

Lab II: PSFGEN & NAMD Minimization

In this lab you will:

- 1) Learn how to use PSFGEN to create PSF and new PDB files.
- 2) Write an input file for NAMD to run a vacuum minimization of a protein.

Working directory: \$HOME/Desktop/aps_2016/exercises/lab_II

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).

1. Examine the CHARMM topology file located on your system.

Using a text editor, open the CHARMM topology file:

```
$HOME/Desktop/aps_2016/exercises/lab_II/common/top_all27_prot_na.inp
```

and search through the file to identify

- a) All of the defined definitions for the residue types, e.g. for RESI ALA ... merely write down ALA (it will be either a two-, three- or four-character name).
- b) Then, write down all of the definitions of the patches, i.e. search for all instances of PRES and write down the three- or four-character patch name (i.e. NTER, GLYP, etc).

2. Reproduce PSFGEN example build projects covered in the morning lecture.

<<<< WORK IN GROUPS OF TWO FOR THIS PART OF THE LAB. >>>>

The purpose of this problem is to reproduce the building examples that were covered in the morning lecture. If you recall, we went through the “single water” example together. The files that you are asked to create in this problem can be generally copied from the lecture notes.

Directories:

```
psfgen/  
  single_water/  
  protein/  
  protein_II/  
  protein_III/  
  protein_dna_complex/
```

Files: each directory under psfgen/ has a directory named output_building that contains a PDB file corresponding to that exercise. In order, (single_water to protein_dna_complex) the files are:

```
single_water/output_building/  
  oh2.pdb
```

```
protein/output_building/  
  icln.pdb
```

```
protein_II/output_building/  
  hydrolase.pdb
```

```
protein_III/output_building/  
    chain_a.pdb  
    chain_b.pdb
```

```
protein_dna_complex/output_building/  
    dna1.pdb, dna2.pdb, dna3.pdb, dna4.pdb, dna5.pdb, dna6.pdb  
    led1.pdb, led2.pdb  
    fixed_int1.pdb, fixed_int2.pdb, fixed_int3.pdb, fixed_int4.pdb
```

Procedure:

- a) Change to the psfgen/single_water/ directory.
- b) Starting with the single_water example create an input file to build a water molecule from the oh2.pdb file that contains a single oxygen atom.
You need to alter the path to the topology file to match that used on your system. The specific commands to build the system are shown in Figure 1 below.

```
topology ../../common/top_all127_prot_na.inp  
  
segment TIP3 {  
    auto none  
  
    pdb output_building/oh2.pdb  
}  
  
coordpdb output_building/oh2.pdb TIP3  
  
guesscoord  
  
writepsf output_building/one_water.psf  
  
writepdb output_building/one_water.pdb
```

Figure 1. Sample psfgen input file for the single water system

You should name this file build_system.psfgen. Then to run the psfgen program you should type:

```
> psfgen build_system.psfgen > build.out
```

Then to inspect the output using a text editor (or just type > cat build.out).

Examine the resulting output from running the PSFGEN program using your input file and the new PDB and PSF files.

- c) In successive steps, in the separate directories (protein/ , protein_II/ , protein_III/ , protein_dna_complex/), write input files and build the associated systems.
 - a. Remember to apply the NTER and CTER patches (proteins) or 5TER and 3TER (DNA) in the segment definition statements.
 - b. After you compose your psfgen input file you will run psfgen by typing

```
> psfgen NAME_OF_YOUR_INPUT_FILE > build.out
```

Where NAME_OF_YOUR_INPUT_FILE will depend on well, what you name your input script. You can name it anything, but it may be helpful to name the files the same way we did in lecture.

- c. Examine the resulting output from running the PSFGEN program using your input file and the new PDB and PSF files.

3. Create a PSF file and new PDB file for the following systems.

New patch types and PDB irregularities will be encountered.

