20.320 Problem Set #1

Due on September 23rd, 2011 at 11:59am. No extensions will be granted.

General Instructions:

1. You are expected to state all of your assumptions, and provide step-by-step solutions to the numerical problems. Unless indicated otherwise, the computational problems may be solved using Python/MATLAB or hand-solved showing all calculations. Both the results of any calculations must be printed and attached to the solutions, and the corresponding code should be submitted on Course website. For ease of grading (and in order to receive partial credit), your code must be well organized and thoroughly commented with meaningful variable names.
2. You will need to submit the solutions to each problem to a separate mail box, so please prepare your answers appropriately. Staple the pages for each question separately and make sure your name appears on each set of pages. (The problems will be sent to different graders, which should allow us to get graded problem sets back to you more quickly).
3. Submit your completed problem set to the marked box mounted on the wall of the fourth floor hallway between buildings 8 and 16. Python codes when relevant should be submitted on Course website.
4. The problem sets are due at noon on Friday the week after they were issued. There will be no extensions of deadlines for any problem sets in 20.320. Late submissions will not be accepted.
5. Please review the information about acceptable forms of collaboration, which is available on the Course website and follow the guidelines carefully. Especially review the guidelines for collaboration on code. NO sharing of code is permitted.

Question 1 - Introduction to PyRosetta

PyRosetta is a python interface to Rosetta – a software suite developed in C++ by labs around the US (and the world) for protein structure prediction and design. This software is used and developed collaboratively, and used to solve problems ranging from predicting a protein's structure from its amino acid sequence, to designing enzymes for performing novel functions, to creating proteins with completely novel folds – and much more. To read more about Rosetta, visit http://www.rosettacommons.org/.

Throughout this course, you will use PyRosetta – a python interface directly to the C++ source code. Like Rosetta, this is a tool used for research, meaning it is constantly in development, and not everything always works perfectly. Through homeworks and a final project, you will all be able to use PyRosetta to probe and manipulate protein structures, both for structure prediction and to design novel protein structures. The following question is a basic introduction to PyMol (a protein viewer) and PyRosetta, aimed to get you acquainted with both pieces of software. In the excercises below, you'll be guided through using PyMol to view protein structures, and using PyRosetta to understand protein structure and energy. An answer sheet is provided to clarify which answers are expected to be turned in.

Installing and Running PyRosetta

PyRosetta as well as any other software you'll need has been installed on a remote computer with a virtual interface. To run it, download nomachine (http://www.nomachine.com/download.php), the client version, for your relevant platform. Once you've installed nomachine, the relevant settings are as follows:

> Host: fraenkel-nsf.csbi.mit.edu
>
> Desktop: Unix (in the first drop down menu), XDM (in the second drop down menu).

Your username will be your **Kerberos id**, and your password is your **student id number** – you SHOULD change this password (if you don't want to type this all semester) by opening a terminal once you've logged in, typing "passwd" and following the instructions.

If you're feeling rusty about your python, here's an additional Python Tutorial (for those that might like a refresher):
http:// http://docs.python.org/tutorial/

Please complete the exercises below which will introduce you to PyMol and PyRosetta, and submit answers where requested.

Exercises: (adapted from http://www.pyrosetta.org/tutorials)

Learning to Use GREP

You'll need to use this to download protein structures from the protein data bank (www.pdb.org) and get them into a format usable in PyRosetta – to do this, make sure you are using a COMMAND LINE and

not using python! (this isn't a python command!). It will make your life easier if you download the pdb files into the directory in which you have installed PyRosetta. For the purposes of the following exercises, use the structures 3IHW and 1YY8 from www.pdb.org

The grep command

grep "^ATOM" filename.pdb > filename.clean.pdb

Will take all lines in filename.pdb that start with "ATOM" and move them into a new file called filename.clean.pdb. When you load a pose into pyrosetta, make sure to use the clean file!!

IN A TERMINAL WINDOW, make "clean" versions of the files 3IHW.pdb and 1YY8.pdb, downloadable from www.pdb.org – when you use these in PyRosetta, make sure to use the clean version.

Now, you're ready to start working with PyMol and PyRosetta!

Basic PyMOL

Pymol is a viewer for macromolecular structures, typically derived from x-ray crystallography or NMR experiments. These structures are typically obtained from the RCSB Protein Data Bank (www.pdb.org),and come in pdb format. Pymol itself is available from www.pymol.org for free for academic use, and should run on any platform. The full pymol manual is available at http://pymol.sourceforge.net/newman/userman.pdf, and is relatively thorough.

