Exercise 1: Essentials

In the field of computational structural biology, we constantly inspect and manipulate molecular structures. This exercise will introduce you to the environment and basic tools we will use to do so.

Contents

The Linux environment 1

PDB 1

PyMOL tutorial 2

- Basic operations in PyMOL 2
- Structural Analysis 3
- Comparing Molecules 5
- Scripting and High-Quality Visualization [2] 5

Protein classification systems 6

- SCOP 6
- CATH 6

The Linux environment

Linux [1] is a free, open-source operating system. It is based on Unix and so is stable, powerful and flexible. Linux comes in handy when we work in the command line, which we will extensively.

If you are not familiar with the Linux terminal environment – go through this tutorial and inform the TA. At the very minimum, you should be able to easily do the following using command lines: move around the directory tree, create/copy/move/delete files, view and edit text files.

From now on: `$> command` will indicate a command you can invoke in the command-line interface.

PDB

The Protein Data Bank is the worldwide repository of information about the 3D structures of macromolecules, including proteins and nucleic acids. Structures are usually determined by X-ray crystallography or NMR experiments, and deposited in the PDB by the scientists involved. Each such structure is assigned a unique 4-character code called a *PDB ID*.

How do we use this data? A structure is described by a PDB file which is essentially a tabulated text file (for more see its Wikipedia page) that describes all the available details about the structure, most importantly every atom's coordinates. Software tools can read, manipulate and write structures in PDB format.

Go to the PDB website and download the coordinates (PDB file) for PDB ID "1YY8" (after you found the entry, use the right panel to download the "PDB File(text)"). This is a structure of the monoclonal therapeutic antibody cetuximab.

Now, examine the PDB file (it's a text file and can be viewed by writing `less 1yy8.pdb`). The main portion of the file is composed of `ATOM` entries that contain the 3D coordinates of each atom in the molecule. Prior to the coordinates section there is a header with ample information about the structure

(molecule solved, experiment conditions etc).

PyMOL tutorial

PyMOL is a molecular visualization tool. We use it to view molecular structures, gain intuition and come up with interesting questions to ask about them.

There are many such tools available (SwissPDBv, rasmol, VMD, MolSim, Insight II, etc.). We use PyMOL since it has excellent features for viewing, it is fast and the display quality is very good. It can handle multiple molecules at once and it is easy to define custom objects such as complexes or sets of atoms. It is also freely available for academic use.

See the PyMOL reference sheet on the course website for more details.

Basic operations in PyMOL

We will now inspect the Cetuximab structure that you have just downloaded.

1. Open PyMOL (`$> pymol`) from the directory where you saved the file, and load it (type `load 1YY8.pdb` in the PyMOL viewer window).

   > ***Tip***
   >
   > If you know the PDB ID you're after, you can skip visiting the PDB website, and load structures straight from PyMOL. Type `fetch <pdb_id>`, and PyMOL will query the PDB repository for you and load the structure immediately.

2. Use the mouse and mouse buttons to rotate, zoom, and translate the molecule. Left button=rotate, middle button=translate, right button=zoom.
3. The buttons at the top right can set the viewing parameters. `A=Actions, H=Hide, S=Show, L=Label, C=Color`. In the line where 1YY8 appears, select "Hide → Everything", then "Show → Cartoon", then "Color → By Chain → By Chain (e,c)", then "Label → Chains."

> ***Tip***
>
> "show → as" replaces your current representation with another, as opposed to show that adds different representations on top of each other.

Proteins are basically long folded chains of amino acids. Many functional proteins are just a single peptide chain. However, some proteins (antibodies for example) are composed of more than one chain.

4. You should see two copies of an antibody, since there were two copies of the antibody in the unit cell of the crystal based on which the structure was determined. Each antibody is composed of two chains. If you click on any atom, the console window will display information identifying that atom, including its chain. Confirm that the four chains are identified as chains A, B, C, and D.

