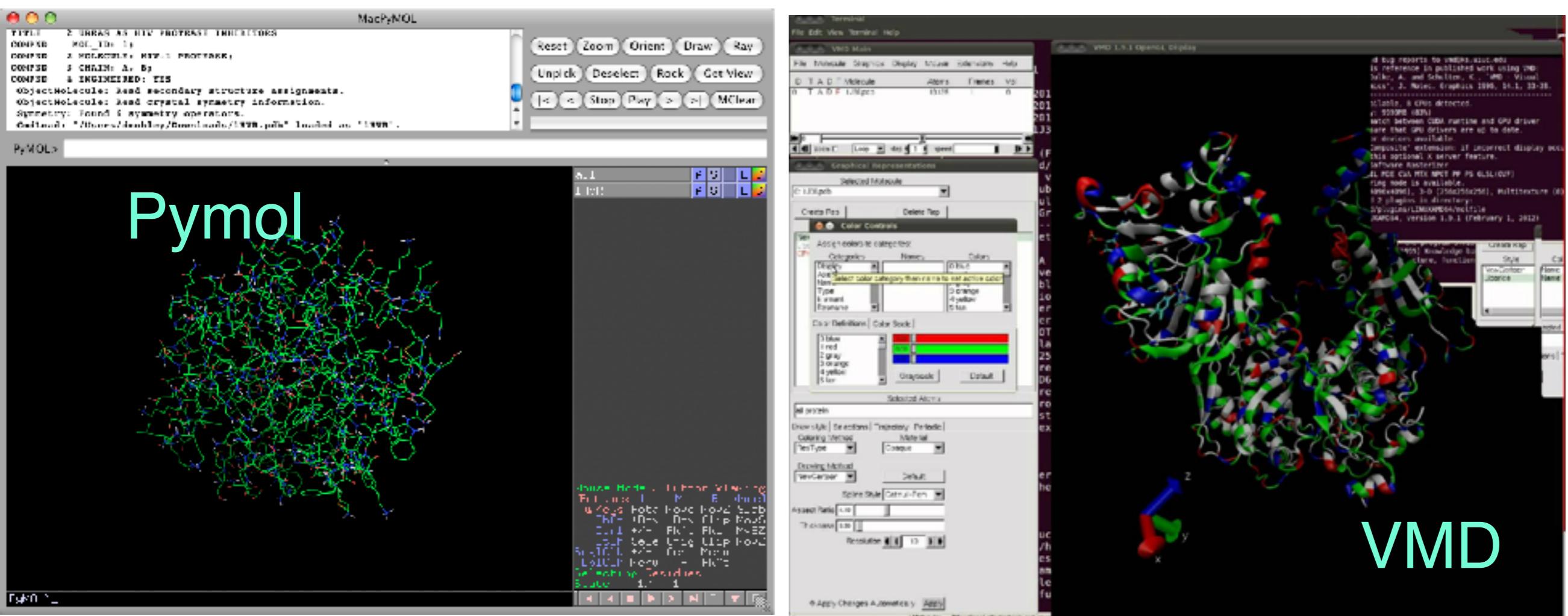


# Notes/Logistics

- Today: Visualization/movies
- Thursday: MDTraj, OpenMM, ...
- Tuesday: Normal modes unless other requests
- Thursday: TBD
- The following week: Lectures by Lim, Gill on drug discovery process, free energy calculations

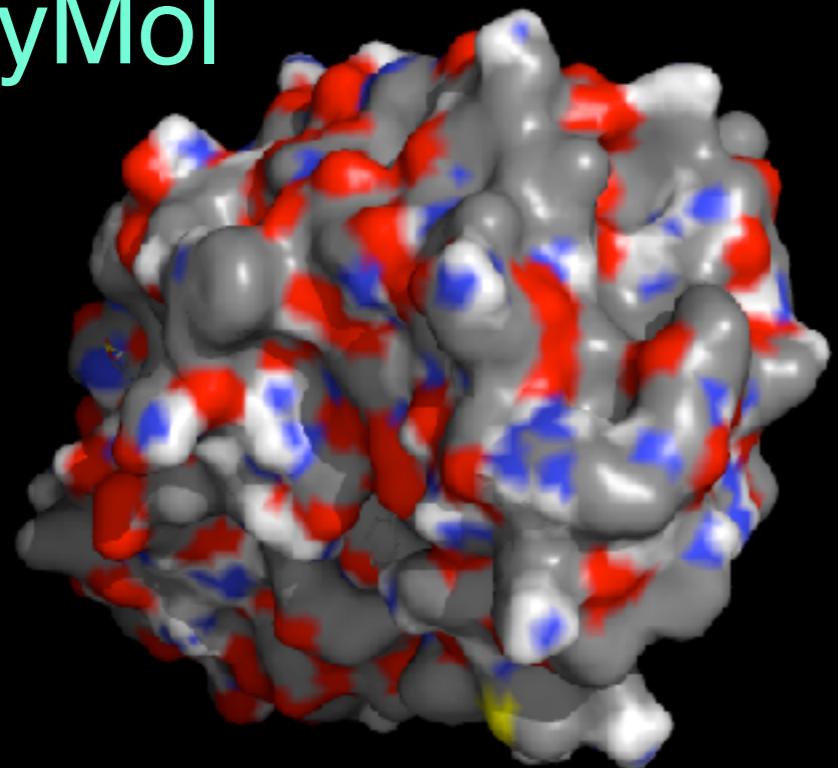
# To work in 3D, you should become familiar with a general purpose viewer



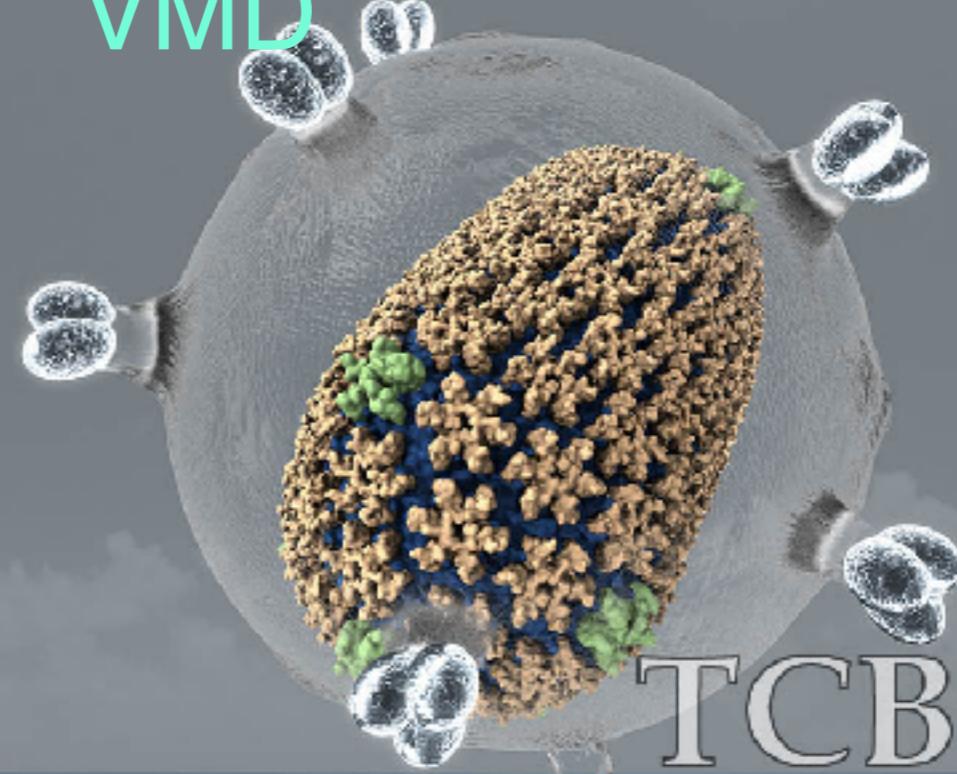
- Pymol and VMD are good, free, general-purpose viewers
- PyMol is python based/VMD supports TCL & Python
  - Can run Python code
  - Easily extensible and scriptable

# Both make high quality, ray-traced images

PyMol



VMD



TCBG

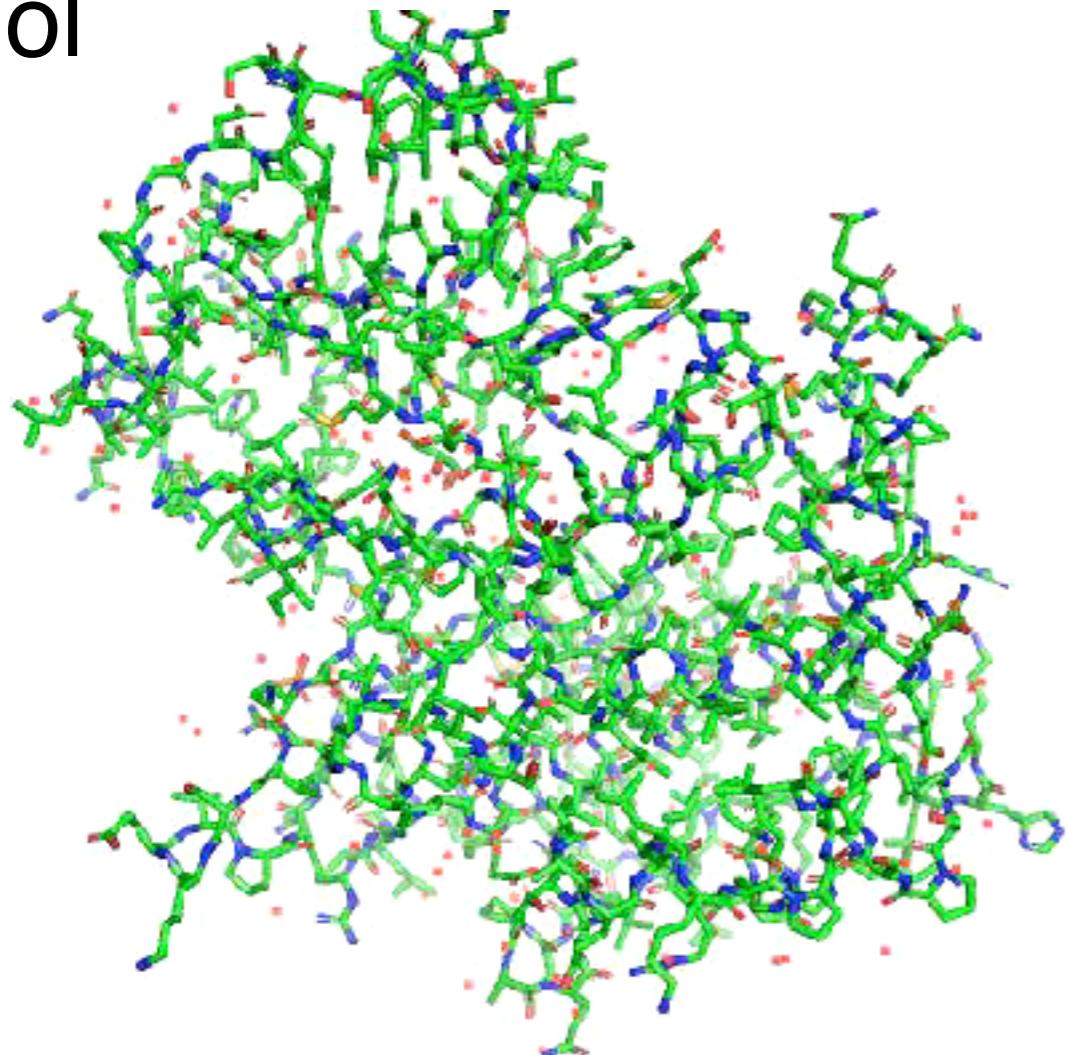
[www.ks.uiuc.edu](http://www.ks.uiuc.edu)

