A new installment

This chapter is part of a comprehensive manual-in-progress, so you will find references to chapters that are not included here. Nonetheless, we think you may find it helpful, especially if you are a new user. Please email help@schrodingerc.com if you find it in any way confusing or incomplete.

Only for sponsors

This is an incentive product for PyMOL sponsors. Please do not post it publicly or otherwise share it with the general public. Incentive products, such as this manual, are exclusively for sponsors, and sponsors are what make possible PyMOL’s continued development, documentation, and support. To confirm or inquire about sponsorship, please email sales@pymol.org.
Y. COMPLETING THE TOUR

This chapter introduces a few more essential PyMOL features. The information here corresponds to PyMOL version 1.2.B5.

Typing Commands 4

Why type commands? 4
Simple scripts 7

Measuring Distances and Angles 9

Comparing Multiple Structures 12

Aligning Protein Structures 13

Alignment based on full protein sequence 13
Alignment based on specific atomic pairs using the Pair Fitting Wizard 17
Typing the pair_fit command 19

Changing Protein Structures 20

Mutagenesis 20
Removing atoms 22

Making Simple Movies 23

Rock & roll 23
Saving QuickTime movies with MacPyMOL 25
Saving movie images on other platforms 25

Appendix A 26

Atom selection macros 26
Why type commands?

In Chapter X, we demonstrated that typed commands can replace certain mouse operations. In fact, typed commands can do everything that mouse operations can, and many PyMOL users prefer typing to pointing and clicking. Once you’re familiar with PyMOL, it can be more efficient to keep your hands on the keyboard. However, mastery of typed commands does require the knowledge of PyMOL syntax, which is the subject of Chapters TC and ES, and the online references at http://pymol.org/dsc/dokuwiki/doku.php.

For now, learning a few typed selection terms will make it much easier to pair-fit sparsely related proteins, to make movies, and to perform other tasks described in this chapter. Also, we’ll see how selection terms and commands can be readily collected in scripts that remove the tedium of repetitive tasks.

Choose your syntax

Like everything else in PyMOL, there is more than one way to type commands for making selections. This section provides the long form of PyMOL syntax. It consists of “natural language” terms, designed to be naturally easy to learn. Advanced users prefer the more compact syntax of PyMOL selection macros. As the name implies, the macros are concisely coded, and they are very efficient. You may see the advantage of learning to use macros from the outset, so we provide Appendix A at the end of this chapter, to get you started with them.

Long form atom selection syntax

Amino acid residues in proteins are generally identified by their sequence number, from residue 1 at the N-terminus to residue n at the C-terminus, where n is the number of residues in the chain. (Don’t take it for granted, however, that the numbering for every chain you load starts at 1. Check the numbering using the Sequence Viewer.) A PyMOL selection language term for a residue identifier is resi. The simplest typed selection command is
PyMOL> select resi 100

This selects any residue identified as 100. If more than one object containing a residue 100 is active when you type the command, all the residues numbered 100 will be selected. This includes water molecules numbered 100, if there are any. To be more specific, exclude some of these possibilities with the Boolean AND. It narrows the selection to atoms that fulfill a selection criterion AND another selection criterion, AND, if desired, another selection criterion, etc.

For example, to avoid selecting water molecules with resi 100, use the selection term polymer, which selects atoms with given amino or nucleic acid residue identifier numbers, using the Boolean AND,

PyMOL> select polymer and resi 100

To be selected by the Boolean AND, the atoms must both belong to a polymer and be identified as 100. You can type and or AND. Like most PyMOL terms, it is not case-sensitive.

If you want to select residue 100 from only a given chain, say chain a, use the selection term chain and the Boolean AND to select it,

PyMOL> select resi 100 and chain a

The chain identifier a is sometimes case sensitive, depending on how the coordinate file was written. If you need to exclude water molecules, another Boolean AND narrows the selection,

PyMOL> select polymer and resi 100 and chain a

In Chapter X, we demonstrated that named selections are useful. For a selection named (my_resid), including all residues with identifier 100, type

PyMOL> select my_resid, resi 100

Don’t type the parentheses. They appear automatically in the objects & selections list. For a selection named (my_single_resid), including residues with identifier 100 in chain a, type

