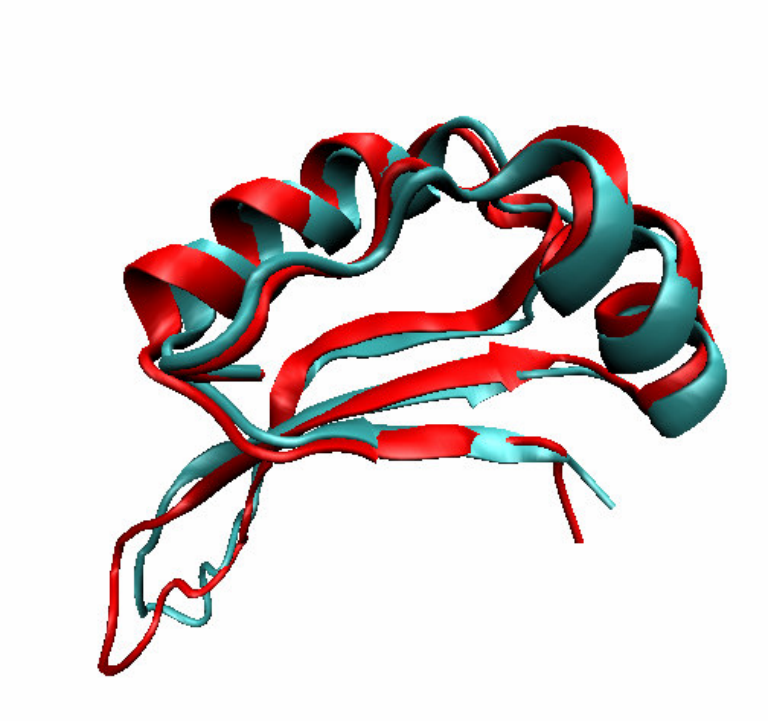
Laboratory 2:
CHARMM Molecular Dynamics. Minimization and Equilibration.

Nathan Hammond
nhammond@mit.edu

Issued: Tuesday, February 17, 2009
Due: Tuesday, February 24, 2009



In this lab you will:

1. Run the CHARMM software for molecular dynamics
2. Convert a protein structure from a PDB file to CHARMM's standard formats (PSF and CRD)
2. Perform minimization and equilibration on your protein structure to prepare it for a "production run" of molecular dynamics

!! Note: once you get this step started, you'll have to leave it to run for several hours before you can do the next step. Plan accordingly.

3. Run a molecular dynamics simulation

!! Note: Again, this will take some processing time. Plan for several hours before you have the final results.

Step 1: Execute CHARMM and examine the basic structure of a CHARMM script

Running CHARMM interactively

Login to the cluster as you did in Lab 1, using PuTTY or another SSH client. CHARMM version 35b1 has already been installed for you in the directory /usr/local/c35b1r1, and the executable "charmm" has been linked from one of the $PATH directories (directories where the system looks when you issue a command).

This means that you can execute CHARMM simply by typing "charmm" in the command line from any directory that you happen to be in. Try it.

You should see text that looks like this:

```
1
          Chemistry at HARvard Macromolecular Mechanics
           (CHARMM) - Developmental Version 35b1     August 15, 2008
    Copyright(c) 1984-2001  President and Fellows of Harvard College
                        All Rights Reserved
   Current operating system: Linux-2.6.21.7-2.fc8xen(i686)@ip-10-250-10-1
          Created on  2/ 6/ 9 at 17:18:53 by user: root

      Maximum number of ATOMS:     60120, and RESidues:       20040
      Current HEAP size:        10240000, and STACK size:  10000000
```

CHARMM is now running and waiting for a command. First, enter a title as follows. Start the lines with an '*' to indicate that this a title, and end the title by entering one line with nothing but an '*'.

```
* This is a test run
*
```

Try a few simple commands.

```
! Comment lines starting with ! are ignored
set x = 5
! The @ symbol recalls a stored value
calc y = @x * 7  !spaces are important
if y NE 35 set mathgrade 0
if y EQ 35 set mathgrade A
```

NE and EQ here are boolean operators for "not equal" and "equal". If we want CHARMM to exit nicely, we can type the following command:

```
stop
```

