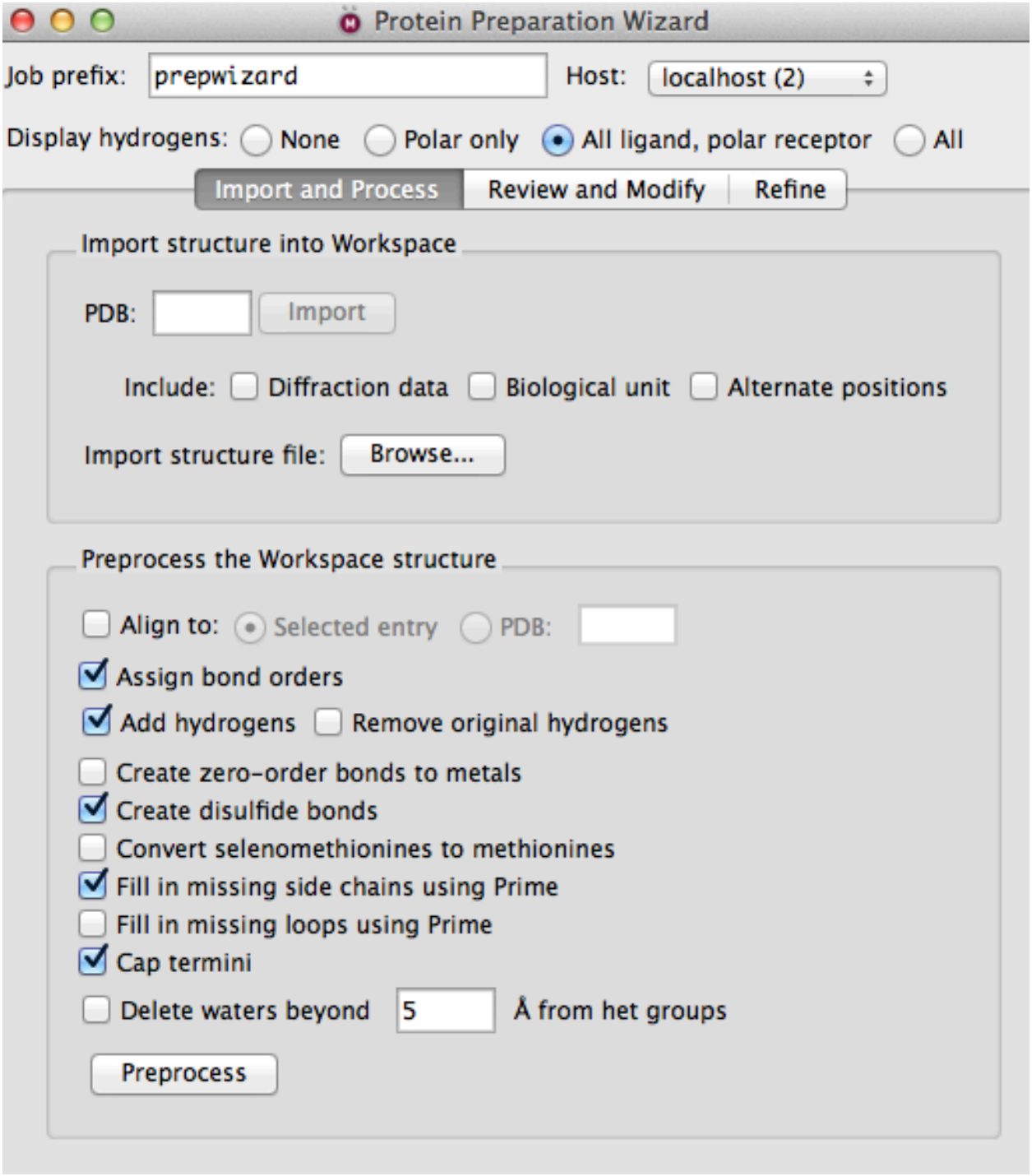
Now that we have all the atoms that we would like to actually simulate, we will open the "Protein Prep Wizard" to facilitate the remaining tasks: filling in any missing side chains, adding hydrogens, assigning bond orders, and assigning protonation states. In the Prep Wiz, first check only "Assign Bond orders" and "Add hydrogens", and de-select all others, in

the Import and Process tab. Now click “Preprocess.” If there are any missing side chains, you will receive a warning to this effect now.



If you do get a warning message, save it for documentation purposes.

Click OK. Now select “Assign Bond Orders,” “Add hydrogens,” “Create disulfide bonds,” “Fill in missing side chains using Prime,” and “Cap Termini.” **Note: if you will be using CHARMM-GUI instead of Dabble, do NOT cap termini.** If you have a FASTA file and want to model missing loops, you can select that as well, but we will not be doing this in this tutorial.