Let's zoom in on one antibody. In the PyMOL console, type the following commands:

```
# (Comments in texts begin with a # sign)
select AB, chain A+B
# this defines a selection AB that contains all atoms from chains A and B
hide all
show cartoon, AB
orient AB
```

Note that in the panel at the top right, you can now manipulate subset AB using the buttons. You can use the mouse or the `select` command with other protein descriptors (for example: `name ca+cb`, `symbol o+n`, `resn lys`, `resi 1-50`, `ss h`. more example are in the course's resources page) to create objects for various subsets of the molecule. You can also combine descriptors (`chain A and hydro`) as in the following:

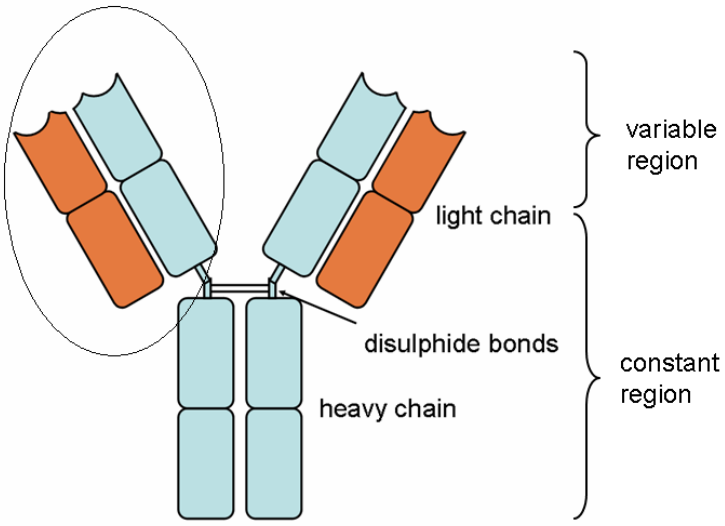
```
select linkerA, chain A and resi 107-112
color red, linkerA
select linkerB, chain B and resi 117-122
color orange, linkerB
```

Note that you can use "File → Save Session" at any time. This will store all your current objects, selections and views in a PSE file. You can later load that file and return to the exact state you saved.

At any problem you might have, you can always type `help select` and `help selections` for full details, or use the PyMOL wiki page for help.

Structural Analysis

The structure you have downloaded is Cetuximab, a therapeutic antibody in development for cancer treatment. Antibodies are composed of two heavy chains and two light chains; this particular construct is known as a `Fab fragment` (circled in the picture) and contains one full light chain (chain A) and the N-terminal half of one heavy chain (chain B). At one end of the antibody are six loops known as the *Complementarity Determining Regions*, or CDRs, which bind a particular antigen.



A full Ab. Fab shown in circle. Taken from Wikipedia

Q1. Find and specify (as sequence ranges) two beta strands in chain A which are adjacent in structure but not in sequence. Hint: select chain A and from the right panel controls hide everything else. Then click

"Color → Spectrum → Rainbow" for that selection (to color along the sequence from blue at the N-terminus to red at the C-terminus).

Q2. Look at the sheet from the side. You can see it is not completely flat. In which direction does it twist (clockwise/counter-clockwise)? Compare the direction of two neighboring strands. Do all strands twist in the same direction? Consult the picture below for help.

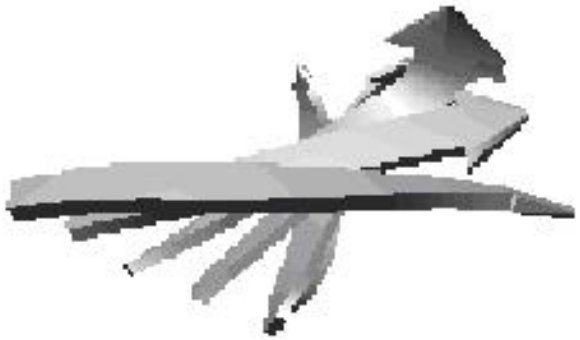


Figure from swissmodel.expasy.org

Let's analyze a couple of strands at the N-terminal domain of chain A. Create a selection for strand 3 (residues 19-25) and a selection for strand 8 (70-75), in a similar way as in the linker selections above, and hide the rest of the molecule.