PyMOL> select my_single_resid, resi 100 and chain a

To select a range of residue numbers in all chains and name it my_range, type

PyMOL> select my_range, resi 100-110
For a selection restricted to chain a and the same range of residue numbers, type

PyMOL> select my_range_a, resi 100-110 and chain a

You can combine ranges and single residue identifiers, as in

PyMOL> select my_collection, resi 100-110+90+82

*Note that there are no spaces in the residue specification 100-110+90+82.*

You can combine ranges and single residue identifiers, and restrict the selection to a given chain as well, as in

PyMOL> select my_collection, resi 100-110+90+82 and chain a

PyMOL uses the three letter amino acid code to name residues with the selection term resn. For example, to select all the tyrosines in all the loaded objects,

PyMOL> select my_tyrs, resn tyr

Use plus signs to collect more than one residue type in your selection,

PyMOL> select my_negs, resn glu+asp

PyMOL uses the IUPAC atom names, so you can collect main chain atoms, as in

PyMOL> select my_main, name n+ca+c+o

The atom names are not unique. For example, in addition to main chain atoms, name n+ca+c+o may select waters given as oxygens, o, and calcium ions given as ca in the PDB file.

To avoid selecting solvent or ions with name n+ca+c+o, include the selection term polymer, which refers to atoms with amino or nucleic acid residue identifiers.

PyMOL> select my_main, polymer and name n+ca+c+o

The plus operator works for collecting chains as well as atom names. For a data set that includes chains a, b, c, d, e, etc., you can choose a group of chains, as in

PyMOL> select my_group, chain b+d+e

Object and selection names can be used in selections. If you have two active protein objects, 1uwh and 1t46, you can select the negative residues in 1uwh alone by typing

PyMOL> select 1uwh_negs, resn glu+asp and 1uwh
To select atoms that already belong to a named selection, such as `my_group`, type

```
PyMOL> select my_group_alas, my_group and resn ala
```

**Simple scripts**

Scripts, which are plain text files containing a sequence of PyMOL commands, can be used to perform repetitive tasks automatically. For example, it may make sense to use a script to prepare a series of standardized figures, using a different PDB file for each figure, but giving them all the same color scheme, using `spheres` to represent all the ligands, and `cartoon` to represent all the proteins.

Standard .doc, .rtf and other file types produced by word-processors generally won’t work for programming tasks, including PyMOL scripting. For a file to function as a PyMOL script, the file name extension must be .pml, and the file format must be plain text.

On a Mac, use the TextEdit utility program for preparing scripts. Under the TextEdit Format menu, select ‘Make Plain Text,’ and type your script. When saving the script, un-check the ‘Hide Extension’ and the ‘If no extension is provided, use “.txt”’ boxes, and type the filename with the extension ‘.pml’ in the File Save dialog box.

On a Windows PC, use the Notepad program to prepare the script. When saving for the first time, use the Save As option and choose ‘All Files (*.*)’ in the ‘type’ field. Type in the .pml file name extension when entering the file name.

The following example script contains five PyMOL commands to color and represent proteins and ligands in a standard way. For each PDB file, `fetch` or `load` the PDB file, then run the script. Before loading another PDB file, you may want to choose File / re-initialize, or simply disable the objects you no longer need to view. To run a script, choose Run... from PyMOL’s File menu and navigate to choose the script file, or type @ followed by the file name path in the command line.

```
# show the ligands as spheres, and show the protein as a cartoon

hide everything
show spheres, organic
show cartoon, polymer
```
# color the protein carbons cyan
# and the ligand carbons yellow

color cyan, polymer and elem c
color yellow, organic and elem c

The lines that begin with hash marks, #, are comments to aid people reading the script, and are skipped by the PyMOL program. The selection term organic selects carbon containing molecules that PyMOL doesn’t recognize as polymer; and elem refers to the chemical symbols of the elements.

In addition to PyMOL language scripts, PyMOL can be controlled by commands in the Python programming language. Python scripts and programs are used to create custom applications integrating PyMOL. Integrating Python commands with PyMOL in scripts is the topic of Chapter ZZ.
Measuring Distances and Angles

PyMOL measures the distances, bond angles, and dihedral angles between atoms in molecular structures. The easiest way to measure is to choose the Measurement Wizard from the Wizard menu (see Figure Y.1).

