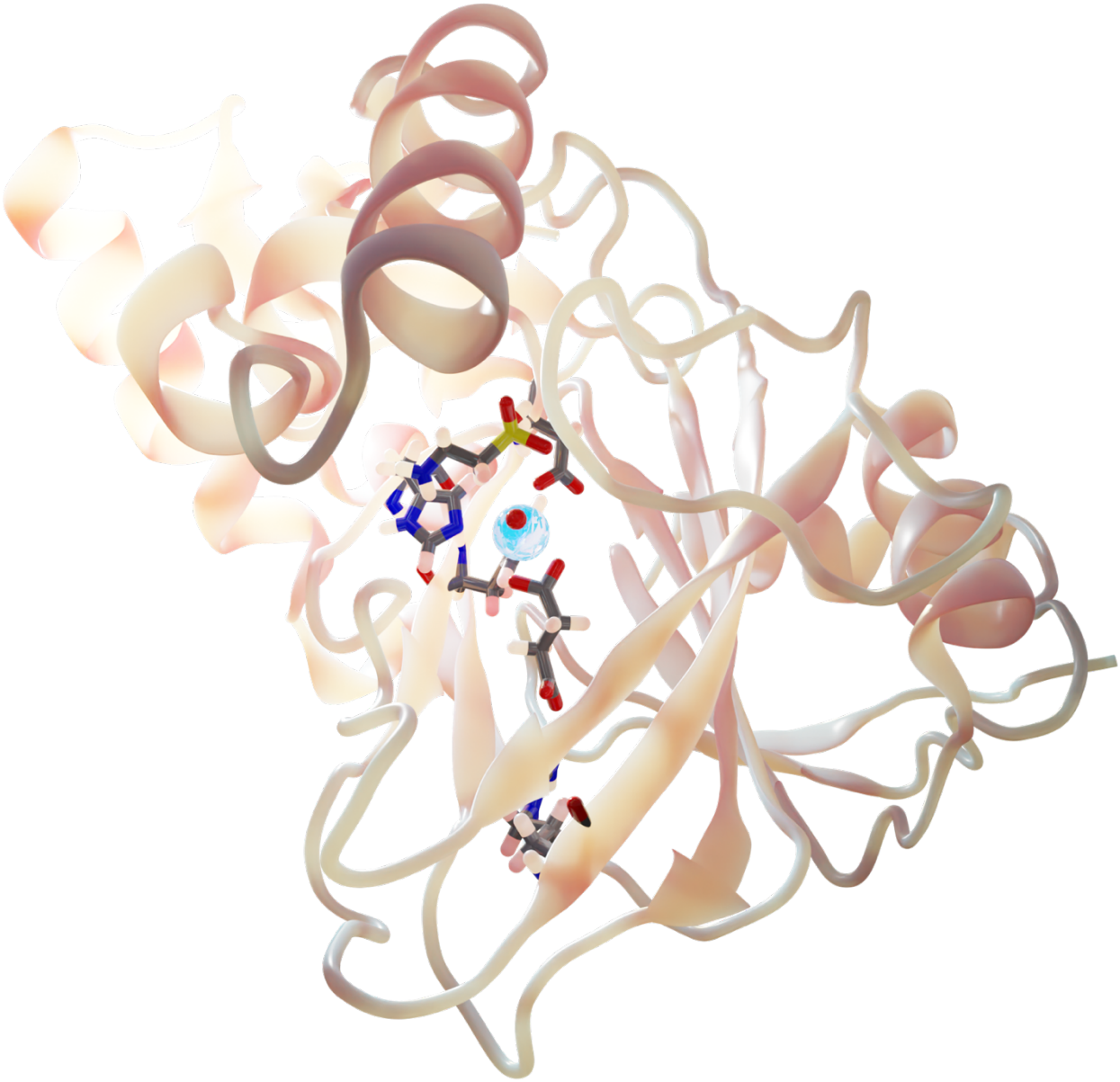

COMPUTATIONAL PROTOCOLS FOR BIOLOGICAL SYSTEMS

Advanced Tutorial for QM and MD Simulations and Calculations
Written by David W. Kastner

CONTENTS

ABBREVIATIONS

AMBER	Assisted Model Building with Energy Refinement
BCP	Bond Critical Points
CREST	Conformer-Rotamer Ensemble Sampling Tool
CSA	Charge Shift Analysis
DCCA	Dynamic Cross Correlation Analysis
EDIA	Electron Density for Individual Atoms
GBSA	Generalized Born Surface Area
HYSCORE	Hyperfine Sublevel Correlation Spectroscopy
MCPB	Python Based Metal Center Parameter Builder
MD	Molecular Dynamics
MM	Molecular Mechanics
MOL2	Tripos Molecules Structure Format
NCI	Noncovalent Interactions
NetCDF	Network Common Data From
PCA	Principal Component Analysis
PDB	Protein Database file
QM	Quantum Mechanics
QM/MM	Quantum Mechanics/Molecular Mechanics
RG	Radius of Gyration
RMSD	Root Mean Squared Deviation
RMSF	Root Mean Squared Fluctuation
TauD	Alpha-ketoglutarate-dependent taurine dioxygenase
TLEaP	Terminal-based Link, Edit, and Param
VMD	Visual Molecular Dynamics
XSEDE	Extreme Science and Engineering Discovery Environment

PROTOCOLS

1 PREPARING INPUT FILES

Note: Since so many files are generated over the course of the preparation process, PDB versions are based on the step in the tutorial. For example, a PDB generated in step 1.1.1 would be called `1os7_v111.pdb`.

1.1 Evaluating a PDB

1. First, we need to choose a PDB. This can be difficult if there are too many or two few choices. However, we can evaluate the quality of a protein structure and compare it to other options using the Electron Density Score for Individual Atoms (EDIA).[1] It can be access as part of the Protein*Plus* database.[2]
    a. https://proteins.plus/
    b. https://proteins.plus/1os7?edias/23052#edia

1.2 Basic PDB modifications

1.2.1 Remove extra chains

2. Your protein will almost certainly have more than one chain. However, we will want to select a single chain to work with. You can do this intuitively, choose the one that looks best to you or as a specific conformation, or select the chain that has the best EDIA score.
3. Open ChimeraX and the PDB you are interest in:
    a. `>> open 1os7`
4. Remove all extra chains:
    a. Select > Chains > α-ketoglutarate-dependent taurine dioxygenase > **Chain A**
    b. Select > **Invert**
    c. Actions > Atoms/Bonds > **Delete**
5. Save as new PDB:

   a. File > **Save**
   b. Save as: **`1os7_v111.pdb`**

6. If you have ANISOU entries in your PDB file these can easily be removed using.
   a. `grep -v ANISOU your.pdb > no_anisou.pdb`

1.3 Adding missing residues with Modeller

1.3.1 Installing Modeller

7. Most PDBs will have residues that were left unresolved during crystallography. These residues should be added back in using the Modeller software.[3, 4]
8. Create a conda environment where you can install the Modeller library. You can call the environment anything that you would like. I will name the environment `md` for molecular dynamics. Then you can install Modeller through condaforge.
   a. `>> conda create --name md`
   b. `>> conda config --add channels salilab`
   c. `>> conda install modeller`
9. Once conda is finished installing Modeller, there will be a printout at the end of the terminal. It will tell you the location of your `config.py` where you will need to update your '`license`' variable.
10. You can request a license by going to the following url:
    a. https://salilab.org/modeller/registration.html

1.3.2 Adding the missing residues

11. Once you have Modeller installed and you have included your license key, you are ready to fill in your missing residues.
12. While I will explain the protocol, you can also visit the Modeller website and walk through the missing residues tutorial on your own:
    a. https://salilab.org/modeller/wiki/Missing%20residues
13. This has been automated and can be found here:
    a. https://github.com/davidkastner/pyQMMM/blob/main/pyqmmm/md/modeller_automator.py

1.4 Modelling the reactive state

1.4.1 Add coordinating oxygen atom

14. For many of the metalloenzymes that we work with there is a key oxygen atom that that is coordinated to the Fe that doesn't appear in the PDB file. When choosing how to place the oxo

group you can do one of three things: 1) there may be a coordinated water molecule that you can use as a reference for the oxo group or 2) use another non-heme enzyme PDB as a guide.

15. In the case of 1OS7 TauD, we will model the oxo position using 6EDH as a reference. I recommend using the software Avogadro.[5]
    a. Draw oxo group.
    b. Selection Tool > **Select oxo group**
    c. Select > **Invert Selection**
    d. Extensions > Molecular Mechanics > **Fix Selected Atoms**
    e. AutoOptimization tool > **Start**
    f. Copy the new coordinates back into the `1os7_v112.pdb`
    g. Save as: **`1os7_v113.pdb`**

1.4.2 Replace a-ketoglutarate and succinate

16. The crystal structure, 1OS7, captured a different step of the reaction than the one that we are interested in. Therefore, we will need to exchange a-ketoglutarate with succinate. We will use the PDB 6EDH as a reference as it contains succinate accurately positioned.
    a. Replace the carboxyl group with an oxygen and adjust the geometry in Avogadro.
    b. Add the hydrogens to succinate while we are at it.
    c. Save as: **`1os7_v114.pdb`**

1.4.3 Protonation state using H++

17. Submit the PDB from modeler to the H++ webserver.[6]
    a. http://biophysics.cs.vt.edu/
18. Use the following conditions to replicate the protein environment:
    a. **`Internal dielectric`** `= 4`
    b. **`External diaelectric`** `= 80`
    c. **`Salinity`** `= 0.15`
    d. **`pH`** `= 7.0`
19. Download the PDB, topology (top), and coordinate files (crd)
20. Save as: **`1os7_v115.pdb`**

Note: The trailing whitespaces on the top file are essential. Make sure that your text editor is not removing it. For example, the text editor Atom will remove it and ruin the top file unless you turn the setting off.

