Chemistry 430 — Simulation in Chemistry & Biochemistry

Laboratory #7 — Folding Simulation of the TrpCage MiniProtein

In this lab we will use molecular dynamics simulations to attempt to fold the 20-residue TrpCage miniprotein. The TrpCage amino acid sequence (NLYIQWLKDGGPSSGRPPPS) was designed as a stable, modified subsequence of the naturally occurring 39-residue protein exendin-4. The original biophysical characterization of TrpCage by the Andersen group at the University of Washington in Seattle is described in a research paper and commentary available on the lab web site (both are in *Nature Structural Biology*, **"nsb"**). This miniprotein exhibits 2-state folding kinetics, characterized by the presence at any given time of only "folded" and "unfolded" states of the protein. The ΔG of unfolding is +8.6 kJ/mol at 3°C, and the folding half-time is thought to be on the microsecond time scale. Due to its small size and simple folding behavior, TrpCage has been the subject of several computational studies. Three reports describing molecular dynamics simulations of TrpCage folding are provided on the lab web site (two are from *Journal of the American Society*, **"jacs…"** and one from the journal *Proteins*, **"proteins…"**). Several other, more recent computational efforts can be found from a literature search via Web of Knowledge.

We will perform a series of computations over the next two labs. During a first lab session, you will start molecular dynamics simulations of both the native (folded) form of TrpCage and an extended (unfolded) chain with the TrpCage sequence. Simulations will run on graphical processing units (GPUs) in the Ponder lab research cluster. These GPUs are much faster for running MD simulations than the CPUs in the lab iMac workstations. Then during a second lab period, you will analyze the results of these simulations.

Note: It is possible to setup the simulations for this lab remotely, via ssh connection to one of the Chem 430 lab machines. Use **"ssh xxxxx.wustl.edu"** from a Mac Terminal, where **xxxxx** is one of the lab machines: **holly, icicle, eggnog, merry, ivy** or **noel**. To login from a remote Windows computer, use the above **ssh** command in a Windows PowerShell terminal window to connect to lab machines.

Protocol

(1) Obtain the **trpcage.pdb** and **tinker.key** files from the lab web site, and place them in a new directory where you will perform the lab. If you are present in the lab to perform this step, just click on the files in your browser, save them to the lab computer, and then move them into the desired directory.

If you are on a remote computer, first use **ssh** to connect to one of the lab machines (see note above). Then create a directory for this lab under your home area, move to that directory, and copy the two required files. For example, you can copy the files by issuing the following two commands in a **ssh** terminal window:

```
cp /user/www-dasher/chem430/labs/lab-06/trpcage.pdb .
cp /user/www-dasher/chem430/labs/lab-06/tinker.key .
```

The **trpcage.pdb** file contains coordinates for the TrpCage miniprotein as determined by NMR spectroscopy, and in Protein Data Bank (PDB) format. These coordinates are the same ones you would get from the official PDB web site located at **https://www.rcsb.org/**. Take a look at the **.pdb** file in Chimera, VMD or FFE so that you get some idea what the structure of this tiny protein looks like. If you have free time at some point, you might want to explore the PDB web site, as it is a treasure trove of information about structural biology. The PDB 4-letter code for the TrpCage miniprotein is **1L2Y**.

The **tinker.key** file will be used by default by all Tinker programs for all calculations on structures in its directory. (You can override the use of the generic **tinker.key** file by creating a **.key** file with the same base name as the **.xyz** coordinates file.) Examine the **tinker.key** file, and note the use of the Amber *ff99sb* force field, use of the **RATTLE** keywords to constrain bond lengths to hydrogens and invoke rigid water molecules, and use of Ewald summation to handle long-range electrostatic interactions.

(2) First convert the PDB file to a Tinker .**xyz** file by running the **pdbxyz** program in a terminal window using the command "**pdbxyz trpcage.pdb**". This will create **trpcage.xyz** containing the coordinates, and **trpcage.seq** with the amino acid sequence. This **.xyz** file will have the same native protein conformation as the PDB file, and can be viewed in FFE, VMD, Chimera, *etc.* Move the XYZ coordinates file to another name. I might suggest using the command "**mv trpcage.xyz folded.xyz**".

(3) Copy the **tinker.key** file to **folded.key**, and add the keyword **CUDA-DEVICE 0** which will select the first GPU card physically located in a particular computer to be used for MD simulation starting from the folded structure.

(4) Build an extended, fully unfolded, structure for the TrpCage sequence using the Tinker **protein** program. Use some unique name as the base file name for these structures. I suggest using **unfolded**. The program will ask for the amino acid sequence which should be entered as the three-letter code. The TrpCage sequence is contained in the **.seq** file created in the previous step. After running **protein** you should have a Tinker XYZ file for the extended protein chain. Check this by looking at the structure in a molecular viewing program such as FFE or VMD.

