L02: Desktop Molecular Graphics PyMol 2

1 *PyMol - Exercise N*: Scene menu - animations and transitions

The **Scene** menu has very nice features to save the current view of a structure in various states of representations and toggle between them. The transition from one scene to the next creates also a beautiful screen animation. This is one method for comparing different versions of structure illustrations, or for creating internal animations to tell a story about the structure.

Preliminary:
√ - **Open** PyMol
√ - **load 2BIW.pdb2**
√ - Change background: menu cascade **Display > Background > White**

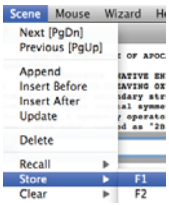
✔ **TASK**

Create Scene 1: a cartoon representation colored by secondary structure

√ - Hide everything: **2BIW.pdb2** > **H > everything**
√ - Show as cartoon: **2BIW.pdb2** > **S > cartoon**
√ - Color by secondary structure: **2BIW.pdb2** > **C > by ss > *pick-a-color***.
√ - Now **orient** the molecule with the mouse **in a way you like**.

Store Scene 1 under the F1 function key:

Scene > Store > F1



Create Scene 2: keep cartoon representation, color e.g. spectrum, and rotate the molecule in a different orientation. For a more stunning effect you can zoom in.

√ - Change cartoon color: **C > spectrum > rainbow**
√ - **Rotate** the molecule in a different orientation.
√ - zoom in.

Store Scene 2 under the F2 function key: **Scene > Store > F2**

Create Scene 3: Zoomed on ligand, and color

√ - Zoom on ligand: **2BIW.pdb2** > **A > preset > ligand sites > solid surface**
√ - change color to blue hue slate: **2BIW.pdb2** > **C > blues > slate**
√ - change ligand color to color wheat
- Cick on ligand: **2BIW.pdb2** (This places the ligand under name (sele))
- (sele) > **C > yellows > wheat**
√ - **Rotate** the molecule in a suitable, pleasant orientation.
√ - Alter clipping planes: menu cascade **Display > Clip > 12 Angstrom Slab**

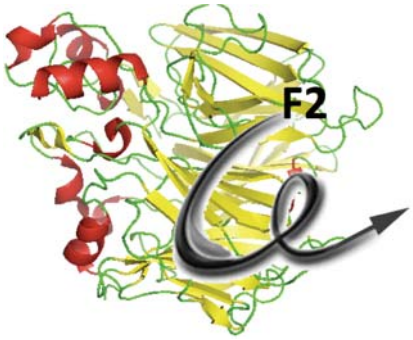
Store Scene 3 under the F3 function key: **Scene > Store > F3**

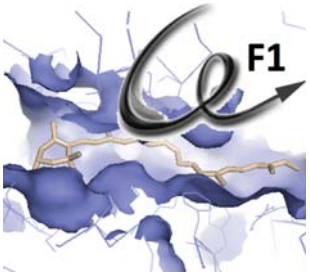
Now you can recall any of those 3 scenes in any order with the **Scene > Recall** Menu.

Note: it is possible to simply click the **F1**,**F2** etc. **key** on the keyboard if it exists. However, if the key has other predefined function by the operating system, use the **FN** or **fn** key to alter the preset definition.

✔ TASK

√ - Recall scene 1: **Scene > Recall > F1**
You should witness a beautiful animation and color change while the scene transitions and twirls to the new scene.
√ - Recall scene 2: **Scene > Recall > F2**
√ - Recall scene 3: **Scene > Recall > F3**
√ - Etc.





Note: Scenes have to be played from within PyMol and are not currently exportable to a QuickTime movie without additional external scripts.

2 *PyMol - Exercise O*: A simple animation within PyMol, and for PowerPoint

There are various ways to animate and create movies within PyMol. This short exercise is meant as a simple option to both create a rotation within the PyMol Viewer, and a simple option for generating a movie suitable for PowerPoint. At this point you should have the simple pocket surface with the carotenoid shown as stick as in the previous exercise.