1.5 Pre-MCPB ligand file preparation

Note: For more questions and alternate descriptions of the MCPB section of this protocol, refer to the following online tutorial: http://ambermd.org/tutorials/advanced/tutorial20/mcpbpy.htm

1.5.1 Preparing the computational environment

21. The following sections involve preparing the input files for MCPB, which we will use to generate molecular dynamics parameters. While this section can be performed on your local, it is recommended to run them on the resource where you will run the molecular dynamics, e.g., a remote HPC cluster.

1.5.2 Install miniconda on Gibraltar

22. First, we will create a virtual environment for the MCPB packages. If you already have a virtual environment, feel free to add to that. You can execute the following set of commands to get started:
    a. `>> conda create -n AmberTools22`
    b. `>> conda activate AmberTools22`
    c. `>> conda install -c conda-forge ambertools`
23. The AmberTools22 package contains everything that we will need for now.

1.5.3 Prepare the H++ output PDB for Amber

24. For additional practice, refer to the amber tutorial for setting up proteins and metalloproteins.
    a. `http://ambermd.org/tutorials/advanced/tutorial20/mcpbpy.htm`
    b. `http://ambermd.org/tutorials/pengfei/`
25. `ssh` into you cluster where Amber is installed.
26. Load the Amber module using your installation of the newest version of AmberTools. You may already have this activated.
    a. `>> conda activate AmberTools22`
27. Since H++ uses atom names that Amber does not recognize, we need to generate an amber compatible PDB using the topology and coordinate file from H++.
28. Add a new line to the end of the `crd` file as a terminal EOL is required but isn't generated by H++.
    a. `>> ambpdb -p 1os7.top -c 1os7.crd > 1os7_v121.pdb`
29. Most of the protonation states should be correct except those around the metal are most likely wrong. In the case of TauD, H++ has protonated the coordinating nitrogen of the two histidines. Make sure to remove these protons by opening the PDB and deleting them. His255 is also missing a hydrogen.

1.5.4 Extract ligand PDB data

30. Now that we generated an Amber PDB for the protein, we will need to do the same for the ligands. We pull the ligand from the PDB using `awk` and protonate it using `reduce`. You can then clean the structure using `pdb4amber` before performing a manual inspection. This step can be skipped if you already have the protonated version of your ligand separate. Repeat this step for all ligands in your system. In TauD, there is a taurine and an alpha-ketoglutarate.
    a. `>> awk '$4=="SUC"' 1os7_v114.pdb > SUC.pdb`
    b. `>> reduce SUC.pdb > SUC_H.pdb`
    c. `>> pdb4amber -i SUC_H.pdb -o SUC_H_clean.pdb`
31. Repeat for taurine:
    a. `>> awk '$4=="TAU"' 1os7_v114.pdb > TAU.pdb`
    b. `>> reduce TAU.pdb > TAU_H.pdb`
    c. `>> pdb4amber -i TAU_H.pdb -o TAU_H_clean.pdb`
32. Repeat for iron:
    a. `>> awk '$4=="FE2"' 1os7_v114.pdb > FE.pdb`

Note: Pdb4amber will alert you to any missing heavy atoms. This can likely be fixed by hand since only a few atoms should be missing, but don't ignore any pdb4amber errors.

Note: It is absolutely essential that you go in and make sure that the atom type and the atom name are the same. For example, use FE for iron or ZN for zinc but not FE2. You will get an error saying 'FE' key not found if you forget to change this.

```
HETATM    1 FE    FE A   1      29.979  79.059  23.710  1.00 61.74          Fe
```

1.5.5 Mol2 and frcmod files from parmchk2 and antechamber

33. The method shown in RED RESP ESP step is more accurate as it uses Gaussian to calculate the parameters. However, we will first use antechamber to find out if parameters already exist for some of our small molecules.
34. Generate a `mol2` file and a `frcmod` file for the ligands. Antechamber will generate the `mol2` file. The `parmchk2` program can then construct the `frcmod` file.
    a. `>> antechamber -fi pdb -fo mol2 -i suc.pdb -o suc.mol2 -c bcc -pf y -nc -2`
    b. `>> parmchk2 -i HMM.mol2 -o HMM.frcmod -f mol2`
35. Repeat for taurine:
    a. `>> antechamber -fi pdb -fo mol2 -i TAU_H_clean.pdb -o TAU.mol2 -c bcc -pf y -nc 0`

b. `>> parmchk2 -i TAU.mol2 -o TAU.frcmod -f mol2`

Note: Add the parameter -nc -1 for the charge of the molecule. A charge of zero is the default if nothing is added.

36. If you open the `frcmod` files, you will notice that while succinate contains parameters, the file for taurine does not. This means that all the parameters need to simulate taurine are already contained within the AMBER GAFF forcefield. That means we will only have to parameterize succinate in the RESP step.
37. The oxo group consists of a single atom and therefore is not supported by antechamber. We will need to write it out by hand as shown below. Make sure to update the coordinates to match your system.

```
@<TRIPOS>MOLECULE
OXO
    1     0     1     0     0
SMALL
No Charge or Current Charge

@<TRIPOS>ATOM
      1 O         -13.3790   65.2270   25.7390 O       285 OXO      -2.000000
@<TRIPOS>BOND
@<TRIPOS>SUBSTRUCTURE
     1 OXO         1 TEMP              0 ****  ****    0 ROOT
```

1.5.6 RED RESP ESP Server for frcmod files

38. The mol2 files for the ligands created by Antechamber and Parmchk2 are a good start, however, we can improve upon the bcc charges that it predicted by using the online Red Resp server.

1.5.7 Follow the subsequent steps to use the Red Resp server

39. You can access the RESP ESP server at the following path.
    a. https://upjv.q4md-forcefieldtools.org/REDServer-Development/
40. Login steps:
    a. Submit
    b. Private account
41. Submission of Data 1/5:
    a. Leave all as defaults.
    b. **Next**
42. Submission of Data 2/5:
    a. Leave all as defaults.

b. **Next**

43. Submission of Data 3/5:
    a. Project name: TauD
    b. Quantum mechanics program: Gaussian 16_C.01
44. Submission of Data 4/5:
    a. Create a new directory called taud
    b. Add `SUC_H_clean.pdb` to the directory.
    c. Rename the PDB to Mol_red1.pdb
    d. Zip the folder.
    e. Choose File: Upload the archive to the RED server.
    f. **Let's go.**
45. When the red server is done, it will provide you with the parameters in a `frcmod` file and a PDB. However, these files are not yet compatible with MCPB. In the dihedral section of the `frcmod` files, you will notice that there are five columns whereas the `frcmod` files produced by antechamber only have four. Delete the column labeled `Path`.

1.5.8 Combine all generated PDBs

46. Now that we have separate PDB files for the cleaned protein, ligands, and metals, we can use the `cat` command to combine them back into a single file.
    a. `>> cat 1os7_v121.pdb FE.pdb OXO.pdb SUC_H_clean.pdb TAU_H_clean.pdb > 1os7_v124.pdb`

1.5.9 Clean the final PDB

47. Now that we have a single PDB file with the protein, ligands, metal, and oxo group will want to do some final manual formatting as we will be using this. As our base PDB.
    a. `>> pdb4amber -i 1os7_v124.pdb -o 1os7_v125.pdb`
48. Manually add TER endings after each grouping for clarity as well as optional chain identifiers. After this protocol you will end up with all the same chain identifiers.

Note: The AmberTools21 implementation of pdb4amber requires that there be a chain listed or it will throw a python types error.

2 METAL COORDINATION SITE PARAMETERIZATION

2.1 Metal file preparation

2.1.1 Generate metal MOL2 file

1. We need to generate a `mol2` file as we did previously for the metal ions. We will use the script `metalpdb2mol2.py`. We can then run the script using the keyword `-c 4` to designate a charge of four for iron.[7]

    a. /home/kastner/scripts/MCPB/metalpdb2mol2.py

    b. `>> python metalpdb2mol2.py -i FE.pdb -o FE.mol2 -c 4`

2.1.2 Generate MCPB and Gaussian input files

49. We now generate the Gaussian and fingerprint modelling files. The input will need to be written manually. A copy of the input used here can be found in the following directory.

    a. `>> /home/kastner/projects/taud/1_prep/3_mcpb/3_metal/1OS7.in`

```
original_pdb 1os7_v125.pdb #The most recent optimized PDB
group_name 1os7 #A prefix designating all files generate for this section
software_version g09 #Use "software_version gms" if you are using GAMESS
cut_off 2.8 #Angstrom cutoff indicating the metal is bond to a neighbor
ion_ids 4536 #The atom ID of the complex's central metal ion in original_pdb
ion_mol2files FE.mol2 #The mol2 file of your metal ion
naa_mol2files OXO.mol2 SUC.mol2 TAU.mol2 #Indicates non-amino acid mol2 files
frcmod_files SUC.frcmod #Indicates non-amino acid frcmod files
large_opt 1 #Optimizes hydrogen positions
```

50. Once the input file has been completed with the necessary commands, we can execute the MCPB.py file.

    a. `>> MCPB.py -i 1OS7.in -s 1`

Note: When we combined the PDBs back together it is convention to add standard residues first, then the metal ions, then ligand, and then water.