The Measurement Wizard defaults to measuring the Distances between two atoms. To measure a distance, follow the prompts in the Display Area: left-click with the cursor on the first atom, and then the second. A dashed yellow line appears, connecting the atoms, and the number of Ångstroms between the atom centers is given near the line. The default object name of the measurement, object01, appears in the list of objects & selections.

Figure Y.1 Sequential menus and prompts appear here in one figure. When you choose the Measurement Wizard from the Wizard menu, it comes up below the list of objects & selections. A prompt to (left-) click on the first atom in the measurement appears in the Display Area. The Measurement Wizard includes two blue pop-up menus, Distances and Create New Object. Here, Distances has been chosen, and the menu cascades to show the Measurement Mode choices.
tions, and you are prompted to choose another first atom. By default, each new measurement is loaded into an object of its own, and the individual object names appear in the list. You can change the names by the usual procedure, choosing rename from the corresponding A Action menu.

You can add a completed measurement to the most recently created object by choosing Merge With Previous from the cascading Create New Object menu. Now Merge With Previous will show in the Wizard until you choose Create New Object again.

Figure Y.2. Angle, dihedral, and distance measurements were each created as new objects so they could be individually colored (using the corresponding C Color pop-up menus). They were renamed angle, dihedral, and distance for future reference. The PDB coordinate file is 1oky.

To measure a bond angle, choose Distances / Angles from the Measurement wizard. Then left-click with the cursor above each of the three atoms that define the bond angle. To measure a dihedral angle, choose Dihedrals and left-click on four atoms.
The next three choices in the Distances pop-up menu are Polar Neighbors, Heavy Neighbors, and Neighbors (Figure Y.1). A neighbor is defined as any atom with its center within 3.5 Å of the center of the first atom, except if it is joined to the first atom by four or fewer bonds. This definition eliminates members in the same ring of six atoms or fewer. A heavy neighbor fulfills the same requirements, and it cannot be a hydrogen.

![Image](image.png)

**Figure Y.3** Using the neighbors and polar neighbors measurement modes, the Measurement Wizard added dashed lines between the central water and all its neighbors. The object names were changed to identify the measurements. Labels were removed from all_neighbors using the H hide pop-up menu, so distances only of polar_neighbors are shown.

A polar neighbor must be within a 3.5 Å radius, and must be a polar atom, positive or negative. Figure Y.3 shows the neighbors and polar neighbors of a water inside the kinase 1oky.

If you make a mistake in a measurement, left-click on the Delete Last Object button in the Wizard to delete the erroneous object. You can also delete all measurements by left-clicking Delete All Measurements.

One common use of the Measurement Wizard is to create a dashed bond between two atoms without any label for the distance value. To eliminate the label from a measurement, left-click on the H Hide menu for the measurement object and choose labels. When you are done, left-click the Done button to close the wizard.
Comparing Multiple Structures

Multiple structures can be opened sequentially using PyMOL’s File menu. However, this can be tedious when there are a lot of files to load. Alternatively, on Macs and Windows PCs, multiple PDB files can be dropped directly onto the PyMOL icon to open them together at launch. Also, the `fetch` command can be used to load one or more structures directly from the protein data bank using PDB identifiers. For example, type

```
PyMOL> fetch 1t46 1oky 1z5m
```

to load the three structures. PyMOL will display them side by side, each in its own coordinate space, and each with its own solvent molecules. You will have to zoom out to see that all the structures are there. The structures are represented by lines colored by element, with a different color for carbon in each structure. The object names for each file appear in the objects and selections list. To see which display is which object, toggle the object names out of activity and back by left-clicking on them.

![Figure Y.4](image)

Figure Y.4 PDB files 1t46, 1oky, and 1z5m contain related protein kinases with ligands bound. When multiple files are loaded, PyMOL displays them next to one another, but not overlapping, until the user gives commands to superimpose them.

The first steps in structural comparison are aligning the sequences and superimposing the structures on a common template. PyMOL can align proteins for superposition based on entire protein sequences, based on selections, or, using the **Pair Fitting Wizard**, based on clicked atom pairs.
Aligning Protein Structures

Alignment based on full protein sequence

The **A** Action menu next to the object name of a structure provides a number of ways to produce protein alignments. Figure Y.5 shows the *A* Actions menu next to **1t46**, cascading to display the choice of **align / to molecule / 1oky**. This command produces an alignment of **1t46** to the template **1oky**.