Today's lecture will be a **VMD tutorial**

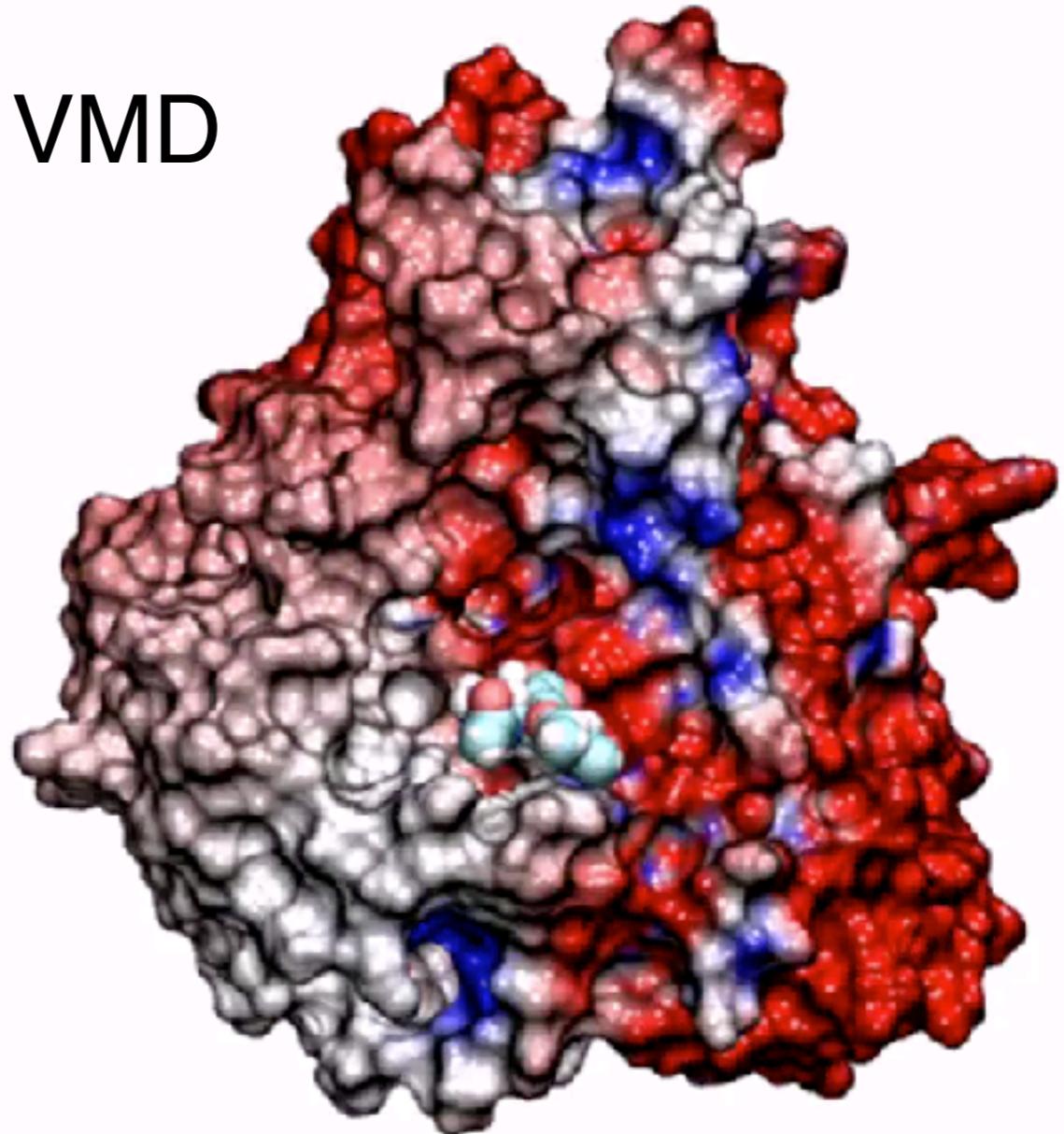
Note: Tutorial material for *PyMol* is also available

# Use PyMol for static structures and VMD for simulations

PyMol



VMD

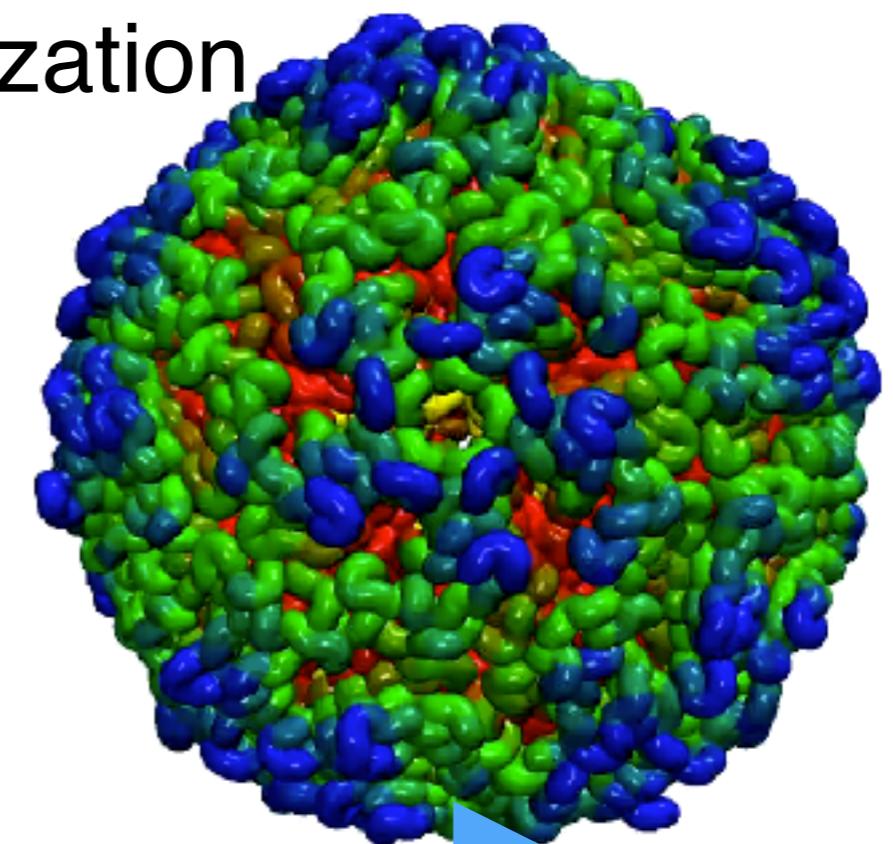
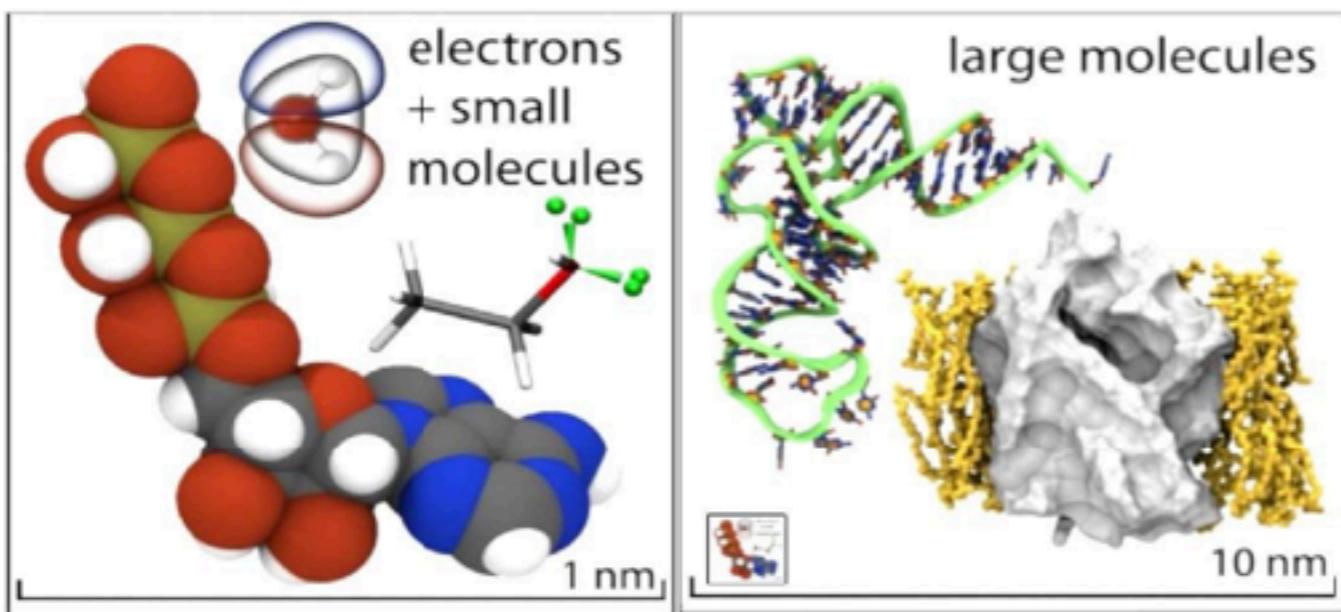