Note: Don't forget to change the ion_ids whenever you change something in your input pdb file. If you don't you will get a `KeyError`.

2.2 QM calculations

2.2.1 Optimize Geometry with TeraChem

51. MCPB.py just produced `1os7_small.pdb`, a very useful file for double checking the quality of the simulation. Open it in a molecular visualizer to make sure that everything looks good. In this tutorial, we have had manually made a lot of changes to the structure that may or may not be accurate. To fix this, we will use `1os7_small.pdb` and optimize the structure with TeraChem.[8]
52. We will need to create three files: `1os7_activesite.xyz`, `1os7_activesite.in`, and `jobscript`.
53. To create `1os7_activesite.xyz`, open `1os7_small.pdb` in Avogadro and save it as an xyz file.
54. The other two files can be easily generated by using mine as templates. You can find them in the prep folder.
    a. `>> cd /home/kastner/projects/taud/1_prep/`
    b. `>> module load sge`
    c. `>> qsub jobscript`
    d. `>> qstat -u kastner`

2.2.2 Update Gaussian Scripts

55. MCPB.py will generate twelve files. While the files are almost ready to submit, we will have to make a few modifications to the Gaussian submit files ending in `.com`. The three files that we will have to modify are:
    a. `1os7_small_opt.com`
    b. `1os7_small_fc.com`
    c. `1os7_large_mk.com`
56. For all `.com` files we will be making similar changes. First, we change row five to an unrestricted functional (`UB3LYP/GEN`), row ten two a charge of -1 and a spin state of five, and at the bottom of the file, we change the method that the Fe will be treated with to `LANL2DZ`. Make these changes to each of the three .com files. The final scripts can be inspected at:
    a. `>> /home/kastner/projects/taud/2_mcpb/succinate/gaussian/comet/`
57. Next, we will generate four `sbash` scripts for submitting the gaussian jobs to Excede. These files can be copy and pasted from the following directory:
    a. `/home/kastner/projects/taud/2_mcpb/succinate/gaussian/comet/`
    b. `/gaussianScript_1os7_large_mk`
    c. `/gaussianScript_1os7_small_fc`
    d. `/gaussianScript_1os7_small_fchck`
    e. `/gaussianScript_1os7_small_opt`

2.2.3 Submit the Gaussian scripts

58. Submit the first and second MCPB scripts using the first and second commands. Watch the status of your job live run the second. The small optimization will likely take around two days and the Merz-Kollman RESP charge calculation will take less than a day.
    a. `>> sbatch gaussianScript_1os7_small_opt`
    b. `>> sbatch gaussianScript_1os7_large_mk`
    c. `>> watch squeue -u kastner`
59. Once the small geometry optimization finishes, we submit the next calculation.
    a. `>> sbatch gaussianScript_1os7_small_fc`

2.3 Working with TLEAP

2.3.1 Final modeling and TLEAP

60. Once the `fc` calculation finishes, we convert the checkpoint file to an formatted checkpoint file. The default checkpoint is a large binary file that is not readable by the utilities we will use next.
    a. `>> sbatch gaussianScript_1os7_small_fchck`
61. Move all the gaussian files back to the working directory in Gibraltar.
62. Use MCPB.py to perform the Seminario method to generate force field parameters. This will generate a `frcmod` file called `1OS7_mcpbpy.frcmod`. I tried using the older version of AmberTools16 and it does not work with some formatting/styling choices.
    a. `>> conda activate AmberTools21`
    b. `>> MCPB.py -i 1OS7.in -s 2`
63. We perform RESP charge fitting and generate a mol2 file for the metal center residues.
    a. `>> MCPB.py -i 1OS7.in -s 3`
64. Generate the `tleap` input file and a new PDB. The new PDB, called `1os7_mcpby.pdb`, should be inspected to make sure everything is correct.
    a. `>> MCPB.py -i 1OS7.in -s 4`
65. The TLEaP file is almost complete. However, as you might have noticed taurine was not part of active site Gaussian calculations so we will have to add a line into the TLEaP file so AMBER knows where to find the TAU mol2 file.
    a. `>> TAU = loadmol2 TAU.mol2`
66. We then run the `tleap` file to get `1OS7_solv.prmtop` and `1OS7_solv.inpcrd`. These are the key files that we will need to run our AMBER simulations. It will also generate a final solvated PDB, `1os7_solv.pdb`, that you should visualize to confirm that everything is correct.
    a. `>> tleap -s -f 1os7_tleap.in > 1os7_tleap.out`
    b. `>> tleap -s -f IC_tleap.in > IC_tleap.out`

c. `>> ambpdb -p mc6sa_solv.prmtop -c constP_prod.restart > constP_prod.pdb`

3 MOLECULAR DYANMICS SIMULATIONS

3.1 MD simulations setup

3.1.1 Amber input files

1. Now we will begin preparing the input files for running an Amber simulation. There are only two files that we need from the previous section to start an MD simulation: 1os7_solv.inpcrd and 1os7_solv.prmtop. Create a new directory to store your MD simulations and move these two files to the new directory.
2. There are seven files that we will now need to write in order to run an Amber simulation. These files tell Amber what you would like it to do. In general, these base files that you will create are highly recyclable and will only need minor modifications when spinning up new simulations. They are as follows:
    a. **amber_multistage.q**: This is the master file that tells Amber which files to run and in what order. When basing your files off these example files, most of the scripts can stay the same. However, you may have to make a few changes to this script to accommodate your file system names.
    b. **restrained_min1.in**: The initial minimization where we fix all atoms except for hydrogens and allow unfavorable interactions to settle. This step is important because the hydrogens were added in by H++ and there may be some unfavorable angles, lengths, or torsionals.
    c. **restrained_min2.in**: In the second minimization, we will fix only the backbone of the protein and the active site. In general, we assume the crystal structure to be correct. Therefore, we want the side chains to settle into favorable conformations that support the backbone conformation found in the crystal structure.
    d. **unrestrained_min.in**: In the next minimization step, we remove the restraints on the backbone but retain the active site restraints. This will allow the protein to settle in such a way that it supports the true configuration of the active site.
    e. **constV_equil.in**: The equilibration steps will begin with a constant volume simulation as we slowly heat the system to 300K.
    f. **constP_equil.in**: Once the system has reach room temperature, we will give it a chance to equilibrate at that new temperature under constant pressure.
    g. **constP_run.in**: Prior to the final production step, we want the system to be as stable as possible. To help stabilize the system we will run a pre-production simulation that is longer than any of the previous steps.
    h. **constP_prob.in**: This is the final production script and tells Amber how we will run a much longer simulation. In general, it is the same as the previous step with the time increased.

3.1.2 Initial simulation analysis

3. The scripts for all these steps can be found in my directory.
    a. `>> /home/kastner/projects/taud/3_md/unrestrainedmd/`
4. Copy these files to your personal directory where you would like to run your Amber simulation.
    a. Open `amber_multistage.q` and change your job identifier
5. You may also want to make changes to some of the commands in the other files if you want Amber to do something to handle a nuance of your system. To find more on each individual command there is no better way than taking a look at the newest manual, which are very informative.
    a. https://ambermd.org/doc12/Amber18.pdf

3.1.3 Running the simulation

6. Now you are ready to submit the simulation. You can submit your job to the queue with the following series of commands. You can occasionally check on the queue with the last to command to make sure it is running and hasn't been killed for an internal error.
    a. `>> module load sge`
    b. `>> qsub amber_multistage.q`
    c. `>> qstat -u kastner`

3.2 Initial simulation analysis

3.2.1 Converting to NetCDF file type

7. The trajectory files that are produced as output have the extension `mdcrd`, which are massive ASCII file. This is how we can convert the `mdcrd` to a NetCDF file which will have the extension `nc`.
8. While a NetCDF will always be faster to work with, it is 50x larger and is in the range of 50GB so only use them when you have to.
    a. `>> cpptraj -p 1os7_solv.prmtop -y constV_equil.mdcrd -y constP_equil.mdcrd -y constP_run.mdcrd -y constP_prod.mdcrd -x combined.nc`

3.2.2 Calculating the RMSD and RadGyr for the trajectory

9. It is a good idea to measure the RMSD and radius of gyration once the simulation has finished. The RMSD will give you a good idea of whether or not the structure has reached equilibrium and the radius of gyration will monitor whether or not the structure is experiencing any unfolding. To do this, we can use the Amber utility CPPTraj.
    a. `>> module load amber/18`
    b. `>> cpptraj`

c. `>> parm 1os7_solv.prmtop`

d. `>> trajin constP_prod.mdcrd`

e. `>> rms ToFirst :1-287&!@H= first out rmsd.dat mass`

f. `>> radgyr :1-287&!(@H=) out rog.dat mass nomax`

g. `>> run`

3.2.3 Basic analysis Perl script

10. Amber provides a quick way to perform analyses in the form of a widely distributed script. Running this script will create a file containing data files for all the most common types of analysis including the temperature over the simulation, the energy, the pressure, the solvent, the volume, and the solvent density.
11. The script can be found in my scripts folder.
    a. `>> cd /home/kastner/scripts/amber/process_mdout.perl`
12. Move the script to your working directory.
13. The script is written in python so we will need to add Perl to our Amber related.
14. Activate your anaconda environment if you haven't already.
    a. `>> conda env list`
    b. `>> conda activate AmberTools21`
    c. `>> conda install -c anaconda perl`
15. Create a new folder called analysis and move into it and execute the script. You will need to give the script the name of your production output file as the second argument.
    a. `>> mkdir analysis`
    b. `>> chmod 777 ../process_mdout.perl`
    c. `>> cd analysis`
    d. `>> ../process_mdout.perl ../constP_prod.out`

3.2.4 Restrained MD Simulations

16. In the case of non-heme iron enzymes, we have found that the parameters of the ligand—Taurine (TauD), L-arginine (VioC), threonine (SyrB2), etc.—often are unable to produce the reactive conformation. This is likely a result of shortcomings in the parameterization for the unique electronic environment of non-heme iron enzymes.
17. As a solution, we can use spectroscopic restraints such as HYSCORE to guide our simulations.[9]
18. HYSCORE is a spectroscopic technique that takes advantage of the metallic center of metalloenzymes. The metal acts like a magnetic probe which can be used to understand the atoms around it. This is specifically useful to gain insights into the bond lengths and angles of the coordinating residues.