Other sequence alignments can be chosen from this menu cascade, including the alignment of **1t46** to the **1z5m** template, **enabled to this**, and **all to this**. Because this *A* action menu belongs to **1t46**, **enabled to this** and **all to this** use **1t46** as the template. For a pair of proteins, it doesn’t matter which one is the template. For wider comparisons, different variations occur, and it does matter.

When you choose an alignment, PyMOL produces an alignment object, adds the object to the objects & selections list, reports the RMSD of the alignment in the command history window, and superposes the structures in the Display Area. The alignment object is displayed in two ways: yellow lines connect the C-alphas of the structures, and unaligned sequence regions are grayed out in the Sequence Viewer (Figure Y.6). Spaces are added to line up the matching regions of the sequence.

Clicking on the alignment object name in the object & selections area toggles it from enabled (displayed) to disabled. If you want to keep the sequence view of the alignment, but hide the yellow lines in the Display Area, choose cgo from the H Hide menu next to the alignment object.

PyMOL has its own dynamic sequence alignment routine, which aims to reproduce the BLAST alignment. To superimpose the aligned chains, PyMOL uses the Kabsch least squares approach. PyMOL minimizes only the distances between the aligned C-alpha atoms, except when alignments typed commands specify the alignment differently.
Figure Y.6. 1z5m, 1oky and 1t46 were loaded using the fetch command, and the sequence viewer was turned on by choosing it from the Display menu. Here, 1oky is disabled and 1t46 is aligned to 1z5m. Proteins are shown as Ribbon, with ribbon_width set to 5.0 and cgo_line_radius set to 0.5. Settings such as these are discussed in Chapter TC.

Figure Y.7 shows an application of alignment to compare binding of Phosphoinositide-Dependent Kinase 1 (PDK1) to Staurosporine (in PDB file 1oky), and a to novel inhibitor (in PDB file 1z5m). Here, the complete coordinate sets were aligned, and the alignment object was disabled, which does not move the structures out of alignment but only removes the display of the cgo lines that highlight the differences in the C-alpha positions.
Figure Y.7 Starting with the same PDB files as in Figure Y.6, 1t46 was disabled. Then PyMOL was given the following commands:

Choose 1oky / C Color / by element with green carbons.
Choose 1z5m / C Color / by element with cyan carbons.
Choose 1oky / A Action / align/enabled to this.

Choose 1z5m / A Action / preset / ligands.
Disable 1z5m_pol_conts.
Choose all / H Hide / ribbon.
Choose all / H Hide / waters.

The image was adjusted in the Display Area by mouse navigation and clipping.

Alignment based on selections

Finer control over the alignment process can be exercised by creating selections that belong to the two regions to be aligned. One selection (sel01) is made from the template molecule, and a second independent selection (sel02) is made from the molecule to be aligned (see Figure Y.8). The alignment and superposition are made by choosing align / to selection / sel01 in the A Action menu of (sel02).

Figure Y.8 For this alignment, the green 1t46 sequence region in the sequence viewer, including the gray F, was selected and named (sel01); the corresponding 1oky region in cyan was named (sel02), and the selections were aligned. 1t46 and 1oky are shown as ribbon, and the selections are shown as cartoon.

The finest control over global alignment can be exercised by typing an align command in the command line. To see this, fetch the kinase structures 1uwh and 1t46. The PDB
file 1uwh contains two enzyme structures, chains A and B. The file for 1t46 contains only one. The default alignment between 1t46 and 1uwh matches only the B (or b, the PDB file is not set up to be case dependent) chain of 1uwh. To force alignment of 1t46 against the A chain of 1uwh, type

```
PyMOL> align 1t46 and name ca, 1uwh and chain A and name ca
```

If you prefer to use selection macros (Appendix A), type

```
PyMOL> align 1t46///CA, 1uwh//A//CA
```

The results of the default alignment to chain B and the alignment to chain A of 1uwh are shown in Figure Y.9.
Alignment based on specific atomic pairs using the Pair Fitting Wizard

Sometimes a local superposition over a few key residues is preferable to a global superposition. We demonstrate local superposition with the same two kinase structures, 1uwh and 1t46. This time we superimpose a few residues in the ligand binding site.

