Lab IV: Intermediate Building

In this lab you will:

1) Add missing atoms to protein structures using PSFGEN and NAMD
2) Add missing atoms to protein structures using CHARMM.
3) Set up a simulated annealing run of a protein with a disordered loop.
4) Use steered molecular dynamics to move a buried N-terminus.

Working directory: /home/mdschool/labs/tuesday/lab_4/

NOTE: in this lab, the ">" symbol means that you are in an active terminal window and what follows this symbol is what you type. Make sure you know what directory you are in (> pwd) and what files are in your current directory (> ls). The symbol "vmd>" means you type what follows inside the VMD console (not a Linux terminal).

1. Correct a PDB structure by adding in missing atoms (PSFGEN / NAMD).

In this problem you will create new PDB and PSF files from a starting structure that is missing several amino acids. You will create a new PDB file by adding in missing amino acid names which will be used in your psfgen build script to generate a complete starting structure (new pdb and psf files). You will then create a fixed atom file to restrain the known structural components while allowing for the newly added atoms to move. With these files you will minimize the structures by running NAMD.

Note that in this problem we are no longer giving explicit commands to run psfgen, or NAMD, etc. If you have trouble, refer to previous labs (II & III) or ASK a question!

Directories:
missing_bits_namd/

Files:
missing_bits_namd/
known_coordinates.pdb
full_length_sequence.txt
extract_sequence.py

Procedure:

a) Migrate to the missing_bits_namd/ and create two new directories: output_building and output.
   a. > **mkdir output_building**
   b. > **mkdir output**

b) Inspect the known_coordinates.pdb file and by comparison to the full_length_sequence.txt file write down the missing amino acids (resid and resname).
    a. If needed, a python script is provided (extract_sequence.py). It is set up to read the known_coordinates.pdb file and write a one-letter listing to a file called "oneletter_sequence.txt". To have the program read other pdb files, you merely have to change the "infile" command on the second non-empty line of the program to point at the file you are interested in. You may want to change the name of the output file as well. To run the script you merely type on the following on the command line.
        i. > python extract_sequence.py

    then compare the known sequence in the file "oneletter_sequence.txt" to the complete sequence in the full_length_sequence.txt file. Note that the numbering in the "oneletter_sequence.txt" does not correspond to the correct amino acid number in the full_length_sequence! See Figure 1 below for additional information.

Figure 1. *Missing residues are highlighted in green.*

c) Using the following command, create a new PDB file that you are going to edit to create a PDB file with the complete amino acid sequence.
    a. **> cat known_coordinates.pdb | grep CA > sequence.pdb**
    b. NOTE the "cat" program merely prints out the contents of the file to the terminal, the "|" command (called "pipe") takes the output the would normally go to the terminal and sends it as input to the commands on the right side of it. In this case, the output of the cat command is evaluated by a program called "grep". The "grep" program then "grabs" only those lines in the file that have the characters "CA" and then we redirect that output using ">" to a file called sequence.pdb.

    c. NOTE2: In other words, the command above makes a new PDB file with just the CA atoms . . . back to work.

d) Edit your newly created sequence.pdb to add in the missing CA atoms that you wrote down in (1b) above. Remember that the coordinates do not matter in this file. See the Figure 2 below, which is taken from the lecture, for an example of adding amino acids to the system.
e) Write a psfgen input file to read your sequence.pdb file and known_coordinates.pdb files and create a new pdb/psf pair. No patches, other than the ubiquitous NTER and CTER in the segment definition are needed. Name your new psf/pdb files: complete_protein.psf and complete_protein.pdb. Run your script to create the new files in the output_building directory.
f) Load the new psf/pdb pair into VMD using the VMD_1.8.6 button on the desktop. With the resid values determined above, type the TCL commands in the VMD console to create a fixed atom PDB file that restrains the known_coordinates. In other words, all amino acids with resid corresponding to new coordinates should have BETA values set to zero (0.00) otherwise they should be set to one (1.00). A sample set of commands is shown in the figure below. Name the fixed atom PDB file fixed_atoms.pdb.
g) Now copy the new pdb/psf and fixed atom pdb file from the output_building/ directory to the output/ directory.
h) Write a NAMD minimization file using the min0 file used in Lab 2 as a template. You will have to copy the file to your current directory. Run the minimization and inspect the output files and final structure. Your final minimized structure should be called min0.coor.
i) Start a second NAMD minimization run where you continue the first (1i) but you first remove the fixed atoms lines in the input file. Copy the min0 input file you generated in (1i) to a file called min1. Edit this file to read the min0.coor file for the initial coordinates and make sure that the output files are named min1. Your final minimized structure should be called min1.coor.
j) Close VMD.

```
> cat coords.pdb | grep CA > ca_sequence.pdb

then, copy known coordinate line (say, ATOM 163 CA VAL ...) and manually
change the index, resname, and resid fields.  Do this for all missing atoms/AA.

ATOM      1   CA THR     1      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      2   CA SER     2      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      3   CA GLY     3      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      4   CA ILE     4      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      5   CA HSE     5      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      6   CA VAL     6      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      7   CA LEU     7      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      8   CA ASP     8      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM      9   CA GLU     9      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM     10   CA LEU    10      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM     11   CA SER    11      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM    163   CA VAL    12      35.989 -22.055 248.035  1.00  0.00      TRAI C
ATOM    179   CA ARG    13      39.541 -23.591 248.463  1.00  0.00      TRAI C
ATOM    203   CA ALA    14      41.070 -23.037 251.954  1.00  0.00      TRAI C
ATOM    213   CA LEU    15      42.765 -25.002 254.835  1.00  0.00      TRAI C
```