19. To set up a HYSCORE guided simulation, you will need to modify some of the input files as well as include a file called `restraints.dat` which specifies what the restrains should be.
20. To each Amber file (except the minimization) we will need to add the command `nmropt=1`. This indicates to Amber that we will be providing NMR restraints.
21. Next, we will need to specify the restraints file in another section.

3.2.5 Restrained MD Simulations

```
/
&wt type='DUMPFREQ', istep1=50
/
&wt type='END'
/
DISANG=restraints.dat
DUMPAVE=distance.dat
```

22. For an example of how these commands are added to each of the Amber job files, you can inspect the follow directory for an example.
    a. `>> /home/kastner/hydroxylases/taud/1os7/3_md/restrained/`
23. Once these additional commands have been added to Amber submission scripts, you can run the simulations as you would an unrestrained simulation.

4 QM/MM SIMULATIONS

4.1 QM System Setup

4.1.1 Choosing a frame for QM

1. We will now need to choose a frame from our MD run to use as our reference structure for the QM/MM calculations. Since we already have a general intuition for which features would be indicative of the transition state, we can screen the entire trajectory for instances that match those criteria.
2. For example, the closer Tau:H7 is to Oxo:285, the closer to the transition state we are.
3. Let's measure the Tau:H7-Oxo:285 distance as a function of time using CPPTRAJ.
    a. `>> module load amber/18`
    b. `>> cpptraj`
    c. `>> parm 1os7_solv.prmtop`
    d. `>> trajin constP_prod.mdcrd`
    e. `>> distance DIST :287@H7 :285@O out distance.dat`
    f. `>> run`
4. This analysis tells us that the smallest distance is 2.09 and occurs at the very end of the trajectory at frame 124876. We will note the top three small distances as well as the largest distance for comparison.
    a. Frame **124876** 2.09 Å
    b. Frame **97127** 2.10 Å
    c. Frame **48726** 2.11 Å
    d. Frame **91340** 6.11 Å

4.1.2 Spherical solvent cap

5. We have more than 9999 molecules. This is problematic for PyMol so we will need to select the closest 9000 waters. I wrote a script to renumber the residues, but it did not help because PyMol only looks at four columns. The best workaround is to use `cpptraj` to select the closest 9000 waters.
    a. `>> module load amber/18`
    b. `>> cpptraj`
    c. `>> parm 1os7_solv.pdb`

   d. `>> trajin constP_prod.mdcrd`
   e. `>> strip :Na+`
   f. `>> autoimage`
   g. `>> closestwaters 9000 :1-287 noimage center closestout closestwat.dat outprefix closest`
   h. `>> trajout closest9000_f pdb multi onlyframes 105626`
   i. `>> run`

6. Open the PDB in PyMol. Note that the PDB will not be generated with the PDB extension.
7. Download the center_of_mass.py script:
   a. https://github.com/Pymol-Scripts/Pymol-script-repo/blob/master/center_of_mass.py
8. Using the command line in PyMol, navigate to the `center_of_mass.py` script and execute.
   a. `>> run center_of_mass.py`
   b. `>> sele prot, resi 1-287`
9. The `prot` selection indicates the name you would like to give your selection. You can leave it as `prot` and 287 can be replaced with whatever the last residue in your system is, not including solvent.
10. Calculate the proteins center of mass with the following command. The resulting coordinates will be printed out.
    a. `>> com prot`
11. Select the solvent cap. Choose a value for the radius that gives a good amount of solvent padding all around the protein. In this example, 35 should be enough. Next, copy the selection to a new object and delete the `prot_COM` psuedoatom.
    a. `>> sele sphere, br. all within 34 of prot_COM`
    b. `copy to object > new`
    c. Select `prot_COM`
    d. `A > remove atoms`
12. Save the selection to a new PDB. We will use this for the QM/MM calculations.
    a. `File > export molecule… > Original atom order (according to “rank”)`
    b. `PDB Options > Retain atom ids`

4.2 Charge Shift Analysis

4.2.1 Background reading

13. To determine the appropriate size of the QM region we can use charge shift analysis (CSA).[10, 11] CSA is a method to systematically determine which residues should be included in the QM regions. A general description can be found on the lab website: http://hjkgrp.mit.edu/csafsa.

4.2.2 Selecting residues

14. First, we need to decide which residues to include in the QM region. This choice will be slightly arbitrary; however, there are a few rules that can help guide us in our decisions. First, we want between 600-1000 atoms and second, we want to maintain net charge neutrality.
15. Start by opening one of your final TauD structures in PyMol. Select all residues within 11.6Å, which will provide you with 1009 atoms. This is good start as we want approximately 1000 atoms. Then move that sphere to a new object.
    a. `>> sele sphere, br. all within 11.6 of resi 284`
16. Save the selection as a new PDB to get a list of the residues in your selection. In this example the selected residues are as follow. However, if this selection is too large and struggles to converge decrease the radius to 11, which will give us around 900 atoms. If possible, it is a good idea to maintain net charge neutrality, meaning the charge for the region should be zero. Luckily, our ligands cancel so if we can achieve net charge neutrality for the apo, then we will achieve it for the holo as well.
17. To get a net charge of zero, we will add the next closest arginine, Arg190, and remove Asp125 and Glu204, both of which would be masked by the solvent.
18. It is a good idea to check the QM selection in PyMol to make sure that you did not select too many residues. We can see from the printout that we have 1005 atoms defined, which is close enough.
    a. `>> sele resi 44+46+57+68+70+71+73+83+84+85+87+93+94+95+96+97+98+99+100+101+102+103+104+108+109+111+112+113+114+115+124+126+127+128+129+155+158+159+203+205+206+240+246+247+248+249+250+251+253+254+255+256+257+258+266 +268+269+270+271+272+273+190`
    b. `Selector: selection "sele" defined with 1005 atoms`

4.2.3 Create holo enzymes

19. Now that we know what to include in our QM region, we will need to generate the structures for the *apo* and *holo* enzymes. The term *apo* refers to an empty active site and *holo* refers to an occupied active site. We will start with the *holo* enzyme:
20. Follow the directions in the section *Clustering a Trajectory* to obtain a representative snapshot of the experimentally observed conformation. In this case we will use frame 1 from the production run.

21. Follow the solvation cap steps in 4.1.2 to obtain a PDB of your snapshot with a spherical water cap.
22. This will be the starting structure used for the final TLEaP steps of MCPB.
    a. 1os7_holo.pdb
23. You can use almost the exact same TLEaP script for the *holo* enzyme. You will need to edit the solvent box line and replace the line indicating our new PDB:
    a. `solvatecap mol TIP3PBOX (40.144937, 41.414312, 45.752116) 34`
24. You will need to use curly braces instead of parentheses 16. Execute the TLEaP script.
    a. `>> tleap -s -f 1os7_tleap.in > 1os7_tleap.out`
25. You can move the generated *inpcrd* and *prmtop* files to your CSA folder.