PyMol better for making movies of **static** structures  
VMD better for making movies of **simulated** structures

# VMD - “Visual Molecular Dynamics”

- Primarily menu driven, but allows scripting
- Trajectories are fundamental to VMD
- Support for very large systems (approaching billions of particles)
- Extensive GPU acceleration
- Parallel and remote analysis/visualization

*complete Satellite Tobacco Mosaic Virus was simulated (1 million atoms for 14 ns).*

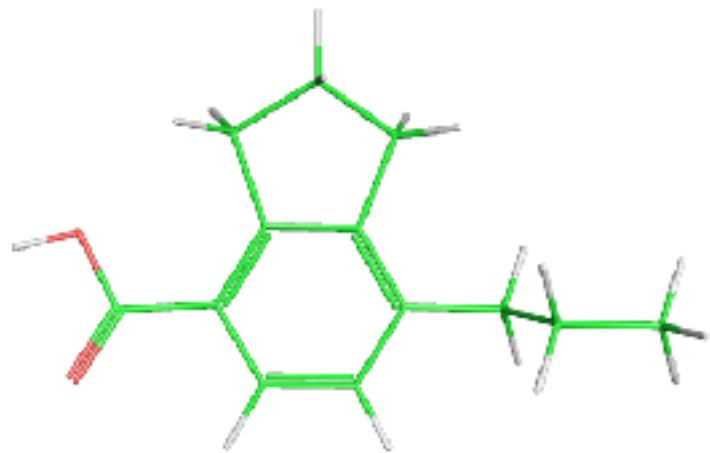


System size

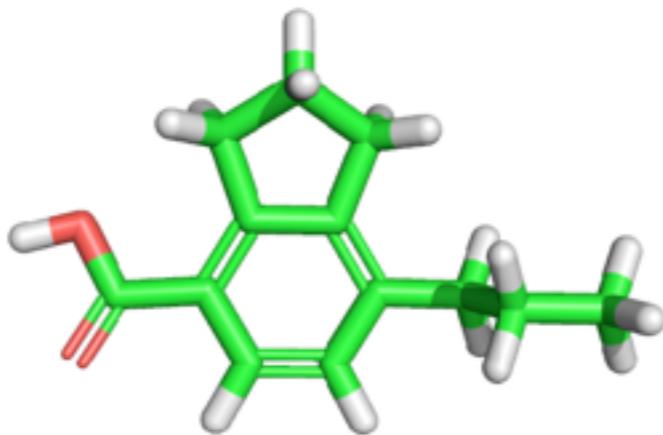
# Both have a pretty full set of features:

- Can build and edit molecules
- Build new proteins from protein sequence
- Molecular movies
  - Morphs and different views of static molecules
  - Viewing molecular simulations
- Structure editing tools, mutagenesis tools
- Structure alignments
- Identify hydrogen bonds, add text labels, etc.
- VMD adds other things:
  - Guided molecular dynamics simulations and training (QuickMD)
  - “Particle” systems and whole cells
  - Cryo-EM densities, volumetric data
  - Quantum chemistry calculations

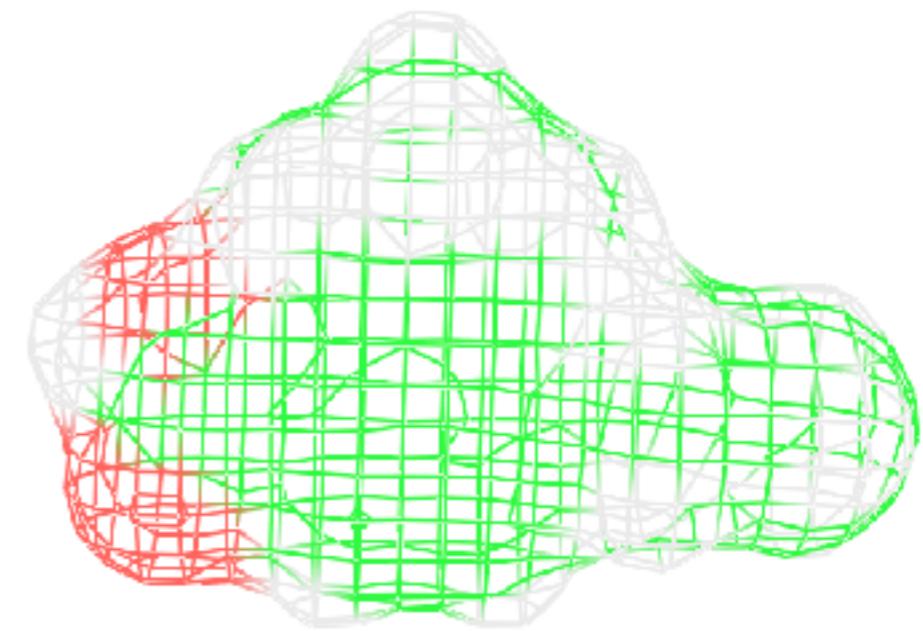
# There are many common representations of molecules



