

PyMOL: Advanced

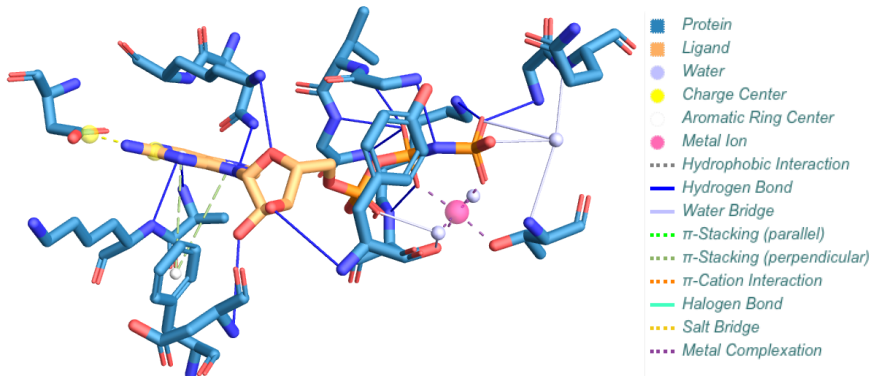
Misc. useful stuff, Using PyMOL scripts, Wizards and Plug-Ins
Making Movies

http://www.pymolwiki.org/index.php/MovieSchool_1

Protein-Ligand Interaction Profiler PLIP

A web server analyzing the interaction between proteins and ligands that uses PyMOL to generate figures and allows to download the results as a PyMOL .pse file

<https://projects.biotec.tu-dresden.de/plip-web/plip/index>

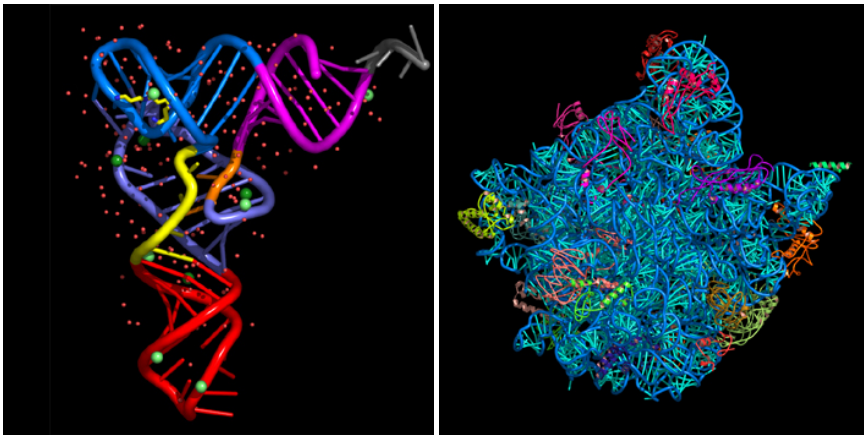


Salentin S, Schreiber S, Haupt VJ, Adasme MF, Schroeder M.

PLIP : fully automated protein-ligand interaction profiler.

Nucleic Acids Res. 2015 Jul 1;43(W1):W443-7. doi: 10.1093/nar/gkv315. Epub 2015 Apr 14.

Program NUCCYL allows to generate highly simplified Nucleic Acid Cartoon Representations in PyMOL



<http://www.biosci.ki.se/groups/ljo/software/nuccyl.html>

Pymol scripts (*.pml)

either:

copy-and-paste script into the command line

or:

run the script with the command

@script.pml

@pml_scripts/AHo_BindingPocket.pml

Running a simple script downloaded from the web

http://pymolwiki.org/index.php/Category:Script_Library

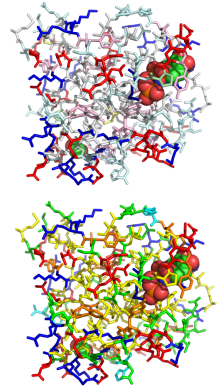
<https://github.com/Pymol-Scripts/Pymol-script-repo>

<http://pldserver1.biochem.queensu.ca/~rlc/work/pymol/>

```
run py_scripts/color_by_restype.py  
color_by_restype 3K8Y
```

```
run py_scripts/AHo_color_by_restype.py  
color_by_restype 3K8Y
```

@AAcolor.pml



If you put the “run” commands into a file called “pymolrc” (visible) or “.pymolrc” (invisible), they will be executed automatically when PyMOL is started, and you will have the commands available in every PyMOL session

Finding a specified Subsequence in a Protein

Using selection operators to find the subsequence “GAD”:

select DDrF0016 and pepseq GAD

also finds “G”, “GA”, “AD” and “D”, but not “A”

Using a python script:

run ~/pymol/py_scripts/findseq.py

findseq GAD, DDrF0016

appears to work fine on a single object, but does not work properly if the selection contains multiple objects

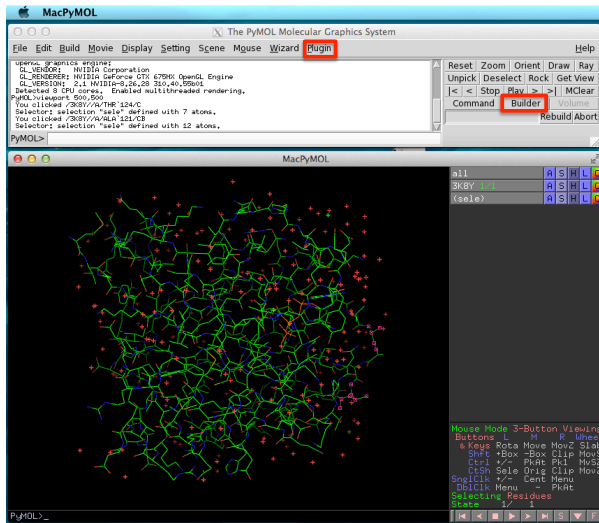
The findseq script interprets **regular expressions***

*: http://en.wikipedia.org/wiki/Regular_expression

Mac: Using the Hybrid Version of PyMOL

The Hybrid mode and X-Windows (Darwin) mode of MacPyMOL offer some advanced options not available in the desktop mode: e.g.

Builder,
Plugin support



Needs X11: <https://www.xquartz.org>

Plugins

MacOSX: Duplicate MacPyMOL.app, rename to PyMOLX11Hybrid.app

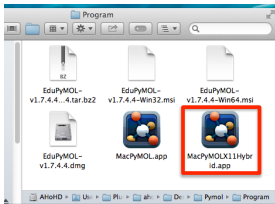
The standard OS X Pymol application, MacPyMOL.app does not run with the Tcl/Tk interface which is required for plugins to work. However, a quick renaming of the program from MacPyMOL.app to PyMOLX11Hybrid.app makes the application run as an X11 application, and plugins are now available.

To rename the executable, right click (or control click) on MacPyMOL and choose "Get Info" in the Panel. Change the Name & Extension to PyMOLX11Hybrid.app. This name can also be changed using the mv command in Terminal.app.

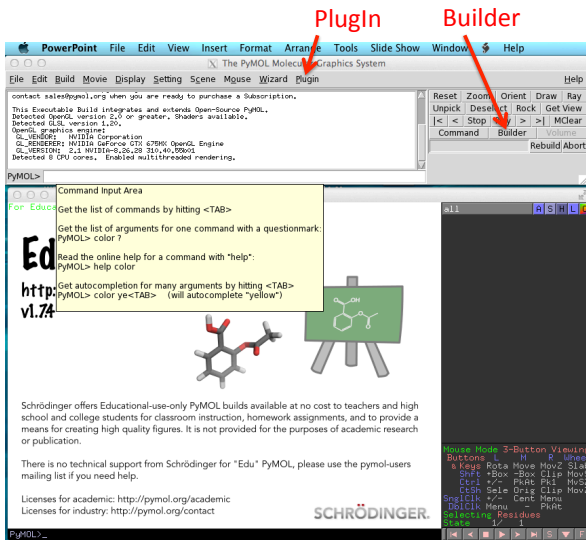
Once this change is made, half of the program will show up under the X11 icon, and half will show up under the MacPyMOL icon.