Now hit "Preprocess" for a second time. This process can take a minute or so. This is not a bad time to pour some hot water for tea.

One error you might get is that there are "Overlapping atoms." This *usually* indicates that water hydrogens are too close to each other or to a protein residue, in which case it is easily fixable by changing the orientation of that water molecule. Hit "OK," and go to the "refine" tab. Click "Interactive optimizer." Click "Analyze Network." Find the offending

water, (in the case of 4DKL, HOH 708), and use the right and left arrows to manually flip it to a qualitatively more sensible position. Once you are satisfied with its position, close the interactive optimizer, click "Preprocess" again, and you should get no error messages now. If you do, keep iterating this process.

Now it's time to choose protonation states. Go back to the "refine" tab of the protein prep wizard, click interactive optimizer, click the box for "Label pKas" and click "analyze network." For each titratable residue – every Asp and Glu (Arg and Lys are less likely to need manual refinement but you can look at these as well) – click on its row in the table, and Maestro should zoom you in to that residue with a labeled pKa. If it has a pKa ≥ 7, you should consider protonating that carboxylic acid. The neighborhood of the residue will really factor into this decision. For example, if the titratable acid will be quite close to a POPC group, it is more likely that that residue will be protonated, or neutral, to avoid a situation where a charged group is near a hydrophobic tail of a lipid or near its negatively charged phosphate. Use a combination of your intuition and Maestro's assigned pKa to make an informed decision. If it's unclear, you might want to consider running two simulations with that particular residue in different protonation states.

One last thing: unless you have a more compelling reason for a different titration state, adjust all histidines from the "HIS" state to "HIE," the epsilon-protonated variant.

At this point, your protein should be prepared for the next stage: system setup. Export this both as an MAE file and as a PDB file, making sure that all atoms in workspace are selected (not just displayed ones). Save a project as well for good measure.

2. System Setup

First, you must properly orient your protein. Predictions for the proper placement of many membrane proteins in lipid bilayers can be found in the OPM, the Orientations of Proteins in Membranes database, run by the Mosberg group at the University of Michigan. (rumor has it that Tom Brady himself worked on this project as an undergrad). Go to the OPM, and download the entry for the protein most closely resembling your own. For the M-OR inactive, there is already an entry for 4DKL. For US28, you can use 3ODU as the closest model.

Open both your prepared protein and the OPM version into the same session of PyMol. Now align your prepared protein to its OPM counterpart, with command:

align [prepared_protein], [OPM_version]

Now save your prepared protein again now that it is aligned.

If you will use Dabble instead of CHARMM-GUI, you can skip the alignment step, as Dabble will do it for you as long as you feed it the pertinent OPM entry.

a. CHARMM-GUI

This browser-based program automates the embedding of a protein in a lipid bilayer, building of a solvent box around it, and generation of PDB/PSF outputs compatible with CHARMM, AMBER, and GROMACS MD codes. However, beware, it has several very serious shortcomings.

CHARMM-GUI (known) shortcomings of which you should be aware:

(1) CHARMM-GUI may change side-chain positions without warning. If all of your side-chains are not filled before uploading, it will use its own algorithm to fill them.
(2) It can be quite cumbersome to handle engineered residues or covalently bound ligands, i.e. to draw a bond between CGenFF and CHARMM Biomolecular force fields. We did not do this for any published simulations, instead we used a workaround with ParmEd (a part of AmberTools14). However, at the end of this tutorial, you can find how to deal with covalently bound ligands in CHARMM-GUI.
(3) Almost any modifications made to your protein in another program (e.g. Maestro) will be reverted and destroyed in CHARMM-GUI, and you will have to manually make those modifications in CHARMM-GUI again, which is rather cumbersome. These include, but may not be limited to, protonation states and disulfide bonds.
(4) Naming is important!!! For example, if you want the epsilon-protonated form of histidine, you must name those histidines in the PDB file as "HSE," not "HIS," or else CHARMM-GUI will revert it to the "HSD" or delta-protonated form (bad, CHARMM-GUI, bad!).

If you have changed your histidines to the epsilon-protonated form in Maestro, you have to manually edit the residue names. Do not do this in Maestro, as it will change them to "HETATM" records when it saves the PDB. Instead, you have two options:

1. Save your PDB, and then open it in a text editor, and find/replace "HIS" → "HSE"
2. Open the PDB in PyMol, and enter command:
    a. alter (resn HIS),resn="HSE"
    b. Then save the result

So with all that in mind, let us begin.

