



UNIVERSITY OF
BATH

Department of
Life Sciences

A self-paced tutorial for visualising protein structures using PyMOL

Dr Charlotte Dodson

Name.....

Reminder of amino acid one- and three-letter codes

Amino Acid	1-letter code	3-letter code
Alanine	A	Ala
Cysteine	C	Cys
Aspartate	D	Asp
Glutamate	E	Glu
Phenylalanine	F	Phe
Glycine	G	Gly
Histidine	H	His
Isoleucine	I	Ile
Lysine	K	Lys
Leucine	L	Leu
Methionine	M	Met
Asparagine	N	Asn
Proline	P	Pro
Glutamine	Q	Gln
Arginine	R	Arg
Serine	S	Ser
Threonine	T	Thr
Valine	V	Val
Tryptophan	W	Trp
Tyrosine	Y	Tyr

Reminder of the first few letters of the Greek alphabet

Letter	Pronunciation
α	alpha
β	beta
γ	gamma
δ	delta
ϵ	epsilon
ζ	zeta

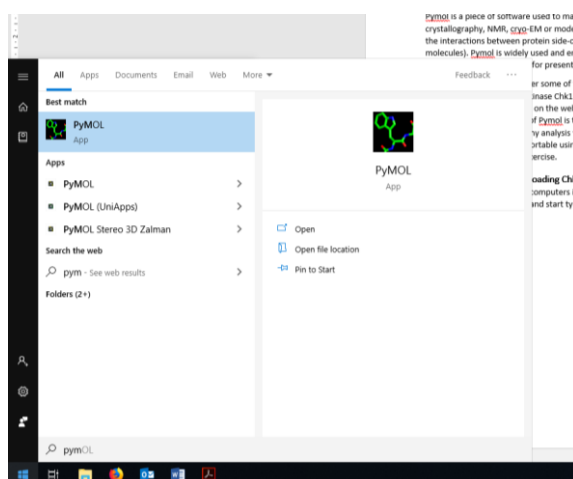
PyMOL workshop

PyMOL is a software package used to manipulate protein structures (pdb files) determined by x-ray crystallography, NMR, cryo-EM or modelling. It is free for academic use and enables us to visualise the interactions between protein side-chains or between side-chains and ligands (including drug molecules). PyMOL is widely used and enables us to create high quality figures for publication. It also enables us to create movies for presentations.

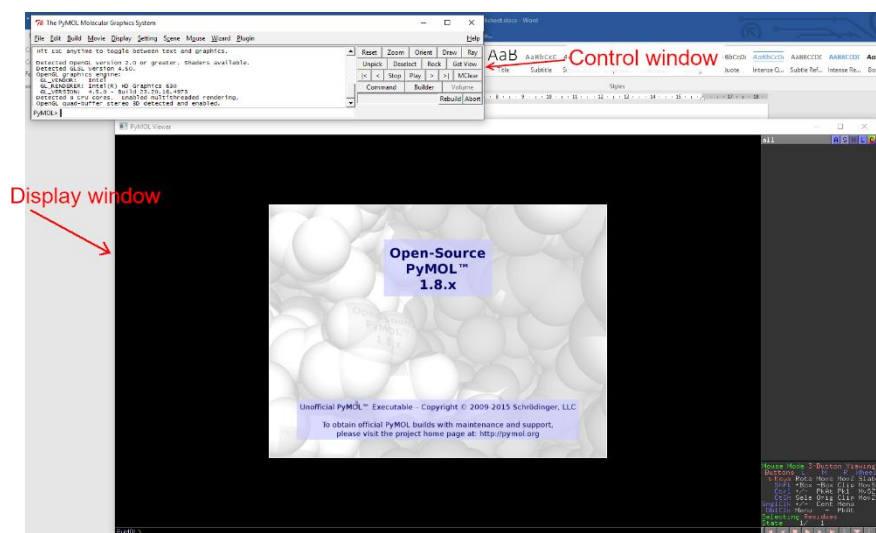
In this tutorial we will cover some of the main features of PyMOL (*eg* opening, saving, manipulating pdb files) using the protein kinase Chk1 (Checkpoint kinase 1) as an example. There is lots of additional material available on the web and lots of advanced functionality in PyMOL beyond the topics we will cover here: one of the strengths of PyMOL is that it has built-in command line access and also the ability to write scripts in the programming language python to carry out nearly any analysis we can think of. The features we will cover here should enable you to become comfortable using PyMOL and will be useful next semester when we come to the virtual drug discovery exercise.

Opening PyMOL and downloading Chk1 structure

This tutorial assumes that you have PyMOL installed on your computer (if you need to install PyMOL, please see p16). To open PyMOL, click on the Windows icon (or press the Windows button) and start typing 'PyMOL'. When the PyMOL app is highlighted, select it with your mouse.

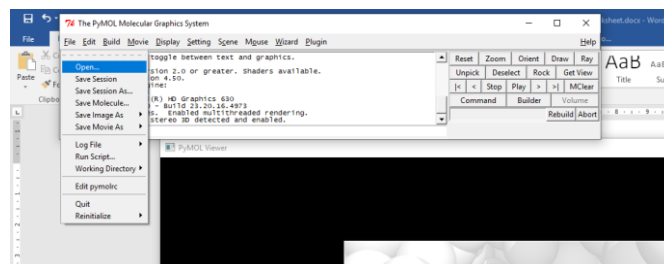


Two windows should open on your screen. One of these is a display window and is used to display and manipulate protein structures. The other (the control window) has a menu at the top and is used to modify how the protein structures are displayed.