In these two PDB files, residue numbering does not match, as is often the case, so we must explicitly specify which atoms to align. The Pair Fitting Wizard simplifies this process. Switch the mouse into 3-Button Editing Mode before launching it from the Wizard menu. The Wizard prompts you to (left-)click on the first atom in the mobile object and then the corresponding atom in the template. Continue to left-click on pairs of
atoms in this order, mobile first and target second. Once three or more pairs have been defined, left-click the **Fit n Pairs** button to perform the superposition (see Figure Y.10).

**Figure Y.10** The **Pair Fitting Wizard** is used to align 1pkg to 1t46 based on the fit of three atom pairs in their ligands. To reproduce this fit,

Type `fetch 1pkg 1t46`.

Choose `all / H Hide / everything`.

Type `show sticks, organic`.

Navigate with the mouse to get a good view of the ligands, as in the top figure. With the mouse in **Selecting** mode, left-click on the two ligands shown to select them, and choose `(sele) / L Label / atom name`.

Notice the ligand names in the History field above the command lines. The green carbon ligand is ADP'1480, and the cyan-carbon ligand is A/STI-3.

Switch the mouse to 3-button Editing mode.

Choose **Wizard / Pair Fitting**.

Follow the prompts in the Display Area, left-click on the following atom pairs:

- STT'3/N3/ and A/ADP'1480/N1/
- STT'3/N8/ and A/ADP'1480/N7/
- STT'3/C6/ and A/ADP'1480/C4/

Finally, left-click on **Fit 3 Pairs** in the **Pair Fitting Wizard**, and PyMOL will display the fit and the RMS in the Display Area. To hide the atom labels, choose `(sele) / H Hide / label`. 

© 2010 Schrödinger, LLC Jan 2010 Draft For PyMOL Sponsors Only 18
Typing the pair_fit command

Local superposition can also be commanded by typing. In typing the `pair_fit` command, remember that the atom list for each object must contain exactly the same number of atoms. Pair fitting commands that include long lists of atom pairs are best prepared as scripts.

The following example `pair_fit` command superimposes the CA atoms for residues 480 to 482 and 528 to 531 in chain a of 1uwh onto the CA atoms of residues 621 to 623 and 670 to 673 in 1t46 (see Figure Y.11).

```
PyMOL> pair_fit 1uwh and chain a and name ca and resi 480-482+528-531, 1t46 and name ca and resi 621-623+670-673
```

![Figure Y.11](image)

Chain a of 1uwh and 1t46 are superimposed here using the typed pair_fit command above. The proteins are shown as ribbon and the superposed active site residue C-alphas are shown as spheres. You can see that the pair fit is excellent: each of the fitted C-alphas is nearly spherical and colored half magenta and half lime.

Using selection macros, the typed commands is

```
PyMOL> pair_fit 1uwh//A/480-482+528-531/CA, 1t46///621-623+670-673/CA
```

For more information on the subtleties of aligning and pair fitting, see the [PyMOL Wiki](https://pymol.org/wiki).
Changing Protein Structures

Mutagenesis

PyMOL’s Mutagenesis Wizard can mutate individual side chains from one amino acid residue to another, and also suggest alternate rotamers (rotational conformations) for existing side chains.

To launch the Mutagenesis Wizard, choose Mutagenesis from the Wizard menu. In the Display Area, the Wizard prompts you to left-click with the cursor over the residue you wish to alter, and it displays five pop-up menus below the list of objects & selections (see Figure Y.12). The Selecting mode is set to Residues. After you choose a residue to mutate, the wizard prompts you to select a mutation, and then a rotamer for the mutation. You can choose an alternative rotamer without mutating to a different residue.

After you choose a residue to mutate, PyMOL will display alternative residues and rotamers using, by default, the by element color scheme with white carbons. Very likely this will distinguish the mutation you are inspecting from the native structure.