Directories:

```
psfgen_II/  
  proper_dna/  
  myoglobin/
```

Files: each directory under psfgen_2/ has a directory named output_building that contains a PDB file corresponding to that exercise. In order, the files are:

```
proper_dna/output_building/  
  dna1.pdb, dna2.pdb
```

```
myoglobin/output_building/  
  3hc9.pdb
```

Procedure:

- a) Change to the psfgen_II/proper_dna/ directory.
- b) Write an input file to create the system. At the top of this file you will need to use the information in Figure 2 below. NOTE, this is only the top portion of the file; you will need to define the segments etc.
- c) Once you have drafted the psfgen input file, you need to write additional lines to apply the patches (DEO1 and DEO2) for *each* nucleotide. DEO1 is applied to pyrimidines (THY, CYT) and DEO2 is applied to purines (ADE, GUA).
 - a. For example, for the first nucleotide of the dna1.pdb molecule you need to apply
patch DEO2 dna:1
 - b. You need to inspect the dna1.pdb and dna2.pdb files and write down which nucleotide is a pyrimidine or purine and its associated residue number. Then you will write a separate patch statement as shown above for each nucleotide for each segment.
- d) Use your script to build the correct system. Examine the structure using VMD. Can you figure out what changed?

```
topology ../../common/top_all127_prot_na.inp  
  
pdbalias residue G GUA  
pdbalias residue A ADE  
pdbalias residue T THY  
pdbalias residue C CYT  
  
pdbalias atom THY O5* O5'  
pdbalias atom THY C5* C5'
```

Figure 2. The top portion of a psfgen input file that you are going to write for the proper_dna problem. You will have to compose remainder of this file.

- e) Change to the psfgen_II/myoglobin/ directory.
- f) Examine the file output_building/3hc9.pdb. Create separate pdb files for the protein and HEME group. For this exercise, you can omit the phosphate, sodium, and water atoms. Name your new pdbfiles myoglobin.pdb and heme.pdb.
- g) It is known that the Fe(III) of HEME forms a covalent bond to the NE2 atom of HIS-93 of myoglobin (Note: you need to be careful about the protonation state of the histidine).

- a. Using this knowledge, write a psfgen input file for this system. Be careful, there will be more patch(es) to apply other than the one needed to attach HEME to the protein. Name your myoglobin segment MYOG and your HEME segment HEME.
 - b. One patch that you need to apply to correct for the addition of HEME is FHEM.

```
patch FHEM HEME:154
```
 - c. There are at least two other patches that you have to apply. Read through the PDB file and your list of patches (PRES) and the CHARMM topology file to determine which additional patches are needed.
- h) Examine the final structure with VMD.

A picture of myoglobin with the HEME bound in the hydrophobic cleft is shown below.