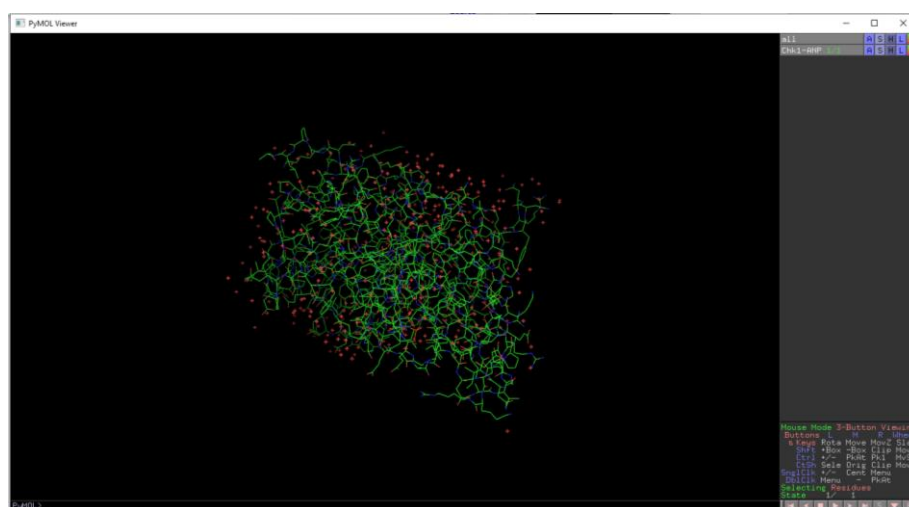


Download the file *Chk1-ANP.pdb* from the NTF and save the pdb file somewhere where you will be able to find it.

Return to PyMOL and click File -> Open in the Control window:



Select the pdb file that you have just saved – it should immediately open in the Display window:



Manipulating your protein structure

Protein structures are manipulated by click and drag with your mouse. There are three buttons on the mouse, each with its own functionality:

- | | |
|---------------|---|
| Left button | Rotate the molecule around a fixed (often central) position |
| Middle button | Translate the molecule (<i>ie</i> move it up, down, left or right on the screen without rotating it) |
| Right button | Zoom in and out |

Have a play and start to get a feel for what each button does.

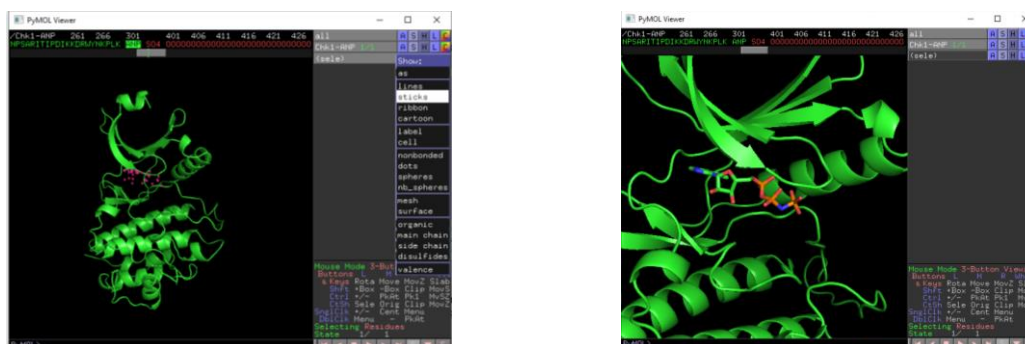
Menus in the Display window

To the right of the display window there are five windows: *Action*, *Show*, *Hide*, *Label* and *Color*. Four of these (*Action*, *Show*, *Hide* and *Color*) are very useful. The *Label* menu is quite clunky – I never use it (I label protein figures in standard graphics software when I need to).

Rather than select an amino acid, we are going to select the ligand. Scroll right using the scroll bar directly underneath the protein sequence until you get to 'residue' 301. This is labelled ANP. Click on ANP – you will find that this is a single object, not three amino acids.

You will also see that a new object called '(sele)' has appeared directly underneath 'Chk1-ANP' at the right hand side of the window. This happens every time any atoms (or combination of atoms) are selected in the window. The menus to the right of (sele) will apply their commands *only* to the atoms currently selected.

Go to the *Show* menu for (sele) and choose stick representation. You will see that the ligand (only) is now visible as sticks.