The command `mset` defines a movie. Since our current level of visualization only contains one PDB file, the value of `mset` will be 1. The command `mdo` defines an action, here this action will consist of a rotation about the Y axis.

Preliminary:
√ - If you are continuing from the previous part, zoom out of the previous exercise view to see the ligand within the pocket, or re-apply a simple preset such as (reload 2BIW.pdb2 if necessary):
√ - **2BIW.pdb2 > A > preset > ligand sites > solid surface**
√ - recolor surface and side chains as above

2.1 Automatic rotation within the PyMol Viewer

The purpose of this set of command is to create a perpetual rotation within the Viewer around the Y axis with an increment of 5 degrees.

✔ **TASK**

Within the top PyMOL> command line, type the following commands**:**

```
mset 1
mdo 1: turn y,5;
```

Once the commands have been typed, the "vcr" buttons at the bottom right of the "Internal GUI become active. We only need to use the Play (triangle - ▶) and Stop (square - ■) buttons to activate the animation.

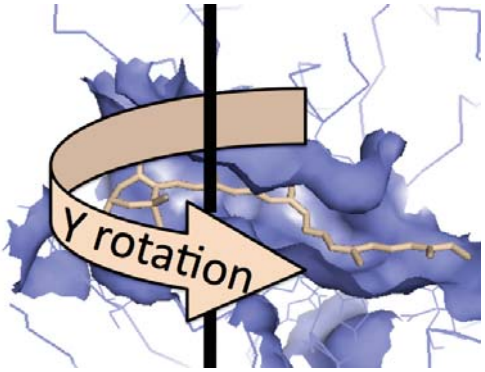
Stop | Play

Note: The **S** button toggles on-screen sequence andthe **F** button toggles full screen display.

a. **Click** the **Play** button

b. watch the rotation happen

c. **Click the Stop** button when done



Note: the mdo command is followed by a colon (:). However in previous versions of PyMol it was followed by a comma (,) a syntax no longer valid. The ending semi-colon (;) is necessary only if other commands are required. For example, a rotation both within the Y and X axes of 5 degrees each would be written as:

```
mdo 1: turn y,5; turn x,5;
```

2.2 Creating a movie suitable for PowerPoint

The purpose of this short exercise is the creation of a single movie file suitable to be played independently or imported within PowerPoint for a presentation.

✔ READ

The creation of movies within PyMol is different depending on the version and on the operating system. The MacPyMol version of PyMol contains a built-in function to tap into the QuickTime™ engine (even if you do not have the PRO license) and can be used to create movie files. On other systems, the movie is saved as a series of images (in PNG format) which then need to be assembled manually with e.g. QuickTime™PRO (Macintosh/Windows), VideoMACH (Windows),or similar movie making software, meant to compile and compress a series of images (frames) into a single movie file. This aspect of movie making as well as the indiosyncracies of PowerPoint and movie compatibility will be reviewed in another module.

Since we were using some movie/animation features above, the first command to use is `mclear` to remove functions and frames that have been saved.

Modification of the command `mset`: since we are going to save the file into a QuickTime movie, we need to define how many frames the movie will contain. For example, to have 36 frames the command will be written as `mset 1 x36`

Modification of the `mdo` command: for each frame we need to specify that we want a rotation around the (e.g.) Y axis of (e.g) 5 degrees. However, this would

require to declare each of the (e.g.) 36 frames and for each frame declare the action we want to take (for example a rotation).

Within PyMol the undocumented python command `util.mroll(start, finish, loop-flag)` does all of this if we have declared the number of frames with `mset`. The actual PyMol command is: `movie.roll`

Note: A complete 360 degrees rotation is assumed by this command.