<http://pymolwiki.org/index.php/Category:Plugins>

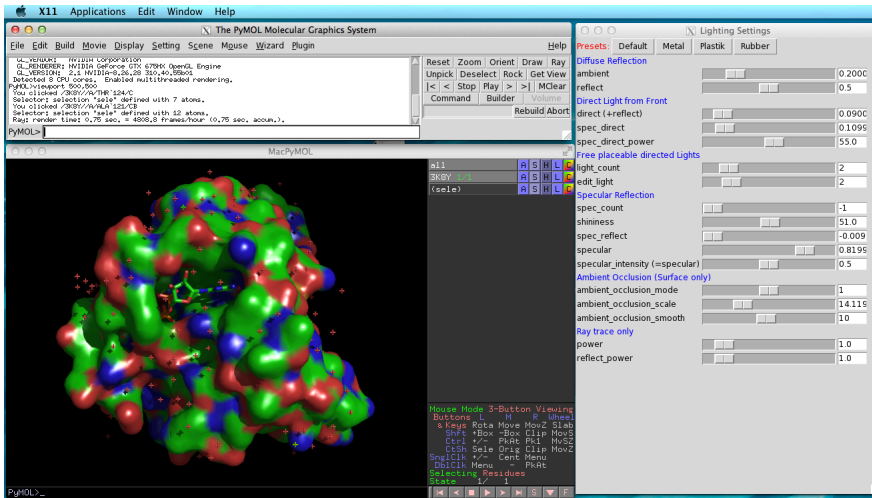
Mac: Using the Hybrid Version of PyMOL



Duplicate MacPyMOL.app
and rename copy to
MacPyMOLX11Hybrid.app



Plugin: Lighting Settings

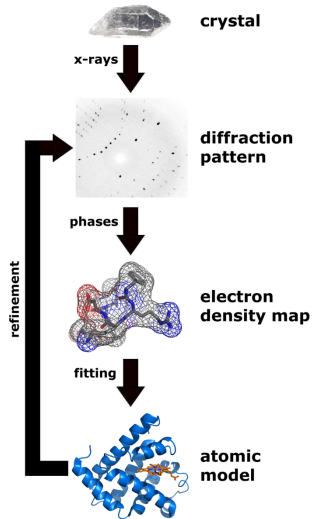


Interactive control over illumination and reflectivity settings

Looking at Electron Density

Using the Density Wizard

X-Ray Diffraction: Display Electron Density




Download Electron Density Map



Find your protein of interest on <http://www.rcsb.org>


Experimental Details Hide

Method: X-RAY DIFFRACTION

Exp. Data:


 [Structure Factors](#)

EDS  

Resolution[Å]:  2.75

R-Value: 0.249 (obs.)

R-Free: 0.273

Space Group: $P 2_1 2_1 2_1$ 

Unit Cell:

<u>Length [Å]</u>	<u>Angles [°]</u>
a = 63.31	$\alpha = 90.00$
b = 89.40	$\beta = 90.00$
c = 212.13	$\gamma = 90.00$

Follow link to Electron Density Server

Download: Maps

Electron-density map generation for 4buo

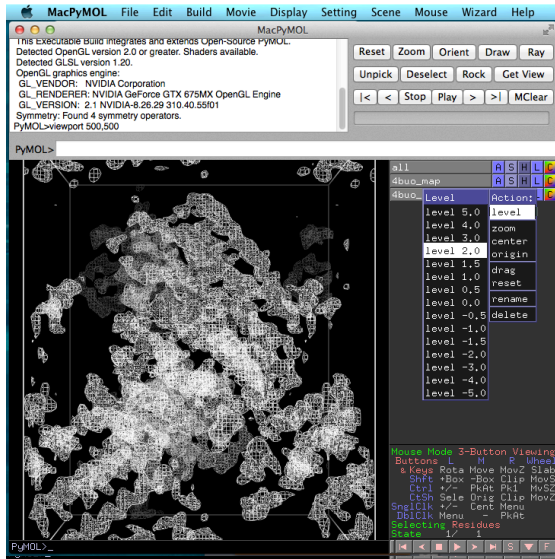
Map format : Type :

(Note: this may take a few seconds, or many minutes, depending on the size of your map.)

Download maps in CCP4-format for use in PyMOL

For 4BUO.pdb (rat NTR1) download files 4buo.ccp4 and 4buo_diff.ccp4
Rename 4buo.ccp4 to 4buo_map.ccp4

Open Map-File with PyMOL



Click on 4buo_map

=> an empty cube is displayed

Open file 4buo_map.ccp4

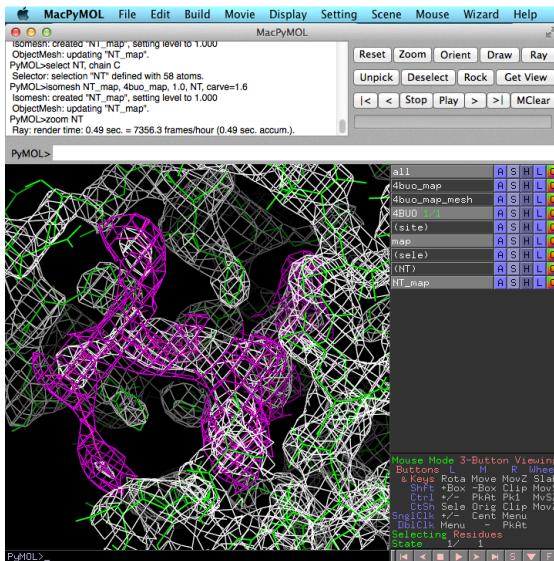
=> nothing visible on screen

**select Action:Mesh:@level 3.0
for object 4buo_map**

=> a new item called
4buo_map.mesh appears

**To change the map level,
select Action:Mesh:@level 2.0
for object 4buo_map.mesh**

Add the Coordinate File



open file 4BUO.pdb

command line:

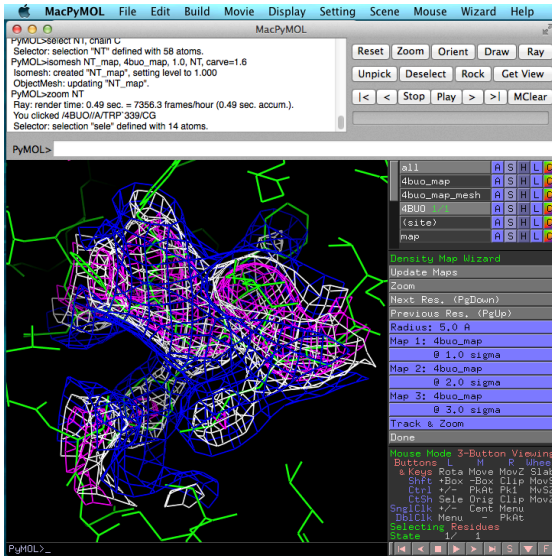
4buo

map, 4buo_map, 2.0, site, carve=1.6
 creates a map that just covers the
 site
 (4buo_map_mesh to see)

=1.6

different color to the local map)

Density Wizard

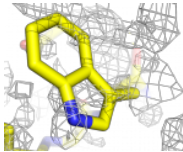


With the density wizard, you can move through the maps residue by residue, using the <pg up> and <pg down> keys

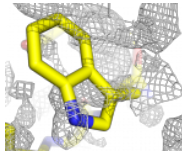
For a publication quality figure the following are suggestions:

color grey50, map	<i># sets map to 50% gray</i>
set mesh_width, 0.5	<i># makes meshes thinner for ray-tracing</i>
bg_color white	<i>#sets background to white</i>
set ray_trace_fog, 0	<i># turns off raytrace fog--optional</i>
set depth_cue, 0	<i># turns off depth cueing--optional</i>
set ray_shadows, off	<i># turns off ray-tracing shadows</i>
ray 1024,1024	<i># this would create a 1024x1024 pixel ray-traced image</i>
png image.png	<i># output final image</i>

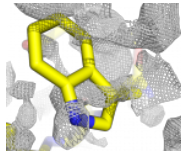
map_double resamples a map at twice the current resolution.
The amount of memory required to store the map will increase eight-fold.