Most pymol commands are available either through drop-down menus or a command line. Here I'll typically address the way to do things via drop-down menus. Let's start by obtaining a pdb file to play with. First download the pdb file 1YY8 (all pdb entries are named with a 4-character code). Then open it with pymol. A mass of sticks should appear on the screen. You can change your viewpoint by left-clicking and dragging to rotate the object.

If you are on a laptop, one-button mode is probably easiest for viewing. Select this from the mouse menu. If you have a 3-button mouse, select 3-button viewing.

Here are some basic mouse actions (these are also noted in the lower left corner of the viewing window).

Desired Action	1-button mode	3-button mode
zoom in and out	control and drag up/down	Right click and drag
translate the object in the viewing plane	alt and drag	Center button and drag

We can change the representation of the protein. On the right side of the viewing screen, click H (everything), click on the S next to 1YY8 and select cartoon. Numerous other representations are also available (spheres, surface, sticks, mesh). 'C' menu can be used for changing the color.

PyMOL will show the sequence of the protein if you select 'sequence' under the Display menu on the command line window. You can then select particular residues by clicking on the letter in the sequence at the top of the viewer window.

You can also select residues by simply clicking on them in the viewer. The atoms will have pink dots on them to indicate they are selected. You can modify the properties of whatever you have currently selected by using the drop down menus to the right of '(sele)' in the right column.

Another way to select things is via the command line. The syntax is

select <name>, <category> <identifier>

<name> is what you want to name your new selection. <category> can be any of:

resn	- residue type. identifiers are either 3-letter codes for amino acids or their sequence representations
res	- residue number
name	- atom in amino acid: N, C, O, Ca, Cb, etc
symbol	- element name
chain	- chain A, B, C, etc
ss	- secondary structure type: 'h', 's' or 'l'

Selection criteria (in the form of <category> <identifier>) can be linked together by boolean operators: and, or, and not. When you create a new selection, it will show up in the right hand panel and you can access the drop down menus to manipulate only the atoms in that selection.

For example type

select chainA, chain A

This way, you can modify only chain A of the molecule. For example, on “all,” click on H and choose everything, then chainA, click S and “cartoon” to show the cartoon version of only chain A.

In addition to creating a new selection, the above syntax for selecting parts of the structure is generally used throughout pymol. For example,

color red, res 40

will color residue 40 red.

Measurement is sometimes also useful. Select Residue 40 on chain A. What type of residue is this? Go to the 'wizard' menu on the command line window and select 'measurement.' Then click on the the alpha carbon, followed by the oxygen on the hydroxyl group.

To make a very pretty picture, type in ray and hit enter. This will ray trace the image. You can also change the color of the background in Display, background. Explore with Pymol, trying the different

types of view (cartoon, sphere, stick, etc), and make an image showing some feature of the protein that is interesting to you. Submit this image with your homework, but using "ray" and saving the image.

Basic PyRosetta

Open a terminal window, and type "iPyRosetta"

To load the PyRosetta library, type

```
from rosetta import *
rosetta.init()
```

Load a protein from a PDB file (the clean 3ihw you made from grep above)

```
pose = Pose()
pose_from_pdb(pose, "3ihw.clean.pdb")
```

Examine the protein using a variety of query functions:

```
print pose
print pose.sequence()
print pose.total_residue()
print pose.residue(5).name()
```

How many residues does this protein have?
What type of residue is residue 5? _____

Note that this is the 5th residue in the PDB file, but not necessary "residue number 5" in the protein. Sometimes the residue numbering follows a convention from a family of homologous proteins, and often several residues of the N-terminus do not show up in a crystal structure. Find out the chain and PDB residue number of residue 5: _____

```
print pose.pdb_info().chain(5)
print pose.pdb_info().number(5)
```

This protein has one chain, labeled chain A. Lookup the Rosetta internal number for residue 91 of chain A:

```
print pose.pdb_info().pdb2pose("A",91)
print pose.pdb_info().pose2pdb(25)
```

Protein geometry

Find the , ψ, and χ1 angles of residue 5:

```
print pose.phi(5)
print pose.psi(5)
print pose.chi(1,5)
```

We can also alter the geometry of the protein. Perform each of the following manipulations, and give the coordinates of the N atom of residue 6 afterward.

```
pose.set_phi(5,-60)
pose.set_psi(5,-43)
pose.set_chi(1,5,180)
```

To see the Cartesian coordinates of the N atom, use the command:

```
pose.residue(5).xyz("N")
```

Using Score Functions

Score functions are rosetta's way of calculating energies – similar to a ΔG (but not including an entropy term, and energies are unitless). Here, we will explore the types of energy Rosetta considers, and use an energy function to score our protein.

```
scorefxn = create_score_function('standard')
```

The option for standard tells Rosetta to load the standard all-atom energy terms. To see these terms, you can print the score function:

```
print scorefxn
```

Set up your own custom score function which includes just van der Waals, solvation, and hydrogen bonding terms, all with weights of 1.0. Use the following to start:

```
scorefxn2=ScoreFunction()
scorefxn2.set_weight(fa_atr,1.0)
scorefxn2.set_weight(fa_rep,1.0)
```

Enter the other weights and then confirm that the weights are set correctly by printing your score function.