Figure 2. *Example of adding missing residues to the sequence file.*

```
vmd > set everyone [atomselect top all]
atomselect32

vmd > $everyone set beta 1.00

vmd > set sel1 [atomselect top "(resid > 341 and resid < 406) or (resid <
12)"]
atomselect33

vmd > $sel1 set beta 0.00

vmd > $everyone writepdb fixed_atoms.pdb

Info) Opened coordinate file fixed_atoms.pdb for writing.
Info) Finished with coordinate file fixed_atoms.pdb.
```

Figure 3. *VMD TCL commands to make the fixed atoms PDB file.*

2. Correct a PDB structure by adding in missing atoms (CHARMM).

In this problem you will be given new PDB and PSF files from a starting structure that is missing several amino acids along with the CHARMM input file taken directly from the lecture. Your task is to run the job and inspect the results.

Directories:

missing_bits_charmm/

Files:

missing_bits_charmm/

coords_traI_10_gap_481.pdb
ca_sequence.pdb
nbuild_structure.inp

Procedure:

a) Migrate to the missing_bits_charmm/ directory.
b) Inspect the nbuild_structure.inp file. See if you can figure out what it is doing, what files will be generated, etc.
c) Run CHARMM using the supplied input file.
    a. > **charmm < nbuild_structure.inp >& cbuild.out &**
d) Monitor the progress of the job by typing
    a. > **tail -90f cbuild.out**
e) When the job is done, hit "control c" to stop the tail process.
f) YOU can start the next problem while this is running.
g) Inspect the final coordinates by loading the temp_3_traI_piece.psf file and the final coordinates contained in the dyn0_temp_3_traI_piece.pdb file into VMD_1.8.6. You can also load the binary trajectory file as well in the same molecule (i.e. load the trajectory.trj file as a DCD file!).

3. **Set up and start a simulated annealing (SA) run using the fixed and minimized structure that you generated using PSFGEN/NAMD in problem (1) above.**

You will copy files from problem 1 above and then edit the NAMD input file to include the commands to carry out the SA run. Then you will start the run and after you confirm that it is running correctly, you will stop the job. It will take too long for you to do this run during the lab (or this week!).

Then you will inspect the results of a pre-cooked SA run of *another system*. To do this you will concatenate files using the "catdcd" program and visualize the trajectory using VMD.

Directories:

simulated_annealing/
output/
sample_data/

Files:

Simulated_annealing/sample_data/
min_*.coor
blah.psf
(a series of 87 minimized structures and their associated psf file)

Procedure:

a) Migrate to the simulating_annealing/ directory.
b) Copy your minimized structure and associated psf and fixed_atoms.pdb files from 1(j) above to the simulated_annealing/output/ directory.
   a. **cp ../missing_bits_namd/output/min1.coor output/**
   b. **cp ../missing_bits_namd/output/complete_protein.psf output/**
   c. **cp ../missing_bits_namd/output/fixed_atoms.pdb output/**
c) Copy the last minimization input script from 1(j) above to the local directory. You will use this to write your SA input script.
   a. > **cp ../missing_bits_namd/min1 sa_min0**
   b. NOTE: you are renaming the file called min1 to sa_min0
d) Edit this file to include TCL commands to perform a short SA run. You must remove the final minimize command from your starting input file and add the information in the figure below to the bottom of your script. Write down the name of the file(s) that are going to be written at the end of each SA cycle.
e) Add the appropriate lines to the file to use the BETA column of the output/fixed_atoms.pdb file.
f) Make sure to change the starting coordinates to use the min1.coor you copied in step 1(b) above and change the generic output name to sa_min0.
g) Start the SA run without saving the output to a file. In other words, you should start the run, confirm that it is running, then hit "control c" to cancel

the job. You can watch the run before you stop it by using IMD if you are interested. Given the time constraints in the lab, we will inspect the results of a SA run performed previously in the steps below.

h) *Open VMD_1.8.6 by clicking the icon on the desktop*. Load the blah.psf file. Located in the simulated_annealing/sample_data/ directory.

i) At the Linux terminal, type the following command to concatenate all of the output coordinate files into a single DCD file. This command assumes that you are in the sample_data/ directory.
   a. **> catdcd –o all_sa_min.dcd –otype dcd –stype psf –s blah.psf – namdbin min_*.coor**

j) Load the binary coordinates into the molecule that you loaded in (f). Inspect the trajectory via VMD.

k) Close VMD.