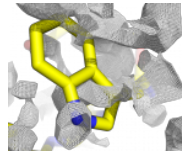
Std. map mesh spacing



Map
doubled

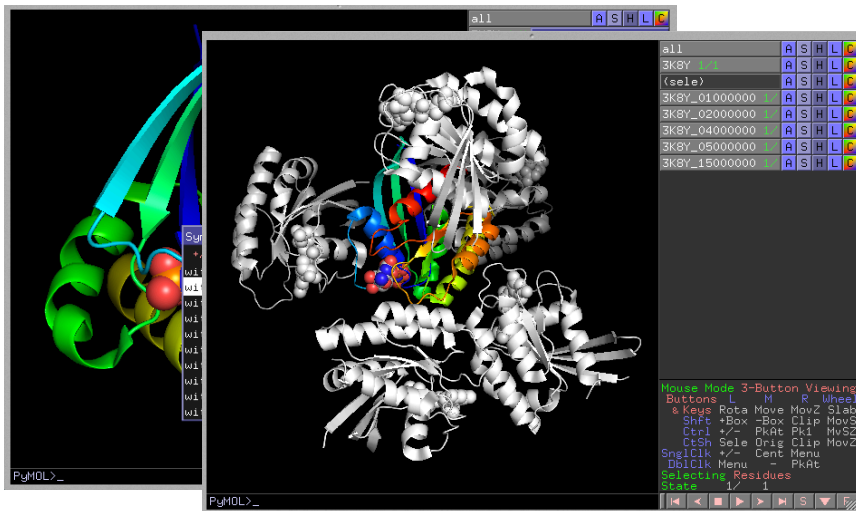


Map
double, doubled



Map
double, double, doubled

Creating symmetry related molecules

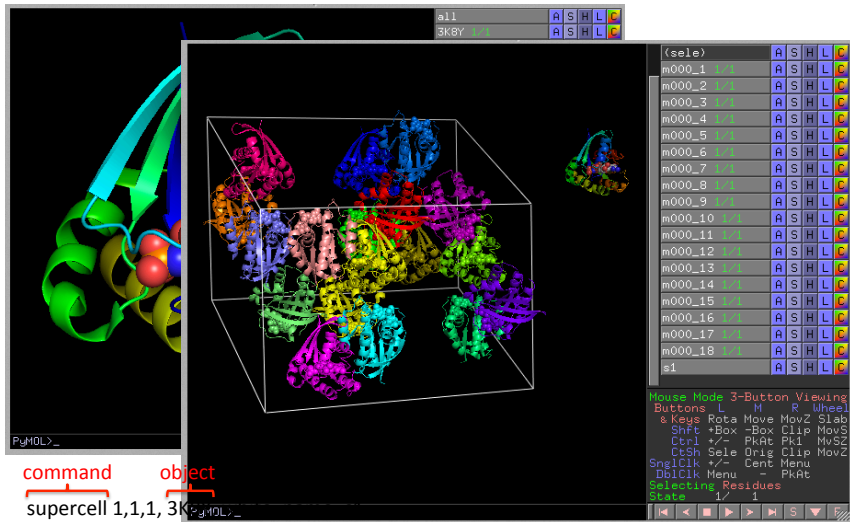


symexp **prefix**, **object**, **selection**, **cutoff** [, **segi**]

symexp 3K8Y_, 3K8Y, 3K8Y, 5.0

<http://pymolwiki.org/index.php/Symexp>

Python Script: Supercell



command

object

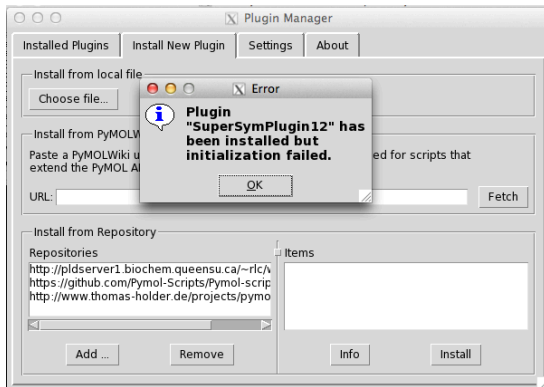
supercell 1,1,1, 3KBY

of unit cells

color & name of box

<http://pymolwiki.org/index.php/Supercell>

PlugIn: SuperSym



This plugin requires **cctbx** and numeric python (numpy).

Computational Crystallography Toolbox (CCTBX)

<http://cctbx.sourceforge.net>

Looking at Electrostatic Potentials

Using the ABPS PlugIn

Adaptive Poisson-Boltzmann Solver (APBS) Plugin

Adaptive Poisson-Boltzmann Solver (APBS) is a software package for modeling biomolecular solvation by solving the Poisson-Boltzmann equation (PBE). The PBE is a popular continuum model used to describe electrostatic interactions between solutes in salty, aqueous media.

<http://www.poissonboltzmann.org>. **Work through the explanations, examples and tutorials on this site to understand what you are doing!**

PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations http://nar.oxfordjournals.org/content/32/suppl_2/W665.full

- Adding a limited number of missing heavy atoms to biomolecular structures

- Determining side-chain pKas

- Placing missing hydrogens

- Optimizing the protein for favorable hydrogen bonding

- Assigning charge and radius parameters from a variety of force fields

PDB2PQR Server: http://nbc222.ucsd.edu/pdb2pqr_2.0.0/

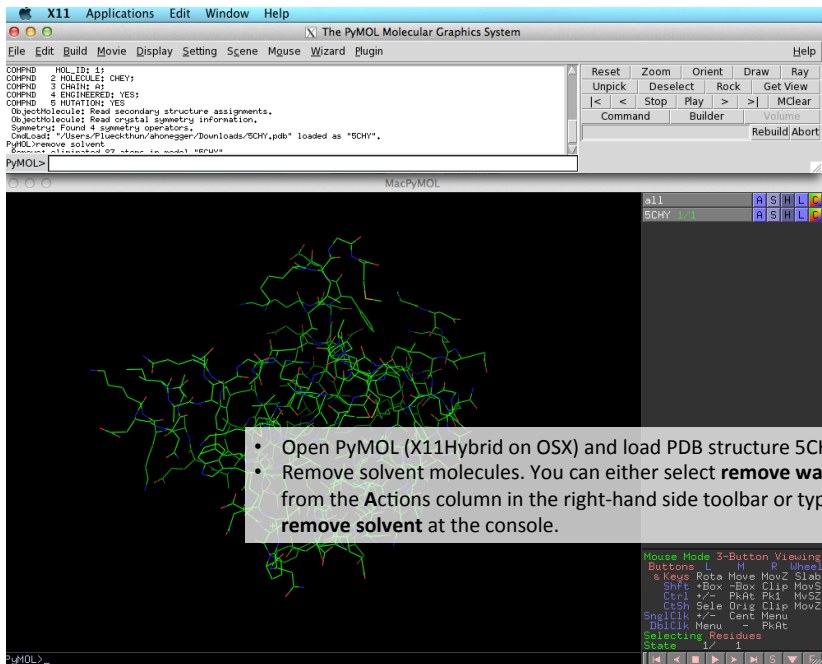
- Uses **PROPKA** (or PDN2PKA) to determine the titration state of proteins at specific pH.

- Uses **APBS web solver** to solve the Poisson-Boltzmann and related equations to calculate solvation energies and electrostatic properties for analysis and visualization.

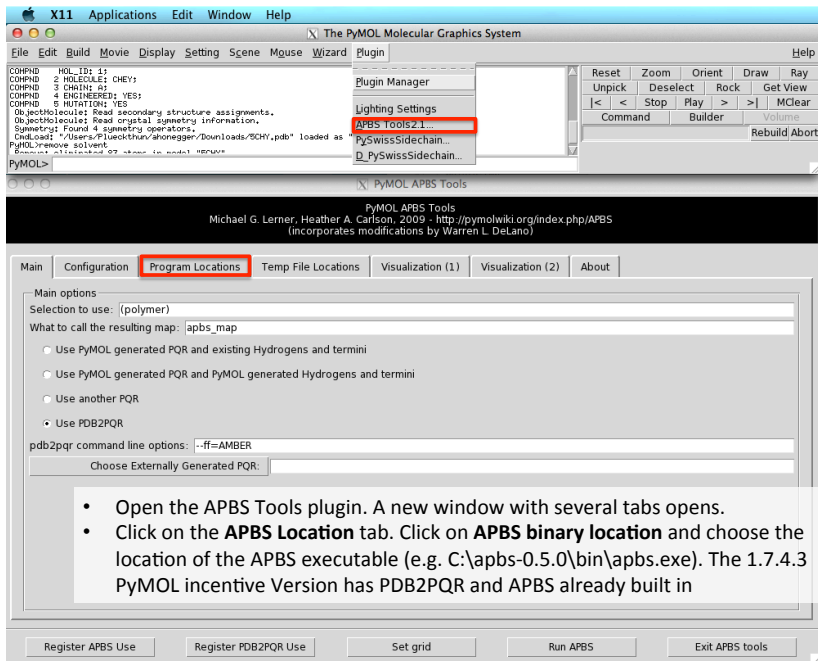
PDB2PQR and APBS are part of the SBgrid package, set APBS-Plugin paths to the appropriate binaries