In this case, CHARMM tells us about memory usage and total run time and CPU time.

```
Total time               17.35 Other:          0.00

                NORMAL TERMINATION BY NORMAL STOP
                MAXIMUM STACK SPACE USED IS      0
                STACK CURRENTLY IN USE IS        0
                NO WARNINGS WERE ISSUED
                HEAP PRINTOUT-  HEAP SIZE  10240000
                SPACE CURRENTLY IN USE IS        0
                MAXIMUM SPACE USED IS          540
                FREE LIST
          PRINHP> ADDRESS:        1 LENGTH:  10240000 NEXT:        0

                $$$$$ JOB ACCOUNTING INFORMATION $$$$$
                 ELAPSED TIME:    17.35  SECONDS
                    CPU TIME:     0.00  SECONDS
```

Chances are, you've already had a typo that made CHARMM crash. If not, try running it again, but instead of typing "stop" type something like this:

garbagenonexistentcommand

CHARMM exits with an error message:

```
***** LEVEL  0 WARNING FROM <CHARMM> *****
***** Unrecognized command: GARB
****************************************
BOMLEV (  0) IS REACHED - TERMINATING. WRNLEV IS  5
```

(Scroll up to see the error message, and take a moment to appreciate the huge ASCII skull and crossbones that CHARMM prints when it crashes. This is not something I created—it is in the original CHARMM code. As if you don't already feel badly enough when you have an error in your script, now you have death staring you in the face.)

Running CHARMM from a script

It's not practical to run CHARMM from the command window, because even a minor typo will halt the program and wipe out the work that you've done. Instead, we write our commands into a CHARMM script file and pass them to CHARMM using the "<" operator.

charmm < *input.txt*

We can go one step further and collect the output from CHARMM in a file, like this:

charmm < *input.txt* > *output.txt*

Let's create a simple script using the Emacs text editor. First create a folder in which to work, perhaps "mkdir lab2", then change to that directory ("cd lab2"), then create a subdirectory like "practice" and change directories into it. *Look at Lab 1 again for a list of some basic bash shell commands for getting around with a Linux command prompt.*

Now create a text file that will serve as our CHARMM script, using the text editor Emacs. Type "emacs testrun.inp". The intro screen will disappear with your first keystroke. Now type the following text:

```
* CHARMM test script
*

date
! This will print the date and time
stop
```

Save the file using "Ctrl-x Ctrl-s". Exit the editor using "Ctrl-x Ctrl-c". You may have to enter 'y' to confirm if prompted.

Emacs is more straightforward to use than some other linux text editors, but it can be tricky if you hit a wrong key. Some of you may find that your "Backspace"key doesn't work, and you'll have to rely on "Delete" instead. If you're having problems feel free to work with the text files on your computer and upload/download using the FTP client.

Now let's run our script. Enter "charmm < testrun.inp". Did it work? The output should be displayed on the screen.

Normally we'll want to write this output to a file, so try entering this: "charmm < testrun.inp > testrun.out". Then take a look at the output file using "more testrun.out". Use *space bar* to scroll through pages, or "q" if you want to quite before you reach the bottom.

If it's a very long file, and you don't want to look through it line by line, you can search for certain words using the "grep" command. Type the following to see all the lines containing the word "SPACE":

```
grep 'SPACE' charmm.out
```

Almost always in Linux, case is important, so use uppercase letters for "SPACE".

What if you want to write the output from grep to a file? (Hint: It's the same way you wrote the output from CHARMM to a file.)

Step 2. Converting a Protein Data Bank (PDB) file to a protein structure (PSF) and a coordinate (CRD) file.

PDB is not a format that is native to CHARMM. We need to convert PDB's to CHARMM's own file formats: the PSF (protein structure file, containing a list of

all atoms present and information on how they are connected) and the CRD file (x,y, and z coordinates of each atom).

