

Tutorial on molecular visualisation

Biomolecular Simulation (2024–2025)

Master in Computational Biology and Bioinformatics

Contents

1	Introduction	2
1.1	How to do this tutorial	2
1.2	Purpose of this tutorial	2
1.3	Structural databases	2
2	Software	3
2.1	Installing PyMOL	3
2.2	Running PyMOL	3
3	Build and view small molecules	5
3.1	Building an organic molecule	5
3.2	Building a peptide	9
3.3	Molecular representations	9
4	View, mutate and compare proteins	10
4.1	Special molecular representations	10
4.2	Stereo images	11
4.3	Making point mutations	12
4.4	Comparing similar proteins	13

1 Introduction

1.1 How to do this tutorial

Follow these steps:

1. You were previously provided with a link to the tutorial's archive, or you can get it at <https://www.itqb.unl.pt/simulation/education>. Download and unpack the archive in your computer, which creates a folder named `visualisation-class` containing all the required material. We suggest that all files that you create during the tutorial are kept inside this folder.
2. Read this introduction (section 1).
3. Install (if needed) the software and get acquainted with it (section 2).
4. Do the practical parts of the tutorial (sections 3 and 4).

1.2 Purpose of this tutorial

This tutorial intends to teach you some very simple techniques of molecular visualisation, skills that can be useful to most people working in a (bio)chemical area. In this way, you will not depend on a Molecular Modelling specialist to perform these simple tasks.

The simpler skills you are expected to develop are viewing and manipulating molecular structures on the computer screen, and producing quality pictures for presentation or publication purposes. More sophisticated but often quite useful techniques are also introduced, namely building small molecules and making mutations in proteins. These are more advanced topics, but simple tasks can be easily done by plain users. If more detailed work turns out to be necessary, you can always ask a molecular modeller.

Although this tutorial is intended for a hands-on class, these notes can also be used as a short reference guide for self-study, until you move on to the program manuals (if that ever happens). Thus, we hope that you “play” with the software yourself and eventually find out additional features not discussed here, which can be useful to your work.

1.3 Structural databases

Although it may be easy to build small molecules (see section 3), quite often you need to obtain the structure of the molecule you are studying, either because your molecule is too big (e.g., a protein) or it is small but not obvious in terms of structure. In the present section we provide some very brief information on structural databases.

The place to look for protein structures is the Protein Data Bank (PDB), which can be freely accessed at <https://www.rcsb.org>. The PDB is easy to search and you can find there all the structures experimentally obtained so far (by x-ray crystallography, structural NMR, cryo-electron microscopy, etc). If you know the 4-character PDB code of the structure you



want, you can easily download it from the PDB and visualize it in your computer, as illustrated in section 4 of this tutorial. The format of these *PDB files* is described in detail at <https://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html>.

Many other structural databases exist, oriented towards specific subjects (small molecules, nucleic acids, ligand binding, etc) and featuring special search methods. Many of these databases are at least partially dependent on the PDB. If you think you need some specific database, you can start by looking at the PDB site, which has links to many databases.

2 Software

This tutorial uses the molecular graphics program PyMOL (<https://www.pymol.org>), which is a freely available and widely used program that we recommend.

2.1 Installing PyMOL

You may need to install PyMOL or not, depending on where you are doing this tutorial:

- If you are using the ITQB classroom desktop computers, they already have PyMOL installed. So, you don't need to install anything.
- If you want to install PyMOL in your own computer (because, for example, you are doing a tutorial remotely), go to its website (<https://www.pymol.org/installers>), download **version 2.2.3** for your operating system (Windows, MacOS or Linux) and install it.

Troubleshooting: If you use a version greater than 2.2.3 you will not be able to produce high-quality (ray-traced) images. Versions different from 2.2.3 can also have some commands in different menu locations, with different names, etc.

2.2 Running PyMOL

After PyMOL is installed, you can start/run it in either of two ways:

- The installation should have created a desktop icon to launch PyMOL. Double-click that icon.
- Alternatively, if you are using a Linux system, you can run PyMOL from the shell terminal. Enter the command `pymol` in the terminal.

When you start PyMOL, an activation window pops-up: just press Skip Activation. You can see that PyMOL's GUI (graphic user interface) consists of several regions, as illustrated in figure 1:

Menu bar This contains several pull-down menus (File, Edit, Build, Movie, etc) to access many actions and settings.