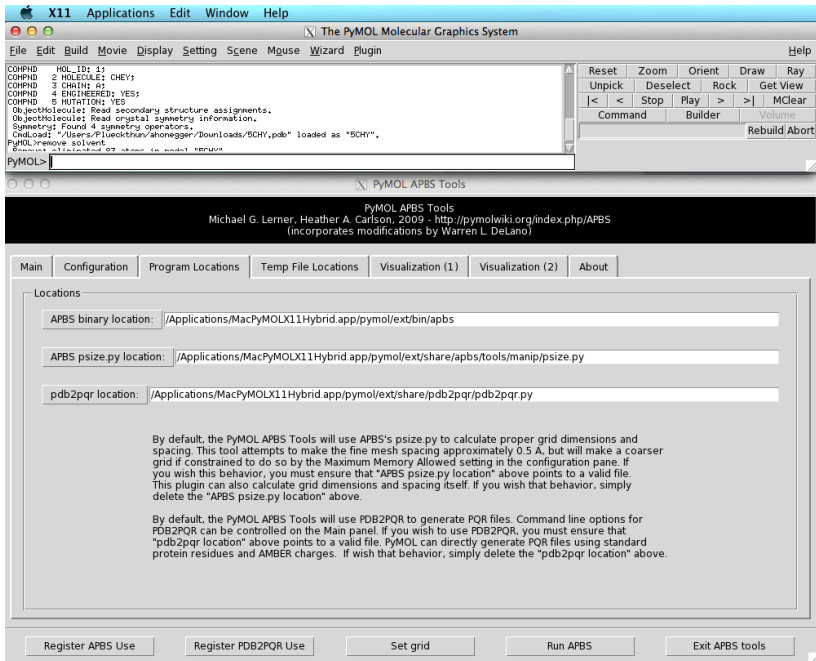
Getting started

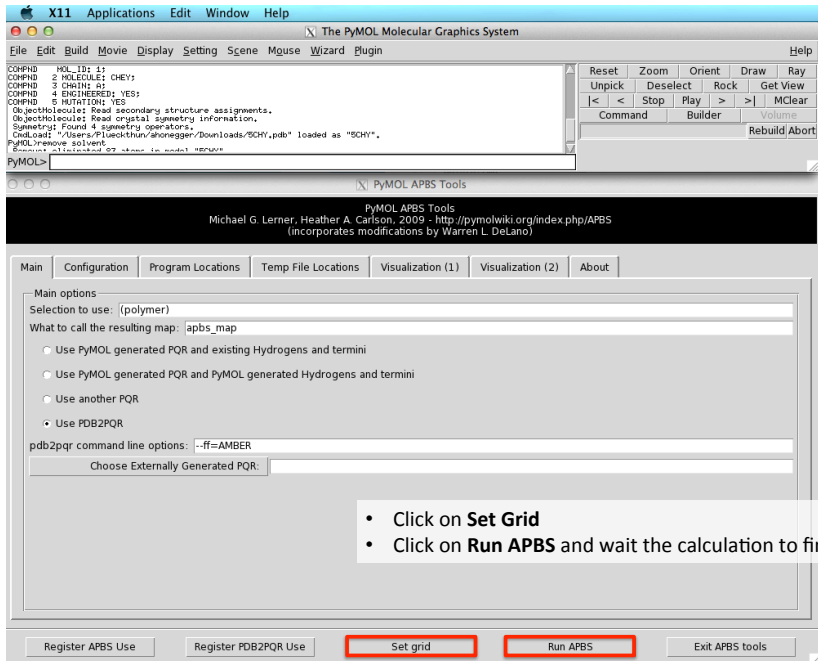
1. Open PyMOL (X11Hybrid on OSX) and load PDB structure 5CHY
2. Remove solvent molecules. You can either select **remove waters** from the **Actions** column in the right-hand side toolbar or type **remove solvent** at the console.
3. Open the APBS Tools plugin. A new window with several tabs opens.
4. Click on the **APBS Location** tab. Click on **APBS binary location** and choose the location of the APBS executable (e.g. C:\apbs-0.5.0\bin\apbs.exe). The 1.7.4.3 PyMOL incentive Version has PDB2PQR and APBS already built in
5. Click on **Set Grid**
6. Click on **Run APBS** and wait the calculation to finish
7. Click on the **Visualization** tab and hit **Update**
8. In the Molecular Surface area, click on **Show**

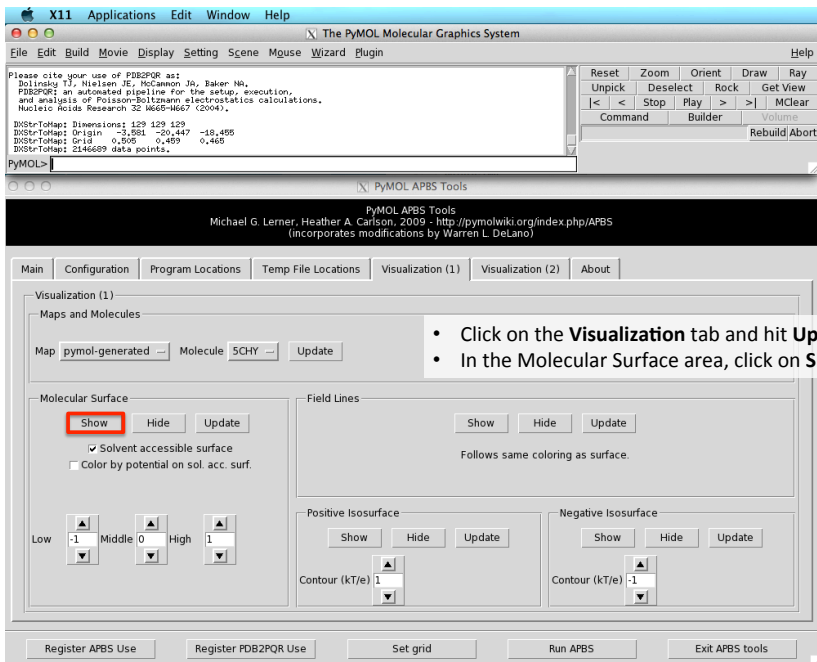


- Open PyMOL (X11Hybrid on OSX) and load PDB structure 5CHY
- Remove solvent molecules. You can either select **remove waters** from the **Actions** column in the right-hand side toolbar or type **remove solvent** at the console.



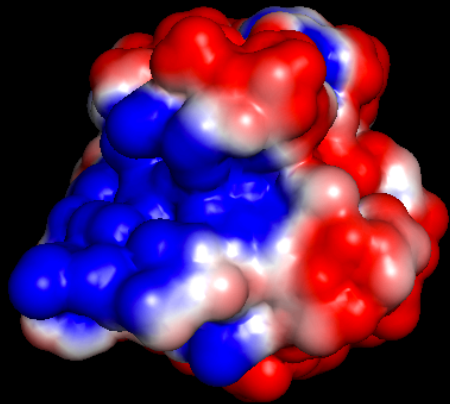




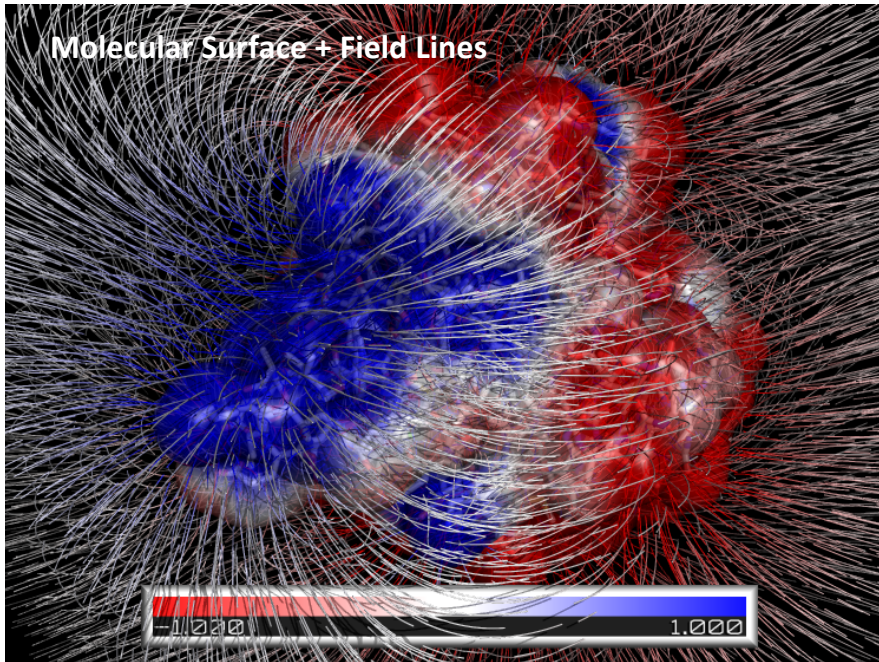


- Click on the **Visualization** tab and hit **Update**
- In the Molecular Surface area, click on **Show**