1. Upload your PDB file to CHARMM-GUI

References for Membrane Builder:

S. Jo, T. Kim, V.G. Iyer, and W. Im (2008)
CHARMM-GUI: A Web-based Graphical User Interface for CHARMM. *J. Comput. Chem. 29:*

S. Jo, T. Kim, and W. Im (2007)
Automated Builder and Database of Protein/Membrane Complexes for Molecular Dynamics S

S. Jo, J.B. Lim, J.B. Klauda, and W. Im (2009)
CHARMM-GUI Membrane Builder for Mixed Bilayers and Its Application to Yeast Membranes

E.L. Wu, X. Cheng, S. Jo, H. Rui, K.C. Song, E.M. Dávila-Contreras, Y. Qi, J. Lee, V. Monje-C
CHARMM-GUI Membrane Builder Toward Realistic Biological Membrane Simulations. *J. Cor*

◉ **Protein/Membrane System**

Download PDB File: [ ] Download Source: [OPM]

Upload PDB File: [Choose File] mor_inacti...d_unk_2.pdb

PDB Format: ◉ RCSB ○ CHARMM

○ **Membrane Only System**

Hit Next Step.

2. Select all the components of your receptor that you'd like to keep:

Model/Chain Selection Option:

Click on the chains you want to select.

Select Model # [1] ☐ Read all models?

				Residue ID		
Type	SEGID	PDB ID	First	Last	Engineered Residues	
☑ Protein	PROA	A	65	263	UNK	
☑ Protein	PROB	A	270	352	None	
☑ Hetero	HETA	A			CLR	
☑ Water	WATA	A				

3. The PDB Reader is where CHARMM-GUI obtains parameters for any non-standard residues as well as makes its own modifications to the protein.

a. Parameters

If CHARMM-GUI ever asks you for parameters for what should be a standard amino acid, then you have likely made a mistake. For example, let's say it lists "HSE" as an Engineered Residue. This means it is seeing epsilon-protonated histidine as an engineered instead of a standard residue. If this occurs, make sure histidine lines in your PDB are listed as "ATOM" and "HSE", not "HETATM".

If you need to generate your own force field parameters for your ligand, please see tutorial "lilo_spec.docx" in the DrorLab Google Drive folder.

For cholesterol, rename it to "CHL1," as this is the standard CHARMM name for this residue.

For other non-standard groups, like palmitoyl cysteine and both covalently and non-covalently bound drugs, you will need to upload topology (.top/.rtf) and parameter (.prm) files in CHARMM format for those groups.

For non-covalently bound drugs, this is easy: simply upload the CHARMM files for that drug (tutorial on how to generate found in a separate document). If it has a three-membered ring, you must choose this option. You must select "Upload CHARMM top & par for hetero chain" for this to upload your own custom files.

For covalently bound drugs (like betaFNA bound to Lys233 in M-OR inactive or for palmitoyl cysteine), it is more complicated. You must create top and prm files encompassing **both** the amino acid and the drug. An example of one that works for betaFNA bound to Lys233 can be found in the Google Drive folder for this tutorial/documentation. Creating this is *not* trivial, and a tutorial describing the process is forthcoming.

Once uploaded, choose to cap with ACE and CT3 the chains of your choosing, to preserve hydrogen coordinates, to mutate any residues if you must (though we did not use this for any of our current simulations), and to protonate any titratable residues as **CHARMM-GUI will remove any protonation on Asp or Glu etc. that you added in Maestro**, and to add Disulfide Bonds, **as CHARMM-GUI will remove these too.**

Your Step 1 page should now look something like this:

PDB Manipulation Options:

☐ Renaming Engineered Residues:

Rename **UNK** to [ ] Leave blank to remove

☑ Has three membered ring — Lowers the bomlev before reading the hetero chain in case three membered ring is present

◉ Upload CHARMM top & par for engineered residue:

Topology: Choose File LYB_2.top

Parameter: Choose File LYB.prm

☑ Reading Hetero Chain Residues:

CLR

◉ Rename to CHL1

○ Use CHARMM General Force Field to generate CHARMM top & par files (using ParamChem service)

○ Use Antechamber to generate CHARMM top & par files

○ Upload CHARMM top & par for hetero chain

☐ Has three membered ring

☑ Terminal group patching:

	First	Last	
PROA	ACE	CT3	☐ Cyclic peptide?
PROB	ACE	CT3	☐ Cyclic peptide?