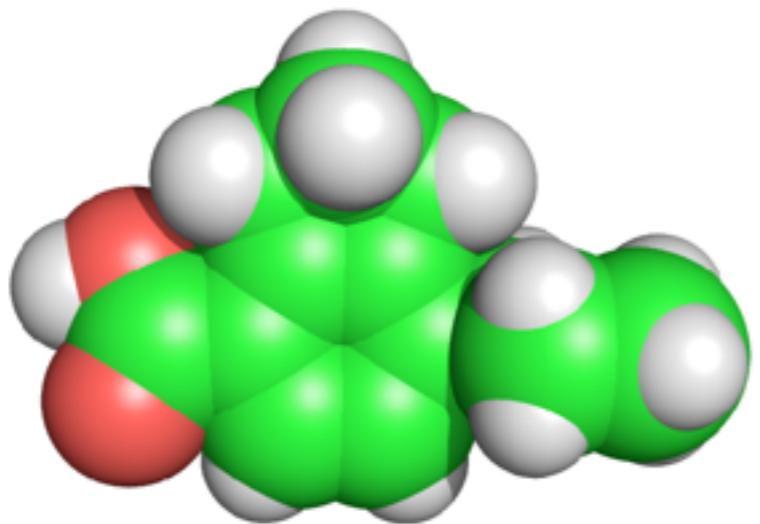
Lines



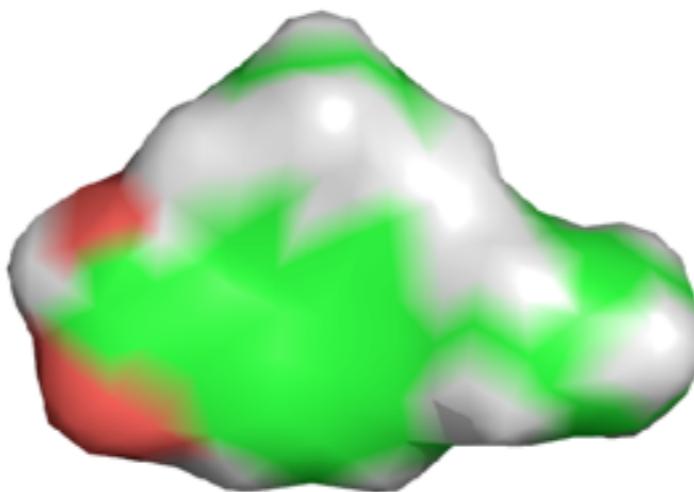
Licorice



Isosurface

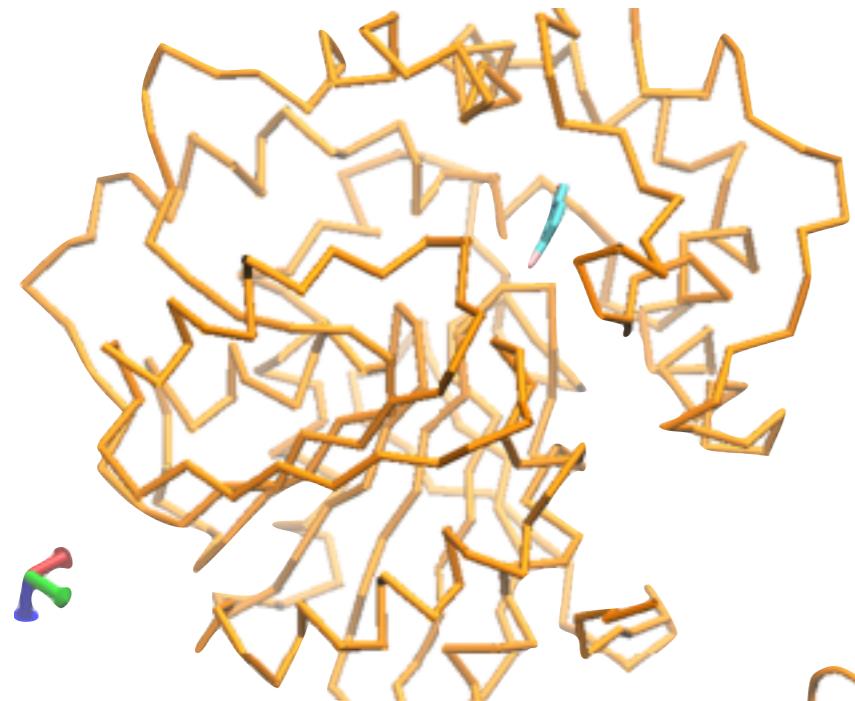


VDW



Surface

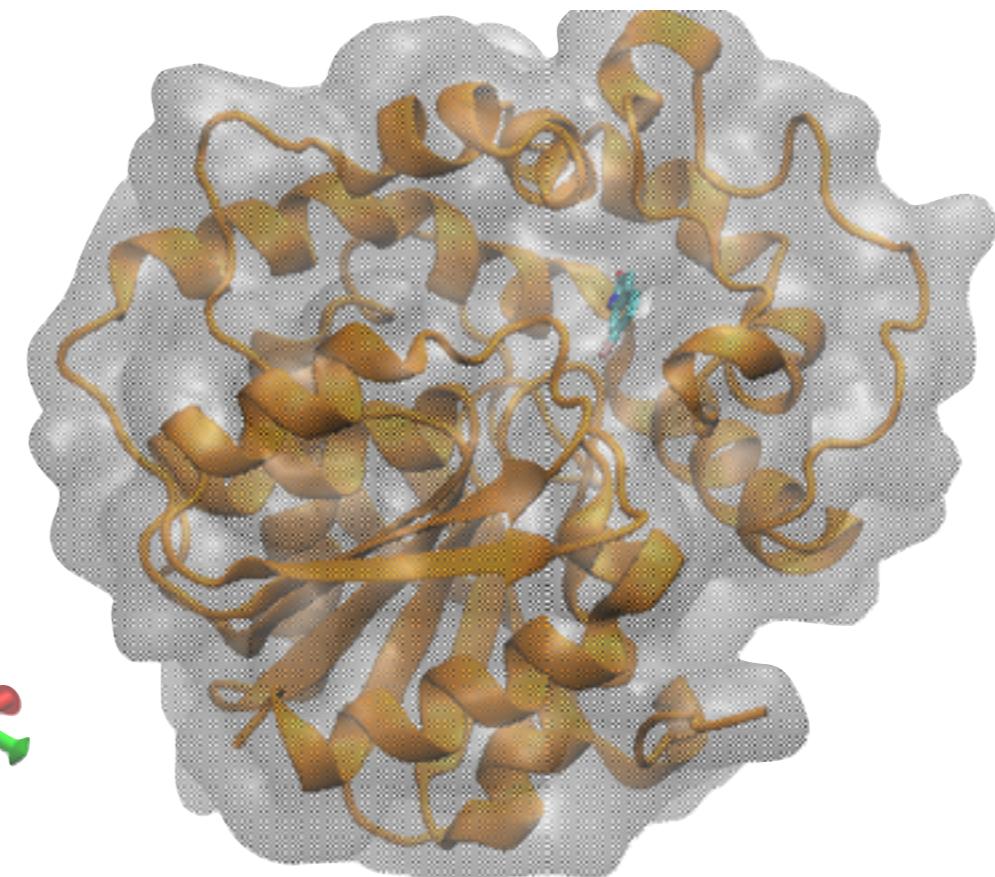
# Proteins add several additional common representations



Trace



NewCartoon



Surface

# Visualizing something in 3D requires a 3D structure, which can be modeled

- Many toolkits can take a 2D structure and automatically generate a 3D structure from it
- 3D structure generation typically requires at least energy minimization (which we will address in another lecture)
- There are many other tools for this
  - Some can use just chemical names as a starting point

# VMD Tutorial

Download: [http://www.ks.uiuc.edu/  
Research/vmd/](http://www.ks.uiuc.edu/Research/vmd/)

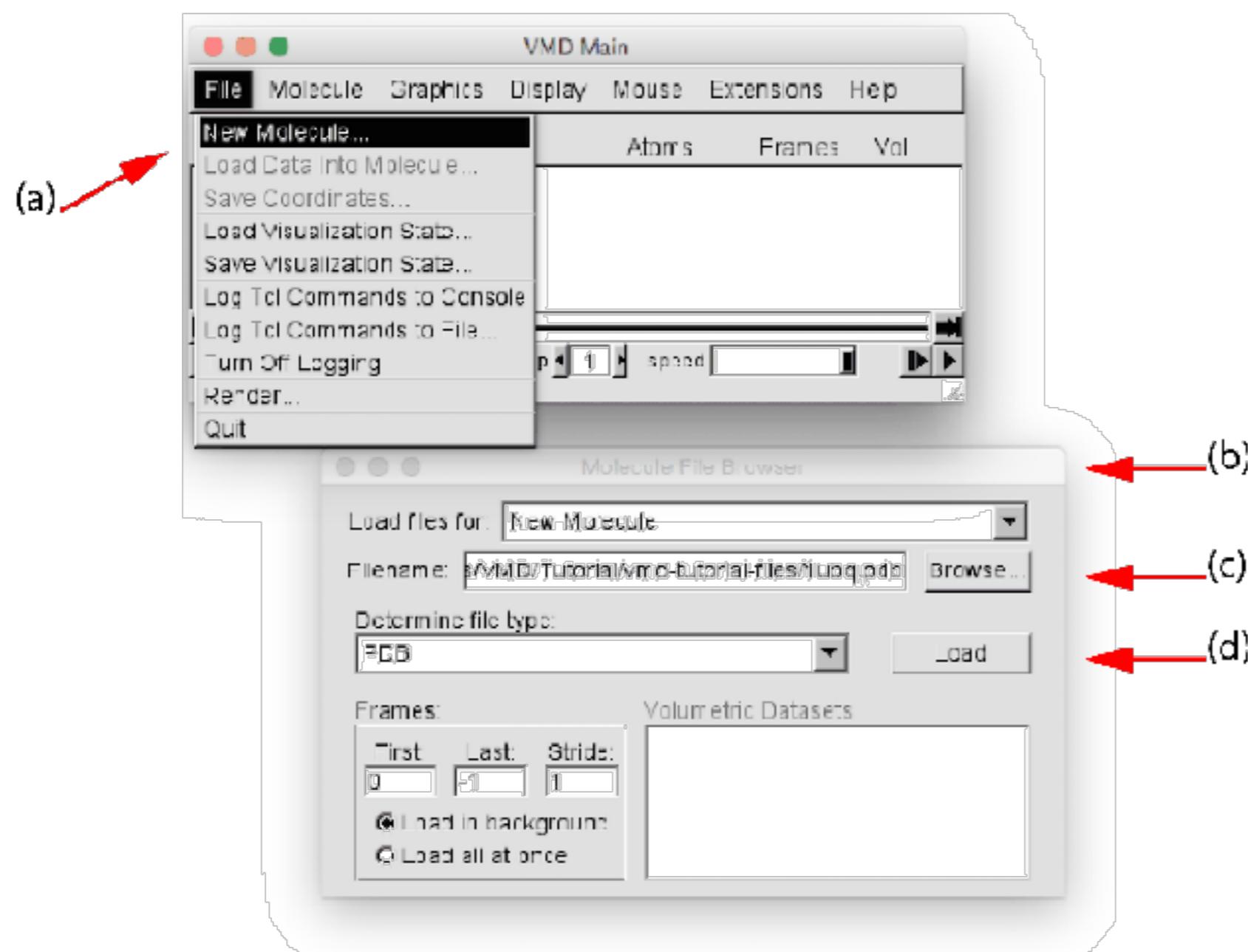
Tutorial: [http://www.ks.uiuc.edu/  
Training/  
Tutorials/vmd/tutorial-html/](http://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/)

# VMD has both command-line and menu-based versions of most commands

- **[Menus]** better for beginners
- **[Command-line]** for power users and for scripting
  - Can be faster than menus
  - Command-line is a Python2.7\* and Tcl ‘interpreter’
    - Launch VMD with `vmd gopython` \**if you compiled with python enabled*
    - Can write Tcl/Python scripts to run in VMD

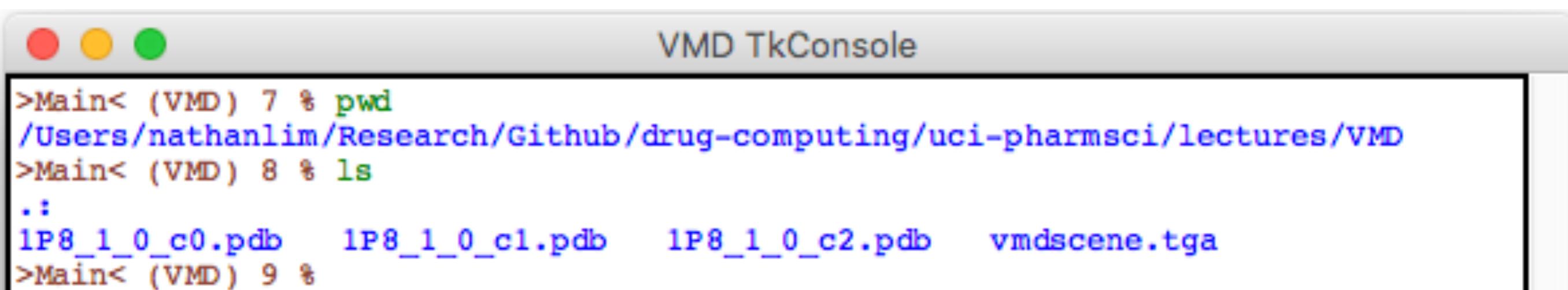
# Task: Loading a 3D structure (PDB)

- **[Cmd]:** `vmd 1P8_1_0_c0.pdb`
- **[Menu]:** *File* → *New Molecule* → *Browse* → *Load*



# Using the Tk Console allows you to run Tcl commands and limited BASH commands

- **[Menu]:** *Extensions* → *TK Console*
- Allows you to run Tcl commands and limited BASH commands
- Tip: Use *File* → *Log Tcl commands to console* to get the Tcl commands when performing menu actions