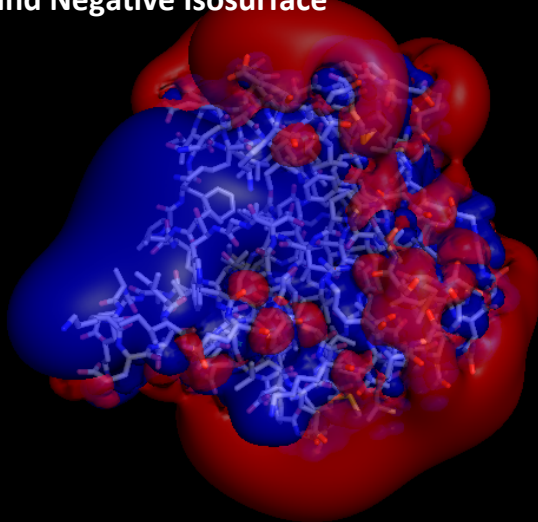
Molecular Surface



Molecular Surface + Field Lines



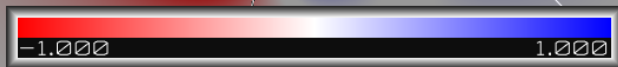
Positive and Negative Isosurface



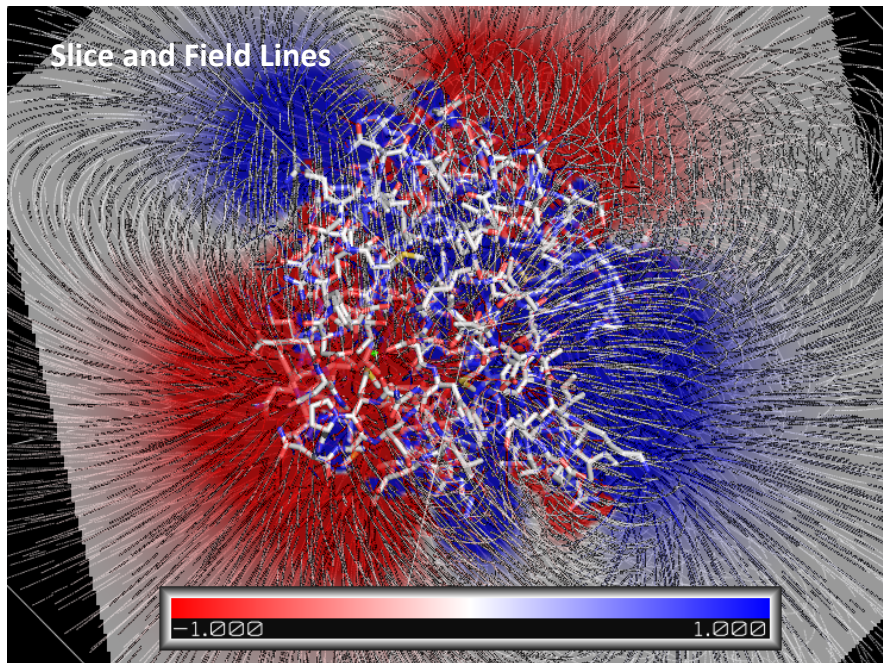
-1.000

1.000

Slice



Slice and Field Lines



APBS tutorial at

[*http://web.chem.ucsb.edu/~kalju/CSUPERB/public/prot_prot_A1.html*](http://web.chem.ucsb.edu/~kalju/CSUPERB/public/prot_prot_A1.html)

goes on to investigate electrostatic interactions between CheA and CheY

- Try it out!

CastP <http://sts.bioe.uic.edu/castp/pymol.php>

CAVER <http://www.caver.cz/index.php?sid=199>

Autodock <http://wwwuser.gwdg.de/~dseelig/adplugin.html>

Gromacs [http://www.pymolwiki.org/index.php/GROMACS Plugin](http://www.pymolwiki.org/index.php/GROMACS_Plugin)

DSSP and Stride http://www.biotec.tu-dresden.de/~hongboz/dssp_pymol/dssp_pymol.html

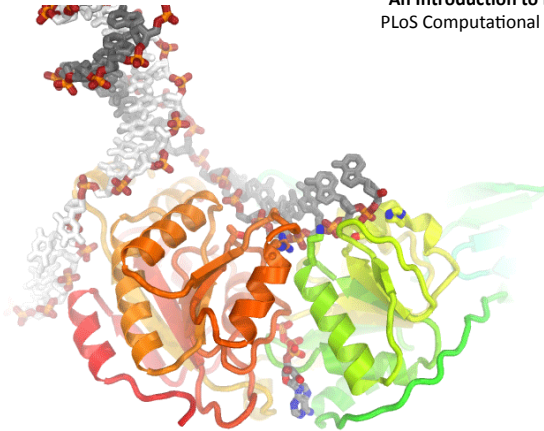
Xpyder <http://linux.btbs.unimib.it/xpyder/>

MLP Tools <http://mlptools.altervista.org/index.html>

An Animation is a Series of Images

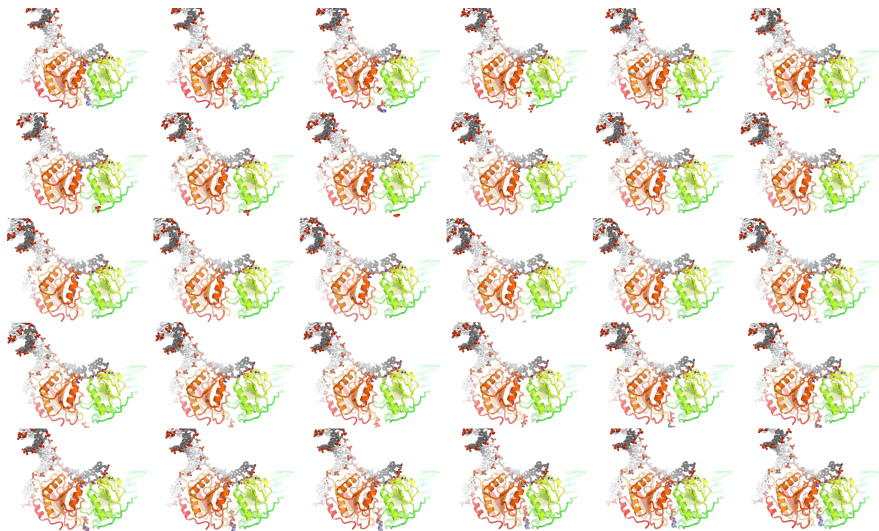
C.Mura, C.M. McCrimmon, J.Vertrees, M.R. Sawaya
"An Introduction to Biomolecular Graphics"

PLoS Computational Biology 8 (2010) Vol. 6, pp 1-11

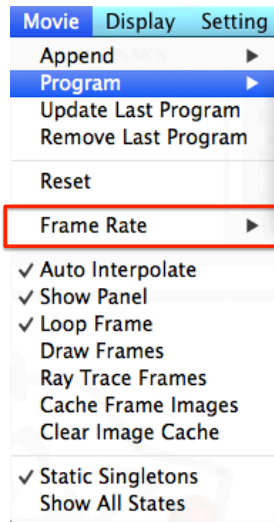


<http://pymolwiki.org/index.php/PLoS>

Movie of a DNA helicase unwinding dsDNA (animated GIF format)



Setting up the Movie



Set up movie, apply standard motions,
undo last addition to movie

Up to 10-12 pictures per second can still be perceived
as individual pictures, the animation appears jerky
Standard for 35mm sound movies: 24 Fps.

PyMOL default: 30 Fps, the animated gif was 15 Fps

http://en.wikipedia.org/wiki/Frame_rate

**Database of Macromolecular Movements
with Associated Tools for Flexibility and Geometric Analysis**

<http://molmovdb.mbb.yale.edu/molmovdb/>

Types of Animations

Camera stationary Molecule immobile

Stationary figure
Stationary phase in movie

Camera stationary **State loop** Molecule mobile **State sweep**

- Multiple conformations of an object provided as multiple states in an object, e.g. MD Trajectories, NMR, Morphs
Moving rigid objects relative to each other

Camera mobile **Camera loop** Molecule immobile **Scene loop**

Animation achieved by changing the transformation matrix (get_view)
Rotate, Rock, Nutate, Zoom, Clip
with or without change in representation

Camera mobile Molecule mobile

Transformation matrix and object coordinates (displayed state) both change during the animation

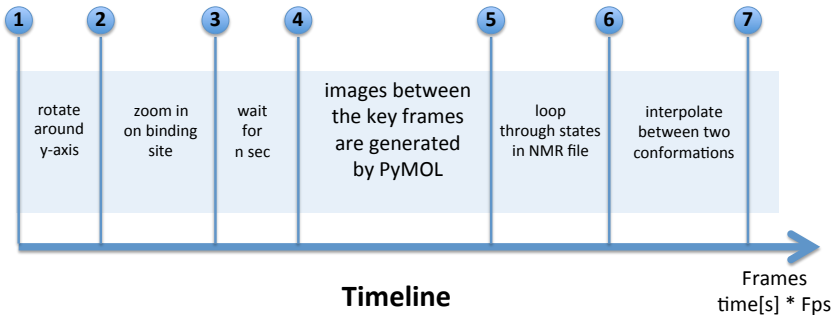
Smooth Transitions between representations

Fading in and out of surfaces, side chains etc.