☑ Preserve hydrogen coordinates:

☐ Mutation:

☑ Protonation:

	Residue	Residue ID		
PROA	ASP	114	-	Add Protonation

☑ Disulfide bonds:

Pair 1	Residue ID	Pair 2	Residue ID		
PROA	140	PROA	217	-	Add Bonds

☐ Phosphorylation:

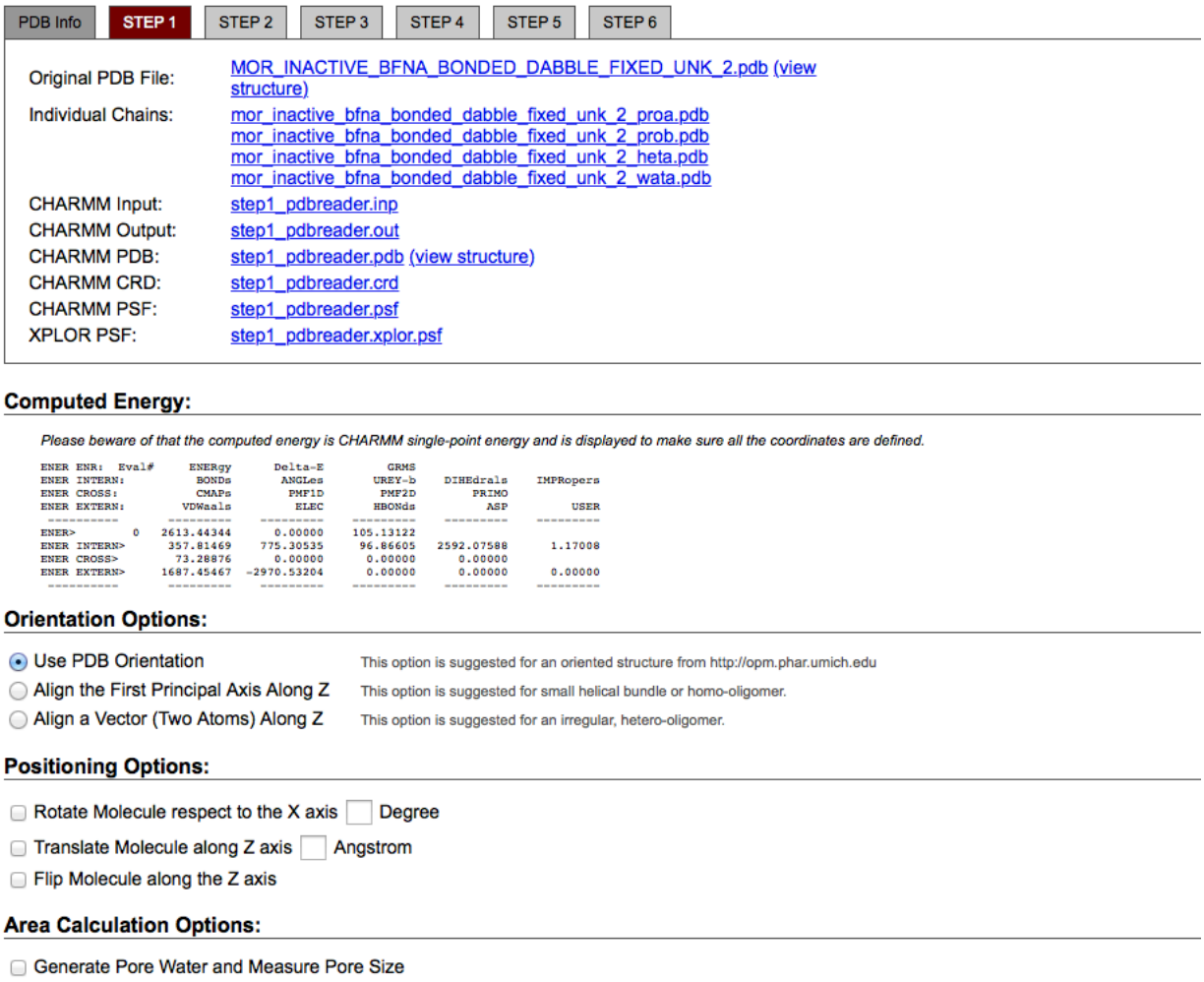
☐ Add MTS reagents: nitroxide spin labels

☐ Add MTS reagents: chemical modifier

☐ Unnatural amino acid substitution:

Symmetry Operation Options:

