BIOC351: Proteins

PyMOL Laboratory #3

Scripting

Some information and figures for this handout were obtained from the following source:

http://www.virology.wisc.edu/acp/Tutorials/PyMol-E-density.pdf

In our last lab we concluded by making a number of images. Some were more complex than others, but all of them took us awhile to find just the right orientation, zoom, colors, representation, etc. If I asked you to do it all again you each would breath a sigh of relief because when you made these images you saved Session files (you did—right?!) which would allow you to bring you self back to exactly the way your image was made. However if you didn't save a Session file you could use your image to try and recreate everything. It wouldn't be fun, but since you are fairly familiar with the program it wouldn't take you long to redo it. If I asked you to do it in two months—I think you wouldn't be so happy.

To investigate scripting with PyMOL we will be modifying a series of scripts to engage in a more in-depth exploration of a number of the commands you used in Lab #2 as well as introduce you to finer control of the representation of your structures. Generally these scripts will investigate:

1. Size, colors and types
2. Images
3. Alignments
4. Electron density

Scripting

The real power of PyMOL is in its ability to be scripted—instead of remembering complex commands or complex GUI maneuvers you can write it out a series of commands in a text file and read them in. This is MUCH better than a Session file because you know what files are being read in and exactly how each object was made, colored, etc.

Script files are text files and as such should not be modified in Microsoft Word, as it (even in text mode) wants to add lots of "stuff" which will confuse PyMOL. Therefore if you are on a:

PC	use	Notepad
Mac	use	Textedit
Linux	use	vi, nedit, etc

To indicate something is a PyMOL script file, by convention, you add a .pml suffix to the file.

With scripting any line that begins with a # is ignored—so I encourage you (like I have done for you) to thoroughly comment your files so that you will recall what each command is doing.

Loading and Displaying Files

To facilitate us all having access at the same time, I have posted these scripts to a directory on the Biochemistry webserver (http://biochem.uvm.edu/BIOC351). Please download them and leave them on your Desktop.

Below is the file Exercise-A.pml (shown in blue) from the server:

```
# Exercise-A.pml
# Written for BIOC351 to teach scripting
# Stephen Everse, Jan 2011
#
#
# initialize PyMOL to bring everything back to baseline
reinitialize

# Fetch diferric pig transferrin (TF) from the PDB and rename it pig
fetch 1h76, pig

# you can also load files from your hard drive.  To do the same command as above
it would look like: (remove the #)
#load 1h76.pdb, pig

# as a general practice hide everything
hide everything

# create some selections
create fe, resn FE
create co3, resn CO3
create sugars, resn NAG

# create representations for the different objects
show cartoon, pig
show spheres, fe
show sticks, co3
show sticks, sugars

# color the objects
color slate, pig
color red, fe
color green, co3
```

```
color yellow, sugars

# lets label the oxygens in the carbonates red
color red, (name O1+O2+O3 and co3)

# lets label the oxygens in the sugars red and the nitrogens blue
color red, (elem O and sugars)
color blue, (elem N and sugars)

# reset the orientation
reset
```

Exercise A:

To run this script, select **File** → Run… from the Menu bar and find the script you downloaded called Exercise-A.pml
What do you think of the image? Anything strike you as odd?