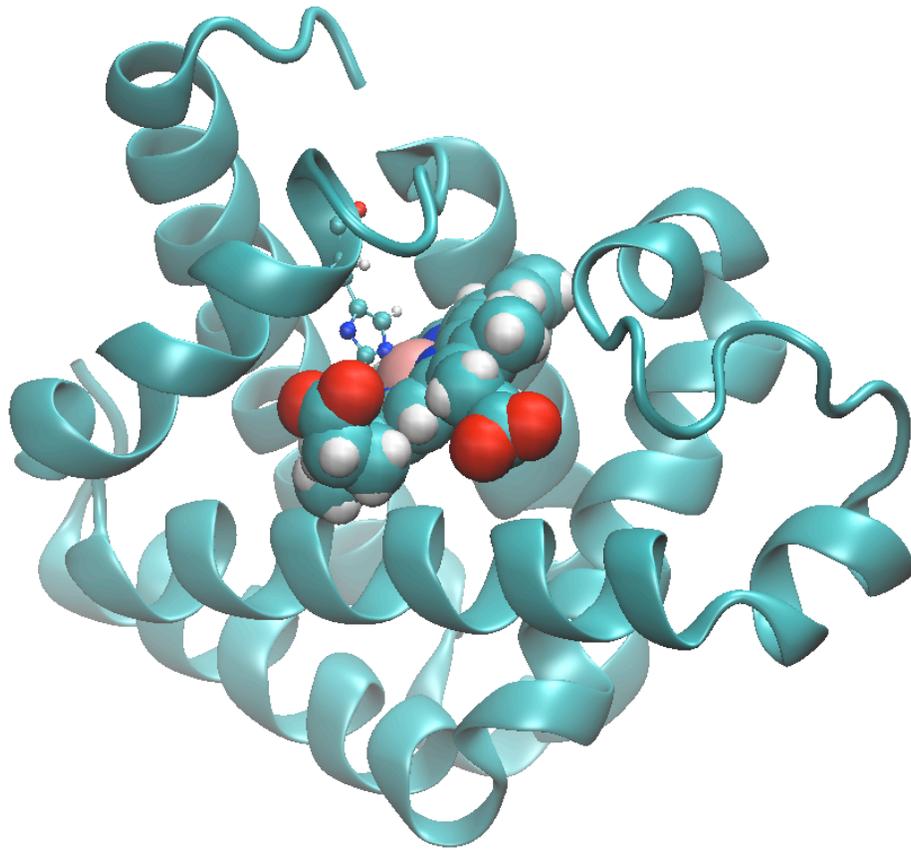


Figure 3. *Myoglobin, with the explicit bond between HIS-93 and the HEME iron.*

4. NAMD vacuum minimization of protein ICLN.

Directories:

namd/minimization

Files: you will generate these!

- a) Change to the namd/minimization directory and then create a directory called output/.
- b) Copy the files new_icln.pdb and new_icln.psf into the output/ directory. You created these files in part 2b above. If you named these files differently, then adjust your commands and inputs below.
- c) Write a NAMD input file to perform a vacuum minimization. Name the file min0. See Figure 4 below for the exact syntax you need in the input file.
- d) Run the minimization using your input file.
 - a. `> namd2 min0 > min0.out &`
 - b. ALTERNATIVELY, since we have dual-core CPUs, you could run this job in parallel using the command:
 - i. `> charmrun +p 2 $(PATH_TO_NAMD)/namd2 min0 >min0.out &`
 - or
 - `> namd2 +p 2 min0 > min0.out &`
 - ii. DON'T DO IT BOTH WAYS AT THE SAME TIME. Choose one. In either case the output is saved to a file called "min0.out".
 - c. NOTE: on some UNIX systems, the addition of an ampersand (&) symbol after the redirect command (i.e. ">"), allows you to redirect the error message to the output file. The final ampersand (&) allows you to run the job in the background. So anything that is normally written to the screen as program runs is sent to this file. Also note that when "charmrun" wrapper is invoked for parallel namd minimization, the full path of namd (\$PATH_TO_NAMD) need to be provided in the command line. You can view this file while it is running by typing
 - i. `> tail -90f min0.out`
 - ii. NOTE: this command looks at the "tail" of the file, and in this case the last "90" lines. The "f" tells the tail program to continually update so that as more lines are written you can see the updated material.
 - iii. NOTE2: the "tail" program you are running does not affect the min0.out file AND it will not quit unless you force it to do so by typing "control c".
- e) Examine the output file and resulting final structure using VMD.
- f) Write a new NAMD input file for one or more of the systems that you created above and minimize the structure(s). Examine the resulting output files and final structure(s) using VMD.

```
# sample NAMD configuration file for Minimization

# molecular system
coordinates      output/new_icln.pdb
structure        output/new_icln.psf
temperature      300

# force field
paratypecharmm  on
parameters      ../../common/par_all27_prot_na.inp

# approximations
exclude          scaled1-4
1-4scaling      1.0
switching        on
switchdist      10
cutoff          12

# output

outputname       output/min0
binaryoutput     no

# run control

minimize        1000
```

Figure 4. *NAMD* input file for the minimization run of the *icln* protein.