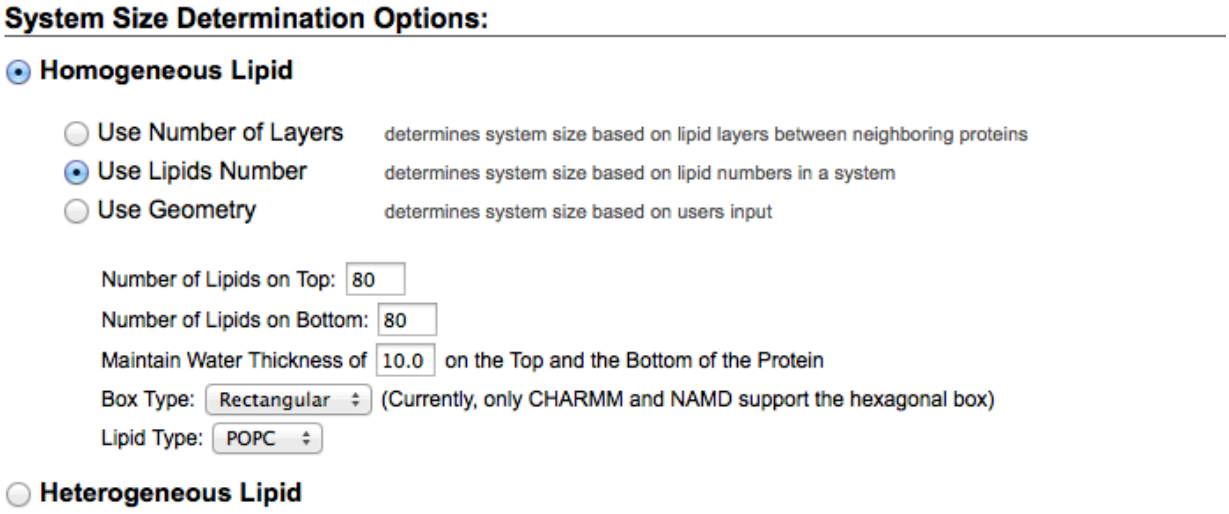
If you (and CHARMM-GUI) did everything correctly, you should get a page like this:



In almost all cases, you should choose “Use PDB Orientation,” as this will preserve the OPM’s prediction to which you aligned earlier.

Now, next step…

For previous simulations, we used a homogeneous lipid bilayer, with 80 lipids in each leaflet, 8.0 A of water above and below the protein, rectangular box type (may be possible to do hexagonal with GROMACS, we are still investigating this), and POPC lipids. However, CHARMM-GUI now makes 10.0 A a lower limit.



Next step…

We usually chose ion concentration of 0.15 M and NaCl as our ion of choice, with "Distance" as our ion placing method, but you should choose what is best for your particular simulation.

Next step…

You should get no protein surface penetration or no lipid ring penetration. If not, keep refreshing in your browser until you get no penetration. If it's being resistant, this is likely related to a cholesterol issue, so make sure you renamed it "CHL1" in the Step 1 PDBReader stage.

Next assemble the components, and click through to Step 5.

Record your box parameters! These are essential for the next step. Write them down IMMEDIATELY since it is rather cumbersome (though possible) to recover them from the files you will download. Download a .tgz of everything. Move it to a new directory on Sherlock with a folder describing the system and simulation you aim to run. Extract it, name the resulting .tgz and extracted folder something descriptive about your system, and move on to the next stage…

[2b. All the preceding information in Step 2 can be done equivalently with Dabble, a superior piece of software written by our very own Robin Betz. Her documentation will be filled into this section]

3. Assigning Force Field Parameters and Writing Input Files for your MD Code

Currently, your output is in a format suitable for CHARMM, an MD package developed in part by Martin Karplus and his group. It was previously named "HARMM," which Karplus thought was appropriate for the amount of harm it inflicted on graduate students, but they settled on a more charming name instead. However, at least for the types of hardware we will be using, packages like AMBER, GROMACS, and Desmond all achieve better performance and have more features than CHARMM at this point. So this section will go over how to generate input files for one of these options: AMBER. AMBER has several advantages, one of which is an excellent cost-to-performance ratio with GPUs

AMBER has a program called "CHAMBER" (a portmanteau of CHARMM and AMBER) that takes coordinate (PDB), topology (PSF), and force field (RTF/PRM/STR) files as input, and outputs coordinate (inpcrd) and topology (prmtop) files ready for MD simulation.

The CHAMBER implementation you will use is incorporated as a command in ParmEd, a software tool that comes with AmberTools14 and that Jason Swails wrote to edit AMBER parameter files. It is an extremely useful utility that you may find helpful throughout your MD activities.

To run ParmEd, first go to the directory containing your CHARMM-GUI output files, and run

/share/PI/rondror/software/parmed/bin/parmed.py

A sample command to run inside ParmEd would be:

```
chamber -top toppar/top_all36_prot.rtf -top toppar/top_all36_lipid.rtf -top toppar/unk.rtf -top
toppar/top_all36_cgenff.rtf -param toppar/par_all36_prot.prm -param toppar/unk.prm -
param toppar/par_all36_lipid.prm -param toppar/par_all36_cgenff.prm  -str
toppar/toppar_water_ions.str -psf step5_assembly.psf -crd step5_assembly.pdb -box
77.6112,77.6112,71.138
```

Change the "box" parameters to be the x,y,z, dimensions that you wrote down in Step 5 of CHARMM-GUI (or output of Dabble). Change "unk" to whatever your engineered residue's name is. You can add as many –top, -prm, and –str lines as needed to make sure ParmEd has all the FF parameter information it needs to build your AMBER-ready system.