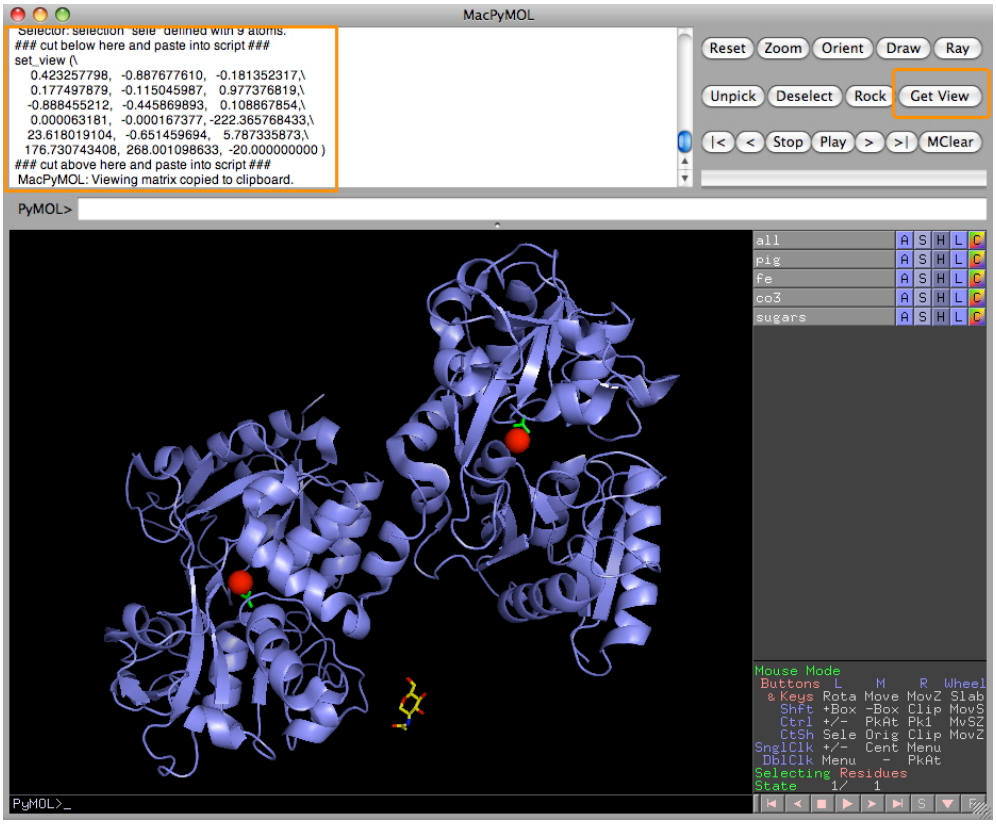
Orientation

What immediately bothers me is the strange orientation that the molecule is left in. You basically can't see any of the features of the structure that we just highlighted. So what I need you do is re-orient (zoom, spin, clip, etc) the molecule until you have what you think is the perfect orientation to highlight the features displayed. Now click on the Get View button at the Upper Controls. You will observe that a large set of numbers appears in the window above the command line (and also in your computers paste memory).

Let's edit our script to include our new orientation. Open the Exercise-A.pml in your text editor.
Go to the bottom of the file and past the new orientation there—it should look something like this:

```
set_view (\
    0.423257798,   -0.887677610,   -0.181352317,\
    0.177497879,   -0.115045987,    0.977376819,\
   -0.888455212,   -0.445869893,    0.108867854,\
    0.000063181,   -0.000167377, -222.365768433,\
   23.618019104,   -0.651459694,    5.787335873,\
  176.730743408,  268.001098633,  -20.000000000 )
```

Save the file in your text editor and re-run (**File** → Run → Exercise-A.pml)



Clicking the Get View button (orange oval) deposits the actual orientation required in the status window (orange box).

What do you see now? What looks odd now? Personally I find the irons to be too large and the carbonates to be too tiny to be recognized. Lets reduce and increase those sizes respectively. (Note these can also be changed **Setting** → Edit All, but they would apply to ALL objects. In the manner shown below we are changing it for named object only.)

Add these lines to your growing .pml file, save and re-run it:

```
set sphere_scale, 0.75, fe
set stick_radius, 0.5, co3
```

Anything else ODD about your resulting image? Why isn't your sugar connected? The reason is that we don't have a side chain displaying to connect to. How do we figure out which side chain we should be connecting to? We could go back to PDB and see there, we could click around nearby until we hit upon a reasonable candidate, or we could create a new object displaying things only within 5 Å of the sugar. A command you could use would be:

```
create neighbors, sugars around 5
show sticks, neighbors
```

You should have noticed that we had two amino acid fragments appear (see image). Click on them to determine which is the appropriate amino acid to be linked to a sugar. First replace the create sugars with the line below:

```
create sugars, resn NAG or (resid 497 and not name n+o+c)
```

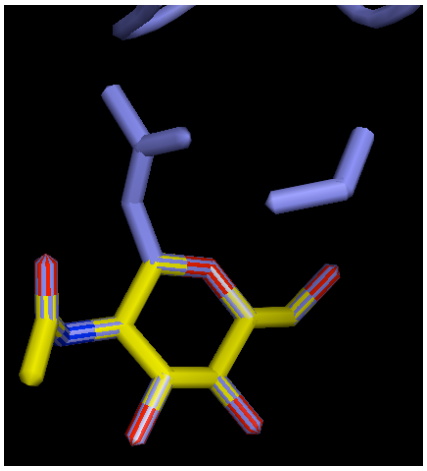
(Note the use of or in the command above)

Modify the appropriate lines in the script file to have a *:

```
color red, (elem O* and sugars)
color blue, (elem N* and sugars)
```

To create a bond between your Asn and NAG:

```
bond /sugars//A/ASN`497/ND2, /sugars//A/NAG`1688/C1
```



Two blue amino acid fragments appearing that are 5 Å from the sugar.