Ligand identity

Zoom in on your structure and have a look at the structure of the ligand. By default, PyMOL colour-codes the atoms in a protein as follows:

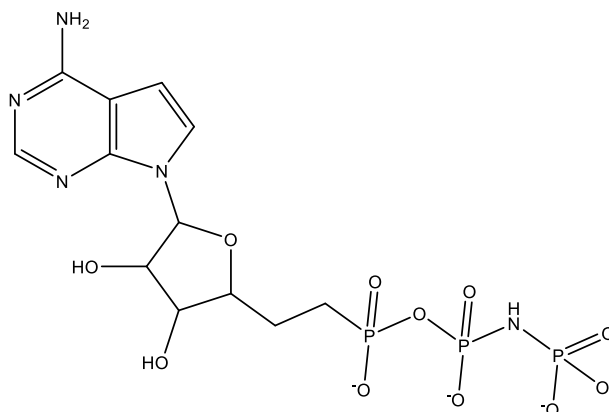
Atom	Colour
Carbon	Colour selected (green in example here)
Nitrogen	Blue
Oxygen	Red
Sulphur	Yellow
Phosphorous	Orange

Note that PyMOL often does not show hydrogen atoms. This is because most protein structures are obtained through x-ray crystallography and hydrogen atoms do not diffract x-rays. This means that no experimental electron density is generated for hydrogen atoms – essentially they are invisible and we have to infer (assume) their location.

The ligand in your protein structure is AMP-PNP, a non-hydrolysable analogue of ATP which is often used in crystallography to mimic the binding of ATP.

Question:

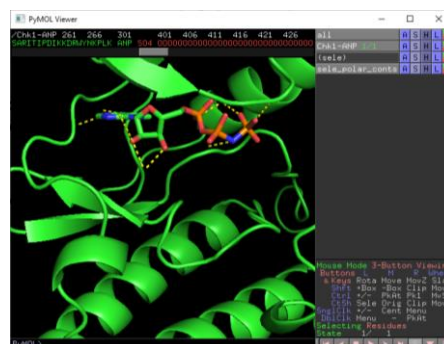
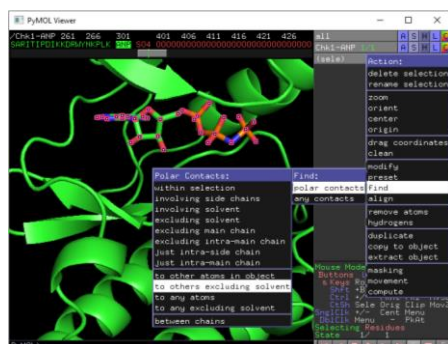
I have made two errors when drawing ANP-PNP. Have a look at the structure shown in PyMOL and correct the structure below. (Ignore differences in the P=O double bonds.)



Finding hydrogen bonds and other polar interactions (and finding the identity of individual atoms)

PyMOL is able to predict likely hydrogen bonds and salt bridges in any protein structure based on atom identity and distance. This helps us find the interactions which are holding our ligand in place.

Select AMP-PNP in the display window by clicking on the molecule. Then go to Action -> find -> polar contacts -> to others excluding solvent.



You will find that some dotted yellow lines appear. These are hydrogen bonds which PyMOL has predicted are found between AMP-PNP and atoms in the protein.

As drawn this is not very informative – we cannot see which protein atoms are involved in the hydrogen bonds. To find these atoms, turn on line (or stick) representation for the protein. You will see that each yellow line ends on a specific atom.

In order to find the identity of an atom, double click on it. A menu will pop up with a slightly cryptic header containing information on the atom selected. (Sometimes you need to rotate the protein or change the zoom so that PyMOL properly understands which atom you intended to click on.)

The header has the format below:

/Molecule_name (eg Chk1-ANP)//chain_name (ignore this for now)/residue_name_and_number (using the three-letter code)/atom_name

PyMOL uses a code to name atoms which is related to the formal chemical naming of amino acid atoms:

C	Backbone carbonyl carbon
N	Backbone amide nitrogen
O	Backbone carbonyl oxygen
CA	C _α (backbone carbon next to carbonyl carbon)
CB	C _β (sidechain carbon attached to C _α)
CG	C _γ (sidechain carbon attached to C _β)
CD	C _δ (sidechain carbon attached to C _γ)
OG	O _γ (sidechain oxygen attached to C _β)
OD	O _δ (sidechain oxygen attached to C _γ)
OE	O _ε (sidechain oxygen attached to C _δ)
NZ	N _ζ (sidechain nitrogen attached to C _ε)
etc	

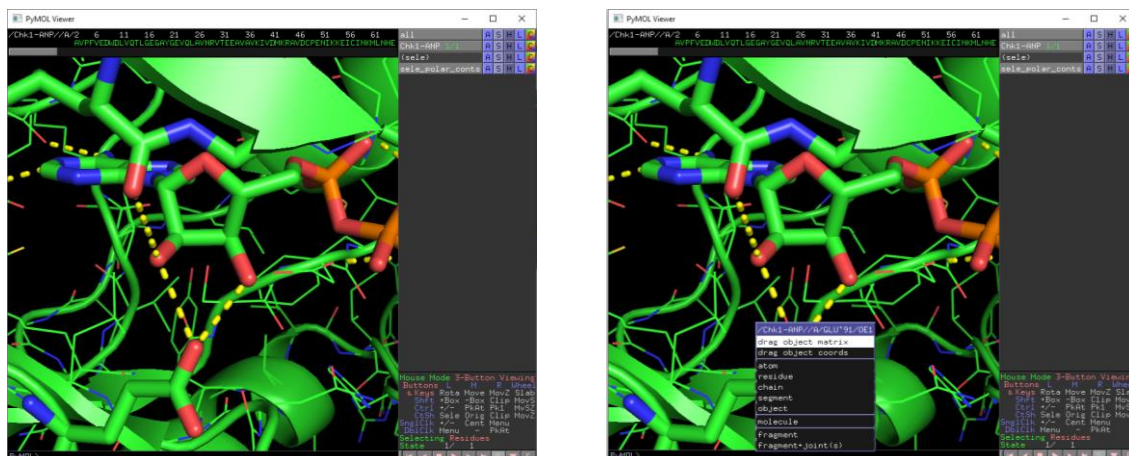
The hydrogen bonds made by the adenine rings (nitrogen-containing rings) of ATP with protein kinases are highly conserved, and are made with residues which are known as the *hinge region* of the protein. These interactions are often maintained by small molecule kinase inhibitors.