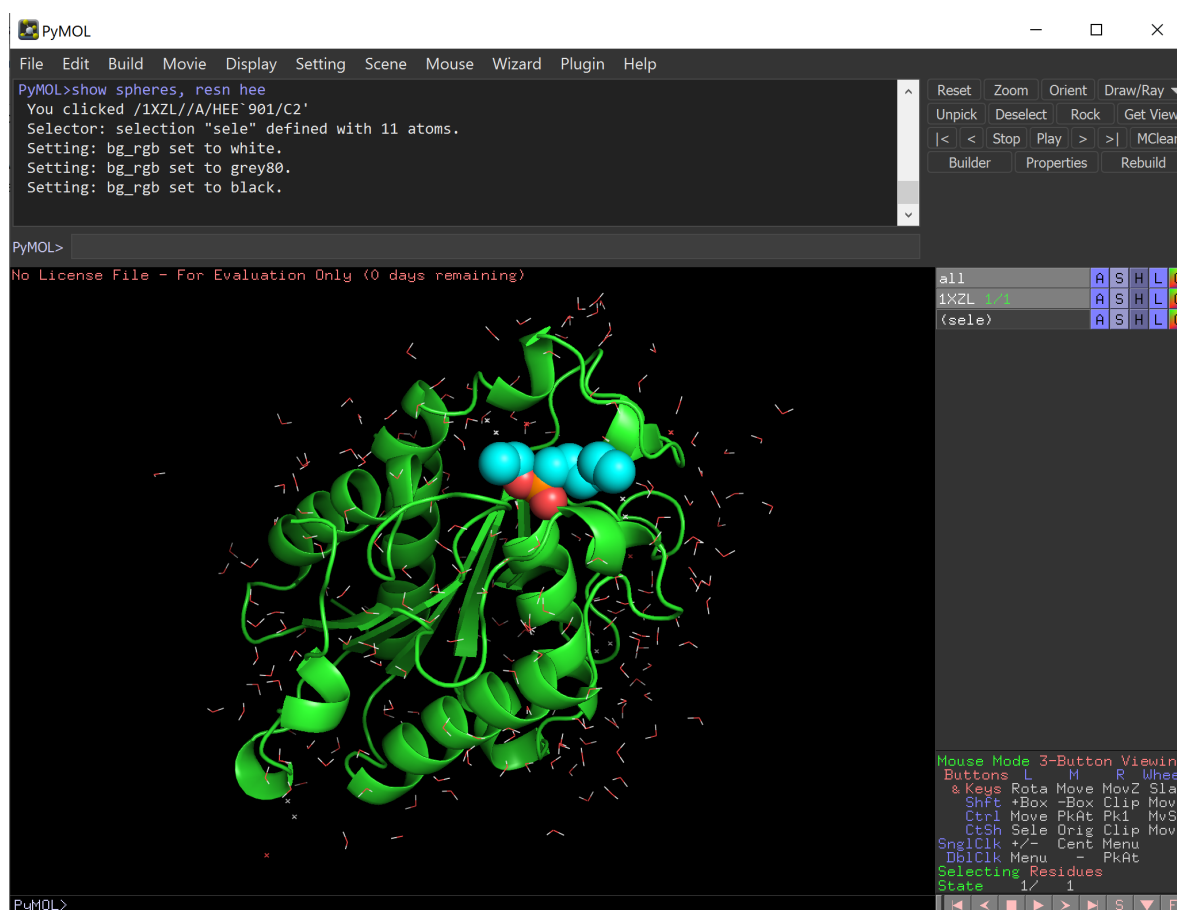


Figure 1: Snapshot of a PyMOL session.

Command area Located below the menu bar, this contains a command line (marked with the prompt PyMOL>) where you can write/enter commands and a region above it where command-related and other messages are displayed. You can use cut-and-paste in this area.

Button panel This is on the right of the command area and contains quick-access buttons for several frequent actions and commands (Reset, Zoom, Orient, Unpick, etc).

Visualisation area This is the larger region of the GUI, where molecules are displayed. It can also work as an alternative to the command area: at its bottom there is a command line (marked with the prompt PyMOL>) and the visualisation area can be turned into a message-displaying area by pressing Esc. Note that, in contrast to the command area, cut-and-paste does not work here.