4.2.4 Create apo enzymes

26. For the *apo* structure, delete the all the ligands from 1os7_v125.pdb and save is as 1os7_apo.pdb.
27. Edit the 1os7_tleap.in file to account for the addition of a solvent cap and change the pdb assigned to the `mol` variable to the new PDB we just created.
    a. `solvatecap mol TIP3PBOX {40.124726 41.397530 45.731101} 34`
28. Execute the TLEaP script.
    a. `>> tleap -s -f 1os7_tleap.in > 1os7_tleap.out`
29. Recreate a TLEaP file for the apo enzyme. In this case, we cannot use the MCPB generated script because we are not suing the same modified coordinating residues now that the metal has been removed.
    a. `>> tleap -s -f 1os7_apo_tleap.in > 1os7_apo_tleap.out`

```
source oldff/leaprc.ff14SB
source leaprc.gaff
source leaprc.water.tip3p
loadamberparams frcmod.ions1lm_126_tip3p
loadamberparams 6edh_mcpbpy.frcmod
mol = loadpdb 6edh_centroid.pdb
check mol
addions mol Na+ 0
addions mol Cl- 0
check mol
solvatecap mol TIP3PBOX {40.144937, 41.414312, 45.752116} 34
savepdb mol 6edh_solv.pdb
saveamberparm mol 6edh_solv.prmtop 6edh_solv.inpcrd
quit
```

4.3 Calculating key quantum properties

4.3.1 Charge of the QM region

30. Once you have run the QM/MM charge calculations for the apo and holo enzymes, the next step is to analyze the charges and determine which residues experience the most charge transfer.
31. I have created a tutorial in video format designed to guide you through a python script that we created called QuickCSA.py that should take care of this process automatically.
    a. https://www.youtube.com/watch?v=Zck8fznmTPA&ab_channel=DavidKastner
32. For the QM calculations we will need to know the charge of our QM region. To calculate the charge, we find the charge of each residue in the QM region and sum the values to find the overall charges. The charges of the residues in the minimal active site region of TauD are as follow:
    a. `Asp94 -1`
    b. `Asn95 +1`
    c. `His99 0`
    d. `Asp101 -1`
    e. `Trp248 0`
    f. `His255 0`
    g. `Fe284 +4`
    h. `Oxo285 -2`
    i. `Suc286 -2`
    j. `Tau287 0`
    k. `Arg270 +1`

4.4 Submitting the QM/MM calculation

4.5 Simulation visualization

4.5.1 Vizualization with VMD

33. To visualize our trajectories, we want to first remove all the waters to decrease the file size by several orders of magnitude. Then we want to convert the `mdcrd` file type to dcd which can be read by a wider range of visualization software's. We will also use this as an opportunity to align all the frames to make it easier to visualize over time.
    a. `>> parm 1os7_solv.prmtop`
    b. `>> trajin constP_prod.mdcrd`
    c. `>> autoimage`

d. `>> rms first :1-287 out rms_pdbsnap.dat`

e. `>> strip :WAT,CL- outprefix stripped`

f. `>> trajout 1os7_dry.dcd`

g. `>> run`

5 DATA ANALYSIS

5.1 General data analysis

5.1.1 Visualizing trajectories with XMGrace

1. You can view these files using xmgrace, just make sure to activate your interactive GUI is activated and that you refer to the lab template.
    a. `>> ssh -Y kastner@gibraltar.mit.edu`
    b. `>> xmgrace /home/kastner/scripts/xmgrace/xmgrace_template.agr`

5.2 Multiwfn

5.2.1 Convert molden to more stable format

2. To run Multiwfn smoothly, SSH into the interactive compute node and load the prerequisite modules. Then move to the tutorial folder to follow along:
    a. `>> ssh Gibraltar-53`
    b. `>> module load intel/19.0.117`
    c. `>> module load multiwfn/noGUI`
    d. `>> cd /home/kastner/tutorials/bcp/BesD/scr/`
3. Convert the molden file to a wfn file which will have more reliable performance. While Multiwfn will accept the molden file there are known issues for calculating BCPs and should be used carefully. Here we will perform the conversion to be safe, which can be formed easily and conveniently directly within Multiwfn.
    a. `>> pwd`
4. You will need the absolute path to the molden file later. When you need it, just scroll back up to this line or copy it now.
    a. `>> Multiwfn *molden`
    b. `>> 100 // Other functions (Part 1)`
    c. `>> 2 // Export various files (mwfn/pdb/xyz/wfn/wfx/molden/fch/47/mkl …) or generate input file of quantum chemistry programs`
    d. `>> 5 // Output current wavefunction as .wfn file`
5. You will be asked for the path of where to save the new file.
    a. `>> ./besd.wfn`
6. Once the conversion is finished, Multiwfn will print Done.

a. `>> Ctrl + C`

5.2.2 Topology analysis to find Bond Critical Points (BCP)

7. Load in the new `wfn` file (if not still loaded) and begin searching for BCPs.
    a. `>> Multiwfn *wfn`
8. Use topology analysis option to search for a variety of different CPs.
    a. `>> 2 // Topology`
9. Since we are interested in a specific non-covalent interaction, we can use option 1 and give the indices of the two atoms that define the hydrogen bond that we are interested in. Don't worry when nothing is printed. The data for the CP will be saved internally.
    a. `>> 1 // Search CPs from a given starting point`
    b. `>> 2 // Using midpoint of two atoms as starting point`
    c. `>> 145,155`
    d. `>> 0 // Return`
10. Print a quick summary of the real space function outputs corresponding to our BCP. It will return the coordinate of our BCP in Bohr (62.7,62.1,72.6), which can convert to angstroms (33.2,32.9,38.4). This will be useful later.
    a. `>> 0 // Print and visualize all generated CPs, paths and surfaces`
11. Multiwfn will print gradient vector field definitions corresponding to the type of bond critical point that was found. Use the following definitions to confirm that you have a BCP. In our case, we get (3,-1), confirming that we have identified a BCP.
    a. **`(3,-3)`**: An atom or atomic critical point
    b. **`(3,-1)`**: A bond/non-covalent interaction or a *bond critical point (BCP)*
    c. **`(3,+1)`**: The center of a ring or a *ring critical point*
    d. **`(3,+3)`**: A cage made up of rings or a *cage critical point*
12. Get more detailed information on the real space functions printed for further analysis.
    a. `7 // Show real space function values at specific CP or all CPs`
    b. `1 // This is the number of our BCP (we only calculated one)`
13. A lot of useful information will be printed but we are interested in the potential energy density or V(r), which is directly correlated to the energy of the hydrogen bond (EHB) via the following relationship.[12]
    a. $E_{\text{hbond}} = \frac{1}{2} V(r)$ $V(r)$ is in Hartree's in the output
14. However, the Multiwfn authors investigated this relationship and found that it has a much wide error range and should NOT be used. We wil6l use the relationship provided in the Multiwfn manual and described in the following citation.[13] The authors found it to be more accurate and universally applicable. You can see page 401 of the Multiwfn manual for more details on the calculation and its implementation.

a. $BE = -223.08 \cdot \rho(\boldsymbol{r_{BCP}}) + 0.7423$ *Neutral H-bond*

b. $BE = -332.34 \cdot \rho(\boldsymbol{r_{BCP}}) - 1.0661$ *Charged H-bond*

15. Returning to our output, the line, `Density of all electrons`, corresponds to $\rho(\boldsymbol{r})$. Note that the outputs of Multiwfn are recorded in hartrees. The conversion is bult into the equation. Therefore, we get the following equation for your hydrogen bond:

    a. $BE = -223.08 \cdot 0.02131044442 + 0.7423 = -4.01$ $kcal/mol$

16. We see that the H-bond binding energy based on the BCP electron density is significant and worth further investigation.

5.3 Vizualizing non-covalent interactions with Multiwfn

5.3.1 Visualizing electron density for an NCI

1. Next, we will visualize the electron density for a non-covalent interaction. In this case, we will be visualizing the electron density for a three-way (cooperative) hydrogen bond.
2. We will calculate the electron density difference for the three-way Asparagine-SuccinateHydroxyl hydrogen bond.
3. First, run single points with each of the species separate and obtain the molden files.
4. Now we will use Multiwfn to calculate the electron density difference. Load in the complete aggregate wfn for all the interactions together.
    a. `>> Multiwfn complete.wfn`
    b. `>> 5 // Output and plot specific property within a spatial region (calc. grid data)`
    c. `>> 0 // Set custom operation`
    d. `>> 3 // We will load in 3 wfn files for Asn, Succ, and hydroxyl`
5. Next, we will use Multiwfn operators to subtract out the wavefunctions for each of the components separately as a first step to visualize only the electron density of the hydrogen bonding interaction. Load in each wfn file with the subtraction operator.
    a. `>> -,asn.wfn`
    b. `>> -,succ.wfn`
    c. `>> -,hydroxyl.wfn`
    d. `>> 1 // electron density`
    e. `>> 4 // Input the number of points or grid spacing in X,Y,Z covering the whole system`
6. We will be asked for the number of points that we would like to be calculated in each direction. For publication quality, you will want to use a value higher than the “high quality” default setting. Our system is small, so we will use 300 in all directions.
    a. `>> 300,300,300`
7. Change the isovalue to 0.003. This is not necessary as it can be adjusted later in VMD but it can speed up the post-modification process.
    a. `>> 4`
    b. `>> 0.003`
8. You can then load the cube file into VMD along with the xyz structure.
    a. `>> 2`

5.3.2 Reading in an ORCA wavefunction file

9. The default wavefunction file from ORCA has the extension of .gbw
10. This file is useable by Multiwfn, however, not directly. It must first be converted to a molden file, which can be done using a script that comes with ORCA called the `orca_2mkl.`
11. For my system, this would be performed as the following:
    a. `>> export PATH="/home/gridsan/dkastner/src/orca_5_0_3:$PATH"`
    b. `>> export LD_LIBRARY_PATH="/home/gridsan/dkastner/src/orca_5_0_3:$LD_LIBRARY_PATH"`
    c. `/home/gridsan/dkastner/src/orca_5_0_3/orca_2mkl qmscript -molden`
12. The output of this command will be a new molden file.

5.4 Visualizing orbitals

5.4.1 Generate cube files

13. Open Gibraltar and load Multiwfn. Execute the following sequence of commands to generate a cube file of a particular orbital.
    a. >> 5
    b. `>> 4`
    c. `>> orbital number`
    d. `>> 4`
    e. `>> 500,500,500`
    f. `>> 4`
    g. `>> 0.003`
    h. `>> 2`
14. Open in ChimeraX for visualization.