Question:

With which atoms on the protein does the adenine ring of AMP-PNP make hydrogen bonds? In your answer, you need to name the atom (or chemical group), and also state whether it is backbone or side-chain atom. Have a look at the example below if you are not sure what this means.

Example:

The hydroxyl groups of the ribose ring of AMP-PNP make hydrogen bonds with the sidechain oxygen of Glu91 and with the backbone carbonyl of Leu15.



Conserved salt bridge

Nearly all protein kinases have a highly conserved salt bridge between an lysine on the β -sheet region ($\beta 3$ strand) and a glutamate on a specific helix (known as the αC helix). In Chk1, this salt bridge is between Lys38 and Glu55.

Find these residues on the protein, select them, and show them in stick representation. Display the hydrogen bond by selecting both residues and then (for Sele) clicking Action -> find -> polar contacts -> within selection.

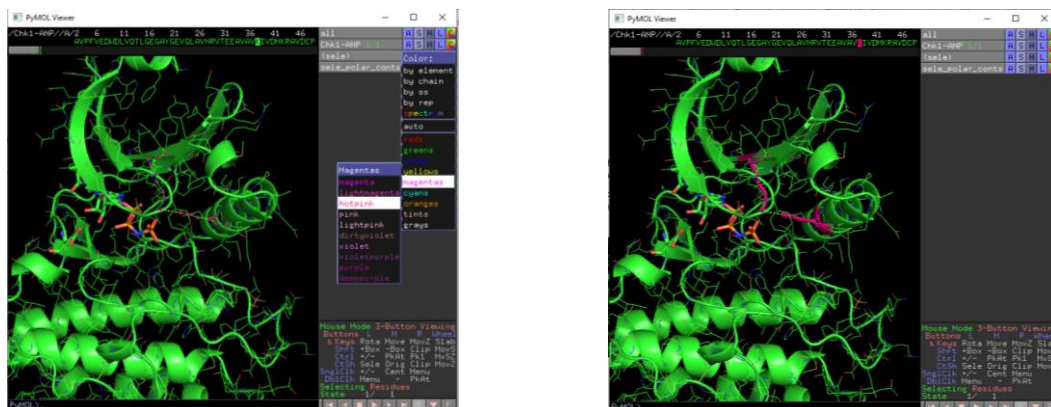
Question:

In this conserved salt bridge, which atoms (or groups) are making the interaction? Don't forget that you need to specify both the atoms and the residue.

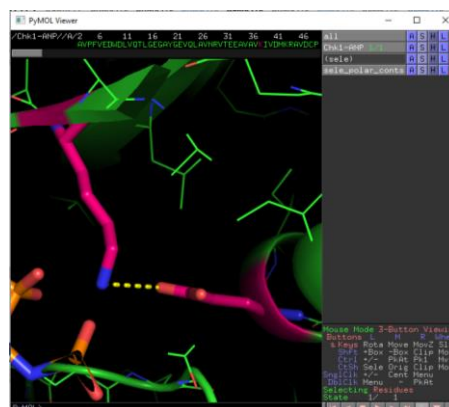
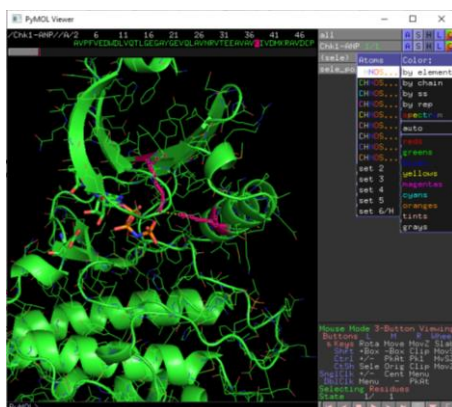
Changing colour

It can be very useful to change the colour of specific parts of a protein to highlight specific residues or ligands. For this exercise, we are going to change the colour of the residues in our conserved hydrogen bond.

Select the residues and show in stick representation. Next, go to the color menu for the selection and select a colour of your choice. I usually choose this particularly hideous pink (it stands out well), but you may well be more tasteful!

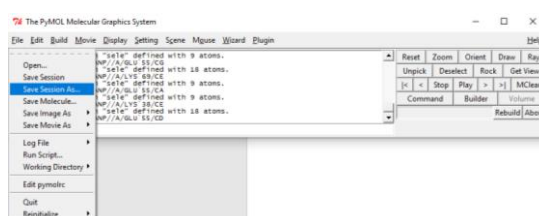


Notice that the entire residue has changed colour, and it is no longer possible to distinguish between carbon, oxygen and nitrogen atoms. In order to fix this, you should go back to the color menu, and select 'by element' and then the top option (H N O S...). This essentially tells the program to use the selected colour for carbon atoms, grey for hydrogen, blue for nitrogen, red for oxygen, orange for sulphur etc.



Saving your work

You have now done a reasonable amount of work on your structure and it would be a shame to lose it. In order to save your work go to File (in the control window) -> Save Session As... . This saves your entire session – the colour changes, the orientation of the molecules *etc.*



Adjusting the depth of view and changing the centre of rotation

By default, PyMOL has a very 'flat' depth of view, and nearly all the atoms of the protein are visible. Sometimes you might want to visualise a thin slice of your protein (perhaps to remove atoms in front – as I did in the right-hand picture of the salt bridge above).

To change the depth of view, scroll the middle mouse button. Scrolling in one direction will make the view shallower, scrolling in the other will increase depth.

Have a go at this to see what happens.

Sometimes the centre of rotation chosen by PyMOL is not where you would like. This can make it very difficult to move the molecule in an intuitive and helpful manner.

In order to change the centre of rotation to an atom of your choice, simply click on the atom with the middle mouse button. Notice that clicking with the central mouse button also resets the selected atom to the centre of the window, and makes it the central point for changes in depth of view.

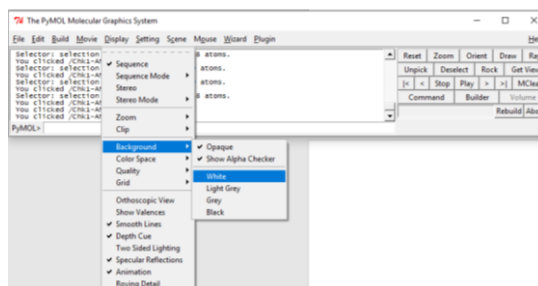
Have a go and see what this means in practice.

Making quality figures

PyMOL is used very widely to make figures of protein structures for publication. These are done using the enhanced graphics capabilities of the program, **not** by making screen shots (which are too low a resolution).

We are going to make a quality figure illustrating the conserved hydrogen bond between Lys38 and Glu55.

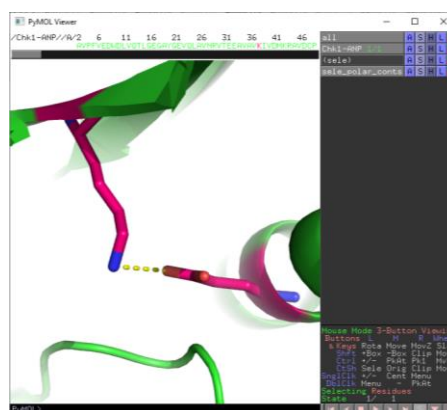
Firstly, we need to set the background colour to white. While a black background is used for day-to-day analysis (and was used historically to make figures when computers and graphics software were less advanced), in the 21st century we **always** have a white background in structural figures. To change the background colour go to Display (in the control window) -> Background -> White



Next, adjust the view in the display window until it is *exactly* how you want (zoom, depth of view, rotation *etc*). Do you want to hide any residues? Are the colours right?

The final figure will be the whole of the contents of this window, so you will need to change the size of the window if you want a different aspect ratio or want to remove white space at either side.

Finally, make sure that the display window is selected, and type *ray*. The text will appear at the bottom of the display window. When you press enter, you will notice that the image on your screen sharpens, and applies some lighting effects. This is called rendering, and essentially creates a high-quality image from the regular quality image used for standard viewing (the lower quality image uses less memory and is therefore faster for interactive viewing on-screen).

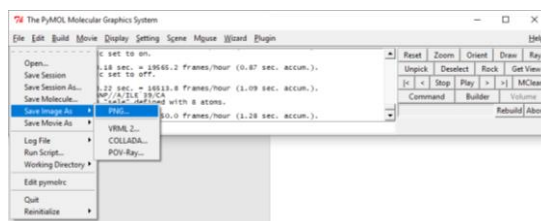


If you click anywhere in the display window or modify the view in any way, the image will revert to regular quality. If this happens, just type *ray* again.

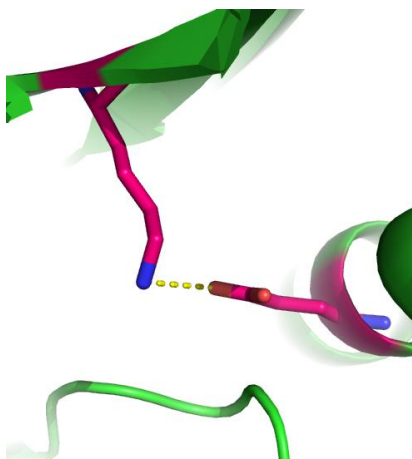
If you want to modify the lighting effects, you can do this using the options on the Display menu of the control window. Have a look at the different options and see what they do.

Saving an image

When you have a ray traced image which you are ready to save, go to the control window and select File -> Save Image As -> PNG...



This will save a high quality copy of the image in your display window as a png file. You can then insert this into powerpoint presentations, word documents etc.