Object menu panel On the right of the visualization area, this panel has one line for each of the currently defined objects/molecules, with menus where you can show (S) and hide (H) graphical features, label parts of the object (L), and apply other actions (A).



Mouse + movie controls This is below the object control panel and contains: (a) a mouse region to control different mouse modes; (b) a region for movie control, at bottom-right corner of the window.

Note that, in older PyMOL versions, the GUI may be split into two windows: (a) a small one containing the menu bar, the command area and the button panel; (b) a large one containing the rest.

In what follows some more-or-less obvious notations will be used to designate keyboard keys and mouse buttons. Thus Ctrl, Shift, Alt, etc., refer to these keyboard keys, while Left-Button, Middle-Button and Right-Button refer to these mouse buttons. A designation like Ctrl + Middle-Button means you should press first the Ctrl key on the keyboard and then the middle mouse button. PyMOL also supports many powerful written commands, but that is not explored in this simple tutorial.

Troubleshooting: This tutorial was initially written assuming that a 3-button mouse is used. In case you are using a 2-button mouse/trackpad, the corresponding commands are also indicated, but before starting the tutorial you need to go to the PyMOL menu bar and select Mouse → 2 Button Viewing.

3 Build and view small molecules

Although many molecular structures can be obtained from databases (see above), it is often necessary to look at a molecule for which there is no structure available, or whose building involves less work than finding it in a database. For these cases it is convenient to know how to build such molecules. The skills you are expected to develop with this building tutorial are very simple and aimed only at small molecules. The building of larger molecules has to deal with the fact that, as size increases, the molecule gains more flexibility (more torsion angles) and a structure build with the simple techniques described here becomes quite arbitrary: for each new torsion you have to make a decision. Much more sophisticated methods exist to model the structure of such large molecules, such as molecular dynamics and comparative modelling. The use of these specialised methods should be left to molecular modellers.

3.1 Building an organic molecule

In this example we will use the hormone 3, 5, 3', 5'-tetraiodothyronine, usually referred to as T4. This hormone is produced by the thyroid gland in response to the thyroid-stimulating hormone released by the pituitary gland. Thyroid malfunctioning can be detected by measuring the T4 levels in the blood. The structural formula of T4 is shown in figure 2. Note that the right-hand side is similar to a tyrosine amino-acid, and therefore we may use a tyrosine as a starting point to build the T4 molecule. This use of other molecules or fragments usually makes building much easier.



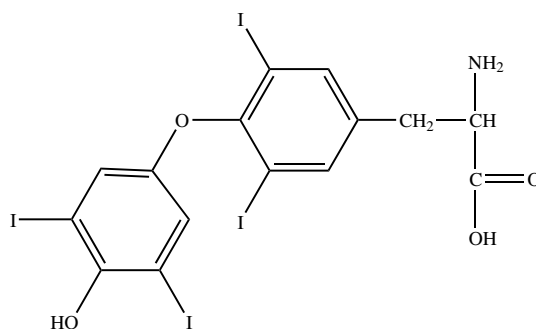


Figure 2: Structural formula of the T4 hormone.

Troubleshooting: To avoid mistakes, make sure to keep checking the structure shown in figure 2 as you build the molecule.

1. Open PyMOL.
2. Start with a tyrosine amino acid residue: on the menu bar select Build → Residue → Tyrosine. A tyrosine residue appears on the screen, using a standard convention for atom colours: green for carbon, white for hydrogen, blue for nitrogen, and red for oxygen. You can select other colour schemes at the “rainbow” squares at the upper right corner.

Troubleshooting: In some installations (Windows) the molecule may appear foggy or dark. In that case, you can recover the right lighting writing in the command line the command `set precomputed_lighting`.