5.5 Using CPPTraj to analyze a trajectory

5.5.1 Using CPPTraj to analyze a trajectory

15. CPPTraj is built as an extension of Amber with the primary purpose of facilitating the analysis of Amber trajectories and generating input files.[14]

5.5.2 Measuring an angle

16. Use this command sequence if you would like to extract the value of a specific angle at each from of a MD simulation.

a. `>> ssh Gibraltar-37`

b. `>> module load amber/18`

c. `>> cpptraj`

d. `>> parm 1os7_solv.prmtop`

e. `>> trajin constP_prod.mdcrd`

f. `>> angle A1 :287@C2 :287@H7 :285@O out angle1.dat`

g. `>> angle A2 :287@H7 :285@O :284 out angle2.dat`

h. `>> run`

5.5.3 Visualizing restart points

17. When your job completes, it is a good idea to visualize the restart points of the minimization and equilibration steps to assure that the simulations aren't blowing up or the active site hasn't moved away significantly from its native position. We can do this using a utility called `cpptraj` that is part of the Amber package.

    a. `>> module load amber/18`

    b. `>> cpptraj`

    c. `>> parm 1os7_solv.prmtop`

    d. `>> trajin constP_prod.mdcrd`

    e. `>> strip :WAT,CL- outprefix stripped`

    f. `>> trajout constP_prod.pdb pdb onlyframes 9625`

    g. `>> run`

18. If you do not want to remove the waters, you can use the following command sequence.

    a. `>> module load amber/18`

    b. `>> cpptraj`

    c. `>> parm 1os7_solv.prmtop`

    d. `>> trajin constP_prod.mdcrd`

    e. `>> strip :NA+`

    f. `>> autoimage`

    g. `>> closestwaters 7500 :1-287 noimage center closestout closestwat.dat outprefix closest`

    h. `>> trajout closest7500_f pdb multi onlyframes 124876`

    i. `>> run`

19. In a single line, this can take the form:

a. `>> cpptraj -p ../6l6x_solv.prmtop -y minimization_5.restart -x min.pdb`

5.5.4 Using CPPTraj to analyze a trajectory

5.5.5 Clustering a trajectory

20. Once you generate a trajectory, it can be difficult to select non-biased snapshots of your simulation. To help address this issue, we will cluster the trajectory into representative sections using CPPTraj and the DBScan clustering method.
21. First, we want to autoimage the trajectory and align the frames. This will make it so we are able to superimpose all of the clusters.
22. The code snippet below shows how we might cluster 1OS7 based primarily on the orientation of the ligand. This will help us identify a representative snapshot of the reactive pose.
23. The values that you can play with are in bold. You can change `epsilon` to control the number clusters that we get. Remember that increasing `epsilon` will give us fewer clusters and decreasing epsilon will give us more clusters. You can also change the `rms`, which allows you do add or remove residues from the selection. The rows are labeled with the atom numbers. The final column of the data frame contains the average RMSF for each residue across all of the residues.

```
parm ../../1os7h_solv.prmtop
trajin ../../constP_prod.mdcrd
strip :NA+,Na+,WAT
autoimage
rms first :1-287 out rmsd.dat
cluster C0 \
        dbscan minpoints 25 epsilon 0.3 sievetoframe \
        rms :284,285,287 \
        sieve 10 random \
        out cnumvtime.dat \
        sil Sil \
        summary summary.dat \
        info info.dat \
        cpopvtime cpopvtime.agr normframe \
        repout rep repfmt pdb \
        singlerepout singlerep.nc singlerepfmt netcdf \
        avgout Avg avgfmt restart
        run
```

5.5.6 Trimming a trajectory

24. The trajectory file associated with a 250ns simulation is often too large to download off Gibraltar. A solution is to use the VMD program CatDCD to break it into smaller pieces. For example, the following command will produce a segment of only 50ns from a simulation of 250ns.
25. First determine how many frames are in your dcd file then extract the number of frames that you need. As you can see, there are 125000 frames in a 250ns simulation.
    a. `>> catdcd -num 1os7_dry.dcd`
    b. `>> catdcd -o crop.dcd -first 1 -last 25000 6edh_50_rest.dcd`

5.6 Generalized Born (GB) trajectory analysis

5.6.1 Generalized Born (GB) trajectory analysis

6 NAVIGATING CLUSTERS

6.1 Navigating Gibraltar

6.1.1 Viewing your scratch directory

1. There are some scenarios where you will need to visit your scratch directory. For example, if you forget to copy a job back to your main folder. To accomplish this, first log into Gibraltar like normal and `cd` to your home directory.
2. You can navigate to your scratch directory using the jobID. For example, your Amber simulations of TauD, will output a file called `taud.o2035308`. In this case, our jobID is 2035308.
3. We also have to determine which GPU the calculations were run on which can be determined from the `taud.o2035308` output file. Once you know where the job ran, we can navigate to the scratch directory using the following series of commands.
    a. `>> ssh gibraltar-37`
    b. `>> cd`
    c. `>> cd ../../scratch/kastner/2092363`
4. You can also check the size of all the files in a given directory. This is useful if you find that your directories are becoming too large.
    a. `>> du -sh *`
    b. `>> qacct -j` (in Gibraltar channel)
    c. `>> rm -r 208*`

6.2 Working with Excede, Comet, and Expanse

6.2.1 Excede

5. At MIT we do not have access to any version of Gaussian past 03. As a work around, we will be using a remote server called Comet which is available through Excede. Log into Xsede. Before you log in for the first time you will have to set up Duo.
    a. `>> ssh kastner@login.xsede.org`

6.2.2 Comet

6. To log into Comet you can use the following command. However, comet has been deprecated and replaced with Expanse so you can likely move to the next section.
    a. `>> gsissh comet`

6.2.3 Expanse

7. To log into Expanse you can use the following command:
    a. `>> gsissh expanse`

6.3 Interfacing with GitHub on Gibraltar

6.3.1 Transferring files to comet

8. Move the all the MCPB generated files into a folder called `gaussian tar zip` the folder using the following command. The command for packing the final gaussian folder is also provided.
    a. `>> tar -zcvf gaussian_in.tar.gz gaussian_in •>> tar -zcvf gaussian_out.tar.gz gaussian_out`
9. Run the following command to transfer the folder to Comet.
    a. `>> scp gaussian_in.tar.gz kastner@expanse.sdsc.xsede.org:/home/kastner/`
10. You can also ssh into comet with a file browser using the path `kastner@comet.sdsc.xsede.org,` which will simplify the process of troubleshooting.
    a. `>> tar -xvzf gaussian_in.tar.gz`
    b. `>> tar -xvzf gaussian_out.tar.gz`
11. To transfer files back to Gibraltar you can use the following command.
    a. `>> tar -pvczf gaussian_out.tar.gz gaussian_out`

6.4 Interfacing with GitHub on Gibraltar

12. Add untracked files.
    a. `>> git add -u . # Adds untracked files`

6.4.1 Push changes

13. You can push new changes with the following:
    - `> git status`
    - `> git commit -m 'message'`
    - `> git push`

6.4.2 How to get an access key

14. You will need to get a fingerprint key which will look something like this:
    a. `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`
    b. `1b:a6:96:f8:bf:4e:7a:45:7a:0f:8c:64:66:59:b2:0a kastner@mit.edu`

6.5 SuperCloud

6.5.1 Generate an SSH RSA key

15. ssh-keygen -t rsa -b 4096

6.6 Archiving Best Practices

6.6.1 Archiving Best Practices

16. If you've made it this far, more likely than not, you've begun to accumulate a number of large files. A good strategy to manage this would be to purchase an external hard drive. You can then transfer your files from Gibraltar to your external drive to archive them, which can free up more space than you might expect.
    a. `>>tar cvf - MyBackups | ssh user@server "cat > /path/to/backup/foo.tgz"`

7 ADVANCED VIZUALIZATION

7.1 Advanced rendering in ChimeraX

7.1.1 Rendering a movie in ChimeraX

17. ChimeraX, created in 2018, is one of the newest additions to the visualization suite.[15, 16] Other common software include VMD,[17] PyMol, and Chimera.[18]

18. Load in a multiframe xyz file or PDB. Set the playback and then record the movie.

    a. `>> open ~/Downloads/optim.xyz coordset true`

    b. `>> size ballScale .6`

    c. `>> transparency #1 100 c`

    d. `>> coordset #1 0,153`

    e. `>> movie record ; coordset #1 1,153 ; wait 200 ; movie encode ~/Desktop/gfn.mp4`

    f. `>> movie record ; coordset #1 1,153 ; label delete #1 ; wait 200 ; movie encode ~/Desktop/gfn.mp4`

    g. `>> movie record supersample 3; coordset #1 1,153 ; wait 153 ; movie encode quality highest ; stop`

```
;color = #9370DB
; radius = 0.074
; dashes = 6
:327@FE :328@O
:327@FE :295@NE2
:327@FE :132@NE2
:327@FE :134@OD
:327@FE :330@O1
:327@FE :330@O2
:328@O :329@H9
```

7.1.2 Tricks for improving ChimeraX renders

19. You can adjust the depth fog with two commands. This is essential when visualizing larger systems.

a. `>> lighting depthCueStart 0.41`

b. `>> lighting depthCueEnd 0.75`

c. `>> lighting qualityOfShadows finer`

20. The pseudo bonds used by ChimeraX will sometimes not accurately find all the coordinating bonds. You can manually add them by writing a pseudo bond file and then loading into your ChimeraX scene.
21. The file should have the extension `.pb` for *pseudobond.*

a. `>> open 6edh.pb`

7.2 Advanced rendering in ChimeraX

22. You can make an abstract protein render using the following commands. It will take a long time to render but produces an interesting effect; however, it will all be the same color.