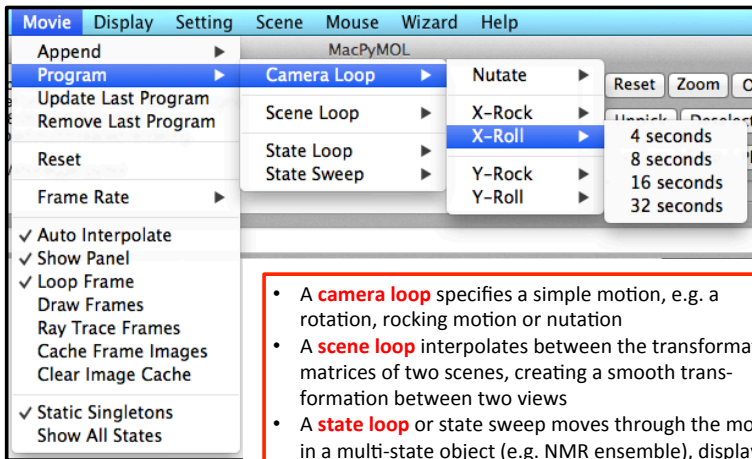
The Movie Timeline

scenes
key frames
defined
by user

The user defines the key frames (individual images saved as scenes)
PyMOL interpolates between the scenes according to the actions and
timing specified by the user



Simple Camera, Scene and State Loops by GUI



- A **camera loop** specifies a simple motion, e.g. a rotation, rocking motion or nutation
- A **scene loop** interpolates between the transformation matrices of two scenes, creating a smooth transformation between two views
- A **state loop** or state sweep moves through the models in a multi-state object (e.g. NMR ensemble), displaying them one after the other

More complex animations require the command line or manipulation of the coordinates outside PyMOL

Camera Loop

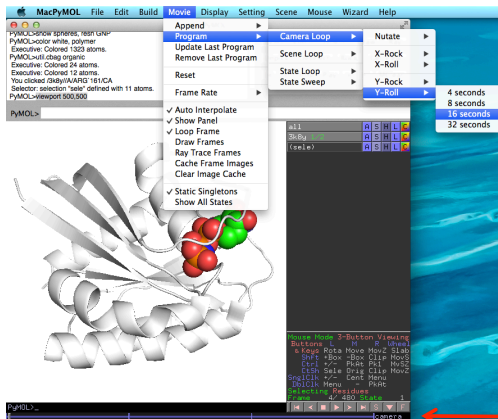
Demo, Play along

Write the script to set up the image,
copy-paste into command line:

```
viewport 500,500
bg_color white
load pdb/3K8Y.pdb
select lig, resn GNP
select prot, polymer
hide all
show cartoon
show spheres, lig
color white, polymer
util.cbag lig
```

Select from pull-down menu:

Movie:Program:Camera Loop:Y-Roll:16 sec



Movie Timeline

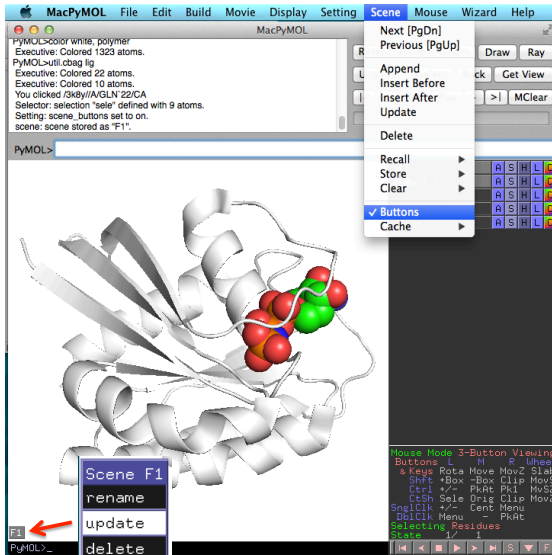
(shown only if “Show Panel” is selected)

use movie control panel to preview the movie:



Try out the effects of other “Camera Loop” options: Nutate, Rock, Roll around X and Y axis

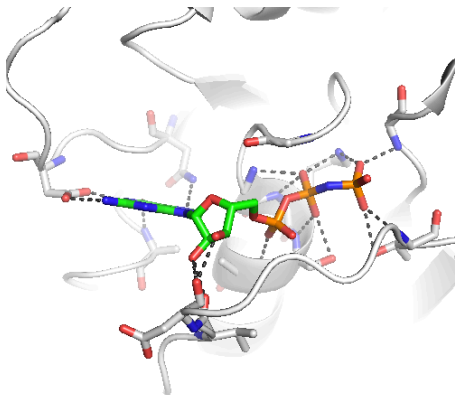
Store Scene for Future Use



Scene buttons can be
renamed and reordered
The first 12 scenes are
mapped to function keys
F1-F12

Right-click for scene button
context menu

Camera Loop



**add to the script and
paste into command line:**

```
hide spheres, lig  
show sticks, lig  
select pocket, br. (prot within 3.2 of lig)  
show sticks, pocket  
util.cbaw pocket  
distance hbond, lig, pocket, 3.2, 2  
hide label, hbond  
color gray20, hbond  
orient lig  
rot y, 180
```

**correct zoom and orientation
to nicely shows the ligand**

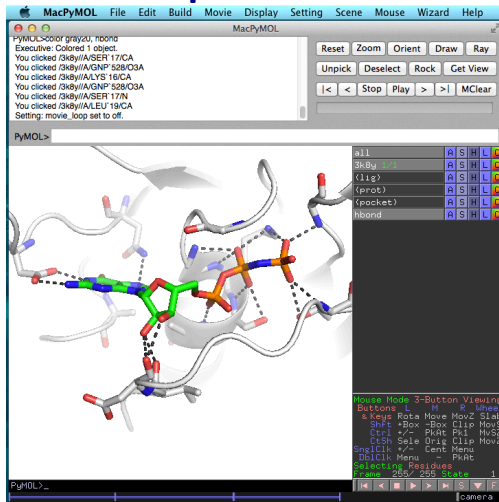
Store to scene F2

select from pull-down menu:

Movie: Reset

Movie: Camera Loop: Nutate: 30 deg. over 12 sec

Scene Loop



Putting the two scenes together:

select from pull-down menu:

Movie: Reset

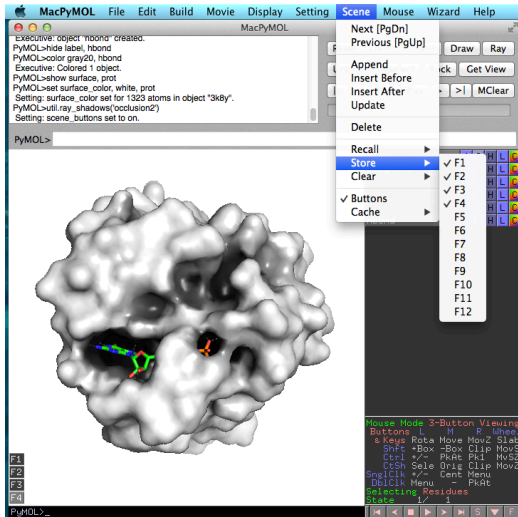
Menu: Scene:Recall:F1

Movie:Program:Scene Loop:Steady:4 sec

The speed of the transition is set by:
 "set scene_animation_duration"
 default is 2.25s

try out different values!

Scenes are your Story Board



```
viewport 500,500
bg_color white
load pdb/3k8y.pdb
select lig, resn GNP
select prot, polymer
hide all
show cartoon
show spheres, lig
color white, polymer
util.cbag lig
### save scene (F1) ###
orient lig
hide spheres, lig
show sticks, lig
select pocket, br. (prot within 3.2 of lig)
show sticks, pocket
util.cbaw pocket
distance hbond, lig, pocket, 3.2, 2
hide label, hbond
color gray20, hbond
### optimize zoom and orientation ###
### save scene (F2) ###
show surface, prot
set surface_color, white, prot
util.ray_shadows('occlusion2')
### save scene (F3) ###
zoom out
### save scene (F4) ###
```

} add
and
apply

Save script as: MyMovie1.pml

PyMOL Show

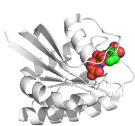
- Scenes are simple PyMOL memory snapshots, in which PyMOL saves the current colors, representations and camera position.
- Scenes can be given informative names and reordered by dragging the scene buttons.
- Settings are not saved, changing settings can affect all previously saved scenes in the show
- A PyMOL show is a series of Scenes that the user can navigate through by pressing the pgUp and pgDown keys
- The transition between two scenes is made by a smooth camera animation
- A scene loop sweeps through all stored scenes, the speed is determined by “set scene_animation_duration”, default 2.25 s
- For better control of the movie, use scenes as Key frames and control the timing through the timeline

Planning your Animation – The Storyboard

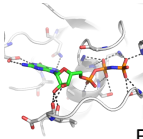
Plan your story: what message do you want to convey to the viewer?

Set up the key images for your movie: what do you want to show?

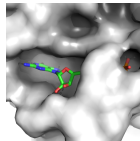
Save the key images as scenes



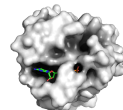
F1



F2



F3



F4

How do you want to transition from one view to the next?

360° rotation
around y-axis

6 sec

zoom in
on ligand

6 sec

rock x

12 sec

show
surface

rock x

12 sec

zoom out
to full protein

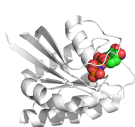
6 sec

360° rotation
around y-axis

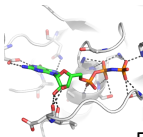
6 sec

Plan the timing – is there a narration?