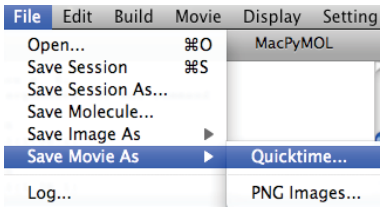
✔ TASK

Within the **top PyMOL> command line, type** the following commands**:**

```
mclear
mset 1 x36
movie.roll 1, 36
mplay
mstop
```

`mplay` and `mstop` activate the movie manually. To save the movie as a QuickTime movie follow these instructions:

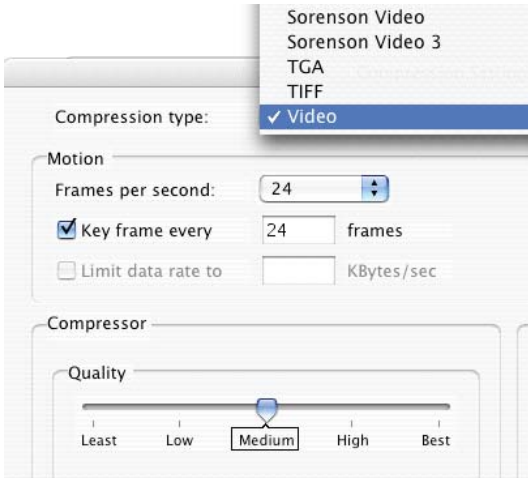
Use the menu cascade:



File > Save Movie As > Quicktime

(Note: this menu available for native Mac version only, usually called MacPyMol)

When saving this way, the default encoding (codec) is “Video” and can work well for this exercise. Other codecs are also available depending on what is installed on your system.



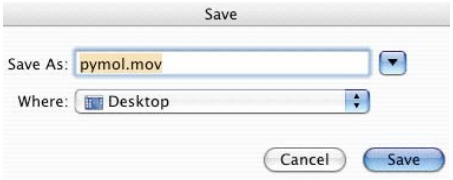
Change the default 12 frames per second to **24** from the pull-down menu.

Keep the quality to **medium**

Press the Save button

The file should be about 2 Mb in size.

Save the file on the Desktop to find it easily. The default name is **pymol.mov** and can be changed here if desired.

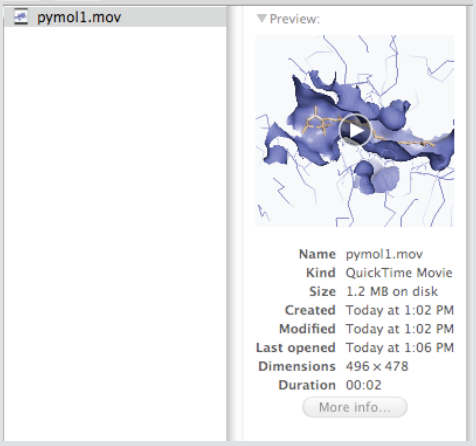


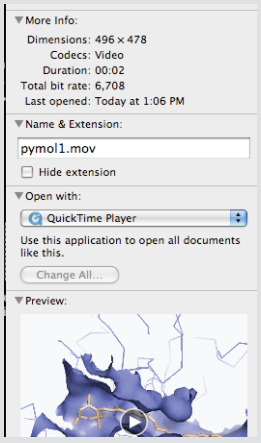
Press the Save button

Double click the **pymol.mov** file and watch it.

INFO

Within the Mac finder the movie can also be previewed. Additionally it tells us the dimensions of the movie: 496 x 478 in this example. (Your movie will have the size of the current Viewer window.) Clicking on the More Info… button opens the information data of the file, portion of which is shown at right.





Clicking the play circled triangular play button  will indeed play the movie within the preview window.

Note: to change the size of the resulting movie file or image files use the following command to set the size of the Viewer: **`viewport 320,240`** to create images and movies of that size (it is 4 times smaller in surface than a 640x 480 image).

2.3 Ray traced movies

The ray tracing option is also available when making movies. The movie making is almost identical as the previous section, but each image is ray-traced before it is saved. The movie can then be saved as a series of images or as a movie file from the File menu.