Evaluate the energy of pose with your score function

```
scorefxn2(pose)
```

Evaluate the energy of pose with the standard score function:

```
scorefxn(pose)
```

Break the energy down into its individual pieces:

```
scorefxn.show(pose)
```

Some basic command line commands you might find helpful

ls = list all things in the current directory
pwd = tells you what directory you're in

cd .. = go up one level in your directory
cd /(filename) = change directory into that file
mkdir = make a new directory
emacs & = emacs is a text editor available on all platforms (aquamacs on a mac) that does markup for python, this command will open an emacs window, the & allows you to still type in the terminal

Question 1 Answer Sheet (be sure to attach your pymol picture)

1) What type of Residue is this? What is the distance between Alpha Carbon and Hydroxyl Oxygen?

2) What is the sequence of 3ihw?

3) How many residues does it have?

4) What is the name of residue 5?

5) What are the phi, psi, and chi1 angles of residue 5?

6) What are the Cartesian coordinates of residue 5?

7) List three terms in the rosetta standard energy function, and what they mean

8) What is the score of your pose (including the modifications made by changing phi and psi angles)? Now reload the pose – what's the score of the pose now? Is it higher or lower, and why?

9) What is the score of the pose with your score function?

10) What's the most important factor (highest value) in the score of your pose?

Question 2 – Protein Secondary Structure

Cells have built in mechanisms to respond to foreign small molecules – including inactivation, alteration of the small molecules target, reduced uptake of the molecule, or increased influx. In general, these mechanisms serve a protective function against potentially harmful agents and are important for organism survival.

In particular, many organisms contain proteins called multidrug resistance transporters, which can transport a wide range of substrates out of the cell. They are mainly ATP-binding cassette proteins, and one very important version in humans is called P-glycoprotein. P-glycoprotein is highly upregulated in cancer patients, and is thought to be responsible for well over half of chemotherapeutic failures. The protein has over 70 known substrates (many of them are chemotherapeutic) and many of the substrates are known to upregulate the protein in tumors. Understanding the structure and function of P-glycoprotein is important in improving cancer patient outcomes.

The structure of P-glycoprotein has been solved and is available from www.pdb.org

a) Download 3G61 from www.pdb.org and make a ramachandran plot to study it's secondary structure – to do this you'll need to write a python script to extract all the phi and psi angles from a pose. Hint: It's encouraged to use a loop structure to access the angles for each of the residues. For better coding practice, create empty arrays before executing loops, and be sure to comment your code! Plotting can be accomplished with pylab.
b) What secondary structural feature is the most common in this protein?
c) What percentage of amino acids are in this secondary structure (to answer this, you'll need to have phi and psi angle criteria for what counts as this secondary structure, and write a script to determine if each residue is in those allowed phi/psi angles)
d) Do you notice any residues (points on your ramachandran plot) that fall outside "allowed" areas of the plot? Circle one. What do you think this might mean?

Please remember to submit your scripts on Course website, and include copies of all plots with your submitted homework.

Question 3 – Understanding Protein Folding Energy

Chymotrypsin Inhibitor 2 is a 64-residue protein. To better understand the kinetics and equilibrium of folding of this relatively small protein, Itzhaki et. al. performed a series of mutations and studied folding equilibrium. A few of their mutations are selected here for further study. (Note, in the notation D52A represents residue D: Aspartic Acid at position 52 is mutated to an Alanine).

	ΔG_{fold}(kcal/mol)	Unfolded/Folded protein ratio
Wild Type	-32.07	
D52A	-14.1	
E14D	-29.9	
F50A	-16.2	
F50V	-22.2	

Data taken from Itzhaki LS et al (1995) *JMB* **254**, 260-288

a.) Calculate the ratio of unfolded to folded protein for the wild type and each mutant (use R = 1.987 cal/K/mol and temperature = 298K) and fill in the table above.

b.) Come up with a biological explanation for why each mutant had the small or large change on energy it had (you can do this by looking at the structure, or simply by knowledge of biology of the mutation made).

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012