To display various rotamers, the video-control buttons come into play. Left-click on the forward play button to display an animation of the residue sampling its library of low-energy rotamers. Left-click on the frame-by-frame forward and backward buttons to display the individual rotamers.
Figure Y.12  All pop-up menus belonging to the Mutagenesis Wizard are shown here simultaneously. The overall Mutagenesis menu, shown here on the left, displays the current choices: the mutation is ARGN, the N- and C-caps are open, etc. The six blue menu items under Mutagenesis cascade. The Mutant menu shows residues to which to mutate, and cascades to show variant ionizable groups. Variations on N- and C-terminal residues can be chosen from the N-Cap and C-Cap menus. Hydrogens can be added or removed from the representations of the mutant residues. Lines, Sticks, Spheres and Dots representations can be chosen. Rotamers are incorporated from the Dunbrack Backbone Dependent and Backbone Independent libraries, http://dunbrack.fccc.edu/SCWRL3.php. Choose Backbone Independent or Dependent Rotamer libraries from the last pop-up menu.

PyMOL displays graphical “bumps” to indicate steric interactions caused by mutations (see Figure Y.13). Once you have chosen the mutation and rotamer, left-click the Apply button in the Wizard. The bumps disappear from view. When you are finished making mutations, left-click Done.
Figure Y.13A In the lysozyme PDB file 1cv4, Met 118, shown as sticks, occupies a large cavity, defined here by spheres. From the Setting menu, Transparency / Spheres / 60% was chosen, and the image in A, B, and C were ray-traced.

Figure Y.13B Stepping through rotamers of Met 118 in the Mutagene-sis Wizard displays steric interference as bumps (red disks proportional to the size of the overlap). Showing the atoms as spheres in the mutation object menu, and as sticks in the Wizard reveals the overlaps involved in the clash. Green disks and yellow lines indicate favorable interactions.

Figure Y.13C Mutating the same residue to Val and stepping through rotamers reveals both favorable (red) and unfavorable (green) interactions.

Removing atoms

PyMOL provides a quick way to remove all waters or all hydrogens from an object. Simply choose A Action / remove hydrogens or remove waters from the menu next to the object. Removing is not the same as hiding. Once an atom is removed, it is no longer available to PyMOL’s memory. However, you don’t have to exercise the same caution
with removing hydrogens from crystal structures as you need for other atoms. Hydrogens don’t appear in electron density maps or the PDB files computed from them. PyMOL computes likely positions for hydrogens based on bond valences and adds them when you type the command

PyMOL> h_add

You can remove a specific unselected atom, residue, chain, segment, or object by right-clicking on an atom in the Display Area to display its pop-up menu. Choose atom, residue, chain, segment, or object / remove atoms.

Selections can be used to remove groups of atoms that may not lie in the same region or object. Simply select the atoms you wish to remove, find the selection in the objects & selections list, and choose A Action / remove atoms in the corresponding menu.

Finally, you can type the remove command to specifically eliminate a specified set of atoms. For example, to remove chains b and y in structure 1uwh, type

PyMOL> remove 1uwh and chain b+y

or, using the selection macro form, type

PyMOL> remove /1uwh//b+y

Making Simple Movies

Rock & roll

PyMOL automates the production of simple movies. Here we describe the most basic of PyMOL’s automatic animation capabilities, rocking (back and forth) or rolling (rotating) a structure to improve the perception of three dimensions. MacPyMOL can export movies directly into QuickTime™.mov format. On other platforms, it is necessary to save the individual movie images and combine them using external software, such as ImageMagick and Adobe Premiere (see Chapter to come).

To practice making a movie, fetch or load a structure file, for example, 1t46.pdb. To ensure that this exercise doesn’t take a lot of compute-power, and can go quickly, choose
1t46 / A Action / preset / simple. The simple preset uses the ribbon representation, which is computed rapidly. Choose Movie / Draw Frames from the top bank of menus. This will also speed things up (compared to Ray Trace Frames). Choose Movie / Frame Rate / 30 fps, which will give a particularly smooth motion (most Hollywood movies run at 24 fps (frames per second)).

Figure Y.14 shows the pop-up menus that cascades from Movie / Program. For this simple movie, choose Static Loop, and then any of the choices of axes (about which to rock or roll), degrees (deg., extent of motion) and seconds (sec., duration in time).