ParmEd's CHAMBER will yell at you about some section being wrong, but this is warning not an error, and Jason Swails informs us that it is nothing to be alarmed about.

To output your inpcrd/prmtop files, use command:

outparm system.prmtop system.inpcrd

If you'd like to make manual edits to your system, e.g. forming a bond between a noncovalently bound ligand and your protein, you can now do this, thanks to a lot of sweat and tears and exchanges with Jason Swails. A sample of such parameterization commands are:

```
strip '(:169 & @HZ3) | @HAW1'
change charge :169&@NZ -0.401
change charge @CAW 0.296
setBond :169&@NZ @CAW 200.00 1.4900
setAngle :169&@NZ @CAW @CAU 43.7 110.0
setAngle :169&@NZ @CAW @CAX 40.0 110.0
setAngle :169&@CE :169&@NZ @CAW 45.0 115.0
addDihedral @NAT @CAU @CAW :169&@NZ 0.6 1 0.0 1.0 1.0
addDihedral @OAV  @CAU @CAW :169&@NZ 0.0000 1 0.00 1.0 1.0
addDihedral :169&@NZ @CAW @CAX :BF0&@CAY 0.2 3 0.0 1.0 1.0
addDihedral @CAU @CAW :169&@NZ :169&@CE 1.7 1 180.0 1.0 1.0
addDihedral :169&@CD :169&@CE :169&@NZ @CAW 0.1 3 0.0 1.0 1.0
```

To parse this, you need to understand AMBER atom selection language. For more information, see the section on "amber masks" in the AMBER14 manual.

Brief description:

Atom names begin with an "@". To select all c-alpha carbons, use mask "@CA". residue names begin with a ":". To select all residues numbered 233, use mask ":233". To select all protein and ligand residues, use command: '!(:POPC | :TIP3 | :SOD | :CLA)', which describes all residues that are NOT named POPC (lipids), TIP3 (waters), SOD (sodium), or CLA (chloride).

Strip = remove atoms matching those mask
Change charge = changes charges on a certain atom
setBond = forms a bond between specified atoms
setAngle = sets angle parameters about specified atoms
addDihedral = sets dihedral parameter about specified atoms

You should view this functionality as a last resort only if truly needed. However, the AmberMask and other functionality is universally applicable to all things AMBER and it is likely worth getting familiar with it.

4. <u>MD Simulation</u>

1. Now that you have topology (system.prmtop) and coordinate (system.inpcrd) files in AMBER format, you are now ready to start minimizing, heating, equilibrating, and running production MD simulation on your system. This process is now nearly fully automated. Copy all files from $PI_HOME/MD_simulations/amber/input_files_2_3_2015/* into your current directory:
    a. cp $PI_HOME/MD_simulations/amber/input_files_2_3_2015/* ./

b. now edit the file “new.sbatch” that you just copied:
    i. change the line `#SBATCH --job-name=sim` to a job name of your choosing, call it `#SBATCH --job-name=TUTORIAL`
    ii. You can choose on which Sherlock GPU node to run this job on by editing these lines:

```
##SBATCH -n 2 -p rondror --gres=gpu:2 --qos rondror
#SBATCH -n 2 -p gpu --qos=gpu --gres=gpu:2 --constraint="titanblack"
```

    iii. Run on the --qos rondror nodes by commenting out line (2) with two hashtags and removing one of the hashtags on the --rondror line. To use only one GPU, change “-n 2” to “-n 1” and -gres=gpu:2 to -gres=gpu:1. Here are the new lines:

```
#SBATCH -n 1 -p rondror --gres=gpu:1 --qos rondror
##SBATCH -n 2 -p gpu --qos=gpu --gres=gpu:2 --constraint="titanblack"
```

    v. Change the line `#SBATCH --mail-user=enf@stanford.edu` and put your email address. [Or else I will get spammed ;-)]
        1. Save the new version of new.sbatch

6. You’re done! To initiate the whole process, just run: “sbatch new.sbatch”
. To check to see it is properly queued and running, type: “squeue | grep ‘enf\|JOBID’ ”, where you need to change “enf” to your SUNet ID.

a. What does “new.sbatch” do exactly? It automatically conducts a minimization, heating, equilibration, and production MD simulation with your system. It currently performs minimization in 3 stages, each involving up to 1000 steps of minimization (500 with gradient descent, 500 with conjugate gradients), and with steadily decreasing harmonic position restraints on all protein, ligand, and lipid atoms. Heating occurs first in the NVT ensemble to 100 K, then in the NPT ensemble to 310 K. Equilibration occurs for 12 ns in six 2 ns increments where the first stage has harmonic restraints of 5.0 kcal/A^2 on all protein [and ligand] atoms and the last stage has harmonic restraints of 0 kcal/A^2, where this is steadily decreased by 1 kcal/A^2 in each equilibration step. You can edit the .in files for each step (in your own directory, not in the master input_files_2_3_2015 directory) to suit the specific needs of your simulation.