3. Note that one of the carbon atoms comes with a ring-made sphere around it, meaning it is “picked” (see below). Press the Unpick button on the menu window to “unpick” everything.
4. Try using the mouse to move (Middle-Button), rotate (Left-Button) and zoom (Right-Button) the molecule. [2-button: move (Ctrl + Left-Button), rotate (Left-Button) and zoom (Right-Button)]
5. “Fill” valences in the tyrosine with hydrogen atoms: at the upper right corner, click (any mouse button) on the A in front of tyr and select hydrogens → add. This adds one hydrogen atom to the nitrogen at the N-terminus, and another to the carbon at the C-terminus.
6. Finish building the carboxyl group at the C-terminus.
 - a) Pick (Ctrl + Middle-Button) the hydrogen atom just added at the C-terminus. [2-button: pick with Left-Button]
 - b) Replace this hydrogen with an oxygen: on the menu bar select Build → Fragment → Oxygen. Note that the oxygen comes already “filled” with one hydrogen atom.



This finishes building the tyrosine amino acid.

7. Add a phenyl group at the hydroxyl group attached to the carbon ring.¹
 - a) Pick (Ctrl + Middle-Button) the hydrogen atom of the hydroxyl group. [2-button: pick with Left-Button]
 - b) In older versions, you could add the phenyl group using Build → Fragment → Phenyl, but now you won't find the phenyl group there anymore. Instead, you can open Builder in the button panel and click over the structure of benzene. Afterwards, you may close the Builder window.

Troubleshooting: Don't select cyclohexane by mistake!

8. You can recenter everything using (all) A → centre. This makes easier to control the motion of the molecule. Do this whenever you find it convenient.
9. Add a hydroxyl group at the tip of the new carbon ring.
 - a) Pick (Ctrl + Middle-Button) the hydrogen atom at the tip of the ring. [2-button: pick with Left-Button]
 - b) Add the hydroxyl group. Since added fragments come "filled" with hydrogen atoms, we just need to add one oxygen: on the menu bar select Build → Fragment → Oxygen.
10. Add iodine atoms. For each of the four hydrogen atoms to be replaced with iodine (see structural formula) do:
 - a) Pick (Ctrl + Middle-Button) the hydrogen atom. [2-button: pick with Left-Button]
 - b) Add iodine group: on the menu bar select Build → Fragment → Iodine.
11. The N- and C-terminal groups are neutral. However, in a solution of neutral pH they would tend both to be ionised (zwitterion). To change them do:
 - a) Pick (Ctrl + Middle-Button) the "filled" oxygen in the C-terminal carboxyl group. [2-button: pick with Left-Button]
 - b) Change the ionisation state: on the menu bar select Build → Make (pk1) Negative.
 - c) Correct the "filling" of the valences: on the menu bar select Build → Fill Hydrogens on (pk1).
 - d) Unpick everything (Unpick button on the menu window).
 - e) Pick (Ctrl + Middle-Button) the nitrogen in the N-terminal amino group. [2-button: pick with Left-Button]
 - f) Change the ionisation state: on the menu bar select Build → Make (pk1) Positive.

¹In older PyMOL versions you may get a different atom colours in the phenyl ring you just added, because a different colour is used for the carbons each time a new fragment is added. If you find that annoying, you can change to a single colour scheme at the rainbow squares at the upper right corner.



- g) Correct the “filling” of the valences: on the menu bar select Build → Fill Hydrogens on (pk1).
- h) Unpick everything (Unpick button on the menu window).
12. When building a molecule, some geometric properties may be set wrongly (i.e., the program is not “smart” enough). In the case of T4, the torsion around the final ring and the hydroxyl group is wrong, since the hydrogen atom should lie on the plane of the ring.² To fix this do:
- a) Switch the mouse mode from 3-Button Viewing to 3-Button Editing: click (with any mouse button) on the lower right panel (e.g., over Mouse Mode). [2-button: switch the mouse mode from 2-Button Viewing to 2-Button Editing]
 - b) Pick (Ctrl + Right-Button) the C–O bond. [2-button: same keys]
 - c) Grab the H atom with Ctrl + Left-Button and move the mouse to rotate around the bond, until the hydrogen atom lies on the ring plane. This is easier if you use the angle indication displayed on the screen each time a bond is selected. [2-button: same keys]
- Troubleshooting:** It is hard to select an exact value for the angle. Just select a value close to zero.
- d) Unpick everything (Unpick button on the menu window).
13. Following the same procedure, try changing other torsions. For example, you can change the torsion around one of the bonds connecting the two aromatic rings. This may be needed to make sure that the atoms of the two rings are not clashing on each other.
- Troubleshooting:** If you leave non-bonded atoms too close to each other and then save the molecule, PyMOL will create bonds between them when you load it again.
14. When you are satisfied with the final structure you can save it:
- a) On the menu bar select File → Export Molecule.
 - b) You will be asked for the selection to be saved. Since your T4 molecule is still called tyr, select that one.
 - c) If you want to save the bond orders (e.g., single, double, etc) in the PDB file, select the tab PDB Options and tick the two options related to CONECT records. In this way, bond orders will be represented next time you open this molecule in PyMOL.
 - d) After clicking Save, select file type PDB from the pull-down list and enter the file name T4.pdb. You can save the file inside the tutorial’s folder, to avoid losing track of it.
15. You are done with this exercise. Clean up things: at the upper right corner, click (any mouse button) on the A in front of (all) and select delete everything.