(5) Obtain the **water.xyz** file from the lab web site. This contains an 70 Angstom cubic box with over 11000 water molecules. We will separately "soak" the folded and unfolded TrpCage structures in this water box. Run the **xyzedit** program on each of the protein structures in turn. First use option 13 to move the center of mass of the protein structure to the origin of the coordinate system (*i.e.,* coordinates of 0,0,0), then without exiting the xyzedit program, invoke option 24 to embed the protein in a solvent box. Use **water.xyz** as your solvent box. After running the program, a new version the Tinker XYZ file will be created with the solvated protein conformation.

(6) Copy the **tinker.key** file to **unfolded.key**, and add the keyword **CUDA-DEVICE 1** which will select the second GPU card physically located in a particular computer to be used for MD simulation starting from the folded structure.

(7) Use the **minimize** program to clean up each of your solvated structures. Choose an RMS gradient at convergence of 1.0 kcal/mol/Ang, instead of the program default of 0.01.

(8) To run the simulations, we will use the **dynamic9** program found in your Tinker distribution. This is a special version of the Tinker **dynamic** program intended to run molecular dynamics on GPU cards. The program will only run on the "elf" nodes in the Ponder lab's research cluster, as described below.

(9) Now login to the head node for the Ponder compute cluster via "**ssh elf**" followed by your password. Once logged onto elf, login a second time to the specific elf cluster node you are assigned via the command "**ssh elfXX**" where "**XX**" is the two-digit number of your node. Assigned elf nodes and temperatures are in the file **assigned-simulations.txt** on the lab website. Your initial password on the numbered elf nodes is **ChEm430**, which can be changed to your usual password for the class by using the "**passwd**" command.

You will run half-microsecond MD simulations as background jobs on your assigned cluster nodes. For simulations starting from the unfolded state, the command will be as follows:

```
dynamic9 unfolded 250000000 2.0 10.0 4 TEMP 1.0 >& unfolded.log &
```

Replace "**TEMP**" in the above command with the assigned temperature for your simulation, for example "**300.0**". Note that we are running the simulation in the NPT ensemble via a thermostat and barostat to maintain the desired temperature and pressure. Start a second calculation to simulate the folded protein on your elf node and at your assigned temperature using the same command with **unfolded** replaced by **folded**. Make sure you understand the meaning of each number and symbol in the commands used to submit the simulations.

(10) These MD simulations will run for several days. You can check to make sure your calculations are running at **https://dasher.wustl.edu/cluster/**. Then during a later lab session, continue with analysis of results starting with step 9 below.

(11) Once the simulations finish, the archive files will be processed for you. In the lab directory, you will find files named **folded-XXX.arc, folded-XXX.pdb, folded-xxx.log** and **folded-XXX.out**, where **XXX** is the temperature value in Kelvin. These files contain information about the trajectories starting from the experimental folded TrpCage structure. There are corresponding files beginning with **unfolded-XXX** are for the simulations started from the extended, unfolded structure at each temperature. If time permits, you will want to examine all these files, and not just the ones for the temperature you were assigned.

(12) The processed **.arc** files are large at about 1 GB each, but roughly 100 times smaller than the original, raw trajectory files. They are Tinker **.arc** files from the MD simulations, and contain only the TrpCage protein (the water has been removed) from each saved trajectory frame of the simulation. Each frame has been superimposed on the NMR-derived experimental structure, provided in the file **trpcage.xyz**. The **.out** files contain the value of the RMS superposition of the α-carbon atoms between each of the 50000 frames of the **.arc** file and the experimental structure. Note that trajectory frames were saved to the **.arc** files every 10 ps, so there are 50000 frames in each archive file, corresponding to a total simulation length of 0.5 μs. The **.log** files are the standard output from the MD simulations, and contain the potential energy, box size, and other instantaneous values for the coordinates saved at each 10 ps frame. Finally, the **.pdb** files contain the same coordinates as the **.arc** files, but in PDB format. These files can be viewed in the VMD program to allow watching a "movie" of the full MD trajectory.

(13) If you are using the computer lab machines, then the **gnuplot** program can a plot of the RMS α-carbon superposition (y-axis) as a function of MD time simulated (x-axis). After invoking the program at the command line, you should use commands similar to the below (where **XXX** is replaced with a temperature value):

```
gnuplot> set style data lines
gnuplot> plot 'folded-XXX.out' using 2:1
```

If you wish, you can try to install **gnuplot** on your Mac, Windows or Linux computer using information readily available on the internet. Alternatively, this type of X,Y-plot can be generated using Excel if you prefer.

(14) Use the **FFE** or **VMD** program to view various of the trajectories as a "movie". Does the folded trajectory started from the PDB structure remain folded? Can you tell by visual inspection whether these simulations ever sample structures similar to the PDB fold? Note the trajectories are not smooth movies since we only saved coordinates every 10 ps, and that is a big enough time interval that significant motion occurs between frames.

(15) The Tinker **superpose** program was used to perform root mean square (RMS) fits contained in the **.out** files via commands like the one given below:

```
superpose trpcage.xyz folded-XXX.arc < trpcage.super | grep Root
```

Try running the program manually, using the "answers" contained in the **trpcage.super** file as input after the file names are entered. Do you understand what the input in the **trpcage.super** file is doing?

(16) Open the file corresponding to the experimentally determined PDB structure **(trpcage.xyz)** in **FFE**. Undisplay the hydrogen atoms, change the representation from "wireframe" to "tube", and color it red (using the Set User Color, and Apply User Color options in **FFE**). Now open and view a superimposed trajectory in a **.arc** file and play it as

a movie on top of the experimental structure. Which parts of the structure are most conserved across the full trajectory?

As mentioned above, original PDB file for TrpCage, **`trpcage.pdb`**, can be opened in **`VMD`**, and then the multi-frame PDB files with the MD trajectory for each temperature can be viewed as a movie on top of the experimental structure. Try other "Drawing Method" options, such as "New Cartoon" for the structures displayed in **`VMD`**.

(17) Repeat steps 10-12 for the trajectories started from the unfolded state. Do the MD trajectories that were started from the unfolded, extended strand fold into more compact structures? In order to share data between class members, the **`folded`** and **`unfolded`** files for all temperatures have been placed on the lab web site. Inspect the **`.out`** files in order to determine which simulation(s) sampled structures closest to the PDB structure.

(18) Of all the unfolded simulations run by the class, the single trajectory frame that is the closest to the experimental NMR-determined structure has an a-carbon RMSD of 0.48 Å. This structure is provided as the files **`closest.xyz`** and **`closest.pdb`**. Use your favorite viewing program to display this closest structure along with the experimental structure. How similar is the overall fold between the two structures? What are the differences?

(19) Inspect the **`.out`** files to find the structure snapshot from the unfolded trajectory at your assigned temperature that has a small RMS against the experimental structure. Extract this structure from its archive using the Tinker **`archive`** program. Display this trajectory frame on top of the experimental structure using **`FFE`** and save a screenshot of the overlapped structures. How similar is the overall fold between the two structures? What are the differences?

Questions

(1) Calculate the protein concentration in the periodic simulation box. How does this concentration compare to the protein concentrations used in typical experiments in a molecular biology wet lab setting?

(2) What kinds of structural changes occur during the MD simulations started from the native, experimental structure? Do these folded simulations ever sample completely unfolded structures?

(3) For which temperatures did the trajectory starting from the unfolded form generate "correctly folded" protein? Structures below 1 Å RMS are quite close to the experimental structure, those below 2 Å RMS are partially folded, below 3 Å RMS have a somewhat similar fold, while those below about 4 Å may still have at least some features in common with the experimental fold.

(4) Generate separate plots of the RMS fit *vs*. MD time for both the folded and unfolded trajectories. Include several different temperatures in each of your plots. Do trajectories

starting from the folded structure sample unfolded states of the protein (say, structures greater than 5-6 Å RMS from the PDB structure)? How many of the trajectories starting from the unfolded chain sample the experimental fold at least once?

(5) Do you observe folding transitions in the trajectory started from the unfolded form? For example, how many times does the structure move from largely "folded" (below about 2 Å RMS) to "unfolded" (above 5-6 Å RMS), or the reverse?

(6) Read the literature papers provided on the lab web site. What is the experimental folding time for the TrpCage protein? What does this say about your ability to see multiple folding transitions during your trajectory? Why do you think the higher temperature simulations run by the class sample structures closest to the PDB structure?

(7) What is the average potential energy across your native simulation (check the **`.log`** files for your particular simulations)? What is the standard deviation in the potential energy for this simulation? Why is the variance not "zero", or at least zero within numerical precision? Does this indicate lack of conservation of energy?

(8) From values in the **`.log`** files, what is the average potential energy different for the early part of the folding simulation, where the structure is extended? How about for the later parts of the folding simulation where the structures are compact? Do you expect the average total energy to decrease as the trajectory reaches compact structures and approaches the experimental fold? Explain.