Protein visualization and modelling workshop – Lecture 11: Introduction to Structural Biology

BIO-212 - Structural Biology

November 27, 2025

1. Introduction to PyMol

To know the structure of the protein means to better understand protein function and, potentially, to be able to intervene with its abnormal activities in diseases. On the website of the Protein Data Bank (PDB) the 3D protein structures are stored as **pdb(protein data bank)/cif (crystallographic information file)** files that contain the coordinates of the atoms in the protein of interest. Programs such as PyMol, VMD, ChimeraX etc. enable us to explore these 3D structures and allow for some manipulation.

Before the workshop, please make sure to install PyMol ahead of time, to speed up the process on Thursday. For that, please go to PyMOL | pymol.org to download the latest version of PyMol for your Operating System.

After the installation process is completed, when you open PyMol, you may be prompted to upload a license file – **you do not need a license file to use PyMol** – so you can ignore this prompt!

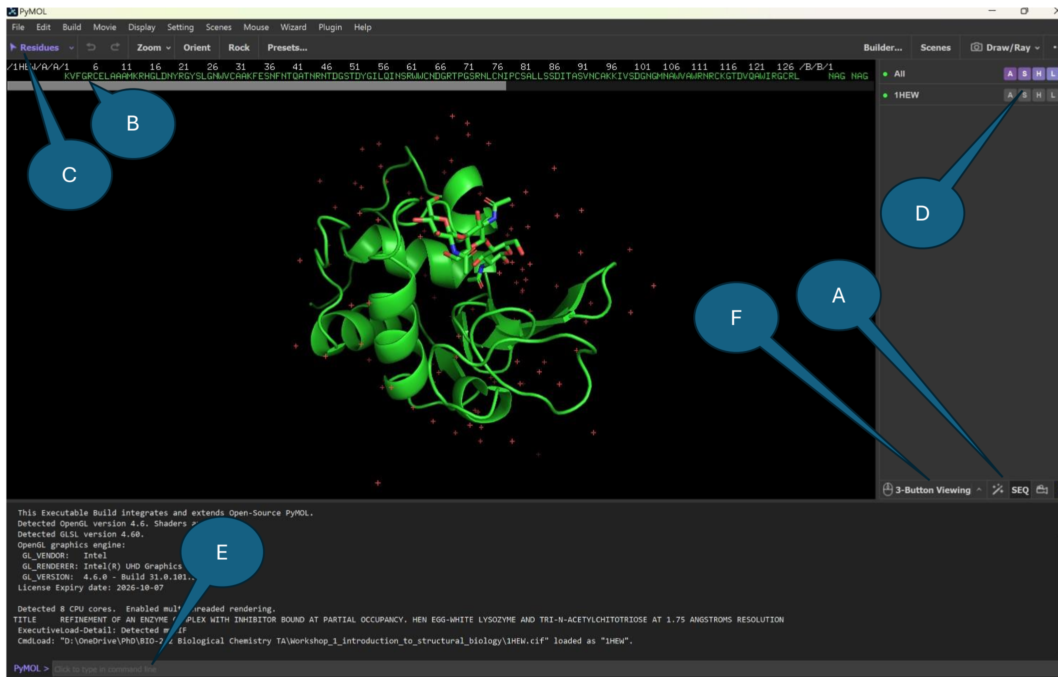
For this exercise session, we will start first by getting our hands on PyMol, and understanding how to navigate it – then we will study actual protein structures: **RanGTPase,** the **F-actin binding domain, SARS-CoV-2 spike protein** and the **RNA Pol II Transcription-elongation complex.** These parts will have questions for you to answer.

1.1 Navigating PyMol

For any tasks required during this week's and next exercise sessions, feel free to consult PyMol's wiki: Help:Contents - MediaWiki or ask a TA.

So that you don't lose any information/selections etc you have generated I strongly suggest you save your session at regular intervals. To do this, go to **File -> Save Session**. This is also useful if you want to save a particular analysis of your protein structure through using **Save Session As…** to generate a new filename. To start the session up at any time in the future, you can go to **File -> Open** and chose the file with the **.pse** extension.

You can undo in PyMol by pressing Ctrl (or Command for Macs) Z, or clicking the undo button on the top left of the screen.