Getting CHARMM to read your PDB file is perhaps the biggest pain in the neck you will ever encounter in life. Really. I wish you well. If the instructions here don't work, email me the structure and ask for help. *But do it soon so you can get started on your project!!*

The problems are enumerated here:

1. Histidine residues may be protonated either on the ε carbon or the δ carbon, so when CHARMM reads a sequence the "HIS" residue name doesn't mean anything. It has to be "HSD" or "HSE".
2. Some of the naming conventions differ between CHARMM and the Protein Data Bank. So while in a PDB file an atom might be named "OCT1", CHARMM might want to call it "OT1" and won't know what "OCT1" means. We'll have to rename some atoms accordingly.
3. PDB structures are often incomplete, because small hydrogen atoms and any part of the protein that has a loose, inconsistent structure won't show up with NMR or X-Ray crystallography. For atoms missing from the PDB, we'll use CHARMM to come up with some reasonable guesses for the coordinates.
4. If the residue (amino acid) numbering in the coordinate file doesn't start on 1, we have to specify an 'offset' value when CHARMM reads the file. For example, if the first residue is numbered 486, we need an offset of -485.
5. When there is more than one protein chain in the structure, CHARMM has to read each chain separately. In our script, this means using separate "read coor" and "generate" statements for each—you'll see what I mean when you open the script file.
6. Where disulfide bonds exist (between Cysteine residues) we have to create those bonds in the CHARMM structure file using a 'patch'.
7. Sometimes the N- and C-termini also have a special structure that needs to be added using a 'patch' in CHARMM.
8. Sometimes proteins are crystallized with other small molecules that may not exist in the CHARMM parameter sets.
9. Who knows what else! But hang with me. We'll get through it.

These problems are listed in the order of pain-in-the-neckness. In this lab, you should be able to handle up to pain level 4. Levels 5-7 are no big deal if you get me to help you, and sometimes it's even possible to work around level 8 by creating our own residue topology file to define the molecule we need.

Get a PDB file

You should have a protein structure (PDB file) to work with. I'm going to work with 1by9.pdb, which is a DNA-binding domain from a protein used by Human Papillomavirus. By now you should really be looking at another protein that

you've downloaded from the Protein Data Bank. *I strongly encourage doing this lab with a structure you want to use for your final report! For a little help selecting a structure, see "Lab 2 Appendix: Finding an easy PDB file to work with" at the end of this document.*

Create a folder where you want to work (perhaps "lab2/pdb") and use FTP to upload the PDB file, (see Lab 1).

Rename Histidine residues in your sequence

I recommend following the convention of *all lowercase letters in file names*, so first I type "mv 1BY9.pdb 1by9.pdb". This will simplify things when we want CHARMM to read the files.

We're going to seek out every use of the word "HIS" in the PDB file and replace it with "HSD" so that CHARMM knows which carbon to protonate. Do that using the 'sed' utility:

sed 's/HIS/HSD/g' 1by9.pdb >1by9-hsd.pdb

Adjust the filename for your own molecule, of course. The "s" tells sed to do a substitution, and the "g" tells it to do the substitution even if there are multiple instances of "HIS" in a single line.

Rename atoms to CHARMM conventions

Next, we need to rename the atoms. There is a script called "efixpdb.awk" to help with this, something I borrowed from a tutorial by Robert Schleif at Tufts University. You'll find it in the /share/lab2 directory. Copy this file to the same directory as your PDB file ("cp /share/lab2/efixpdb.awk ./")

(Note the "/" before "/share/lab2". This means we're starting from the root directory. If you omit the initial "/" then it will look for a directory called "share" in whatever directory you are currently in and won't find the file you're looking for.)

"awk" is a special utility for reading data files and manipulating the data they contain. Run this awk script by typing

awk –f efixpdb.awk segid=*1by9* chainID=*A <1by9-hsd.pdb >1by9-fix.pdb*

Make sure you use the file with the corrected histidines. (Mine is 1by9-hsd.pdb.) You'll have to adjust the segid, chainID, and file name for your molecule. The segid is the PDB name (e.g. 1by9). If you start scrolling through your PDB file, eventually you'll start to see lines that look like this:

ATOM 1 N THR A 283 34.813 12.355 19.026 1.00 52.06 N

```
ATOM      2  CA  THR A 283      34.436  12.849  20.387  1.00 53.35           C
ATOM      3  C   THR A 283      32.958  13.228  20.447  1.00 51.77           C
```

The letter in column 5 is the chainID. Typically it's A. *Check to see if your molecule has more than one chain—say A and B—if it does, you can do this lab with just one chain, but you should email me for help on getting the whole structure*.