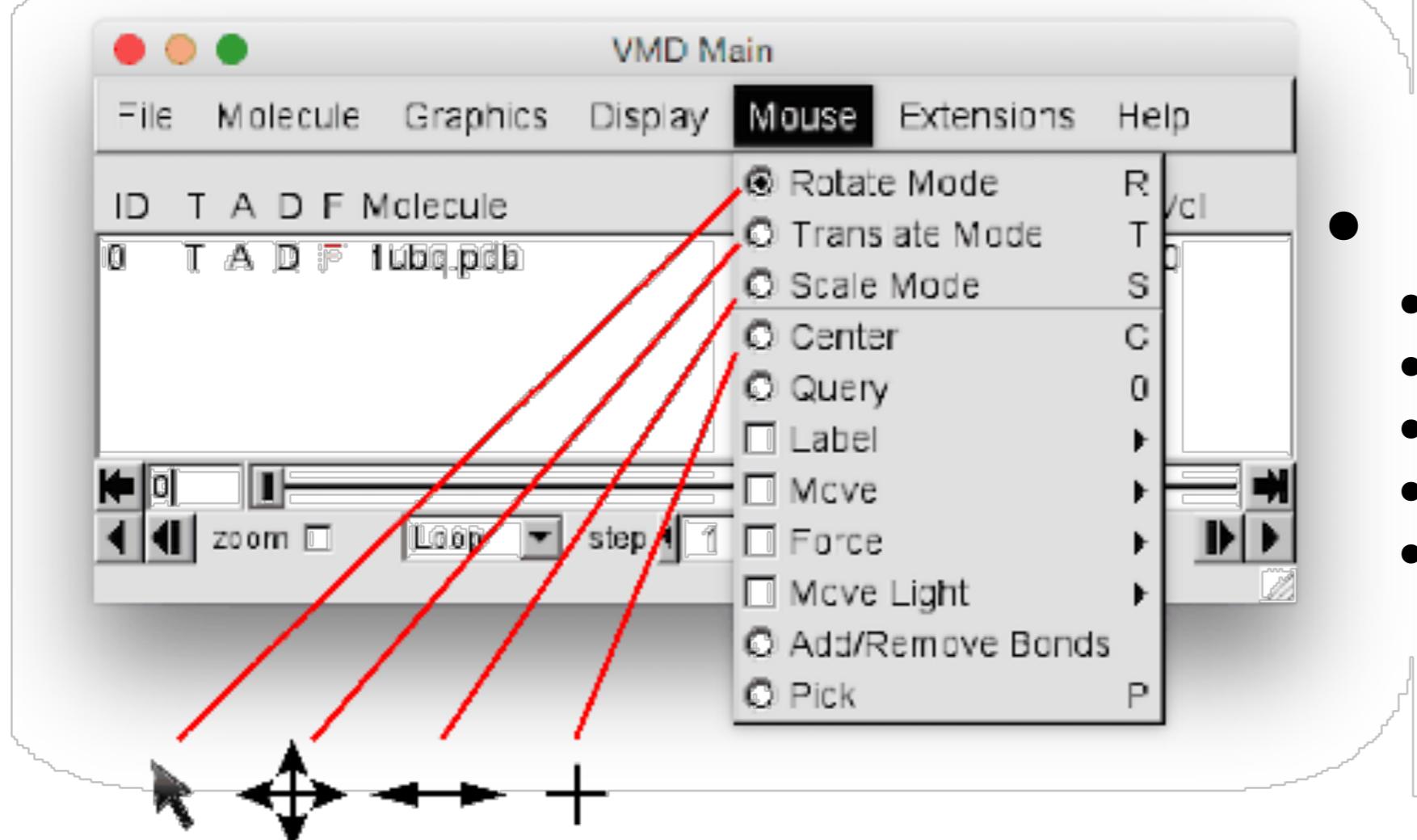


The screenshot shows a window titled "VMD TkConsole". The title bar has three colored circles (red, yellow, green) on the left. The main area is a terminal window displaying the following command-line session:

```
>Main< (VMD) 7 % pwd  
/Users/nathanlim/Research/Github/drug-computing/uci-pharmsci/lectures/VMD  
>Main< (VMD) 8 % ls  
.:  
1P8_1_0_c0.pdb  1P8_1_0_c1.pdb  1P8_1_0_c2.pdb  vmdscene.tga  
>Main< (VMD) 9 %
```

# Task: Molecule manipulation and display

- [Menu:] Mouse → *Rotate/Translate/Scale*

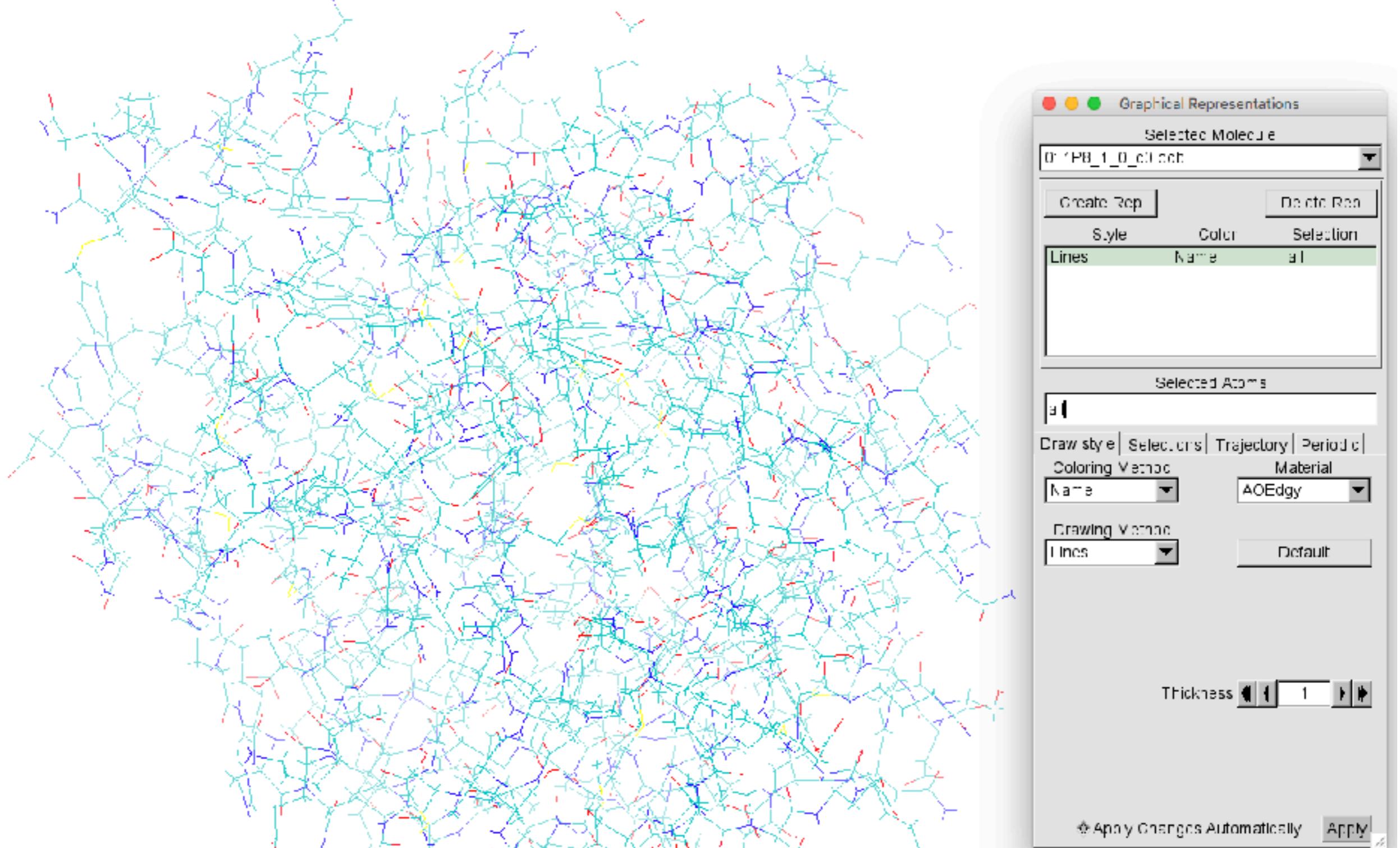


- **Keyboard shortcuts:**
  - Rotate (R)
  - Translate (T)
  - Scale (S)
  - Center (C)
  - Reset view (=)

- [Menu:] Display →
  - *Perspective*: near camera appear larger
  - *Orthographic*: preserves scale