You will notice that we haven't set the resolution (or size) of the output image. Using the default settings (which is what we have done) means that the final size of the file depends on how large the window is on our screen. Often this is sufficient (the image can be resized in standard graphics software).

However, we can set the width, height or both of our image by using the command:

ray width, height

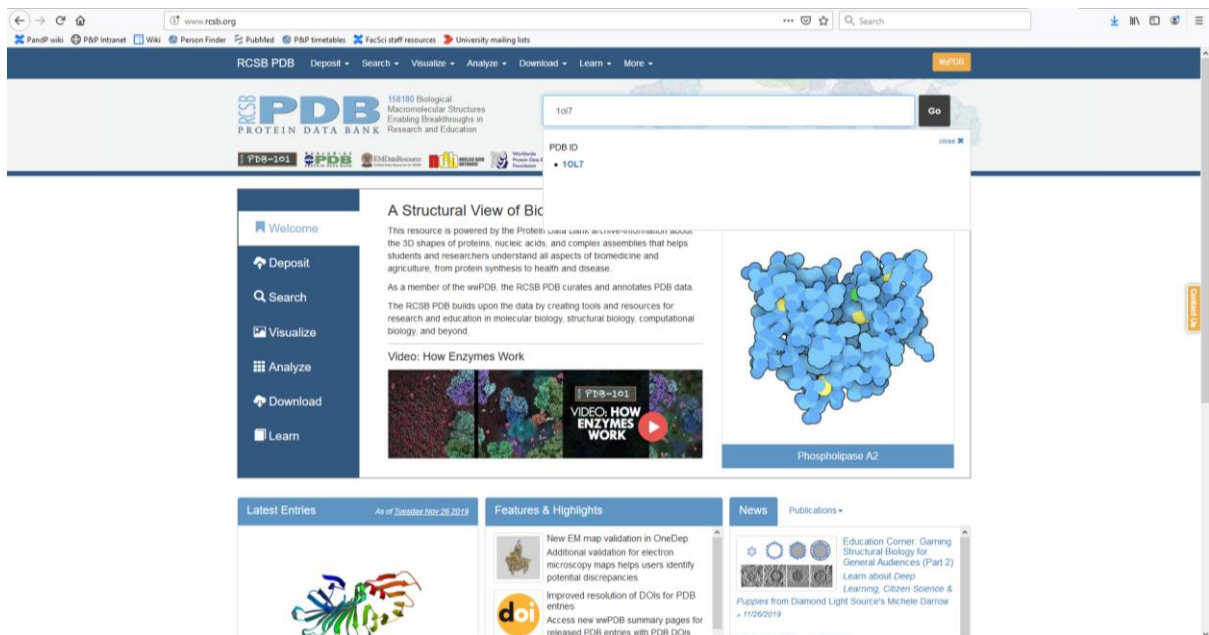
In this case, width and height are the size (in pixels) of the final image. If a value is left out for either width or height, PyMOL will automatically use a value that keeps the aspect ratio of the image the same as shown on screen.

Downloading a pdb file from the protein databank

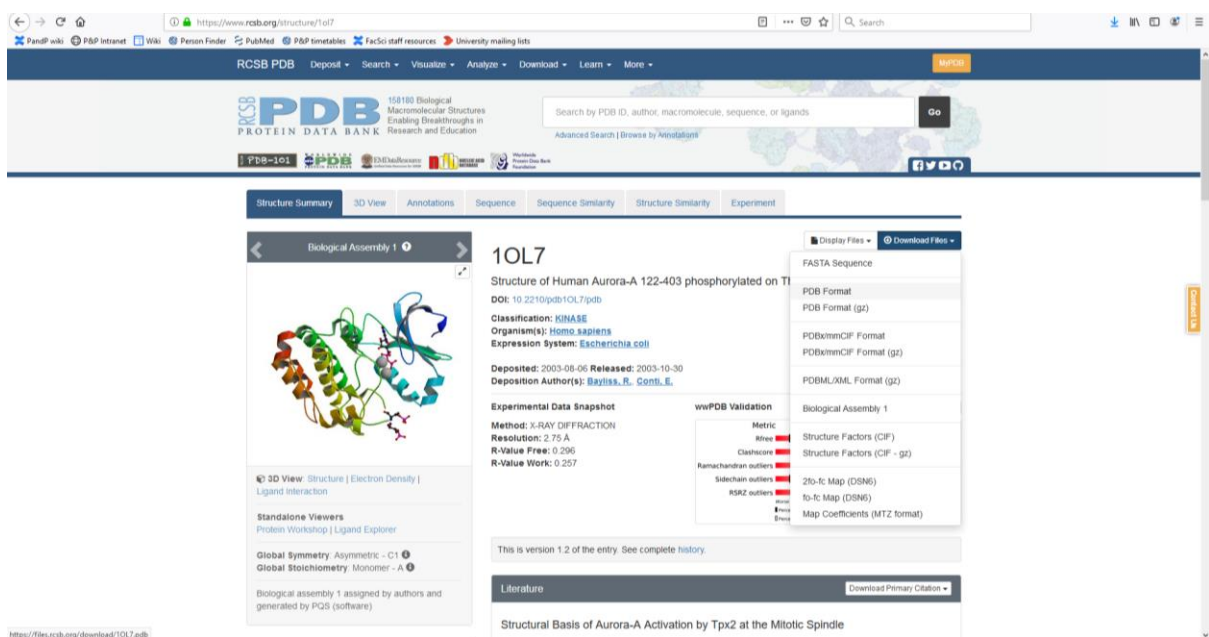
The pdb files for nearly every protein structure published in the literature are stored online in the protein databank (the PDB) and are freely available. Most journals make submitting the pdb file to the protein databank a condition of publishing the work. When a structure is accepted into the PDB, it is allocated a unique four-letter/number code. This means that anyone can look up a particular structure and know that they are looking at exactly the same 3D-coordinates as someone else.

We are going to download a different kinase structure (Aurora-A kinase) and overlay it (align it) with the Chk1 structure that we have open in PyMOL.

Open a browser window and go to www.rcsb.org. In the search window at the top of the web page, type 1ol7 (please note that this code includes a letter o, not a number zero; it also contains a lower case l, not an upper case I).



You will then get a page which gives lots of information on structure 1ol7. We are going to download the pdb coordinates, so click on Download files -> PDB Format.



Save the file somewhere you will be able to find it (eg on your H-drive, on the Desktop).

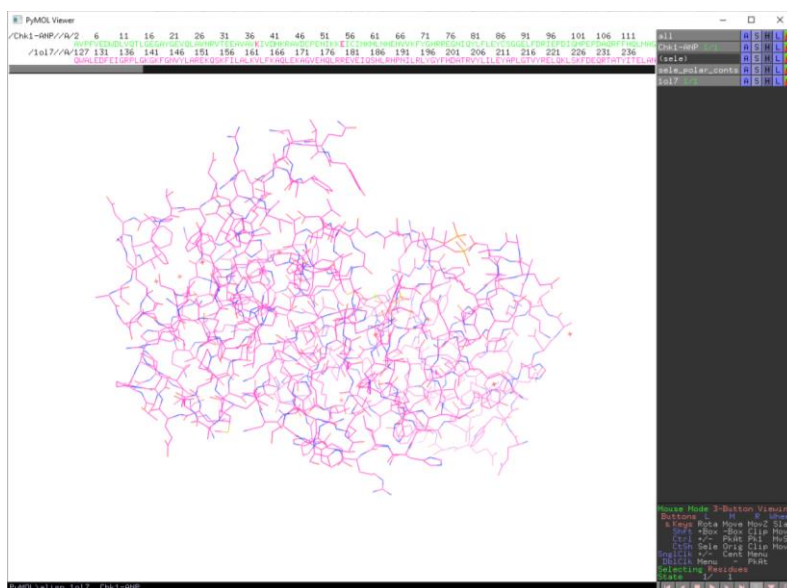
Opening a new file in PyMOL and aligning it

Return to PyMOL. Go to the control window and select File -> Open. Select the pdb file that you have just saved.

Notice that PyMOL has opened the new file and created a new entry for it on the menu on the right of the display window. However the old pdb is nowhere to be seen! This is because the two pdb files have different absolute positions for the molecules within them (although the relative 3D positions of the atoms are important, the absolute 3D coordinates for each atom can be anywhere within 3D space). To align the structures type:

align 1ol7, Chk1-ANP

This command tells PyMOL to find the best structural alignment between the two molecules, using 1ol7 as a template. Slightly unhelpfully in this case, it will move both molecules out of our field of view.



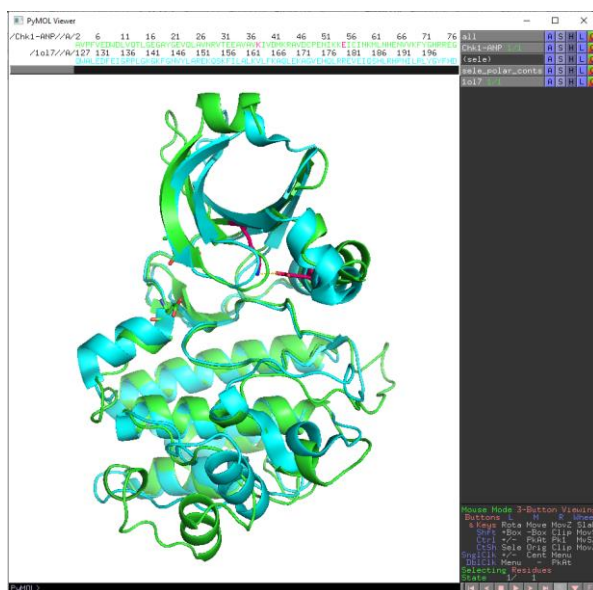
In order to bring both molecules back into the field of view, type *center* (note North American spelling).

Note: If you don't like typing the molecule names out in full, PyMOL has an autofill option. Start typing the molecule name and then press <tab>. PyMOL will fill in the molecule name as far as it can. If two molecule names start the same and then vary (eg 1ol7 and 1ol5), PyMOL will fill up to the point at which ambiguity starts (ie it will autofill 1ol but wait for you to finish with either 7 or 5).

Comparing two molecules

In this last part of the workshop, we are going to use our aligned structures to find the residues in Aurora-A kinase which form the conserved salt bridge.

Firstly, adjust the colour and representation of Aurora-A until you are comfortable with it (I find cartoon representation helpful).



Next, turn on lines representation for Aurora-A and zoom in on the region of the salt bridge. In order to find the residues structurally equivalent to Chk1 Lys38 and Glu55 you need to look at the Aurora-A structure and find the side-chains which overlay Chk1 Lys38 and Glu55 (or almost overlay these residues).

You may find it helpful to switch one of the structures off temporarily so that you can see the other structure more clearly. To do this, click on the structure name on the menu at the right size of the display window. Clicking on the structure name a second time will restore the structure to exactly how it was.

When you have found the residues you think are important, double click on them to find their identities. Confirm that a polar interaction might be possible by selecting both residues and asking PyMOL to find polar contacts: Action -> find -> polar contacts -> within selection

Question:

Which residues in Aurora-A form the conserved salt bridge between the $\beta 3$ strand and the αC helix?

Congratulations! You have now reached the end of the workshop. Please keep hold of this booklet – it is likely that you will find it useful in the virtual drug discovery exercise which will run throughout semester 2.

Installing PyMOL on your own machine

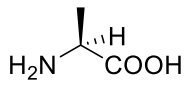
If you want to install PyMOL on your own machine, the education-only build is available (in 2022) at no cost for classroom use. Instructors should register at <https://PyMOL.org/edu/> to receive a suitable licence.

Once licence details have been received, PyMOL can be downloaded from <https://PyMOL.org/ep>.

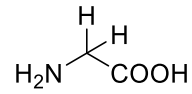
On the download page there are links to follow. One is to the software installers, one is the licence file. You will need to download the licence file and save it somewhere on your computer. When you run PyMOL for the first time it will ask you where it can find the licence file - in the absence of a licence, the software will run in demonstration mode for 30 days only.

Reminder of amino acid structures

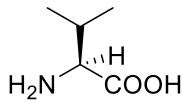
Amino acids with aliphatic side-chains



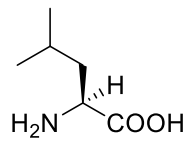
Alanine
(Ala; A)



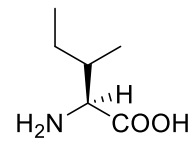
Glycine
(Gly; G)



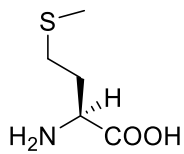
Valine
(Val; V)



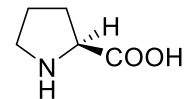
Leucine
(Leu; L)



Isoleucine
(Ile; I)

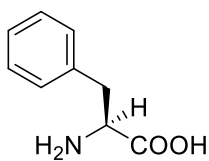


Methionine
(Met; M)

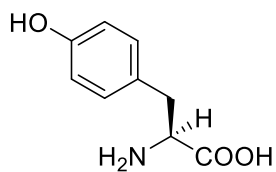


Proline
(Pro; P)

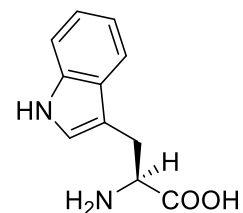
Amino acids with aromatic side-chains



Phenylalanine
(Phe; F)

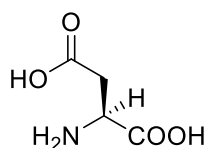


Tyrosine
(Tyr; Y)

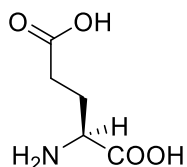


Tryptophan
(Trp; W)

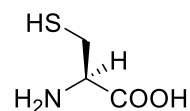
Amino acids with acidic side-chains



Aspartic acid
(Asp; D)
side-chain pK_a = 3.65

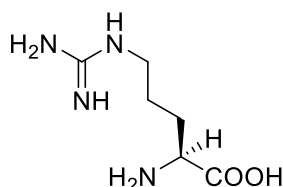


Glutamic acid
(Glu; E)
side-chain pK_a = 4.25

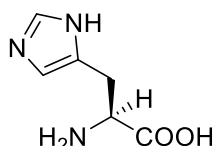


Cysteine
(Cys; C)
side-chain pK_a = 8.18

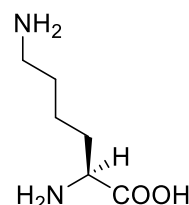
Amino acids with basic side-chains



Arginine
(Arg; R)
side-chain pK_a = 12.48

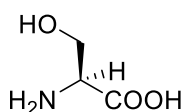


Histidine
(His; H)
side-chain pK_a = 6.00

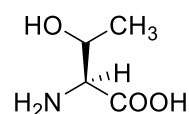


Lysine
(Lys; K)
side-chain pK_a = 8.95

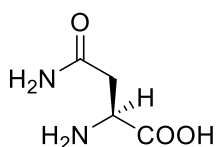
Amino acids with neutral polar side-chains



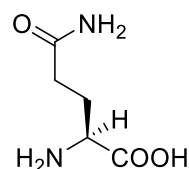
Serine
(Ser; S)



Threonine
(Thr; T)



Asparagine
(Asn; N)



Glutamine
(Gln; Q)

The material in this booklet was created using PyMOL 1.7.1.0. It is © Charlotte Dodson and is published under a CC-BY-SA 4.0 licence <https://creativecommons.org/licenses/by-sa/4.0/>.