Now I have a new PDB file, 1by9-fix.pdb, with the corrections made. Look through it ("more 1by9-fix.pdb", then "q" to exit) to confirm that your atoms and their coordinates are listed there. Notice that our awk script discarded all of the remarks and even the sequence. That's fine, because CHARMM ignores everything prior to the "ATOM" lines. Even when CHARMM reads just the sequence from a PDB, it does so by scanning the lines that start with "ATOM".

Use CHARMM to build missing atoms and create a protein structure file (PSF) and coordinate (CRD) file

Get ready, because we're going to use CHARMM to do something real. Close your eyes and take a deep breath. *Ahh*. Again. *Ahh*.

There is a CHARMM input script in the /share/lab2 folder, called "build-psf-crd.inp". Copy it to the same folder as your pdb files.

Open it up to make edits. (You know how to by now—"emacs build-psf-crd.inp"). There are some notes in the script to help you know what to edit. You'll have to change the file names for the PDB file and the output files. The topology and parameter files are ok—don't change those. Look through the script line by line to see what it does. Save your edits (Ctrl-x Ctrl-s) and exit Emacs (Ctrl-x Ctrl-c).

Run the script by typing a command like this:

```
charmm < build-psf-crd.inp > build.out
```

If all goes well, two new files will appear, a PSF and a CRD file. (You'll see them when you type "ls" or "ls -l". The OUT file is a bit messy, but browse through it for a minute to see if you can pick out the commands we issued in the INP file and check that it didn't end in an error (with skull and crossbones).

Download the PSF and CRD files to your computer and open them in VMD to see if everything looks right. (After opening VMD, choose "File→New Molecule. Browse for your PSF file, confirm that the "Determine File Type" window says "CHARMM,NAMD,XPLOR PSF" and click "Load". Now Browse for your CMD file. Change the file type to "CHARMM Coordinates" and click "Load". Then close the "Molecule File Browser" window.) Does the protein structure look good? Good.

Step 3. Perform minimization and equilibration on your protein structure to prepare it for a "production run" of molecular dynamics

What are Minimization and Equilibration?

Once you have your PSF and CRD files, let's do some real molecular dynamics.

The first thing we have to do with almost any structure is a minimization. When we open a new structure, it's almost certain that there will be some bad contacts between a few atoms. Atoms that are overlapping or bonds that are stretched too far will make a molecular dynamics simulation go unstable and crash. Minimization just relaxes these high-force points so we can safely run MD.

Until you assign velocities to your structure, you could say that it's at absolute zero. Heating the structure to standard temperature requires us to get the atoms moving and bring the mean kinetic energy up to a certain level. Until we heat the structure up and allow it to equilibrate for a while, we shouldn't start collecting data for analysis.

In the script you'll run for this lab, the heating and equilibration are accomplished in the same step. We use something called "Langevin dynamics", which adds a random thermal force to the simulation to bring it up to temperature. Once the temperature reaches the setpoint, Langevin dynamics maintains the temperature constant while the structure equilibrates.

Minimization and equilibration your structure

If you want to avoid clutter, create a new directory to work in, perhaps "lab2/run". You'll need your PSF and CRD files in this directory, and you'll need the script called "min-equil.inp" from the "share/lab2" directory.

Open "min-equil.inp" and look through it. Read the notes that explain the script. Make whatever edits are necessary. Again, check that the file names match your molecule.

Run the script with this command:

```
charmm <min-equil.inp >min-equil.out &
```

Notice the ampersand ("&") at the end. This tells the machine to run this command in the background, so you can issue other commands or even logout while it continues to work.

If you enter a command with no '&' and then decide you want to let it run while you do something else, press 'Ctrl-z' to pause the process, then enter 'bg' to make it run in the background. This does the same things as the '&'.

Look to see if your process is running with the command "ps". If needed, you can kill the process by looking at its process id number output by the "ps" command and typing "kill -9 *pid*". If you logout of SSH and log back in, ps may not show you all of your processes unless you use the -Af option. You can also use 'grep' to see only processes run by you.

ps -Af | grep *team1*

The '|' operator takes the output from the "ps" command and "pipes" it into grep.

You may want to see just the last few lines of an output file to see how much progress has been made. For this use the "tail" command.

tail min-equil.out

Or, to specify the number of lines,

tail -100 min-equil.out

This can take a while to run, so check up on it for a while to ensure that you're not getting error messages. Make sure that it has written your minimized coordinate file (1by9-min.crd) within 10 minutes or so, and that it has started writing a dynamics trajectory file (1by9-equil.dcd). Then leave it for an hour or two while it runs the equilibration and generates a trajectory.