²Due to resonance with the benzene ring, the oxygen of this hydroxyl group has sp^2 hybridization, not sp^3 .



3.2 Building a peptide

To build peptides in PyMOL is quite simple. You just add the amino-acid residues in the order specified by the peptide sequence, specifying one of three standard secondary structures.

This exercise can be performed with any peptide you choose. An example of a very small peptide is kyotorphin (Tyr-Arg), a neuropeptide. An example of a large peptide is glucagon, a 29-residue hormone which stimulates the increase of glucose concentration in blood. For simplicity in building, you can use a peptide consisting of 10 alanine amino-acid residues. This can be regarded as a fragment of polyalanine, a well-studied example of helix–strand–coil transition, which adopts an α -helical conformation in a nonpolar environment and a β -type conformation in aqueous solution.

1. Build the peptide with uniform secondary structure:
 - a) On the menu bar select Build → Residue and choose one of the standard secondary structures allowed by PyMOL: Helix, Antiparallel Beta Sheet or Parallel Beta Sheet.
 - b) On the menu bar select Build → Residue and choose your first residue, Alanine (as shown in the menu, you can also press Alt-A). Repeat this step for all residues. Note that the chain is build with the specified secondary structure.

Usually, you would also fix the N- and C-terminal groups as done above. But you can ignore that step.

2. You can delete everything ((all) A → delete everything), select a different secondary structure, and rebuild the peptide. Note the different final conformation of the peptide.
3. You can rebuild once more the peptide, this time switching to different secondary structures during the building process. This allows mixing different types of secondary structure in the same peptide chain.
4. Save your peptide in a PDB file (e.g., Ala10.pdb), as done above, and clean up things ((all) A → delete everything).

3.3 Molecular representations

Different molecular representations can be very useful in identifying particular features in a molecule. A good setting can usually be obtained by combining suitable representations for bonds, atoms and surfaces. This is important both for screen visualisation and for production of figures.

1. Load the T4 molecule you have build above: on the menu bar select File → Open, and then choose the corresponding file (T4.pdb).
2. Use a good graphical quality: on the menu bar select Display → Quality → Maximum Quality. (If your computer has not a good GPU, this may cause move/rotate/zoom actions to become sluggish.)



3. The loaded molecule is drawn in the default way, as lines representing bonds. However, different representations can be used. To show or hide (S or H) each of them, go to the upper right corner, click (any mouse button) on the S or H in front of T4 and select one or more of the following:
 - a) lines or sticks to represent bonds.
 - b) dots or spheres to represent atoms.
 - c) mesh or surface to represent surfaces (obtained by rolling a sphere over the atoms).
4. When you are satisfied with the final representation, you can make a realistic “ray-traced” version of it: click the button Ray on the button panel. You can define the size, resolution and transparent background. Then, choose Ray (slow) and select Save Image to File. Select PNG format and choose the file name T4.png. This image can be included in most standard documents formats.

Troubleshooting: Note that if we move/rotate/zoom the molecule, you loose the ray-traced image (you have to hit Ray again).
5. If you think that the black background is not appropriate for your presentation or publication, you can switch to a different one: on the menu bar select Display → Background → White (or one of the other colours). You can then repeat the last steps.
6. Clean up things ((all) A → delete everything).

4 View, mutate and compare proteins

Protein molecules have special needs in terms of visualisation and modelling techniques. Here we discuss how to view some particular structural features and the qualitative analysis of mutation effects.