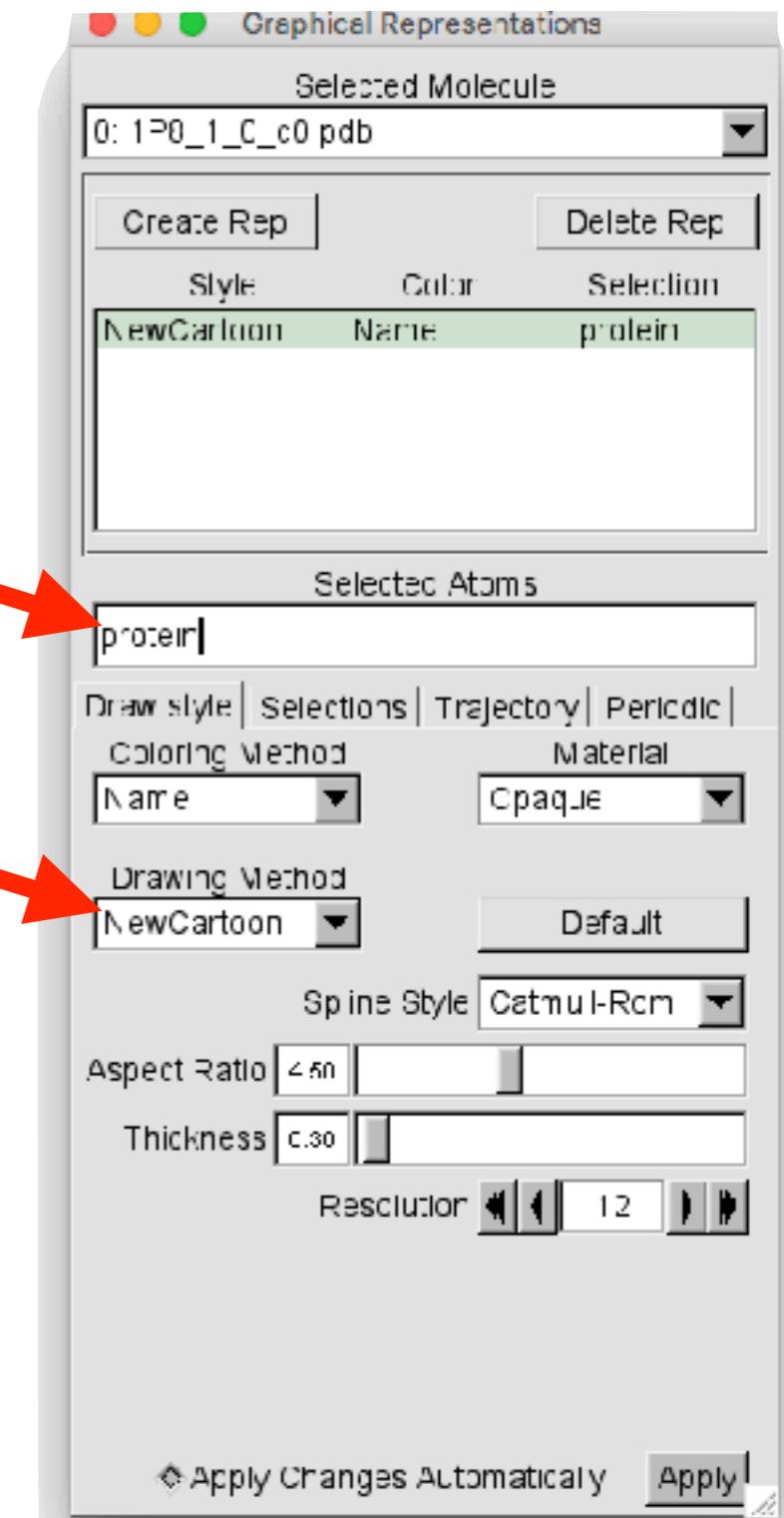
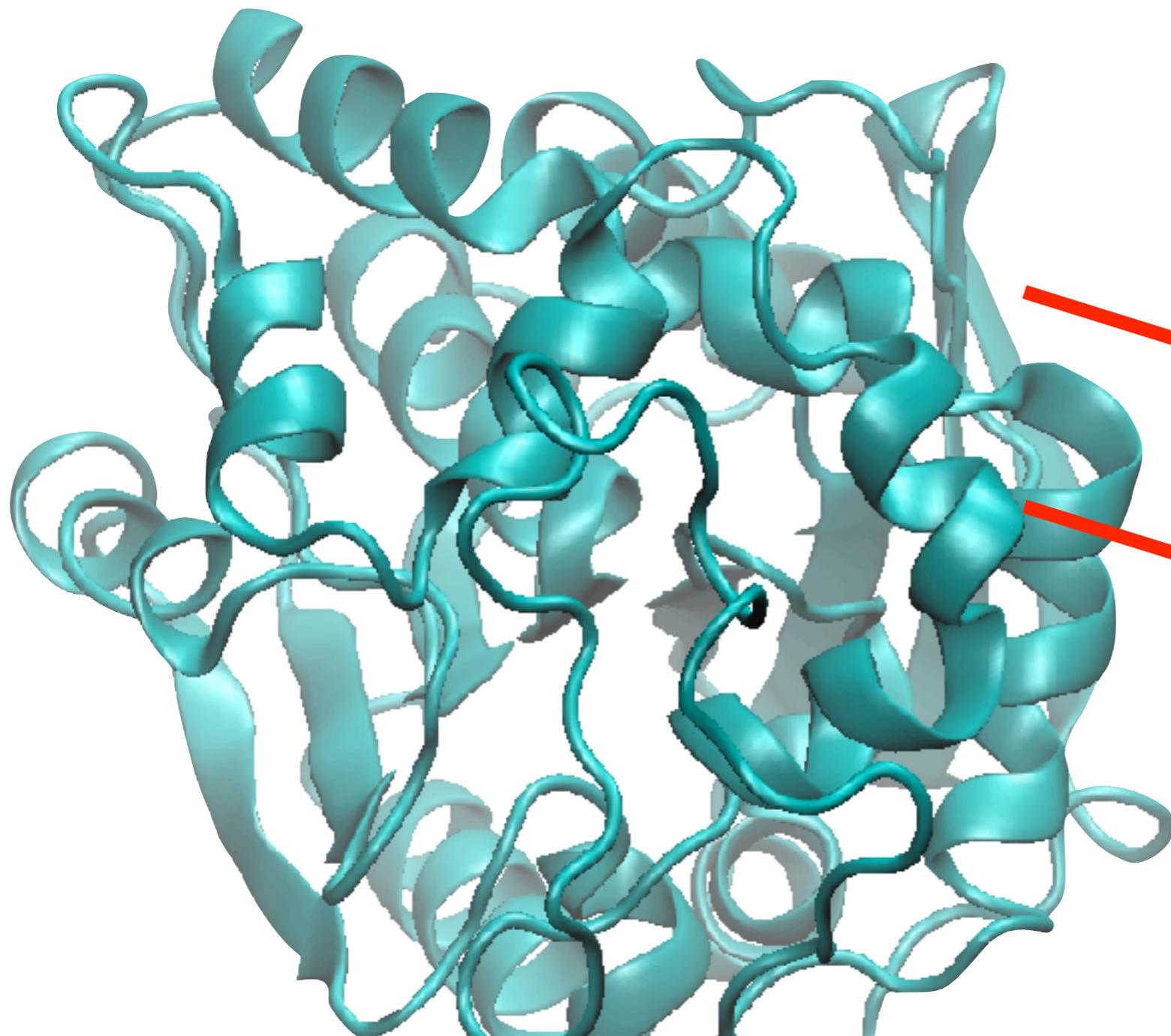
# Task: Changing the molecular rep.

- **[Menu:]** *Graphics* → *Representations*



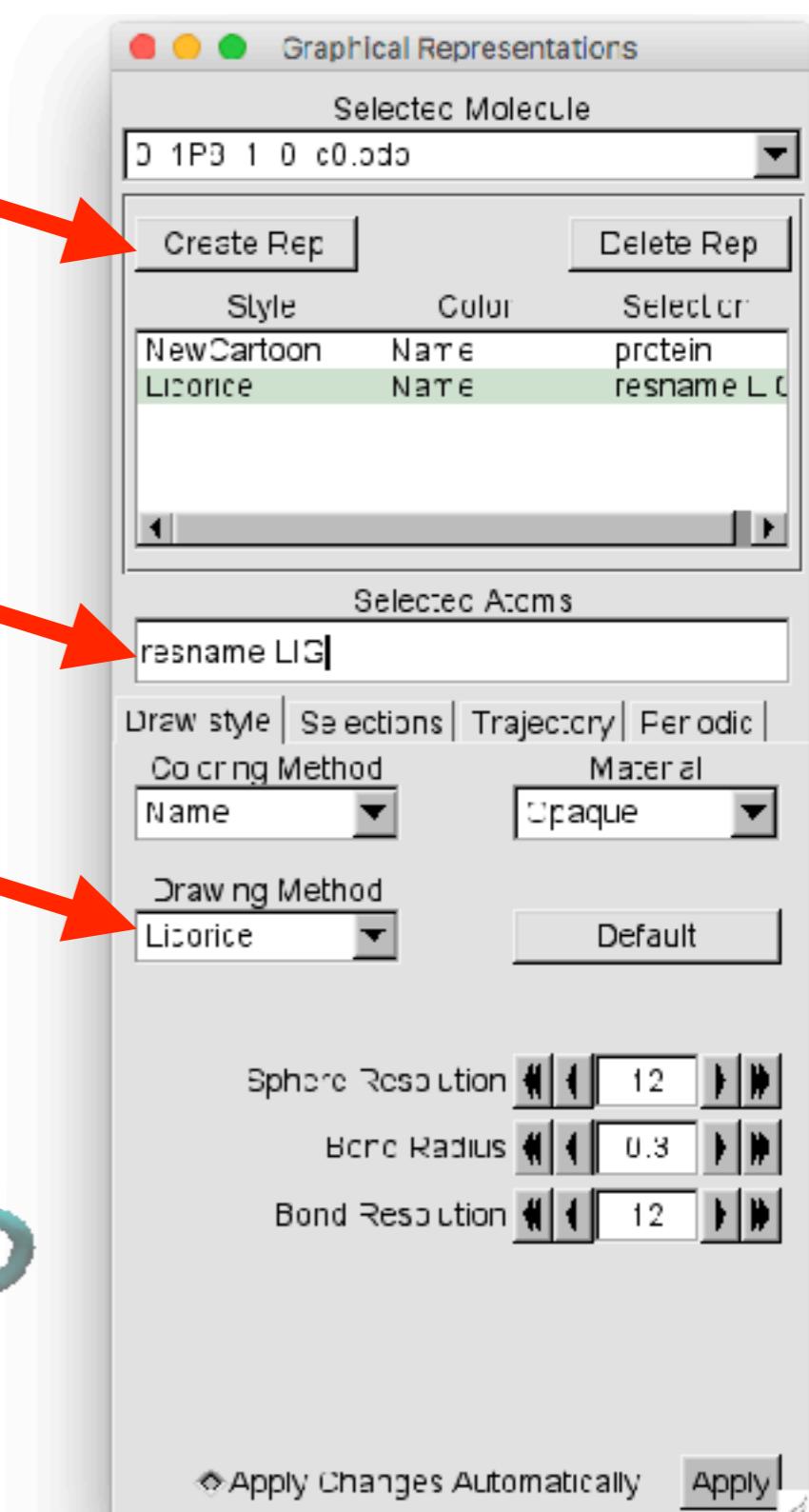
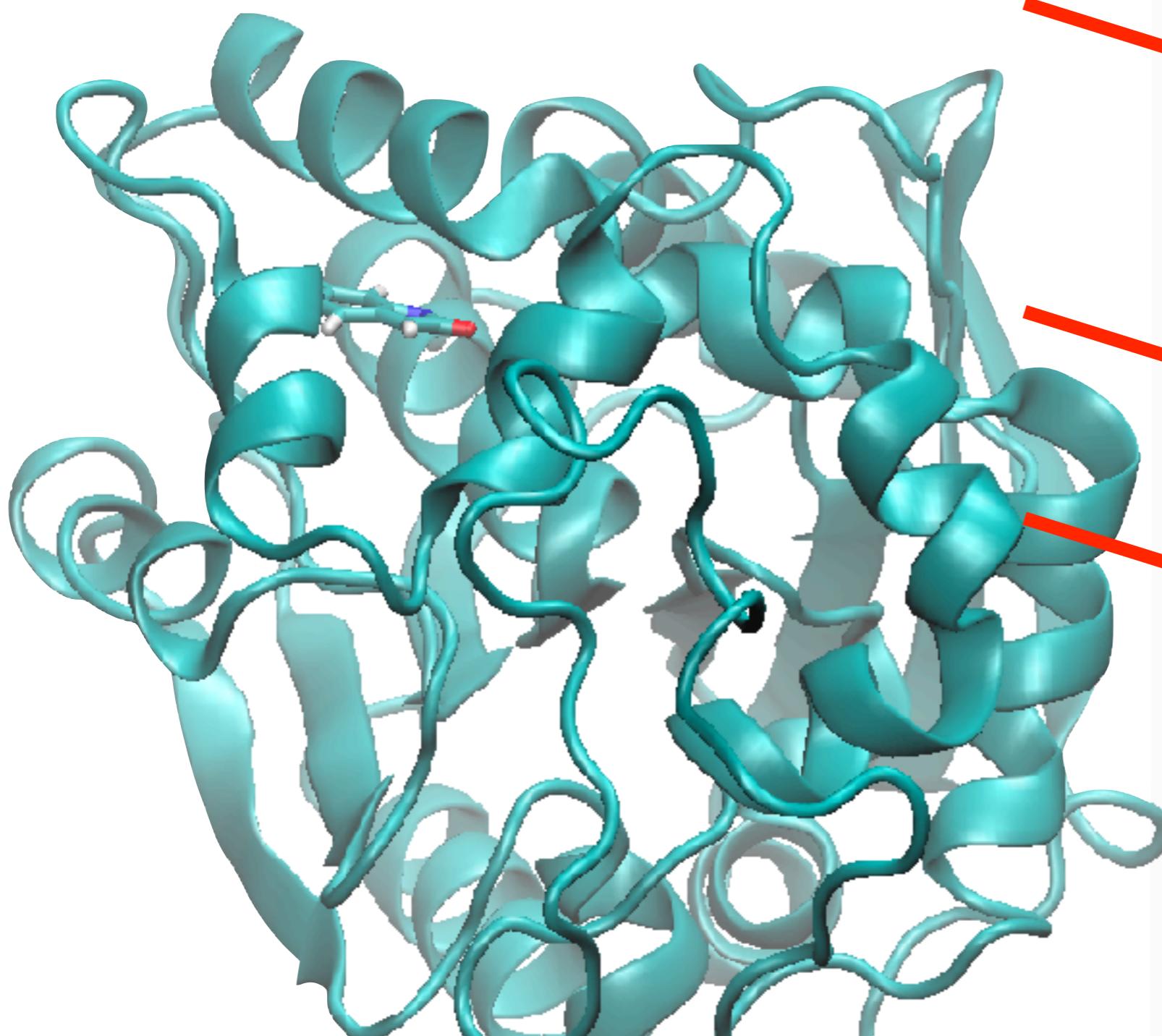
# Task: Displaying a protein in cartoon rep.