Let's add a surface to our object pig, color it differently and then make it transparent so you can see the other objects through it. To do this we actually need to make a second object. As always add these commands to the correct areas of the script.

```
set transparency, 0.75
create pigsurf, pig
show surface, pigsurf
color limon, pigsurf
```

Save and re-run the script. If you don't like the color, find another that is more pleasing to you!

Creating an Image

You can direct a script to save an image for you by adding just the following lines:

```
# ray traces your image – just like the button does in the upper controls
ray
# write out a image file called filename.png
png filename.png
```

With a script PyMOL uses the default background color (black). You can change this using the following command.

```
bg_color tv_orange
```

(or any other color you find under the **C**)

Generally this line is placed near the top of the file (after reinitialize). Add this line to your growing .pml file, save and re-run it.

Most often I find it best to maintain a transparent background, then I can overlay the image on any background I want in PowerPoint™. To create a transparent background replace your bg_color line in your script file with:

```
set ray_opaque_background, off
```

and run your script again.

Altering Your Cartoons

In the standard cartoon mode helices are spirals and strands are arrows. PyMOL also gives us other ways of displaying cartoons. Try these on the Command line and see what happens:

```
set cartoon_cylindrical_helices, 1
```

To return it to normal use:

```
set cartoon_cylindrical_helices, 0
```

What do you think these do?

```
cartoon tube
cartoon putty
cartoon oval
```

To return it to normal:

```
cartoon automatic
```

Rotations

I'm sure you have seen structural images that are rotated by 90° or 180° around the x, y or z axes. To rotate the orientation use the following command:

```
rotate y, 90
rotate x, -180
```

Most frequently I use this when I am making images—like:

```
#
ray
#
png filename.png
#
rotate z, 90
#
ray
#
png filename_z90.png
```

Exercise B: Let's experiment!

Take a few minutes to adjust Example-A.pml to fit what you find visually appealing. Make a couple of images.

Alignments

In PyMOL Lab #2 we played a little bit with alignments. I thought we should take it to the next level. First run Example-B.pml and see what happens (it appears you need to run it twice to get it to load the PDBs correctly).

What is the RMS (root mean square) deviation between the two structures? _____

What has happened is that the program found the same sequence (nlobe) in the larger structure (apo). What we'd like to do see how well it aligns with the other half of the apo. To be able accomplish this we need to tell PyMOL how to accomplish this. We will do this with residue ranges:

align (nlobe and resi 20-200), (apo and resi 348-528)

Edit Example-B.pml and replace the align command present with the one above, save and run.

What is the RMS deviation now between the two structures? _____

As we have seen, often you find more than a single molecule in a PDB file. Are they related crystallographically? Are they all the same? To be able to figure this out we need to be able to overlay them. To do this you must load a PDB file for each structure—see how we accomplished that in Example-C.pml. What are the RMS deviations for the 3 chains to A?

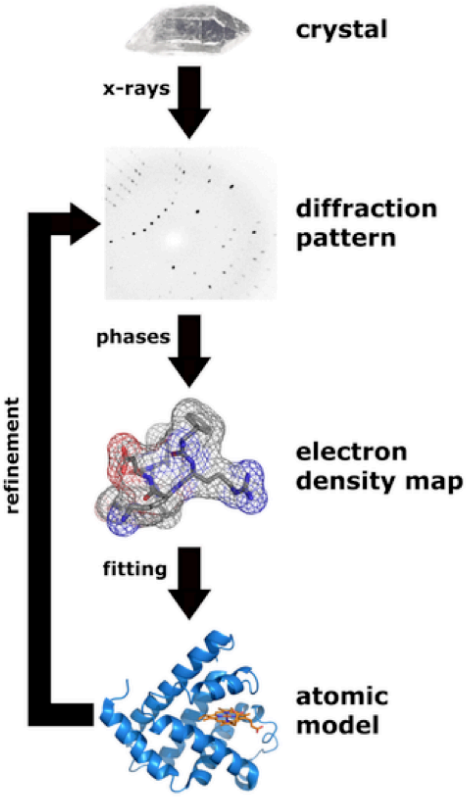
B to A _____ C to A _____ D to A _____

What do you notice that is different?