TASK

To create a ray-traced movie **type the following commands** within the top PyMOL> command line:

viewport 320,240
set ray_trace_frames=1
set cache_frames=0
mclear
mset 1 x36
movie.roll 1, 36

a. if you want to save the individual images use the following command:
mpng MyMovie

This command will create images MyMovie.0001.png, MyMovie0002.png etc. The images can then be assembled into a movie by an independent software such as QuickTime™Pro.

b. If you are using MacPyMol and want to save the movie with QuickTime™Pro follow the same procedure as above with the menu cascade:
File > Save Movie As > QuickTime…

Note that the progression bar () will remain active, indicating that the ray-traced images are still available for other commands (for example the mpng save command above).

Note: ray-tracing is computationally intensive. The bigger the images (viewport size) and the more complex its contents, the longer it takes. You can see which frame is being rendered by looking at the bottom right of the “Internal GUI.”

✔ **TASK**

Save this PyMol **session** for later and **Quit PyMol**.

From the top menu follow the menu cascade:

File > Save Session As…

Save file on the Desktop as e.g. **MySession2.pse**

3 *PyMol - Exercise P*: Advanced movie commands

Preliminary:
√ - Open a new session of PyMol.
√ - Load 2BIW.pdb2 or any other molecule of your choice.
√ - Change the default representation from green lines to e.g. ribbons colored by secondary structure

3.1 PyMol movie set-up and frame

Some command are useful during the preparation of a movie, such as the command `rewind` or `frame`.

INFO

PyMol "frame"commands	Description
`rewind`	goes to the beginning of the movie.
`middle`	goes to the middle of the movie.
`ending`	goes to the end of the movie.
`forward`	moves the movie one frame forward.
`backward`	moves the movie back one frame.
`frame frame-#`	sets the viewer to the indicated movie frame. e.g. `frame 5`
`mmatrix {clear|store|recall}`	sets up a matrix to be used for the first frame of the movie and orient the view in a specific way. Example 1: `mmatrix store`. Example 2: `mmatrix recall`

3.2 PyMol movie motion commands

The purpose if this exercise is to explore the movie programs built within PyMol.

Movie movement commands are summarized below from their definition within PyMol software:

INFO

Movie programs	*Definitions and complete parameters* .
movie.rock	`movie.rock(first,last,angle=30,phase=0,loop=1,axis='y')`
movie.roll	`movie.roll(first,last,loop=1,axis='y')`
movie.zoom	`movie.zoom(first,last,step=1,loop=1,axis='z')`
movie.screw	`movie.screw(first,last,step=1,angle=30,phase=0,loop=1, axis='y')`
movie.sweep	`movie.sweep(pause=0,cycles=1)`
movie.pause	`movie.pause(pause=15,cycles=1)`
movie.nutate	`movie.nutate(first,last,angle=30,phase=0,loop=1, shift=math.pi/2.0,factor=0.01)`
movie.tdroll	`movie.tdroll(first,rangex,rangey,rangez,skip=1)`
movie.timed_roll	`timed_roll(period=12.0,cycles=1,axis='y')`
movie.load	`movie.load(*args,**kw)`

For many of these commands supplying the first and last frame is often enough, and all the other variables are used per the defaults listed in the table. We have already used `movie.roll` in the previous exercise.

Some of these commands are not documented. For example as of this date there are only 64 Google entries for "*movie.zoom pymol*" and only 4 entries for "*movie.screw pymol*" 2 of which are a listing of all PyMol commands

3.3 movie.roll

We have already seen `movie.roll` in an exercise above. Here is an example where we take advantage of choosing the axis of rotation (default was y).

TASK

Try the following mini movie commands. The `viewport` command is optional, it would set the image size to the standard of a 15 inch screen.

We have already encountered `mclear` and `mset` previously. `mplay` will start the movie and `mstop` will halt it. These actions can also be performed from the VCR controller as explained previously.

```
viewport 640,480
mclear
mset 1 x360
movie.roll 1,120,1,axis=x
movie.roll 121,240,1,axis=y
movie.roll 241,360,1,axis=z
mplay
mstop # to halt the movie
```

3.4 movie.rock

The top right panel of PyMol offers a button called "Rock" which rocks the molecule back and forth very slowly providing insight for depth perception. The command `movie.rock` provides the same functionality. The first 2 arguments are the first and last frame numbers, and the third is the degree of rotation. The movement is faster for larger, wider degrees of rotation. This movement is useful to present a molecule which interesting features are on one side, such as a cavity or an active site. The final movie can be exported to a QuickTime movie or a series of PNG files as well.