- Selection: `protein`
- Drawing Method: `NewCartoon`



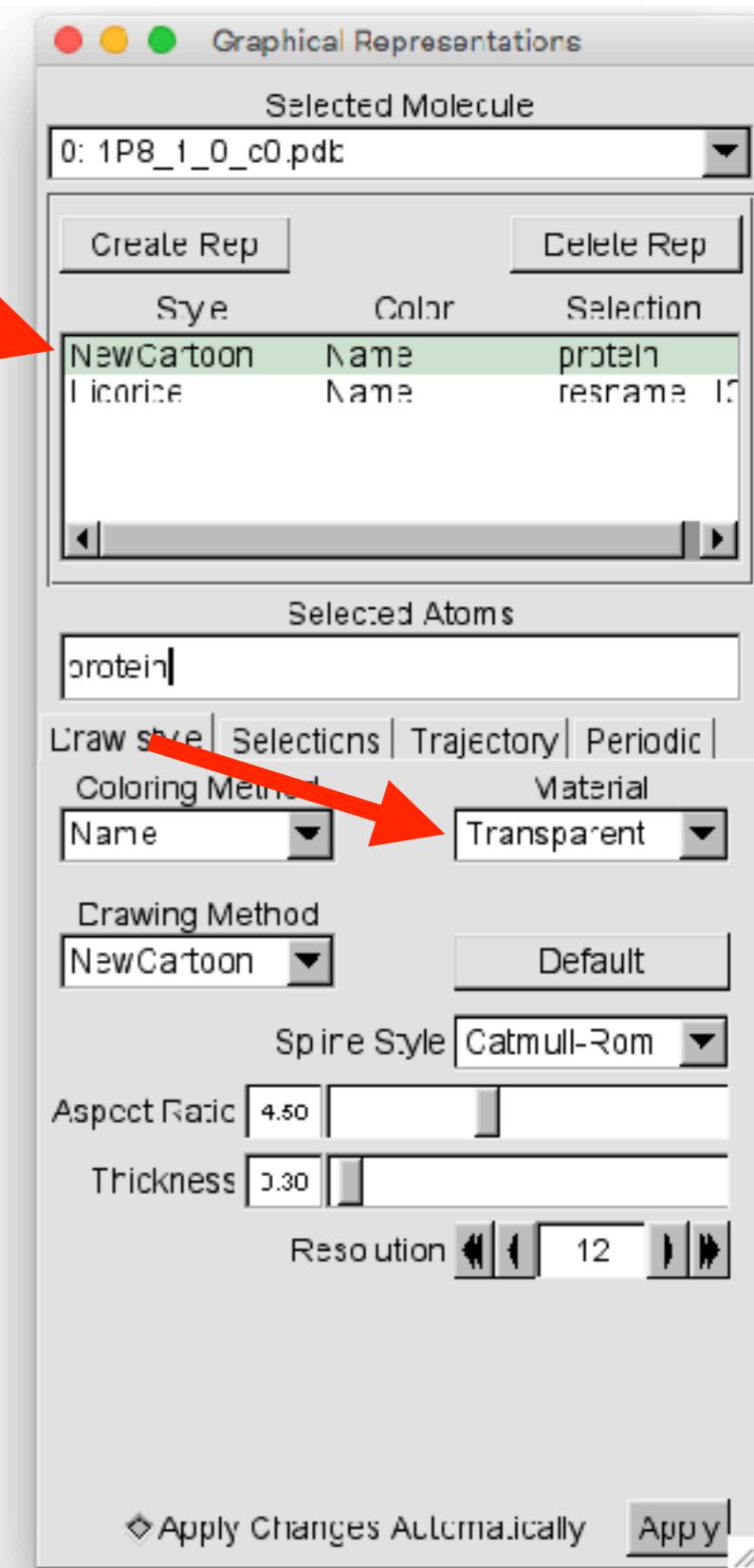
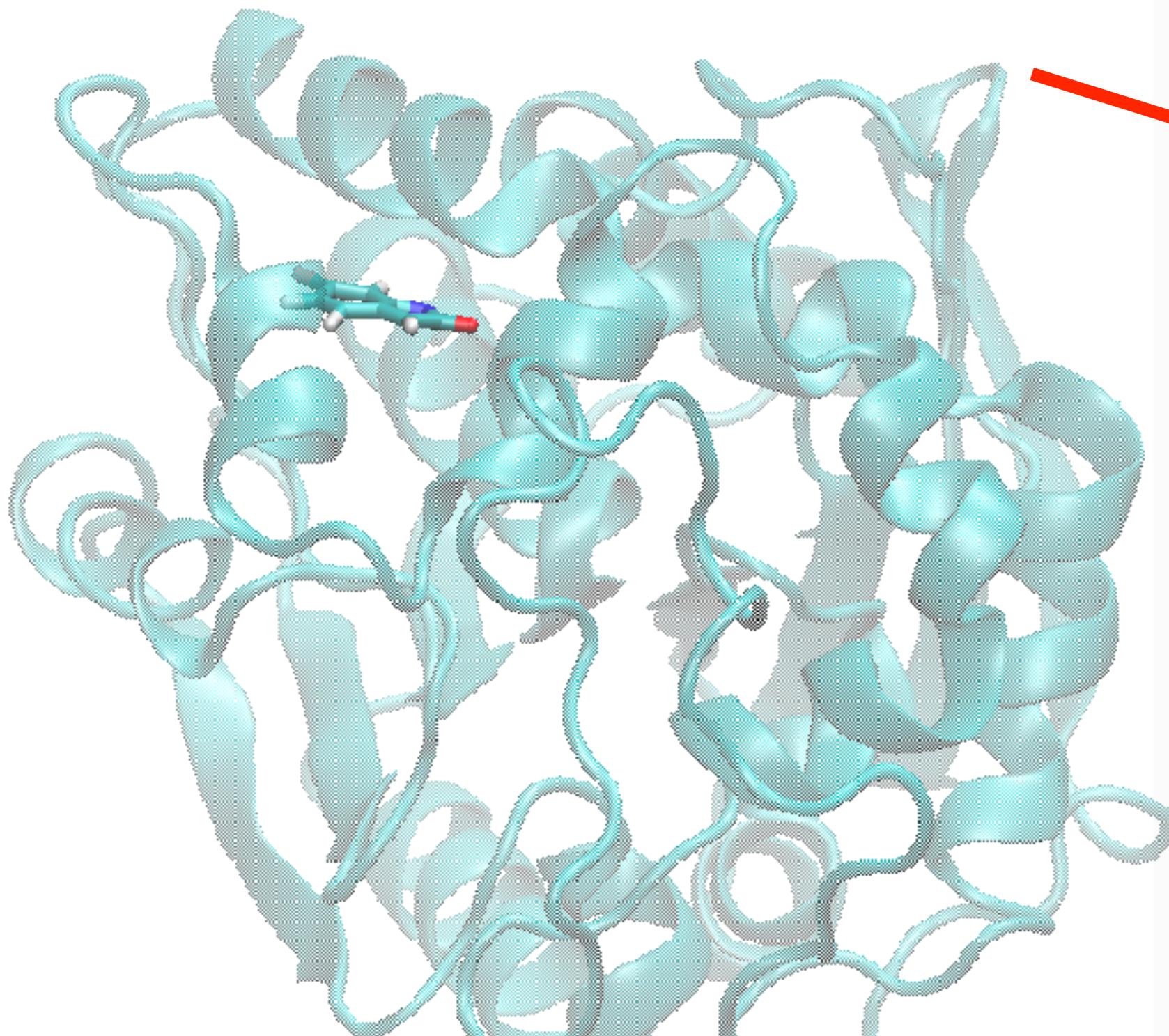
# Task: Adding a ligand representation