1. Download and open the file **1HEW.cif** from Moodle. This file contains a structure of an enzyme in complex with an inhibitor molecule. We will use this file simply to get used to navigating PyMol
2. To view the amino acid sequence click on **SEQ (A)** in the bottom right hand corner. You can also click on **Display** on the top main menu then **Sequence.** This will open up the **sequence panel (B)** where you can see the amino acid sequence of the protein. The text written above the sequence: /1HEW/A/A/1 refers to the pdb 4 character entry name that corresponds to this protein **(1HEW).**
3. We can select a number of things on PyMol. If you click on **"Residues"** in the top left hand corner **(C)**, you can change between selecting individual protein residues, or atoms or entire protein chains.

4. The **object panel** on the top right **(D)** allows you to perform basic activities with the molecules

   a. **Action** A: perform viewing and computation activities

   b. **Show** S: add the representation of molecules e.g. display carbon-alpha backbone, show surface, amino acid side chains etc.

   c. **Hide** H: hide the representation of molecules

   d. **Label** L: set label on selected molecules e.g. display the name and number of a particular amino acid

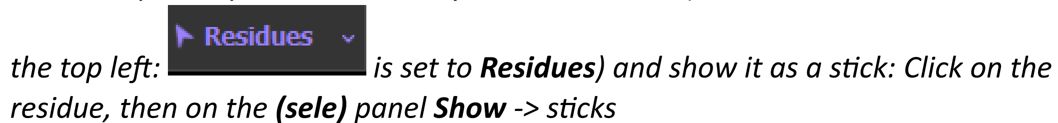
   e. **Colour** C : change colour appearance of the molecule

It is important to know to which molecules to apply changes. You can do it either to only one of the depicted molecules (e.g. 1HEW) or a particular selection (sele) or all molecules (All)

Let's play around with a few representations:

*On the 1HEW tab, click on **Show** -> sticks (All the atoms (except hydrogens) will now appear)*

*On the 1HEW tab, click on **Hide** -> sticks (only the cartoon representation should remain)*

On the sequence panel, click on any random residue (make sure that the selection on the top left: 

*is set to **Residues**) and show it as a stick: Click on the residue, then on the **(sele)** panel **Show** -> sticks*

5. PyMol is a really powerful software as users can also interact with its command line **(E)** to carry out functions. PyMol has its own command language – For more information consult the wiki: [Category:Commands - PyMOLWiki]

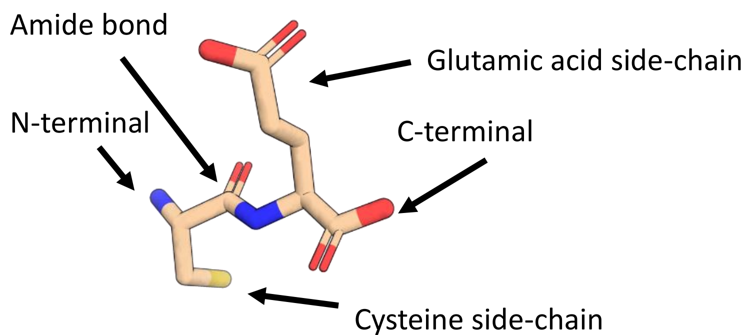
   Let's try a few simple commands. Type the following commands in the command line of PyMol, press enter after every command and observe what happens

   select phenylalanines, resn PHE
   show sticks, phenylalanines
   color magenta, phenylalanines

6. On the lower right **(F)** You can see how the mouse buttons will function with combinations of keyboard keys. Clicking on the arrow next to **3-Button Viewing** will allow you to change mouse settings.
   By default, these are the ways to navigate with a mouse:
   ***Rotate**: press and hold left mouse button and move mouse*
   ***Zoom in and out**: press right and hold mouse button and move mouse*
   ***Mouse wheel**: scroll to see the "slab" view of the molecule change. The "slab" view allows you to look inside a protein molecule.*

Portions of this tutorial were adapted from: Jones, Dafydd, A Simple Tutorial for PyMOL: Visualising Proteins Using Molecular Graphics

1.2 Atom Colours in PyMol

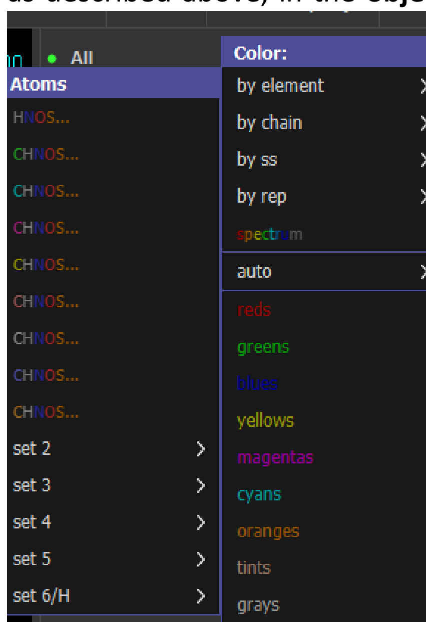


When you look at a molecule in PyMOL, each atom is given a **standard color** so you can immediately tell what type of atom it is. For example, in the sticks representation of a Cys–Glu dipeptide (as shown above) you will see:

- **Nitrogen (N) → blue**
- **Oxygen (O) → red**
- **Sulfur (S) → yellow**

These colors follow common conventions in structural biology and help you quickly recognise functional groups (acidic groups, amines, thiols, etc.). The carbon skeleton here is shown in light brown. Hydrogen atoms are not shown in most PDB structures, and PyMOL does not display hydrogens by default.

If you have an object visualised as sticks, and would like to see the colours of each atom type as described above, in the **object panel**: Colour -> by element -> Click the first colour set:



This is usually the default way of visualising each atom type, however will change after changing the colours of each object.

2. Ran GTPases – 1st Exercise

Now that we're more familiar with PyMol's interface, we can get started with an exercise. In the file uploaded on Moodle, you should find a pymol session called **Ran.pse** in which you will find two pdb files: **1rrp** and **1byu.**

Ran is a small **GTP-binding protein** (a member of the Ras superfamily) that acts as a **molecular switch** in the cell.
It cycles between two states:

- **Ran–GTP** → "active" form
- **Ran–GDP** → "inactive" form

Ran is essential for:

- Nuclear transport (import/export through the nuclear pore)
- Mitotic spindle assembly
- Nuclear envelope formation

Like all GTPases, Ran switches states through GTP binding, hydrolysis, and exchange, and its structure changes depending on which nucleotide is bound.

You will explore two structures of Ran which were solved in its two states: **1rrp** – Ran bound to GNP (a non-hydrolysable analogue of GTP) and **1byu** – Ran bound to GDP.

1. These two structures were taken from the Protein Data Bank. Go to RCSB PDB - 1RRP: STRUCTURE OF THE RAN-GPPNHP-RANBD1 COMPLEX and RCSB PDB - 1BYU: CANINE GDP-RAN.
    a. Which structural biology method was used to solve the structure of these proteins? Describe the method briefly.
    b. Take a look at the resolutions of the deposited structures, in your opinion are they high? The websites also give information about the quality of the deposited structures. Which values should we look for to discern the quality of the two structures? For **1rrp** and **1byu**, can you evaluate these values?
2. Open **Ran.pse.** First thing we can do is align the two structures to see the main differences between the two states. In PyMol's command line type in:

align 1rrp, 1byu

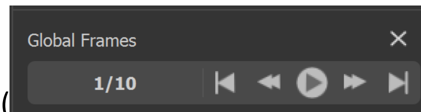
The align tool in PyMol allows you to superimpose two structures, while minimising the **root mean squared deviation (RMSD)** between the two structures.

  a. What is the RMSD value given in PyMol's output console (directly above the command line)? In your opinion, is it large?
  b. What major conformational differences do you observe on the structure of these two Ran isoforms? Take a screenshot and add arrows pointing to the major differences. Speculate or consult literature on whether it can have biological relevance.

3. Now we will analyse some interactions between the ligand and Ran. For this, we will only need **1byu**, so unclick **1rrp.** Make sure that you are selecting residues (check that the selection in the top left is set to **residues** and not **chains**). Open the **sequence panel** and click on **K23 <u>and</u> GDP** (scroll to the right to find GDP).
    a. Show them both as sticks, and colour them in magenta. Then to again view the colours of each atom type, go to Colour -> by element -> click on the first option. **The phosphorus (P) atom here is coloured orange.**
    b. Then, to identify interactions between K23 and GDP, For **(sele)** click on Action (A) -> find -> polar contacts -> within selection
    c. Take a screenshot of the 3 interactions that are found between the phosphates in GDP and K23. Recalling from the first Lecture, name each type of interaction. Are they all attractive interactions?
    d. In both **1byu** and **1rrp** there is a large sphere sitting in the ligand binding pocket of Ran. What is this sphere? Why do you think this protein needs it?
    e. Histidine is a very interesting residue because it has a pKa (6.0-6.5, for an isolated amino acid in aqueous solvent) which allows it to sample between protonated (positively charged) and deprotonated (neutral) states at pHs near physiological pH (7.4). It also has a “flexible” pKa, which means that its tendency to be protonated or not depends largely on its local environment. In the **GDP** bound state **(1byu)**, find H30. Based on its surrounding hydrogen bond environment, can you determine the protonation state of this Histidine at physiological pH? (Hint: show the entire structure as sticks, and on the **1byu object**, find polar contacts “within selection”).

3. F-actin Binding Domain – 2nd Exercise

Next, we're going to take a look at the structure of the F-actin binding domain of a kinase. In the files from Moodle, download the file called **1ZZP.cif.** 1ZZP contains the F-actin binding domain of the ABL kinase, a small helical bundle responsible for anchoring ABL to the actin cytoskeleton. This domain helps regulate ABL's localisation and function by mediating direct interactions with actin filaments, which is important for cell movement and signalling.

1. Open the file. On the bottom right (

), you can play through the "states" of 1zzp.
    a. Click play – what do you observe?
    b. Given this observation, can you reason as to which structural biology method was used to solve this structure? Describe the method briefly, what are the advantages of this method over others?
2. Below are a sequence of commands for you to enter in the PyMol command line, one by one. Press enter after each command:

    color gray
    select hydrophobes,(resn ala+gly+val+ile+leu+phe+met+trp+tyr+pro)
    show sticks, hydrophobes
    color orange, hydrophobes
    disable hydrophobes

    a. Now, have a look at the distribution of hydrophobic residues. Where are most of them located? Take a screenshot.
3. Now we will analyse the distribution of polar contacts across the helical bundle. Hide the sticks. Then, for the **1ZZP** object, go to Show (S) -> main chain -> sticks. Now you see only the backbone atoms as sticks. Then, again for **1ZZP**, go to Action (A) -> find -> polar contacts -> within selection.
    a. Take a screenshot of what you see. Analyse the distribution of interactions. Where do you find most of the hydrogen bonds?
    b. Given your answer to question 2a and 3a, and your knowledge about protein folding can you explain two major thermodynamic properties that lead to the stabilisation of this structure?
    c. We saw in Lecture 5 that alpha helices are very ordered structures with well-defined helical parameters. Can you recall what the length of one helix-turn is?
    d. Since the F-actin binding domain is a very well-defined helical bundle, we can also check ourselves! For this, you can use the **Measurement** tool. Go to: Wizard (top of your screen)-> **Measurement**. Here you can click on two atoms, and the Wizard outputs a distance in **Angstroms.** In the way you think is best, measure the distance of one turn of a helix in **1ZZP**.

4. SARS-CoV-2 spike Glycoprotein – 3rd Exercise

Now, for the next exercise, we will briefly look at a very large protein complex, solved by Cryo-EM. Cryo-EM images "snapshots" of particles frozen in vitreous ice. By averaging across many of these snapshots, cryo-EM can produce density maps for which a model can be fitted. Based on how well represented certain regions of the biomolecule is across these snapshots, the resolution may vary. **6vxx** is the structure of the SARS-CoV-2 spike glycoprotein, solved by Cryo-EM.

Open **6VXX.cif** from Moodle. Enter the following commands one by one into the PyMol command line:

color red, chain A

color blue, chain B

color yellow, chain C

1. How can you tell that the SARS-CoV-2 spike protein is a homotrimer?
2. Based on your knowledge of SARS-CoV-2, can you tell which part of the protein is most important for recognizing host cells (take a screenshot of the domains)? Consult literature if necessary.

5. RNA Pol II Transcription-elongation Complex – 4th Exercise

The last structure we will analyse is one we all know very well! **RNA Polymerase II** in the process of transcription elongation. Open **7OL0.pse** from Moodle.

1. Which structural biology method was used to solve this structure, and why is it suitable for such a large complex?
2. Looking at the nucleic acids, can you identify the **DNA–RNA hybrid** region in the active site? Which strand is DNA and which is RNA? (Take screenshots)