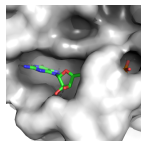
Putting the Scenes together



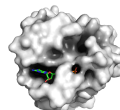
F1



F2



F3



F4

360° rotation
around y-axis

zoom in
on ligand

nutate

show
surface

nutate

zoom out
to full protein

360° rotation
around y-axis

Scene:Recall:F1

Movie:Program:Camera Loop:Y-Roll:16s

Movie:Append:12s

right-click on end of timeline and select: Store with scene F2

Movie:Program:Camera Loop:Nutate:30 degrees over 4s

Movie:Program:Camera Loop:Nutate:30 degrees over 8s

right-click on end of timeline and select: Store with scene F3

Movie:Program:Camera Loop:Nutate:30 degrees over 8s

Movie:Program:Camera Loop:Nutate:30 degrees over 4s

Movie:Append:12s

right-click on end of timeline and select: Store with scene F4

Movie:Program:Camera Loop:Y-Roll:16s

Camera Motion:	Scene:
store	F1
store with scene	F2
store with state	F3
clear	F4
reset camera motions	
purge entire movie	
smooth key frames	
interpolate	
reinterpolate	
uninterpolate	

play movie

Changing the Timing of a Transition

<ctrl> left drag mouse
to the right (**green**):
These frames get added
to the timeline, this phase
of the movie is lengthened

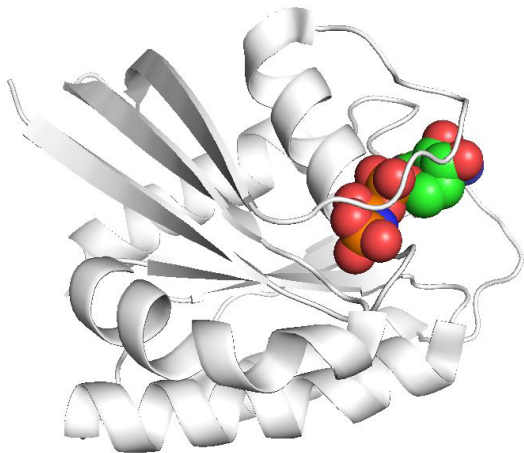


<ctrl> left drag mouse
to the left (**red**):
These frames get deleted
from the timeline,
this phase of the movie
is shortened

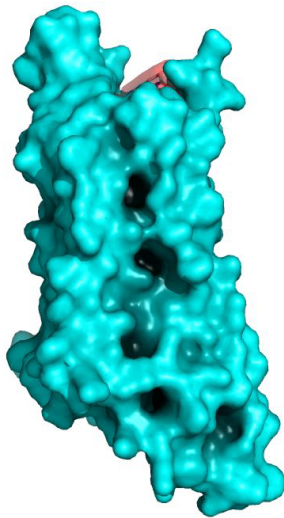
Play the movie: altering the timing of nutation does not work,
motion gets jerky – redo the movie with different time settings,
or using rock instead of nutation

Movie: Cache Frame Images
Movie: Ray Trace Frames
File: Save Movie as: Quicktime Movie

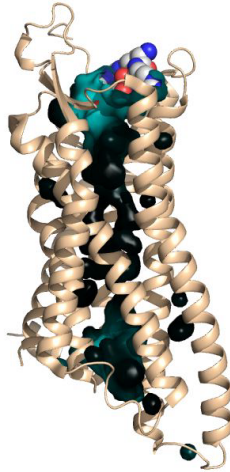
Save session as: MyMovie1.pse



Animating the Clipping Plane



Cavities

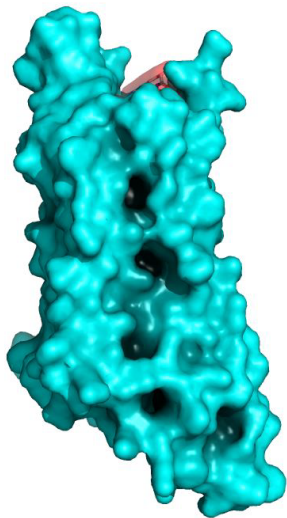


Putting the Movie together in Quicktime

- Changing settings such as transparency or surface cavity mode cannot be changed in the course of a Pymol movie, since they affect all states. Produce separate movies and put them together in Quicktime or an other movie editing program.
- Ray-tracing movie sequences can be quite time-consuming, if you want e.g. several rotations of the molecule without changing anything else, do a single rotation and insert it several times into the final movie

DEMO

The two Movies Combined



Types of Animations

Camera stationary Molecule immobile

Stationary figure

Stationary phase in movie

Camera stationary **State loop** Molecule mobile **State sweep**

- Multiple conformations of an object provided as multiple states in an object, e.g. MD

Trajectories, NMR, Morphs

- Moving rigid objects relative to each other

Camera mobile **Camera loop** Molecule immobile **Scene loop**

Animation achieved by changing the transformation matrix (get_view)

Rotate, Rock, Nutate, Zoom, Clip

with or without change in representation

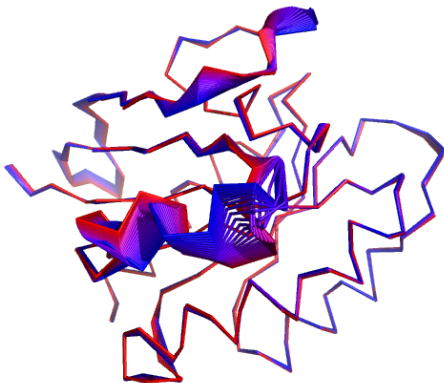
Camera mobile Molecule mobile

Transformation matrix and object coordinates (displayed state) both change during the animation

Smooth Transitions between representations

Fading in and out of surfaces, side chains etc.

Morph between the GDP and GTP bound Ras



```
viewport 500,500
bg_color white
load pdb/1Q21.pdb
load pdb/1QRA.pdb
align 1Q21, 1QRA, cycles=0, object=aln
morph QQQ, 1Q21, 1QRA
```

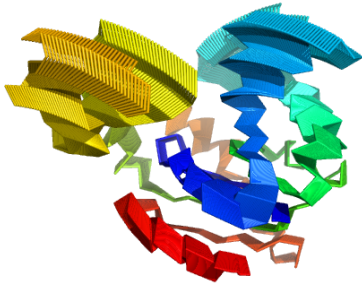
```
hide all
show ribbon
run py_scripts/spectrum_states.py
spectrum_states QQQ, ribbon, red purpleblue blue
```

@pml_scripts/rasMorph1.pml
requires py_scripts/spectrum_states.py

Movie: Program: State Sweep: ¼ Speed: No Pause

```
morph name, sele1 [, sele2 [, state1 [, state2 [, refinement [, steps [, method [, match]]]]]]]]]
```

Interpolating Between two Conformations



1ake, 4ake: two conformations of
adenylate kinase from *E. coli*

@pml_scripts/MakeMorph.pml

```
viewport 500,500
bg_color white
load pdb/1ake.pdb
load pdb/4ake.pdb
create 1akeA, 1ake and chain A and polymer
create 4akeA, 4ake and chain A and polymer
disable 1ake
disable 4ake
align 1akeA, 4akeA, cycles=0, object=aln
morph Make, 1akeA , 4akeA

hide all
cmd.spectrum("count",selection="Make",byres=1)
set all_states, on
ray
png figures/all_states.png

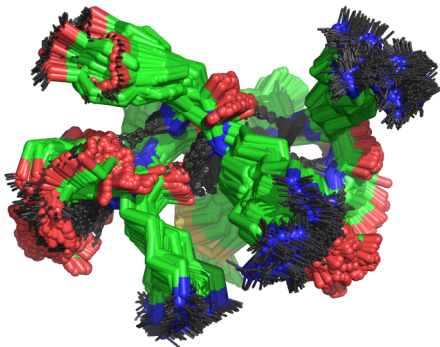
set all_states, off
```

Movie: Program: State Sweep: ¼ Speed: No Pause

Try other representations, e.g. cartoon, sticks, spheres

<http://www2.molmovdb.org>

Animating a MD Trajectory



@pml_scripts/Trajectory.pml

```
load pdb/SampleTrajectory.gz, ST
show sticks, ST and not h.
color gray25, hydrogen
show sticks, ST and not hydrogen
```

```
Action:Find:Polar Contacts: Within Selection
(distance hbond, all, all, 3.2, 2; hide label )
set dash_color, gray25
```

you can already run the trajectory as movie.
To see all states at once, choose:
Movie: Show all States.