4.1 Special molecular representations

Due to their size and complexity, protein molecules often require special visual representations in order to highlight the features of interest. These representations will be illustrated with cutinase, a small serine esterase produced by some pathogenic fungi to hydrolyse cutin, a constituent of the cuticle that protects aerial plant organs.³

You can find the required files in the tutorial’s folder.

1. Load cutinase: on the menu bar select File → Open, and then choose cutinase.pdb.
2. You can use with cutinase the same general representation types used above for T4: lines, sticks, dots, spheres, mesh, and surface.

³Suggestion: if you want to try later a really large molecular system get the chaperonin GroEL (PDB code 1AON) or the large ribosomal subunit (PDB code 1JJ2).



3. You can show and hide (S and H) the main chain and the side chains of the protein, for any of the representations.
4. You can also use some specific representations that apply only to proteins (best seen after hiding all other representations):
 - a) The ribbon representation displays a smooth line following the protein main chain (i.e., the chain formed by the peptide bonds).
 - b) After the assignment of the protein secondary structure has been done (cutinase A → assign sec. struc.)⁴, the cartoon representation draws a schematic representation of the molecule.
5. If you want to make very fancy images, you may want to explore the options Cartoon, Transparency and Rendering under the Setting item at the menu bar. You can easily spend hours tweaking the different combinations, so don't mess with this unless you really need it. The default settings are perfectly good for most cases.
6. Your final image can be ray-traced and saved, as done above.

4.2 Stereo images

Stereo images are quite helpful in visualising the structure of proteins and other large molecules, and are becoming quite common in the literature. They consist of two slightly different images that your brain may perceive as a three-dimensional object. There are four modes of viewing stereo images:

Anaglyph To be used with the type of “red-and-green” glasses used for some 3D movies. These glasses are very cheap, but the stereo quality is not very good and the colours are somewhat distorted.

Cross-eye You cross your eyes so as to put the two images on the top of each other. (Hint: if you move a finger or pen from the screen towards you, the focusing will be correct when it is 10–15 cm from your eyes.)

Wall-eye (or Wide-eye or Parallel-eye) You look “through” the image, as if you were focusing a distant point. (Hint: avoid each eye to see the image at the other side by putting a sheet of paper perpendicular to the screen, from your nose to the screen.)

Quad-buffered and Zalman These modes require special hardware equipment (e.g., glasses, graphics card, monitor).

Seeing stereo images without using special glasses or lenses requires some training. If you can see decorative stereograms, you should be able to see stereo in wall-eye mode. Some people can see in both wall-eye and cross-eye modes, others in just one of them, and some are never able to do it.

⁴Actually, the assignment of the secondary structure is usually done within the PDB file, and rarely needs to be done with PyMOL.



1. To activate stereo viewing go to the menu bar and select Display → Stereo, which displays a cross-eye stereo image of the molecule.
2. To switch to wall-eye stereo, go to the menu bar and select Display → Stereo Mode → Wall-Eye Stereo.
3. As done above, you can ray-trace this stereo view and save it to an image file.

4.3 Making point mutations

It is often important to have an idea of the structural effect of a point mutation in a protein. Specialised modelling methods exist to allow relaxation of the structure around the mutated residue, but these require more advanced methods. Fortunately, useful qualitative information can often be obtained by simple replacement of the residue and a visual analysis of its structural effect. This can be very useful in planning mutation experiments.

This example uses again the structure of cutinase, which was actually obtained with an inhibitor (N-hexylphosphonate-2-ethyl-ester) that is an analogue of the substrate and remains covalently bound at the active site. Since enzyme structures are difficult to obtain with a bound substrate (because the enzyme tends to transform it), locked analogue inhibitors are often very useful to infer substrate-related structural features. The purpose here is to mutate a tyrosine residue near the active site of cutinase, trying to make the enzyme accommodate a larger substrate.