a. `>>surface #3 resolution 1.4 gridspacing .05`

b. `>>surface #3 gridspacing .1`

23. When clipping, you will likely want the protein surface clipped but not the substrate. You can turn off clipping for a given model with the following command.

a. `>> clip model #1 false`

b. `>> clip model #!1 false`

24. You can select only the sidechains, which can make simplify complicated renderings.

a. `>> select :132,134,157,159,193,295,310 & sidechain`

25. The hex codes for two of my favorite carbon colors.

a. `Dark orange: 1,.59,0`

b. `Cornflower blue: .427,.580,.946`

26. To get a more natural and organic shape for the ribbons use:

a. `>> cartoon style all xsection round`

b. `>> cartoon style all thickness .4`

c. `>> rainbow structures palette RdBu`

7.2.1 Preparing structures for advanced ray tracing in blender

27. There are several ways to prepare a structure for post processing in blender. As we have explored in this section, ChimeraX is a modern molecular visualizer and currently the best bridge to blender. However, using PyMol and VMD is also a good option.
28. First, set up your scene in ChimeraX. Get it as close as you can to what you would like to see in Blender but remember that certain characteristics of your model such as material will not be ported to blender.

a. Spend some time to make sure your composition is what you would like. Once it is in blender the structure cannot be changed.

29. Here are some generally useful commands:

a. `>> clip model #!1 false`

b. `colors = ['#8ecae6', '#219ebc', '#023047', '#ffb703', '#fb8500']`

7.3 Advanced rendering in PyMol

7.3.1 Settings to get started

```
cmd.color("grey50","all") util.cnc("all")
set orthoscopic, 1
set max_threads = 2
set fog, 0.2
set ray_trace_fog, 1
set ray_trace_fog_start, 0.6
set ambient, 0.2
set antialias, 1
set spec_power = 200
set spec_reflect = 1.5
set depth_cue = 1
bond (elem fe), (elem n) within 2.5 of (elem fe) set stick_h_scale, 1
set sphere_scale, 0.3
ray 1280, 960
cmd.color("grey50","all") util.cnc("all")
set orthoscopic, 1
set max_threads = 2
set fog, 0.2
set ray_trace_fog, 1
set ray_trace_fog_start, 0.6 set ambient, 0.2
set antialias, 1
set spec_power = 200
set spec_reflect = 1.5
set depth_cue = 1
set stick_h_scale, 1
set sphere_scale, 0.3
ray 1280, 960
set valence, 0
select (name FE or name fe or name Fe)
```

8 ADVANCED DOCKING AND MODELLING

8.1 AutoDock Vina v4

8.1.1 Installation

1. The program AutoDock Vina is a substantial improvement over the original AutoDock4 software and should be used whenever possible. This tutorial section will focus on the use of AutoDock Vina of binding the ligand CABA to its parent protein ScoE.
2. Create a new conda environment called `autodock` and install both AutoDock Vina and MGTools with the following commands. MGTools is a script base that is often as a part of the AutoDock ecosystem. It will help us in preparing our input files. Run the following commands:
    a. `>> conda create -n vina python=2.7`
    b. `>> conda activate vina`
    c. `>> conda install -c bioconda mgltools`
    d. `>> conda install -c bioconda autodock-vina`

8.1.2 Docking

3. First, we will convert the ligand and receptor PDBs to PDBQT, which is contains charge information for the AutoDock software. We will reference the scripts which have been installed in our conda environment.
4. You will have to install MGLTools. https://ccsb.scripps.edu/mgltools/downloads/
5. One really annoying observation is the prepare_flexreceptor4.py code that comes with MGL is broken! You can get the correct script here:
    a. https://github.com/ccsb-scripps/AutoDock-Vina/tree/develop/example/autodock_scripts
    b. `>> pythonsh /home/kastner/packages/mgltools_x86_64Linux2_1.5.7/MGLToolsPckgs/AutoDockTools/Utilities24/prepare_ligand4.py -l ligand.pdb`
    c. `>> pythonsh /home/kastner/packages/mgltools_x86_64Linux2_1.5.7/MGLToolsPckgs/AutoDockTools/Utilities24/prepare_receptor4.py -r protein.pdb`
    d. `>> pythonsh /home/kastner/packages/mgltools_x86_64Linux2_1.5.7/MGLToolsPckgs/AutoDockTools/Utilities24/prepare_flexreceptor.py -r protein.pdbqt -s ARG83`
6. All the commands can be called from the command line; however, they are easier to change and reference in a config file as shown below. Create a submit script called `conf.txt`.

8.2 CREST Ensemble Search

```
receptor = protein.pdbqt
ligand = ligand.pdbqt
out = all.pdbqt

center_x = 32.906
center_y = 48.047
center_z = 1.264

size_x = 15
size_y = 15
size_z = 15

energy_range = 4
```

7. We can now submit the AutoDock Vina job with the following command:
    a. `>> /home/kastner/packages/vina_1.2.5_linux_x86_64 --config conf.txt`
8. AutoDock will produce a file called `all.pdbqt`. This is a very useful file as it has all the predicted ligand orientations as a single ensemble. Load the `all.pdbqt` file and the original ScoE PDB into ChimeraX to visualize the results.

8.3 CREST Ensemble Search

8.3.1 Overview

9. Conformer-Rotamer Ensemble Sampling Tool (CREST) utilizes the semi-empirical GFN2-xTB method to accurate sample many conformations of a given small molecule.
10. In this tutorial section, we will be using CREST to find alternative conformations of the small molecule CABA that may allow alternative binding modes to its parent enzyme ScoE.

8.3.2 Installation

11. First, we will need to create a new conda environment and install the xTB package.
    a. `>> conda create -n xtb`
    b. `>> conda activate xtb`
    c. `>> conda install xtb`
12. Install the crest package from its git repository wherever you like to store packages. This will give you an executable that will run crest. For easy access, add an alias to your `bashrc`. I have provided an example of what the alias might look like.
    a. https://github.com/grimme-lab/crest/releases

b. `>> tar -xvzf crest.tar.gz`

c. `alias crest='/home/kastner/packages/crest/crest'`

8.3.3 GFN2-xTB Optimization

13. Before running crest, we will need to optimize our initial structure at the GFN2-xTB level of theory. If you do not do this, you will get an error that says `WARNING! Change in topology detected.`
14. An optimization can be easily run from the command line with the following command.
    a. `>> xtb --opt tight ligand.xyz --charge -2 --alpb water --threads 8`
    b. `>> xtb vvv.xyz --opt tight --charge 0 --alpb water`
15. The final structure will be written to the file `xtbopt.xyz`.
16. Another important file is the `xtbopt.log` file. This is a series of `xyz` files so it can be viewed as an animation by changing the file name to `xtbopt_log.xyz` and opening it with Avogadro.

8.3.4 Running CREST

17. To run CREST, you will need to `ssh` into the head node `Gibraltar-53`. If you don't, you will get an error saying `FATAL: kernel too old.`
    a. `>> ssh Gibraltar-53`
18. The CREST program is run via the command line. There are several input values that will need to be changed depending on the type of job that you are trying to run.
19. The following example command will run the CREST conformation search on the ligand xyz file using the GFN2-xTB method in the solvent water.
    a. `>> crest ligand.xyz --gfn2 --chrg -2 --alpb water --threads 8`
    b. `>> crest xtbopt.xyz -gfn2 -chrg -2 -g h2o`
20. The final ensemble of the predicted conformers will be written to the file `crest_conformers.xyz`

8.4 NCIPlots

8.4.1 Running an NCI calculation

1. NCI Plots allow you to qualitatively visualize the intermolecular interactions.
2. Install the newest version of NCI Plots. At the time of this writing, it is version 4.
3. Go to its installation location and create a new job directory.
    a. `>> mkdir /packages/cniplot/job/`
    b. `>> cd /packages/nciplot/job/`

4. NCI Plots has an active site feature. It takes the ligand the protein as two separate files and then will calculate the contact points. The most basic version of this type of job will take the following form.

```
2 protein.xyz ligand.xyz LIGAND 2 4.0
ONAME crystal
ULTRAFINE
```

   a. The input is traditionally given the extension `.nci` and the output is denotated with `.nco`

   b. You can run an NCI job from within the base nci directory after establishing two environmental variables.

   c. `>> export NCIPLOT_HOME=/home/kastner/packages/nciplot/`

   d. `>> alias nci='/home/kastner/packages/nciplot/src_nciplot_4.0/nciplot'`

   e. `>> nci crystal.nci > crystal.nco`

5. Once you have executed the above command, nci plots will run for around 5 minutes on a small cluster model and longer on a full protein.

8.4.2 Plotting the results using GNU plot

6. This script can be found on page 28 of the manual.

   a. `>> /Users/kastner/Documents/Packages/gnuplot-5.4.2/src/gnuplot plot.gnu`

8.4.3 Visualizing the results

7. Download the job directory from Gibraltar and open a terminal from within the directory. We will want to execute vmd from within that directory so we can call the VMD state that is generated automatically by the NCI program. If you are using mac, the command would be some variation of the following.

   a. `>> /Applications/VMD\ 1.9.4a51-x86_64-Rev9.app/Contents/MacOS/startup.command -e crystal.vmd`

8. You can also visualize the results graphically using gnuplot.

   a. `>> /Users/kastner/Documents/Packages/gnuplot-5.4.2/src/gnuplot plot.gnu`

   b. `>> draw line {29.511000 44.723999 0.284000 } {30.888000 46.353001 1.296000 } style dashed radius .5`

8.5 Trajectory analysis with Bio3D

8.5.1 Setting up the Bio3D environment

9. When a residue moves in a trajectory, it is not an isolated event. The movements of residues in the protein are often correlated with one another. In other words, when one part of the protein moves another nearby region might move with it. We can visualize these internal correlations using DCCA.
10. The best way to perform this calculation is using a package called Bio3D which uses R.
11. The following series of commands will generate the dataset. Install Bio3D on Gibraltar via a conda environment:
    a. `>> conda create -n bio3d`
    b. `>> conda activate bio3d`
    c. `>> conda install -c conda-forge r`
    d. `>> conda install -c conda-forge r-bio3d`
    e. `>> conda install -c conda-forge r-n`
12. Initialize an R session and perform the following commands
    a. `>> R`
13. These commands can also be found as a script here:
    a. https://github.com/davidkastner/pyQMMM/blob/main/pyqmmm/bio3d_dcca.R
14. Load the Bio3D library.
    a. `>> library(bio3d)`

8.5.2 Reading in and preparing initial files

15. Read in your protein and DCD files.
    a. `>> pdb <- read.pdb("6edh_solv.pdb")`
    b. `>> dcd <- read.dcd("constP_prod.pdb")`
16. Align each from based on the protein's backbone.
    a. `>> ca.inds <- atom.select(pdb, elety="CA")`
17. If you would like to include the entire backbone than use this command instead. Use this when performing PCA.
    a. `>> ca.inds <- atom.select(pdb, elety=c("CA","N","C"))`
    b.
```
>> xyz <- fit.xyz(fixed=pdb$xyz, mobile=dcd,
fixed.inds=ca.inds$xyz, mobile.inds=ca.inds$xyz)
```

8.5.3 Calculate the RMSD

18. Calculate the RMSD and write the data out to a new file.
    a. `>> rmsd <- rmsd(xyz[1,ca.inds$xyz], xyz[,ca.inds$xyz])`
    b. `>> write.csv(rmsd, "rmsd.csv")`

8.5.4 Perform DCCA analysis

19. Perform DCCA and write the data out so we can format it how we want. We will have Bio3D write out a python script that we can open in PyMol on our own machine for rendering.
    a. `>> cij<-dccm(xyz[,ca.inds$xyz])`
    b. `>> write.csv(cij, "dcca.csv")`
    c. `>> pymol.dccm(cij, pdb, type="launch")`
20. If you are working with proteins that contain nonstandard amino acids, as you likely are, and try this procedure you will get a dimension mismatch error, which means the final PyMol command can't parse the non-standard amino acids.
21. To fix this, go through the PDB and remove the ligands and change the MCPB names back to their default names

8.5.5 Analyzing the principal components of a trajectory

22. We can better understand the multidimensional landscape of the trajectory by using principal component analysis.
    a. `>> pc <- pca.xyz(xyz[,ca.inds$xyz])`
    b. `>> write.csv(pc, "pc.csv")`
    c. `>> p1 <- mktrj.pca(pc, pc=1, b=pc$au[,1], file="pc1.pdb")`
    d. `>> p1 <- mktrj.pca(pc, pc=1, pdb=bb.pdb, file="pc1_bb.pdb")`
    e. `>> p2 <- mktrj.pca(pc, pc=2, b=pc$au[,2], file="pc2.pdb")`
23. The last command will produce PDBs that can be opened in VMD; however, you can also generate them as netCDF trajectory formats although you cannot currently open these in VMD as of the newest version.
    a. `>> write.ncdf(p1, "trj_pc1.nc")` ○ `>> write.ncdf(p2, "trj_pc2.nc"`
24. You can also include all backbone atoms and not just the alpha carbons.
    a. `>> pdb_orig <- read.pdb("1fdo_prot.pdb")`
    b. `>> dcd <- read.dcd("1fdo_prod.dcd")`
    c. `>> pdb <- trim.pdb(pdb_orig, "back")`
    d. `>> ca.inds <- atom.select(pdb, elety=c("CA","N","C"))`

e. `>> xyz <- fit.xyz(fixed=pdb$xyz, mobile=dcd, fixed.inds=ca.inds$xyz, mobile.inds=ca.inds$xyz)`

f. `>> pc <- pca.xyz(xyz[,ca.inds$xyz])`

g. `>> write.table(pc$z[, 1:2], file='pc_12.txt')`

h. `>> write.table(pc$z[, 2:3], file='pc_23.txt')`

i. `>> p1 <- mktrj.pca(pc, step=0.05, pc=1, b=pc$au[,1], pdb=pdb, file="pc1.pdb")`

25. You can also calculate the RMSF by residues.

a. `>> rf <- rmsf(xyz[,ca.inds$xyz])`

b. `>> write.csv(rf, "rmsf.csv")`

9 REFERENCES

1. Meyder, A.; Nittinger, E.; Lange, G.; Klein, R.; Rarey, M., Estimating Electron Density Support for Individual Atoms and Molecular Fragments in X-ray Structures. *Journal of Chemical Information and Modeling* **2017,** *57* (10), 2437-2447.
2. Fahrrolfes, R.; Bietz, S.; Flachsenberg, F.; Meyder, A.; Nittinger, E.; Otto, T.; Volkamer, A.; Rarey, M., ProteinsPlus: a web portal for structure analysis of macromolecules. *Nucleic Acids Research* **2017,** *45* (W1), W337-W343.
3. Webb, B.; Sali, A., Protein structure modeling with MODELLER. *Protein Structure Prediction, 3rd Edition* **2014,** *1137*, 1-15.
4. Fiser, A.; Do, R. K. G.; Sali, A., Modeling of loops in protein structures. *Protein Science* **2000,** *9* (9), 1753-1773.
5. Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R., Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of Cheminformatics* **2012,** *4*.
6. Anandakrishnan, R.; Aguilar, B.; Onufriev, A. V., H++3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations. *Nucleic Acids Research* **2012,** *40* (W1), W537-W541.
7. Li, P. F.; Merz, K. M., MCPB.py: A python based metal center parameter builder. *Journal of Chemical Information and Modeling* **2016,** *56* (4), 599-604.
8. Seritan, S.; Bannwarth, C.; Fales, B. S.; Hohenstein, E. G.; Isborn, C. M.; Kokkila-Schumacher, S. I. L.; Li, X.; Liu, F.; Luehr, N.; Snyder, J. W.; Song, C. C.; Titov, A. V.; Ufimtsev, I. S.; Wang, L. P.; Martinez, T. J., TeraChem: A graphical processing unit-acceleratedelectronic structure package forlarge-scaleab initio molecular dynamics. *Wiley Interdisciplinary Reviews-Computational Molecular Science* **2021,** *11* (2).
9. Casey, T. M.; Grzyska, P. K.; Hausinger, R. P.; McCracken, J., Measuring the Orientation of Taurine in the Active Site of the NonHeme Fe(II)/alpha-Ketoglutarate-Dependent Taurine Hydroxylase (TauD) Using Electron Spin Echo Envelope Modulation (ESEEM) Spectroscopy. *Journal of Physical Chemistry B* **2013,** *117* (36), 10384-10394.
10. Kulik, H. J.; Zhang, J.; Klinman, J. P.; Martínez, T. J., How large should the QM region be in QM/MM calculations? The case of catechol O-methyltransferase. *The Journal of Physical Chemistry B* **2016,** *120* (44), 11381-11394.
11. Karelina, M.; Kulik, H. J., Systematic Quantum Mechanical Region Determination in QM/MM Simulation. *Journal of Chemical Theory and Computation* **2017,** *13* (2), 563-576.
12. Espinosa, E.; Molins, E.; Lecomte, C., Hydrogen bond strengths revealed by topological analyses of experimentally observed electron densities. *Chemical Physics Letters* **1998,** *285* (3), 170-173.
13. Emamian, S.; Lu, T.; Kruse, H.; Emamian, H., Exploring nature and predicting strength of hydrogen bonds: A correlation analysis between atoms-in-molecules descriptors, binding energies, and energy components of symmetry-adapted perturbation theory. *Journal of Computational Chemistry* **2019,** *40* (32), 2868-2881.

14. Roe, D. R.; Cheatham, T. E., PTRAJ and CPPTRAJ: Software for processing and analysis of Molecular dynamics trajectory data. *Journal of Chemical Theory and Computation* **2013,** *9* (7), 3084-3095.
15. Goddard, T. D.; Huang, C. C.; Meng, E. C.; Pettersen, E. F.; Couch, G. S.; Morris, J. H.; Ferrin, T. E., UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein Science* **2018,** *27* (1), 14-25.
16. Pettersen, E. F.; Goddard, T. D.; Huang, C. R. C.; Meng, E. E. C.; Couch, G. S.; Croll, T. I.; Morris, J. H.; Ferrin, T. E., UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Science* **2021,** *30* (1), 70-82.
17. Humphrey, W.; Dalke, A.; Schulten, K., VMD: Visual molecular dynamics. *Journal of Molecular Graphics & Modelling* **1996,** *14* (1), 33-38.
18. Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E., UCSF chimera - A visualization system for exploratory research and analysis. *Journal of Computational Chemistry* **2004,** *25* (13), 1605-1612.