7.343 Life from Light: Photosynthesis
Assignment #1: Create a structural illustration of a photosynthesis protein
Due: In class, Session 4

In this assignment you will use freely available software and data to illustrate the macromolecular structure of a protein involved in photosynthesis. Completing the assignment will take several steps:

1. download the structural coordinates of your chosen protein
2. view the protein in a .pdb viewer, such as Swiss-PDB
3. choose representation and orientation of your molecule that best illustrates a designated feature or function of the protein
4. capture or render image
5. answer some questions about your molecule

The product of this assignment should be either a color printout or electronic presentation of your illustration together with your answers to the questions posed below. On Session 4, you will share your images in class, either in the form of handouts, or digitally projected onto the wall, and discuss your images and aspects of the protein structure and function which you learned about in doing this exercise.

The files:

Protein	**PDB id***	**Focus**
LHCII	`1rwt`	pigments
PSII	`1s5l`	special pair
cyt b_6f	`1vf5`	heme groups
PSI	`1jb0`	special pair
F0F1 ATPase	`1bmf, 1efr, 1qo1`	ATP binding site

Questions (answer any five):
1. what does your protein do?
2. what is the resolution of your structure?
3. how many polypeptide chains does your structure file include?
4. how many polypeptide chains are in the native protein (*in situ*)?
5. what are the different cofactors in your protein?
6. what do the cofactors do?
7. what are the different pigments in your protein?
8. what do the different pigments do?
9. what is the size of your protein complex in kD?
10. what are the approximate dimensions of your protein in Å?

If you get stuck, please call, write, or see me! I can give more tutorials just about anytime over the next two weeks.

The tools:
Swiss-PDB Viewer (aka Deep View): http://ca.expasy.org/spdbv/
SPDBV Manual: http://ca.expasy.org/spdbv/program/DeepViewManual.pdf
POV-Ray: http://www.povray.org/
Other tools you can try: VMD, Chimera, and RasMol

Hints on representation:
Proteins can be represented in many ways: ball and stick, alpha-carbon trace, backbone only, backbone and side chains, space filling, ribbon, cartoon, and molecular surface, just to name a handful. Your illustrations can combine different representations; e.g. ribbon diagram for protein and space filling or ball and stick for a heteroatom (chlorophyll, FeS cluster, ATP, etc...). You can increase the amount of information in your image if you color each chain differently.

Hints on rendering in POV:
(Mac): for some reason, mac doesn't deal well with the path to the ".inc" file listed in the text of the ".pov" file that is generated by Swiss-PDB Viewer. I'm not sure if the same bug occurs in the Windows version.

In the .pov file, go to:

```
//************ OBJECTS *********
#include "yourmachine/yourdirectory/yourfolder/1q90.inc"
```

and chop off the path leaving just the filename:

```
//************ OBJECTS *********
#include "1q90.inc"
```

POV automatically generates shadows, which can be quite distracting in a complicated protein structure. You can modify the light source to be shadowless by inserting "shadowless" into the statement below

```
//************ LIGHTS *********
#declare Intensity = 2
background { color rgb < 0.000, 0.000, 0.000 >}
object { light_source {< -20.000, 20.000, -20.000 > color rgb
Intensity*1.000 }}
//END
```

giving:

```
//************ LIGHTS *********
#declare Intensity = 2
background { color rgb < 0.000, 0.000, 0.000 >}
object { light_source {< -20.000, 20.000, -20.000 > color rgb
Intensity*1.000 shadowless }}
//END
```

The X,Y,Z position of the light source can be changed to provide more even illumination. To do so, change:

```
//************ LIGHTS *********
#declare Intensity = 2
background { color rgb < 0.000, 0.000, 0.000 >}
object { light_source {< -20.000, 20.000, -20.000 > color rgb
Intensity*1.000 shadowless }}
//END
```

to:

```
//************ LIGHTS *********
#declare Intensity = 2
background { color rgb < 0.000, 0.000, 0.000 >}
object { light_source {< -40.000, 40.000, -40.000 > color rgb
Intensity*1.000 shadowless }}
//END
```