1. In addition to cutinase, load also the inhibitor from the file inhibitor.pdb, which allows you to easily locate the active site of cutinase (hint: use different representations and/or colours for cutinase and the inhibitor).
2. Identify the tyrosine residue near the active site of cutinase. If you have trouble finding it, you can try the following:
 - a) Hide the main chain in order to more easily visualise the side chains.
 - b) Display the protein sequence by selecting Display → Sequence on the menu bar. Now you can select the tyrosine residues by clicking over them in the sequence bar. Remember that Y is the one-letter code for tyrosine.
 - c) If you still have trouble deciding which tyrosine seems closer to the inhibitor, measure the distances between them. Select Wizard → Measurement on the menu bar and you will be asked to pick the atoms that you want to measure the distance between. To finish, click over Done.
 - d) If you still cannot find that tyrosine, click on residue 119 in the sequence bar.
3. On the menu bar select Wizard → Mutagenesis.
4. Pick (Ctrl + Middle-Button) the tyrosine residue. A white copy of the residue appears with a different conformation of the amino-acid side-chain (see footnote 5) and surrounded by red disks; ignore it.



5. Click under the green header Mutagenesis (which shows No Mutation), select an alanine (ALA) and click Apply and Done. This replaces the tyrosine with an alanine (whose carbon atoms are now displayed in white).⁵
6. Save your mutant protein molecule as mutant.pdb, and then delete it (cutinase A → delete object).
7. Load both cutinase.pdb and mutant.pdb, and represent them both as surfaces.
8. If you now click over cutinase at the upper right corner, this will switch off the display of the wild-type cutinase, showing only the mutant you just made. By clicking again over cutinase, it is again displayed. In this way you can effectively switch between the wild-type and the mutant, showing the large cavity created by the mutation.
9. Select an orientation that clearly displays the difference between the wild-type and the mutant. Hide the mutant and save a ray-traced image of the wild-type. Hide the wild-type and show the mutant, and save also a ray-traced image of it (keeping the same orientation). These images could be used to illustrate the (approximate) structural effect of the mutation.
10. Clean up things ((all) A → delete everything).

4.4 Comparing similar proteins

It is often important to make a visual comparison of proteins that are similar in terms of structure and/or function, because it may help you to rationalize about their structure–function determinants. There are specialised modelling methods to do this, but a simple visualization can be done using PyMOL, at least for proteins sharing a reasonable amount of homology.

This example uses again cutinase. The structure you used in the previous sections belongs to *Nectria haematococca* Mpvi⁶ and corresponds to the PDB entry 1XZL. The purpose of this section is that you compare the 1XZL structure with two other cutinase structures, one from *Glomerella cingulata* (PDB code 3DCN) and another from *Aspergillus oryzae* (PDB code 3GBS).

Since the entry 1XZL has both the protein and inhibitor structures, they were split beforehand for the previous sections of the tutorial. But, to better simulate a real-life situation, here you will load the complete structures (including 1XZL) directly from the PDB.

1. Load the 1XZL cutinase structure from the PDB: on the menu bar select Plugin → Legacy Plugins → PDB Loader Service and then enter the code 1XZL.

⁵If you choose an amino-acid besides alanine, several side chain orientations (rotamers) are possible. You can use the back-and-forth movie controls (lower right corner) to display (in white) each of the rotamers available for this residue in PyMOL, whose current and total numbers are shown in the (green) Frame info. The rotamers are ordered according to their frequencies of occurrence in proteins, shown as a green percentage at the mutation object, which exists while mutagenesis is being performed. After selecting the rotamer you think better fits your structure, proceed to Apply and Done the mutation, as above.

⁶Note that this organism was previously called *Fusarium solani* pisi.



2. Load the 3DCN structure from the PDB in a similar way and then enter the following command in one of PyMOL's windows:

align (3DCN and name ca), 1XZL

This makes a sequence and structural alignment of 3DCN onto 1XZL. The structural alignment is done using the C_α atoms of the peptide chain ("name ca").

Troubleshooting: After the alignment, you may get an empty window. This happens when the two structures are initially at different positions in space and the one used as target was already off-screen. Just center them all using (all) A → centre.

3. Repeat the previous step for the 3GBS structure. The three structures are now aligned. To facilitate their comparison you can start by removing all water molecules and hydrogen atoms (see the A options).

Troubleshooting: Make sure to use the same target, aligning 3GBS onto 1XZL.

4. Compare the structure of the main chain of the three molecules using a main-chain representation and find the major structural differences between them. Try both ribbon and cartoon, since the "beautification" done by cartoon can be misleading, and make sure you selected maximum quality in the Display menu.
5. Locate the active site (you may use the inhibitor to do that).
6. To compare the overall shape of the molecules you can switch to an all-atom representation (such as lines or surface).