TASK

Type the following commands:

```
mclear
mset 1 x120
movie.rock 1,120,45
mplay
mstop # to halt the movie
```

3.5 movie.zoom

This function will zoom by default along the z axis (running perpendicular to the plane of the screen.)

✔ **TASK**
Type the following commands:

```
mclear
mset 1 x120
movie.zoom 1, 120
mplay
mstop # to halt the movie
```

Now change the default axis to x rather than the default z:

```
mclear
movie.zoom 1, 120, axis = x
mplay
mstop # to halt the movie
```

The resulting movement is in fact a translation along the x axis.
Note: since we are using the same amount of frames it was not necessary to restate the `mset` command again.

3.6 movie.screw

This commands operates a movement blending the rocking and the zooming options of the previous commands.

✔ **TASK**
Type the following commands:

```
mclear
movie.screw 1, 120
mplay
mstop # to halt the movie
```

3.7 movie.nutate

This commands is similar to the VMD molecular graphics software default movie movement called “lemniscates” and represents a rotation following the edges of a cone facing the viewer by its opening. The best is to simply view it…

✔ **TASK**
Type the following commands:

```
mclear
movie.nutate 1, 120
mplay
mstop # to halt the movie
```

3.8 movie.tdroll

This command is short for "Three-Dimensional roll" and rotates the molecule along multiple axes.

```
Adapted from the PyMol source code:
AUTHOR
      Byron DeLaBarre
USAGE
      movie.tdroll(first,rangex,rangey,rangez,skip=1)
      first is the first frame to apply command
      rangex/y/z = rotation range on respective axis
      enter 0 for no rotation.
      skip is angle increment in each frame
      Use skip to reduce final movie size or to speed up
      rotation.
EXAMPLE
      movie.tdroll 1,360,360,360,5

Note: in older versions of PyMol, the sequence of number was
different and did not include the “first” frame in arguments.
```

✔ **TASK**
Type the following commands for a rotation around the 3 axes.
Each axis requires 180 / 5 = 36 frames.
Thress axis requires mset to be at 36 x 3 = 108

```
mclear
mset 1 x108
movie.tdroll 1, 180, 180, 180, 5
mplay
mstop # to halt the movie
```

PyMol will echo:

```
(' tdroll: defined rotations for', 108, 'frames, starting at frame 1')
```

Note: if you wan to change the skip, then the number of frames has to be altered also! For example to skip only 2 because 5 is a bit fast:

```
mclear
mset 1 x270
movie.tdroll 1, 180, 180, 180, 2
mplay
mstop # to halt the movie
```

In the following example multiple commands are used successively to create a slower paced rotation:

TASK

Type the following commands:

```
mclear
mset 1 x300
movie.tdroll 1, 0, 180, 0, 2
movie.tdroll 100, 180, 0, 0, 2
movie.tdroll 200, 0, 0, 180, 2
mplay
mstop # to halt the movie
```

INFO

PyMol commands are sometimes found in two forms: the actual python commands which is built within PyMol or the equivallent PyMol command itself. Python is the programming language with which PyMol is built.

For example, we saw above that the python command `util.mroll` is the PyMol command `movie.roll`.

The python command is called an API command. ***API is the abbreviation of application program interface***, a set of routines, protocols, and tools for building software applications.

Within the PyMol wiki (`http://pymolwiki.org/`), commands are typically shown as both the API command and the PyMol command. For example the command "`frame frame-number`" positions the movie to the designated frame number (an integer):

DESCRIPTION
frame sets the viewer to the indicated movie frame.
USAGE
`frame frame-number`
PYMOL API
`cmd.frame( int frame_number )`

However, some commands exist only as an API command: