

1 PyMOL Quick Reference

Below are commands and directions you may find useful to navigate PyMOL for this assignment. If you are interested in the full list of existing PyMOL commands, please see [PyMOL Command Reference](#).

1.1 Moving around the structure

Rotating, translating (sliding the structure sideways or up and down) and zooming are done with the mouse in the viewer window:

Rotate	Click and drag with the left click button.
Translate	Mac: hold command and click and drag with the left click button. Windows/Linux: Middle-click + drag.
Zoom	Click and drag up and down with the right click button.
Clipping planes	Scroll the mouse wheel to slice away the front and back of the scene. If your structure suddenly disappears or is cut in half, this is usually why. reset will bring it back.
Select an atom (pkAt)	Double-click on the atom with the right click button.

There are also many helpful command-line commands for manipulating the view and saving images:

<code>zoom</code>	Fit everything currently shown into the window.
<code>zoom <selection></code>	Fill the window with one selection, e.g. <code>zoom chain A</code> .
<code>center <selection></code>	Center the view on a selection without changing the zoom level.
<code>orient <selection></code>	Rotate so that the selection's longest axis lies across the screen. Usually the best single command for "show me this clearly".
<code>reset</code>	Return to the default view.
<code>png image.png, dpi=300, ray=1</code>	Save the current view as an image file, which is the cleanest way to produce the screenshots this assignment asks for.

1.2 Commands used in this assignment

Loading structures and managing your session

<code>cd <path></code>	Change PyMOL's working directory.
<code>load <file>.pdb</code>	Load a structure file from disk.
<code>fetch <PDB ID>, <name></code>	Download a structure from the PDB by its ID and give it a name. Needs an internet connection.
<code>reinit</code>	Clear everything and start over.
<code>save session.pse</code>	Save your whole session, objects and view included.
<code>load session.pse</code>	Reload a saved session.
<code>run analysis.py</code>	Run a Python script and add the commands it defines. Re-run this after every edit to the script.
<code>pip install matplotlib</code>	Install a Python package into PyMOL's own Python.
Visualization	
<code>show <representation>, <selection></code>	Show a representation for a selection.
<code>hide <representation>, <selection></code>	Hide a representation for a selection; everything works in place of a representation name.
representations used here	cartoon, licorice, lines, sticks, spheres, surface, wire.
<code>color <color>, <selection></code>	Color a selection. For example, <code>color red, ss h</code> colors all alpha helices red.
<code>util.cbc element C</code>	Color the carbon atoms of each chain a different color.
<code>h_add</code>	Add hydrogen atoms, which are missing from most PDB files.
<code>remove <selection></code>	Delete atoms from the session entirely, rather than just hiding them.
Selecting atoms	
<code>select <name>, <selection></code>	Create a named selection, for example <code>select bb, name c+n+ca</code> . sele is accepted as an abbreviation of select .
<code>resid <number></code>	Select by residue number.
<code>name <atom name></code>	Select by atom name.
<code><residue>/<atom></code>	Short form for one atom of one residue, e.g. <code>159/ca</code> .
<code>resn <residue name></code>	Select by residue name; resname is the long form.
<code>chain <chain name></code>	Select one chain.
<code>polymer.protein</code>	Select all protein; <code>polymer.nucleic</code> selects all DNA and RNA.
<code>bymolecule <selection></code>	Expand a selection to the whole molecule that each selected atom belongs to.
<code><sel1> beyond <n> of <sel2></code>	Everything in the first selection further than <code><n></code> Å from the second.
Measuring and comparing structures	

<code>get_dihedral <atom1>, <atom2>, <atom3>, <atom4></code>	Dihedral angle across four atoms, given in order.
<code>align protein1, protein2</code>	Superimpose one structure onto another.
<code>intra_fit <selection></code>	Align every frame of a trajectory to its first frame.
Commands you add yourself, from <code>analysis.py</code>	
<code>ramachandran <selection></code>	Ramachandran plot for a selection; you write this one in Exercise 1.
<code>plot_rmsd <reference>, <trajectory></code>	RMSD of each frame against a reference.
<code>plot_distance <trajectory>, <atom1>, <atom2></code>	Distance between two atoms over a trajectory.