b. This job will last for 48 hours. To queue more jobs, run “sbatch new.sbatch” in the directory of the simulation as many times as you wish. Each time you run that command will add 48 more hours to the job. On –qos rondror nodes, you can run for up to two weeks.
    1. This time length should be written in minutes
    2. Alternatively, in “Prod.in”, you can change the length of a simulation by changing the “nstlim” variable to the number of steps you would like to use in your simulation. The number of picoseconds that your simulation will run will be nstlim * dt. If you want to know how long this will take in real time, you will need to know the performance of that system in the given GPU configuration. For example, a 58,000 atom system will run at ~90 ns/day on 2 Titan Black GPUs. So with dt = 0.0025 ps, nstlim=72000000 will approximately correspond to 48 hours in real time.

c. To kill a job, type “scancel 1234”, where “1234” is your job id. Alternatively, you can do “scancel --name=[job_name]”, which will kill all jobs with that name (so use carefully)
d. To combine the resulting trajectory files when you are ready to analyze the simulation, use the command “sh auto_reimage.in” (described further below)

Further Documentation of Scripts

new.sbatch

This script, executable from the command line with “sbatch new.sbatch” automatically minimizes, heats, equilibrates, and conducts production MD simulations on your system. To proceed, you must have “system.inpcrd” and “system.prmtop” files in your directory, as well as all necessary .in files (contained in input_files_2_3_2015 directory).

This script will check if Min_3.rst already exists. If so, it will assume that minimization is done, and will move on to heating/equilibration/production.

This script will also check if Prod_1.rst already exists. If so, it will assume that all equilibration and preceding steps are done, and will move on to more production runs, incrementing the integer in the file name by one, i.e. if Prod_3.rst already exists the next production window will have name Prod_4.rst (checkpoint file) and Prod_4.nc (trajectory file).

To conduct a longer simulation than Sherlock’s 48 hour time limit, you can submit multiple “new.sbatch” jobs in a given directory. I.e., enter “sbatch new.sbatch” into the command line multiple times consecutively. This will make each new job dependent on the last job completing, so they will all have a dependent queue. You can submit as many as you like (I convinced Kilian to raise our lab’s limit to a couple hundred I believe).

auto_reimage.in

This script, executable from the command line with “sh auto_reimage.in”, concatenates and centers/reimages all heating, equilibrating, and production MD files in a given directory, using the following convention:

One crystal structure, named system.inpcrd
Two heating steps, named Heat_i.nc ($1 \leq i \leq 2$)
Two equilibration steps, named Eq_i.nc ($1 \leq i \leq 6$)
Production trajectories named Prod_i.nc ($1 \leq i \leq \$LAST$)

And outputs traj_script.in (which it then executes), a combined trajectory called 1_thru_$LAST_skip_$SKIP.nc, and another combined and reimaged trajectory called 1_thru_$LAST_skip_$SKIP_reimaged.nc, where $LAST denotes the total number of production MD trajectory windows and $SKIP denotes the number of times the trajectory has been downsampled (i.e. $SKIP=100 means it skipped every 100 frames).

This script works by writing a separate cpptraj-compatible input file called **traj_script.in**, which is in turn executed by cpptraj. This is because cpptraj doesn't access the normal bash commands (for loops, etc.). auto_reimage.in circumvents this issue by automatically generating a cpptraj-compatible script and then executing it (called traj_script.in).

By default, it skips every 100 frames, it skips any heating or equilibration frames, and it centers/reimages all atoms.

To skip less or more frames, change line SKIP=100 to a value of your liking.

To include crystal structure, heating, and equilibration, change EQ="NO" to EQ="YES"

strip_reimage.in

This essentially serves the same function as auto_reimage.in but also strips out any non-protein and non-ligand atoms. This is useful if you have a massive trajectory file that you want to reduce to a more tractable size. It outputs a "_stripped.pdb" file as well as a combined trajectory .nc file that can be used for visualization and analysis in VMD.

Analysis with do_calc.in and calculation.in

If you would like to also run an analysis or do a calculation with this trajectory, you have two provided options.

The recommended option is to use the script "**do_calc.in**", located in input_files_2_3_2015 directory, which you can execute from the command line with "sh do_calc.in". It works by creating and executing another script called "**calc_script.in**" which is readable by cpptraj. It reads in all specified trajectory (.nc) files in a given directory. You can choose to include equilibration (default) by setting the line EQ="YES" or EQ="NO" in the script. For this to work, you must supply your own "calculation.in" file containing a calculation you would like to do in cpptraj. A sample calculation is found in input_files_2_3_2015 folder in the file calculation.in. This file computes and outputs the dihedral angle of four atoms in two residues, and also calculates the interplanar angle between those two residues. There are many more types of calculations you can do with cpptraj. To learn more, I recommend opening cpptraj, entering "help", and exploring the plethora of functions it provides. You can also consult the Amber14 manual, which is quite helpful in the topic. You can also email enf@stanford.edu to get ideas.