_____

_____

_____



Workflow for solving a crystal

Electron Density Maps

You may recall that on the first day I made a comment about the structures (models) that you have been viewing is someone's interpretation of the primary data (electron density). Therefore I encourage everyone to download and view the electron density associated with their structure. To do this you need to go to PDB and look up 1RYO. What is the structure you are looking at?

_____

What is its resolution? _____Å

Most structures these days deposit the structure factors required to generate the electron density. To download the desnity, click on EDS in the Experimental Details panel. It takes you to the Electron Density Server at Uppsala University.

Click on Maps within the Download area at EDS. This will open a new window, you need to select:

Map format: CCP4
Type: 2mFo-DFc

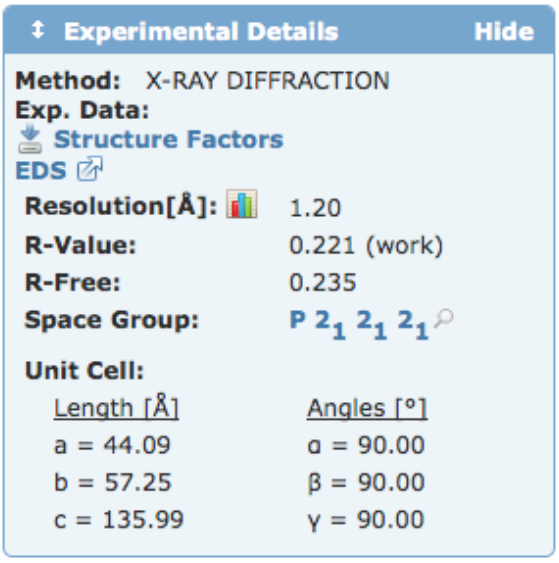
And click Generate map to make your map. When it completes, download, uncompress it and leave it on your desktop.

I have made a script for you (Example-D.pml) that displays and highlights some features of 1ryo. Please examine the script and see an example of when select is required over create. Please run the script and make notes of what you are looking at. I am especially interested in the strange appearing amino acid on the left of the image. _____

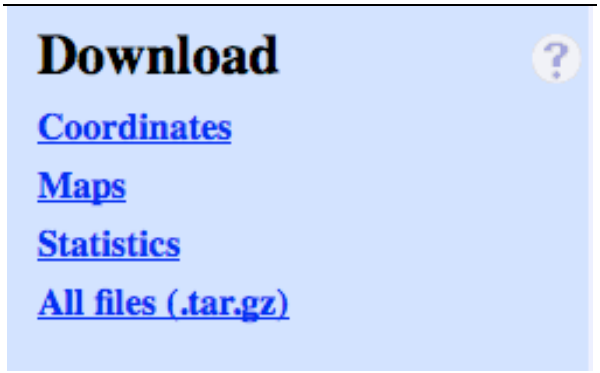
_____

_____

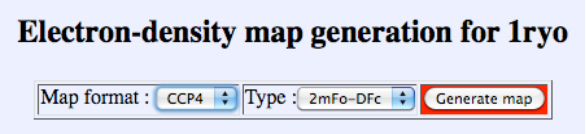
_____



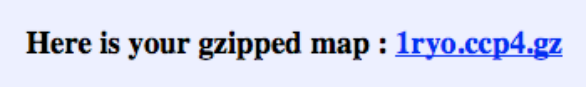
Experimental details at PDB.



Download region at EDS.



Type of map to generate at EDS.



EDS map file is compressed.

Add the following lines to your script to get a map to display:

```
# load and display the map
load /Users/sje/Desktop/1ryo.ccp4
isomesh map, 1ryo, 2.0, all, carve=1.6
color white, map
```

What do you observe? _____
_____
_____
_____
_____

What are the unfilled balls everywhere? _____