Now add sticks ("Show → Sticks"). "Color → ByElement" is recommended - this will color different atom types according to the CPK coloring scheme: oxygen atoms in red, nitrogen atoms in blue and carbons based on your choice. Make sure you can distinguish between a residue's backbone atoms and side-chain atoms. To see the whole sequence, choose from the main menu "Display → Sequence" or click on the "S" button on the bottom right.

Q3. What is the sequence of the strands? What is the sequence pattern in these strands? Hint: try looking at the strands from different angles.

Let's analyze some geometry. From the main menu, select "Wizard → Measurement". You should see a panel on the right in which you can select the measurement of your choice (distances, angles, dihedrals or neighbors). Instructions on the left prompt you to select atoms for measurement. When finished, press done in the right panel. "Label → Residues" from the right-side panel might also be helpful.

Q4. a. What is the distance between the N of L73 and the O of F21 (i.e., the hydrogen bonding distance across the β chain)?

b. Measure 5 of the backbone hydrogen bonding distances between these two strands. What is the range of distances you observe? Recall the definitions of bond angle and dihedral angle from class. Dihedral angles are defined by 4 points in space ("around a line"). In a protein the dihedral angles are defined as:

φ - describes rotation about the N-Cα bond and involves C(O)-N-Cα-C(O) atoms.

ψ - describes rotation about the Cα-C(O) bond and involves the N-Cα-C(O)-N atoms.

ω - describes rotation about the C(O)-N bond and involves the Cα-C(O)-N-Cα atoms.

χ1 - describes rotation about the Cα-Cβ bond and involves the N-Cα-Cβ-Cϒ atoms.

c. On residue F21, what is the bond angle around the Cβ atom?

d. On residue F21, measure ϕ, ψ, ω, χ1.

Comparing Molecules

Q5. In the PDB site, find a second PDB structure of Cetuximab bound to its antigen (Hint: similar PDB ID). What is the antigen?

Clear your current PyMOL session by typing "delete all" and load your new PDB file. Use the cartoon view, and color and label by chain to get an overview of the Fab fragment and the antigen. The antigen also has several post-translational glycosylation modifications.

Now load 1YY8 into the same session. Create a selection for chains A and B and hide chains C and D (you will now need to specify the molecule since you have more than one PDB file loaded: `select unbound_fab, 1YY8 and chain A+B`). Similarly, create an object (call it `bound_fab`) for the Fab fragment of the bound complex (look at the PDB page to find out the relevant chains). Now, superimpose the two structures using `align unbound_fab, bound_fab` (This can also be performed from the "Actions → align → to selection → ..." side bar. When selected from the side bar, the sequences on top are also shown aligned.

The structural difference between the two molecules is measured by *root mean squared deviation* (RMSd) between corresponding pairs of atoms in both molecules (typically in proteins Cα atoms). The `align` command in PyMOL fits two structures one on top of the other, and then calculates RMSD by:

$$RMSd = \sqrt{\frac{1}{n}\sum_i (x_i - y_i)^2}$$

PyMOL performs sequence alignment to define pairs of atoms to compare, and then finds and executes the coordinate transformation which yields the minimal RMS deviation between the structures.

Q6. In the command window, there should be a few lines describing the alignment process. What is the RMSd calculated for this structural alignment?

Q7. Is there much difference between the bound and unbound forms of the antibody? In particular, are there differences in the six complementarity-determining loops *(CDRs)* at the far end of the N-terminal domains (where the antigen binds)?

Scripting and High-Quality Visualization [2]

Your commands can be saved to a file or read in from a file. Click "File → Log" to create a script and record your steps. You should give your log-file-name the extension `.pml` so you can later load the file as a script, to repeat your commands in a new session.