In addition, you can use *auto_reimage.in*, which will allow you to do a calculation simultaneously with its main function if you un-comment the line "`## echo "readinput calculation.in" >> traj_script.in`" by removing the hasthags and providing your own "calculation.in" file in the same directory.

Further Analyze and Plot Analyses

Suppose you conduct several such calculations, analyses, or measurements, on many trajectories. This can constitute a vast quantity of data about your trajectories. Each one will be in a separate text file, where each row represents a single frame of simulation and each entry represents one measurements (e.g., a dihedral angle, or an interplanar angle, etc.). I've written a template script that you can modify for your own purposes for analyzing this data in R, a statistical programming environment. This is in the file **analysis.R**.

analysis.R walks through the working directory and reads all data exported by cpptraj in the previous step into data vectors, then combines those vectors into a single data frame. The do_analysis() function then subsamples and filters each column in the data frame and makes a plot of the data, and subsequently saves it to a PDF file.

The smoothing function, defined in **smooth-4.py**, uses a moving average triangular window approach to smoothing your data timeseries. As a corner case, near t=0, the smoothing window will be equal to double the size of t, such that, at t=0, the smoothing window is of size 1.

5. Analyze the resulting trajectories in VMD

[These instructions were written by AJ Venkatakrishnan and edited by Evan Feinberg]

Starting VMD:

1. Open VMD by double-clicking the application icon. This will open three windows: 'VMD Main', 'VMD Display', and 'Startup Command'.

Loading protein into VMD:

2. In the 'VMD Main' window, click on File > New Molecule. This will open the 'Molecule File Browser' window.
3. In the 'Molecule File Browser' window, under the 'Load files for' menu, please select 'New Molecule'. Then click on the 'Browse' button, locate the protein structure file ('.psf') on your computer, and load it. In the 'VMD Display' window, there will be no visible changes yet at this stage.
4. Using the same 'Browse' button used in step 3, locate the corresponding trajectory ('.nc') file on your computer, and load it. In the 'VMD Display' window, a dynamic display of the protein, lipids, and waters should appear now.
5. The simulation can be played, rewound, paused using the buttons at the bottom of the 'VMD Main' window.
6. It is recommended that you also start a "New Molecule" and load a .PDB file containing the crystal structure of your system, or the input system for MD, to use as a reference.

Customizing the Display:

Display only the protein, turn off waters and lipids

7. In order to show only the protein, in the 'VMD Main' window, please click on Graphics > Representations. This will open the 'Graphical Representations' window. In order to show only the protein, under the 'Selected Atoms' field, please erase 'all', type 'protein', and click on the 'Apply' button at the bottom of the window.

Change the representation

8. By default, all the molecules will appear in the 'Lines' representation. In order to change the representation, in the 'Graphical Representations' window, click on the 'Drawing Method' menu and choose the representation of your choice (e.g. 'New Ribbons').
9. In order to display amino acid residues of interest, in the 'Graphical Representations' menu, click on the 'Create Rep' button. This will create a replicate of your current display, which can be modified to show the residues of interest using a new representation of your choice. For *e.g.*, in order to show Histidine 297 as sticks, under the 'Selected Atoms' field, type 'protein and resid 297' and hit 'Enter' key. For showing multiple residues, for *e.g.* Histidine 297 and Tryptophan 293, under the 'Selected Atoms' field, type 'protein and resid 297 293'. For displaying residues by name, for e.g. all Tryptophan residues, under the 'Selected Atoms' field, type 'protein and resname TRP'.

Align your trajectory

To center your molecule in the visualization window as well as to align it to a reference structure, go to Extensions → Analysis → RMSD Trajectory Tool. Type in the residues on which you would like to align (see Google Doc Simulation Analysis Tools for some examples). Make sure to change "Top" to "Selected" so that alignment will take place on the selected molecule and not merely on the top frame of your selection. Now select align.

Change the color

10. For changing the color of the displayed molecule(s), in the 'Graphical Representations' window, click on 'Coloring Method' menu and choose 'Color ID'. A dropdown menu with numbers will appear adjacent to the 'Coloring Method' menu, which can be used to change the color.

<u>6. Congratulations, you have traced the arc all the way from an unprepared protein from the PDB to a visualized all-atom MD simulation.</u>

Happy simulating ☺ from Evan Feinberg et al.