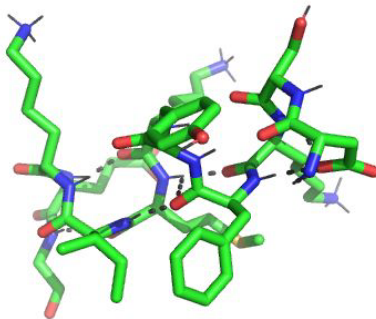
To suppress high frequency
motions, type:
smooth

to keep peptide from wandering type:
intra_fit ST

Movie:Program:State Loop:Full Speed: No Pause

```
smooth [ selection [, passes [, window [, first [, last [, ends]]]]]]]
```

Animate a Trajectory



Types of Animations

Camera stationary Molecule immobile

Stationary figure

Stationary phase in movie

Camera stationary **State loop** Molecule mobile **State sweep**

- Multiple conformations of an object provided as multiple states in an object, e.g. MD

Trajectories, NMR, Morphs

- **Moving rigid objects relative to each other**

Camera mobile **Camera loop** Molecule immobile **Scene loop**

Animation achieved by changing the transformation matrix (get_view)

Rotate, Rock, Nutate, Zoom, Clip

with or without change in representation

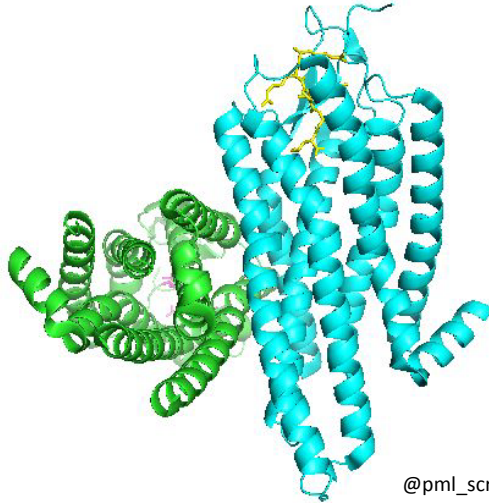
Camera mobile Molecule mobile

Transformation matrix and object coordinates (displayed state) both change during the animation

Smooth Transitions between representations

Fading in and out of surfaces, side chains etc.

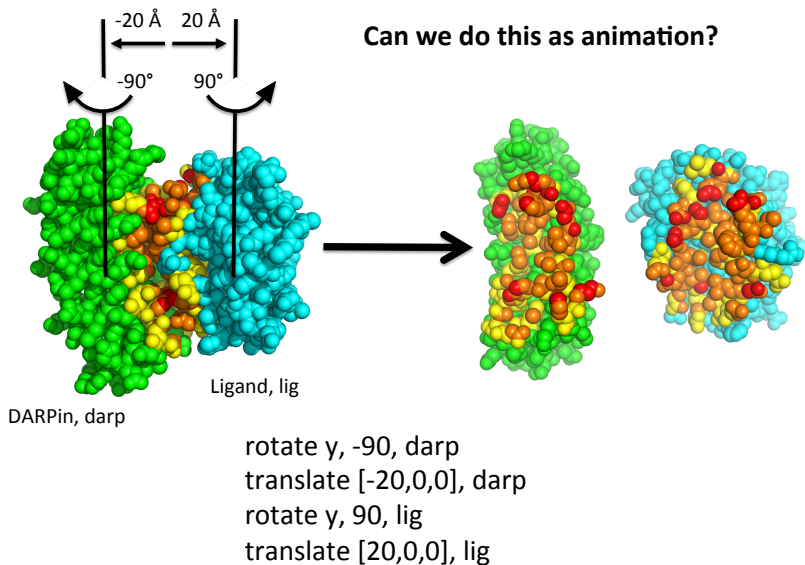
Animating a 3D-Superposition



@pml_scripts/Alignment1.pml

Cannot be done with a scene animation, should theoretically work by timeline editing, it works for dragging objects around, but not for alignment=> Do it by scripting

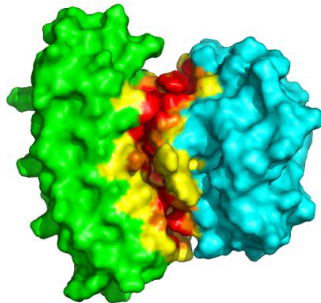
Example: Looking at a Binding Interface



Moving two Objects Relative to Each Other

Animation in object coordinates, not camera coordinates

DARPin-BCL-W complex



@pml_scripts/darpins.pml

```
translate vector [,selection [,state [,camera [,object ]]]]  
rotate axis, angle [,selection [,state [,camera [,object [,origin]]]]]
```

Doing it all in a script

open **DARPin.s.pml** in text editor

load the coordinates

```
load pdb/S4K5B_B-C.pdb, Cmplx
extract darp, Cmplx and chain B
extract lig, Cmplx and chain C
delete Cmplx
```

change this
to load a
different
molecule

representation and coloring

```
viewport 500, 500
bg_color white
color green, darp
color cyan, lig
select C1, br. ((darp within 5.0 of lig) or (lig within 5.0 of darp))
color yellow, C1
select C2, ((darp within 5.0 of lig) or (lig within 5.0 of darp))
color orange, C2
select C3, ((darp within 3.6 of lig) or (lig within 3.6 of darp))
color red, C3
hide all
show surface
util.ray_shadows('occlusion1')
select none
```

change this
for a different
representation

setup PyMOL for movies

```
set matrix_mode, 1  
set movie_panel, 1  
set scene_buttons, 1  
set cache_frames, 1  
set move_auto_interpolate, 1  
config_mouse three_button_motions, 1
```

Settings affecting movie production and manual editing

initialize the movie

```
mset 1 x840
```

The entire movie will be 840 frames long (28 s)

nothing happens in frames 1-60

```
frame 1  
mview store, object=darp  
mview store, object=lig  
frame 60  
mview store, object=darp  
mview store, object=lig
```

darp and lig do not get changed between being stored in frame 1 and stored in frame 60

move darp and lig apart in frame 60-120

```
frame 120
rotate y, -90, object=darp
translate [-20,0,0], object=darp
mview store, object=darp
rotate y, 90, object=lig
translate [20,0,0], object=lig
mview store, object=lig
```

Darp and lig are rotated and translated in different directions before stored in frame 120. These transformations are evenly spread out over frames 61 to 120

nothing happens in frames 120-300

```
frame 300
mview store, object=darp
mview store, object=lig
```

Darp and lig do not get changed between being stored in frame 120 and stored in frame 300

move darp and lig together in frame 330-360

```
frame 360
translate [20,0,0], object=darp
rotate y, 90, object=darp
mview store, object=darp
translate [-20,0,0], object=lig
rotate y, -90, object=lig
mview store, object=lig
```

Note that the order of “rotate” and “translate” have to be reversed to undo the transformation that occurred between frames 60 and 120

frames 360-420

rotate complex around x-axis

```
frame 420
rotate x,-90,object=darp
mview store, object=darp
rotate x,-90,object=lig
mview store, object=lig
```

nothing happens in frames 420-480

```
frame 480
mview store, object=darp
mview store, object=lig
```

move darp and lig apart in frame 480-540

```
frame 540
rotate z, 90, object=darp
translate [-20,0,0], object=darp
mview store, object=darp
rotate z, -90, object=lig
translate [20,0,0], object=lig
mview store, object=lig
```

nothing happens in frames 540-660

frame 660
mview store, object=darp
mview store, object=lig

move darp and lig together in frame 660-720

frame 720
translate [20,0,0], object=darp
rotate z, -90, object=darp
mview store, object=lig
translate [-20,0,0], object=lig
rotate z, 90, object=lig
mview store, object=lig

nothing happens in frames 720-780

frame 780
mview store, object=darp
mview store, object=lig

frames 780-840

rotate complex back into initial orientation

frame 840

rotate x, 90,object=darp

mview store, object=darp

rotate x, 90,object=lig

mview store, object=lig

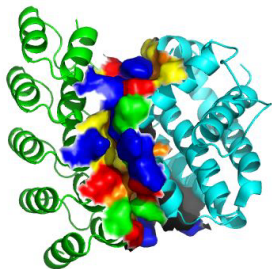
play movie

mplay

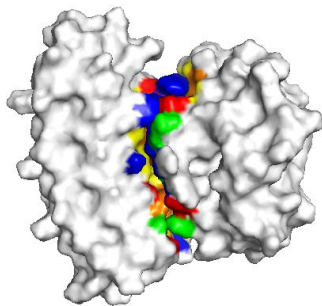
Changing the Representation of the DARPins

define representation

```
viewport 500, 500
bg_color white
color white, all
color orange, resn Tyr+Phe+Trp
color yellow, resn Leu+Ile+Val+Pro+Ala+Met+Cys
color green, resn Ser+Thr+Asn+Gln
color red, resn Asp+Glu
color cyan, resn His
color blue, resn Arg+Lys
color limon, resn Gly
select C1, br. ((darp within 5.0 of lig) or (lig within 5.0 of darp))
color white, not C1
hide all
show surface, C1
show cartoon
set cartoon_color, green, darp
set cartoon_color, cyan, lig
util.ray_shadows('occlusion1')
select none
```



@pml_scripts/darpins1.pml



@pml_scripts/darpins2.pml

Showing Movies in Powerpoint

Movies do not become part of the Powerpoint file!!!

- Create a folder for your ppt file
- In this folder, create a subfolder for the movie files
- Now insert your movies into your ppt file
- When copying your ppt presentation to a memory stick or a different computer, always copy the entire folder containing both your presentation and your movies!

The safest method is to create your presentation on the laptop you intend to use for the presentation!

Avoid switching presentations from Win to Mac or vice versa !!!