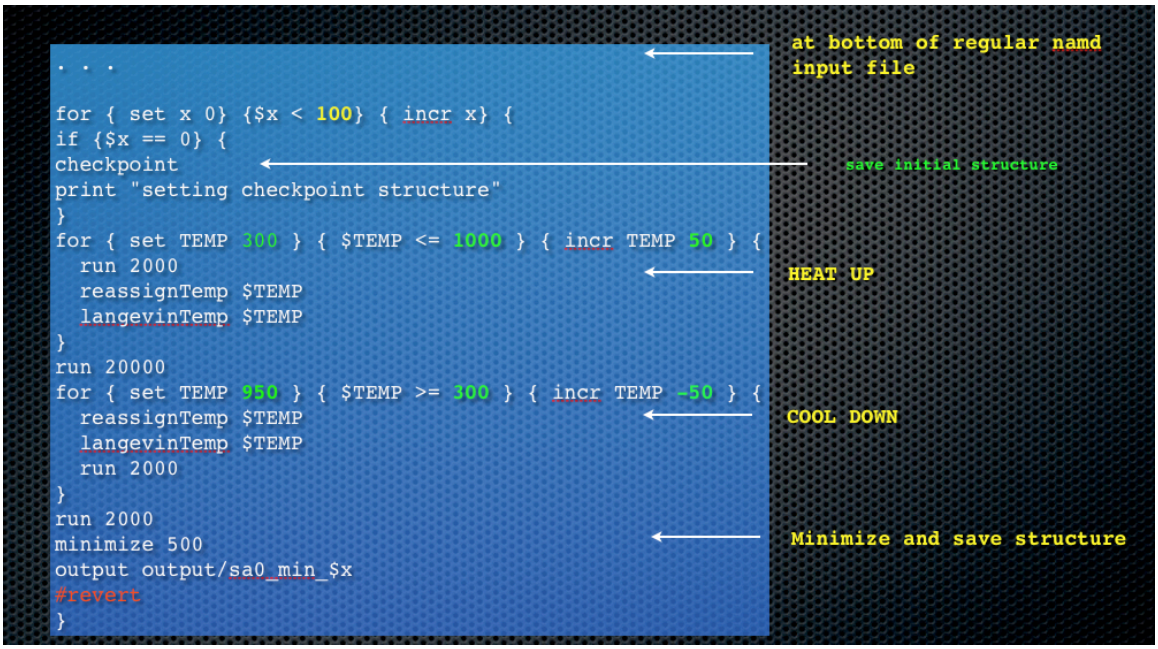

Figure 4. *Example TCL commands to be added to the bottom of a NAMD input file.*

4. Steered molecular dynamics.

In this problem you will be given the psf and pdb files along with a NAMD input file to experiment with the steered molecular dynamics run shown in the lecture. You will load the psf/pdb file, inspect the NAMD input file, run NAMD, connect to the simulation via VMD (using IMD connect). Then you are going to applied residue level forces to the amino terminus.

Directories:

steered_md/
    output/

Files:

steered_md/
    smd_input

steered_md/output/
    new_blah.psf
    fixed_atoms.pdb
    new_blah.pdb

a) Migrate to the steered_md/ directory.
b) Inspect the smd_input file. Write down the name of the psf and fixed atom pdb files. Write down the "port" number for the IMD connection.
c) Start VMD. Load the psf and *fixed atom* pdb files for the system. Color the system using BETA.
d) Start the NAMD run
    a. namd2 smd_input
e) The output screen should indicate that it is waiting for an IMD connection.
f) In VMD, use the main menu to open up the IMD GUI (Extensions → Simulation → IMD Connect (NAMD)). Enter the IP as 127.0.01 and the port number that you wrote down in section (b).
g) After the minimization routine is done (step > 100), you should notice that the output format changes and perhaps you may notice that the molecule is "shaking".
h) Using the VMD main menu, apply forces to residues using the (Mouse→ Force → Residue) function. Play around with the system, apply forces in different directions, remove applied forces, etc. Have fun.