So far nothing has moved. To set the movie in motion in the Display Area, left-click on the video button control. While the movie is in motion, you can still navigate the structure with the mouse, and give commands to change the representation. Using the video buttons as you would on a VCR, you can stop, step backwards and forwards, rewind and go to the end of the movie.

Movies can also be created from scripts. The following script creates a movie that rolls the structure of 1t46 about its y-axis.

```python
# fetch a structure and apply the "simple" preset
fetch 1t46.pdb
preset.simple()

# define a 360 frame movie of a simple Y axis rotation loop
```
mset 1 x360
movie.roll 1, 360

# start the movie
mplay

Saving QuickTime movies with MacPyMOL

MacPyMOL can write QuickTime movies directly. Choose **Save Movie As**... from the **File** menu and choose **QuickTime**.... Then set the **Frames per second** to 30, uncheck the **Keyframe** and **Limit data rate** options, and adjust the quality to High. Click the **OK** button and choose the file name and destination. Once you click **Save**, MacPyMOL will begin the process of creating the movie.

Saving movie images on other platforms

On all platforms, PyMOL can write series’ of MPEG or PNG files containing the images for animation. On Windows PCs, you can write MPEG files directly, which can be played by the Windows Media Player. Write an MPEG movie or PNG images by choosing **File / Save Movie As / MPEG** or **PNG Images**: a dialog box will prompt you to navigate to the folder where you want to save the movie, and to give it a file name. PyMOL will create the files. When you choose **PNG Images**, PyMOL will output the images in a series of separate numbered files, incorporating the given file name, and ready for further processing by ImageMagick or Adobe Premiere.
Appendix A

Atom selection macros

PyMOL’s powerful atom selection language is covered in depth in Chapter ES. Here we quickly run through some macro syntax to help get you started. This will be more useful if you first read the Typing Commands section at the beginning of this chapter.

Macros make it possible to represent a long selection phrase such as

PyMOL> select luwh and chain b and resi 600 and name ca

in a compact form, using forward slashes, as in

PyMOL> select /luwh//b/142/ca

The five fields defined by the slashes are

/entity/segment/chain/residue/atom

Each field may contain an identifier or name that narrows the specification of the selection. An entity is simply an object or selection name.

PyMOL has to be able to recognize the macro as one word, so no spaces are allowed in it. The macro must contain at least one slash for PyMOL to recognize it.

The fields of a macro are related by the boolean AND. That is, atoms are excluded from the selection if their identifiers do not match any one of the identifiers in the macro.

Omitting a field is the same as omitting a restriction. You can omit fields within a macro, as in /luwh//b/142/ca, and you can also omit them from the end. The segment field is frequently omitted because few source files use it. The entity field is frequently omitted when only one object is enabled.

Macros come in two flavors: those that begin with a slash and those that don’t. If the macro begins with a slash, PyMOL interprets the fields starting from the left, or top of the hierarchy.
Selection syntax translations

Here we translate some selections from the Long form atom selection syntax section of this chapter into macros.

PyMOL> select resi 100
PyMOL> select 100/

PyMOL> select polymer and resi 100
PyMOL> select polymer and 100/

PyMOL> select resi 100 and chain a
PyMOL> select ///a/100
PyMOL> select a/100/

PyMOL> select my_resid, resi 100
PyMOL> select my_resid, 100/

PyMOL> select my_collection, resi 100-110+90+82 and chain a
PyMOL> select my_collection a/100-110+92+82/

PyMOL> select my_negs, resn glu+asp
PyMOL> select my_negs, glu+asp/

PyMOL> select my_main, polymer and name n+ca+c+o
PyMOL> select my_main, polymer and */n+ca+c+o

(An asterisk is a wild card. Here it means all residues.)
PyMOL> select my_group, chain b+d+e
PyMOL> select my_group, b+d+e/

PyMOL> select luwh_negs, resn glu+asp and luwh
PyMOL> select luwh_negs, /luwh///glu+asp
More Examples

To select a numbered residue in a specific object (e.g. 1t46):

select 1t46///640/

To select all residues with a given residue identifier in any object:

select 606/

To select atoms in a given object by name:

select 1t46///N+CA+C

To select atoms in two out of several active objects:

select luwh///CA or 1t46///CA