Look at the output files

Even if the molecular dynamics run hasn't finished, you can still download the file and see the trajectory (DCD) generated so far while CHARMM continues to run.

Let's take inventory and see if everything is working. Check to see that your output file didn't terminate in a skull and crossbones. Download the psf (1by9.psf), the minimized coordinates (1by9-min.crd), and the trajectory (1by9-equil.dcd).

Take a look at the minimized coordinates in VMD to see if they seem ok. (You should load the PSF first, then the coordinates.) Take a look at the trajectory file. (Again, load the PSF first). At the bottom of the "VMD Main" window, VMD should have a play button and slider bars that you can use to scroll through the frames and adjust the play speed. You can go open the "Graphical Representations" window and do all the fancy stuff you did in Lab 1, only now it's animated.

Equilibration time

I'll confess, the time we allowed for equilibration here was probably too short. I set the "time" variable in the "min-equil.inp" script to 10 picoseconds. In lab 3 we're going to evaluate whether this equilibration was adequate, and we'll probably find that we need an equilibration of more like 100 ps at least. Keep this in mind as you're prepping your molecules for your final project.

Step 4. Do a "production run" of molecular dynamics using the equilibrated structure.

For this last step, you'll have to wait until the equilibration is completely finished. Use "ps" to see if the process is still running and look at the output file to see what step the dynamics are on (it should run to 10000 and then stop).

After the molecule is equilibrated, we can finally do a real Molecular Dynamics run for the purpose of making observations about the molecule. In this lab, we'll simply do molecular dynamics with no extra forces or constraints. The difference between this and the equilibration that we did earlier is that we will use the restart file we generated after equilibration to run a simulation that is up to temperature and equilibrated from the start.

Copy the file 'run.inp' from the '/share/lab2' directory to your working directory. Look through the file and make the necessary edits to file names. Look to see what the script does. Run the script in CHARMM and write the output to a file such as 'run.out'.

After the run, again look at the output to see that there are no errors. Download and play the trajectory in VMD.

In Lab 3 we'll do some quantitative analysis of these trajectories. For now, pat yourself on the back and call it a day, you CHARMMer.

Lab 2 Appendix: Finding an easy PDB file to work with

I don't want to limit your projects in any way, but let me give you fair warning that some protein structures will be harder to work with than others. Some things that may cause difficulties:

- Large proteins (>100 residues, let's say) will require a long time to do minimizations and molecular dynamics runs.
- Proteins with multiple chains, disulfide bonds, or small molecules bound to the protein will take some extra work when converting the PDB file to CHARMM's format (PSF and CRD files).
- Some proteins may have special residues outset the standard set of 20 amino acids, and CHARMM may not have the parameters to deal with these.

Don't let these things stop you! If there is a structure that you really want to work with that falls into one of the "difficult" categories above, just talk to me for help when these issues come up.

However, if you wish to select a structure that will be a bit simpler to work with, here's how to do it. Go to the www.pdg.org website, and just to the right of the "Site Search" function at the top of the screen, click on "Advanced Search".

First let's limit size. In the drop-down menu labeled "Choose a Query Type", select "Chain Length" under "Sequence Features". Set it between 1 and 100. To the right of this, click on the "+" button to add another search criterion.

Now let's limit the "Number of Chains" under "Sequence Features" to 1, so set it to be between 1 and 1. Hit "+" again.

Let's say no disulfide bonds. Under "Structure Features" select "Disulfides" and set it to between 0 and 0.

Finally, we'll enter our search terms under the field "Advanced" under "Keywords". If you know which protein you want, you might instead enter its name as "Structure Title" below the caption "Structure Summary".

For each of these constraints on our search, we can click "Evaluate Subquery" to see how many structures exist. I query the keyword "mechanotransduction" and find only 5 hits, which tells me my search might be too narrow. I try "focal adhesion" instead and see 43.

At the lower right, click on "Evaluate Query" to see a list of possible structures to browse. If you don't see a protein that interests you, you'll have to broaden your search terms a bit—allow longer chains, multiple chains, or disulfide bonds, and just plan on a little extra work, and ask for my help when you need it.