All your commands, typed or clicked, will be recorded in the log-file, but not the changes you make by reorienting the molecule with the mouse. Therefore, to record the current screen orientation matrices in your script, type get_view (so you will be able to retrieve this specific orientation later on). To stop recording your commands, click "File → Close log" or type `log_close`. Now you can move your molecule and return to the original orientation by typing `set_view` followed by the screen orientation matrices that `get_view` provided you (the matrices are found in the log file), e.g. if `get_view` saved inside the log -0.561639726, -0.827382088, -0.000185494, then `set view -0.561639726, -0.827382088, -0.000185494,` will return you to that conformation.

The command `ray` uses a ray-tracing algorithm to compute the lighting on the molecule (`ray 800,800` will set the image size to 800 × 800 pixels). Use it before saving an image using "File → Save Image" to create publication-quality results. It often makes sense to save a figure with a white background (e.g. for publication). For that purpose use the command `bg white`.

Q8. Create a ray-traced, white-background, publication-quality figure of the Cetuximab antibody fragment: choose an interesting feature of Cetuximab (e.g. β-sheet structure, the antibody complementarity-determining regions, a comparison of bound and unbound antibody loops or the CDR H3 loop in detail). Then, use whatever combination of techniques you learned (e.g. colors, shapes, labels, measurements) to emphasize that feature. Use a log to record the commands that produce the view you want. You should produce:

a. PNG image of your figure, ray-traced in 800x600 resolution and with a white background.

b. PML file that reproduces exactly that figure (test it!).

c. text file with a brief statement (1-3 sentences) of the structural feature your figure is designed to illustrate.

Protein classification systems

SCOP

The Structural Classification of Proteins (SCOP) database is a comprehensive ordering of proteins of known structure, according to their evolutionary and structural relationships. We will explore it using an X-ray structure of ubiquitin (PDB ID 1UBI), an important small protein that is attached to other proteins mostly as a label for degradation.

Q9. Go to the SCOP website and look for ubiquitin. What is its place in the fold hierarchy? Trace its full lineage (you can search by PDB id's).

SUMO (Small Ubiquitin-like Modifier) proteins are a family of small proteins that are covalently attached to and detached from other proteins in cells to modify their function. SUMOylation is a post-translational modification involved in various cellular processes, such as nuclear-cytosolic transport, transcriptional regulation, apoptosis, protein stability, response to stress, and progression through the cell cycle.

Q10. Where would you look for SUMO proteins in SCOP? Find an example of a SUMO protein in SCOP, download it's PDB file and compare it to 1UBI in PyMOL. How similar are they?

Select a SCOP superfamily that has the same SCOP fold as ubiquitin (but not the same superfamily) and choose a random protein from that superfamily. How similar is it to ubiquitin? Write the PDB IDs you found and their respective RMSd to ubiquitin.

CATH

CATH is a similar hierarchical fold classification scheme, but opposed to SCOP the classification is completely automatic (and therefore often more up to date with the current status of the PDB). Similarly to SCOP, Ubiquitin and SUMO share most of their CATH lineage (Ubiquitin 3.10.20.90.1, SUMO 3.10.20.90.14). Why then are two such classification schemes needed?

It turns out that most of the domain assignments are indeed similar between SCOP and CATH, however there are some disagreements. For instance the structure of papain (PDB: 1ppo). SCOP classifies the structure as one domain (SCOP code: 4.3.1), whereas CATH splits the structure into two (CATH code: 1.10.190.10), (3.10.160.10).

Further reading: Caroline Hadley, David T Jones, A systematic comparison of protein structure classifications: SCOP, CATH and FSSP, Structure, Volume 7, Issue 9, 15 September 1999, Pages 1099-1112.

---

1 Linux is named after Linus Torvalds, its original developer.
2 adapted from tutorial by Jeff Gray