- Create Rep. → Selection: `resname LIG` → Drawing Method: 'Licorice'



# Task: Making the protein transparent

- Select the representation for the protein
- Material: `Transparent`



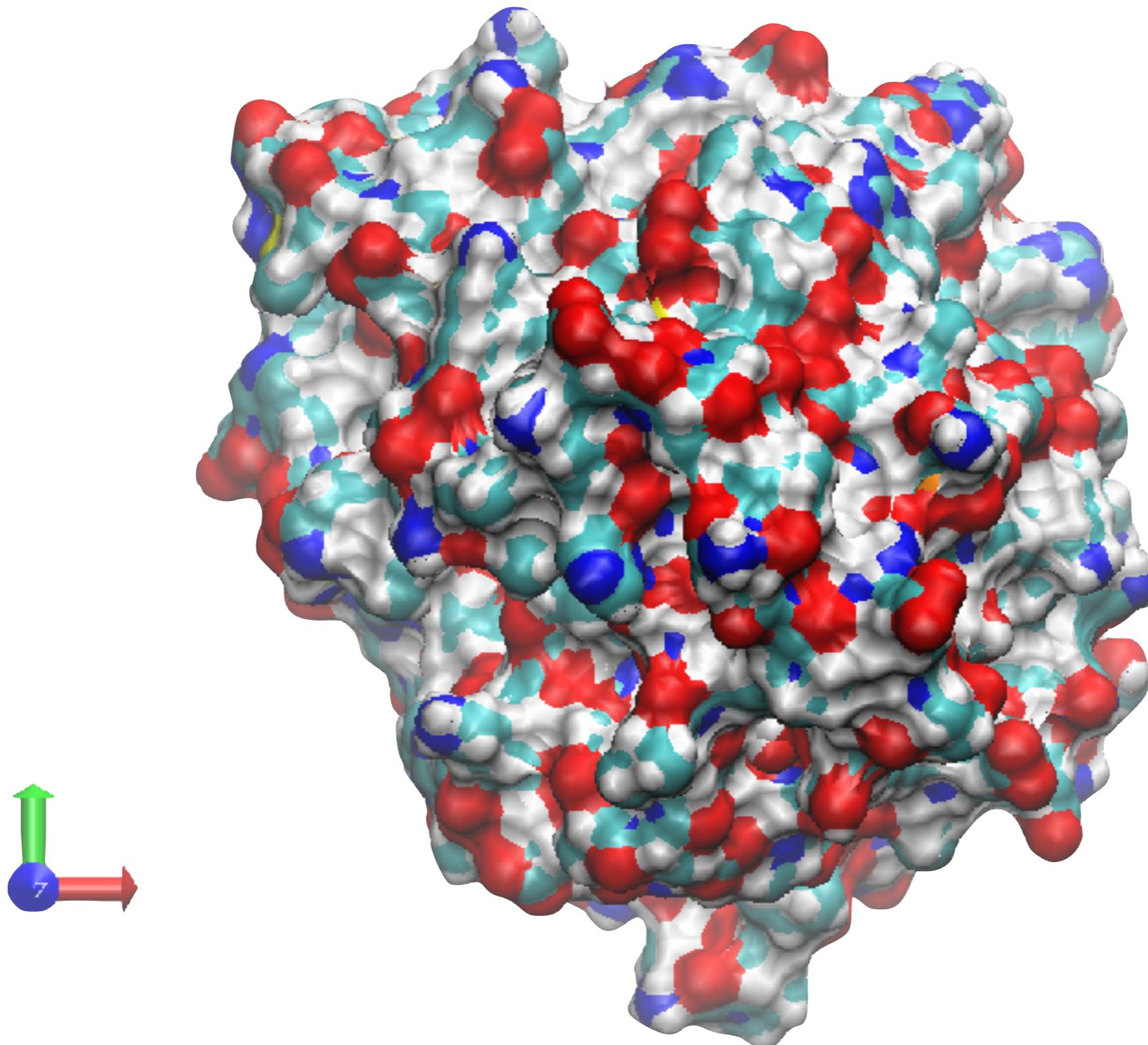
# Reference for atom selection and color methods

- Atom Selection Language: <http://www.ks.uiuc.edu/Research/vmd/vmd-1.3/ug/node132.html>
- Coloring Methods: <http://www.ks.uiuc.edu/Research/vmd/vmd-1.7.1/ug/node74.html>

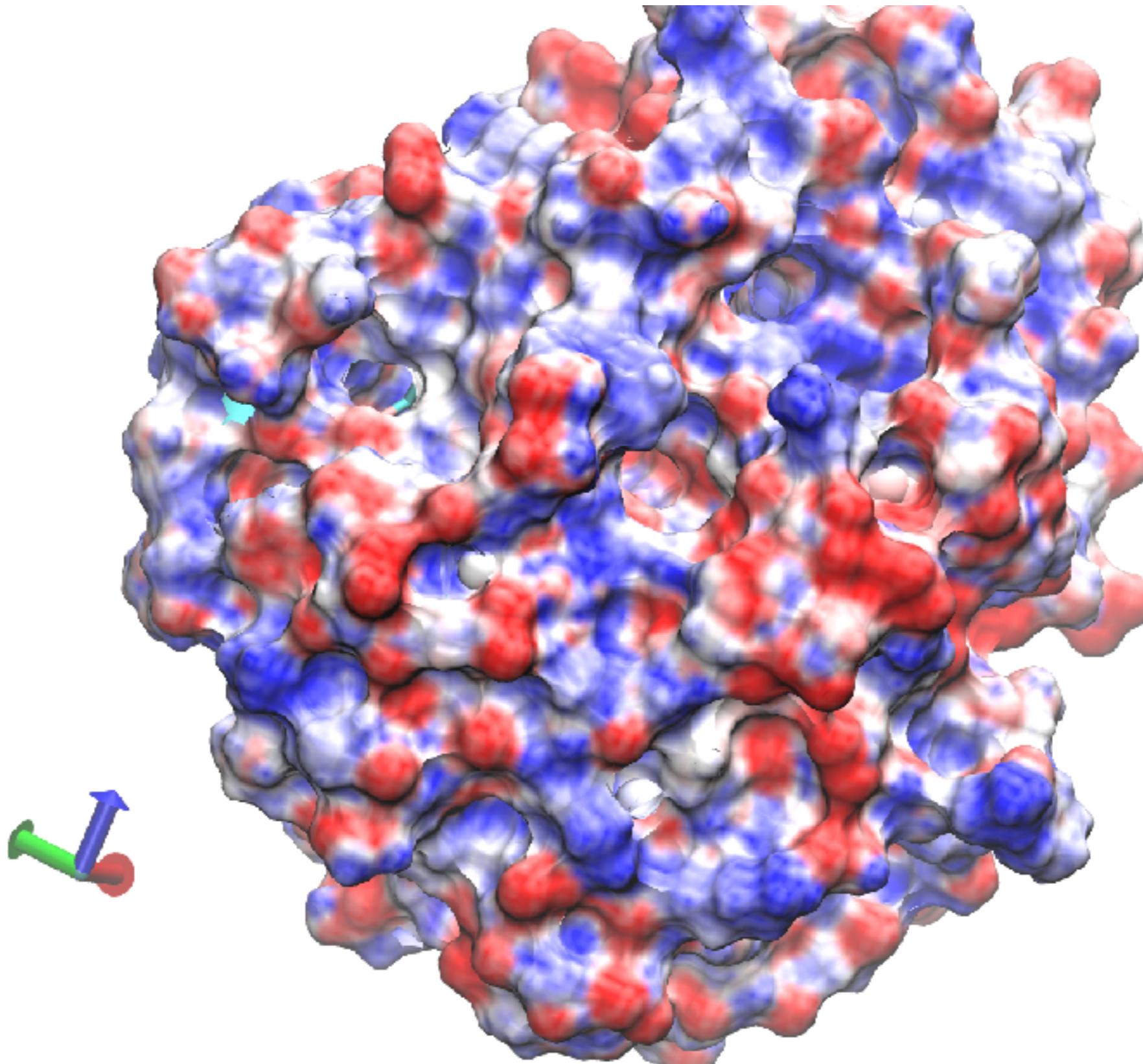
# We are also interested in the electrostatic potential on the surface of molecules

- Could be important functionally
  - Binding sites
  - Protein-protein interactions
  - Protein-DNA binding (nucleic acids highly charged)
- Particularly positive or particularly negative regions are unlikely by chance
- Can impact binding rates, strengths, ...

# VMD/PyMol colors things by their element type (by default)



We can instead use colors to show electrostatic potential



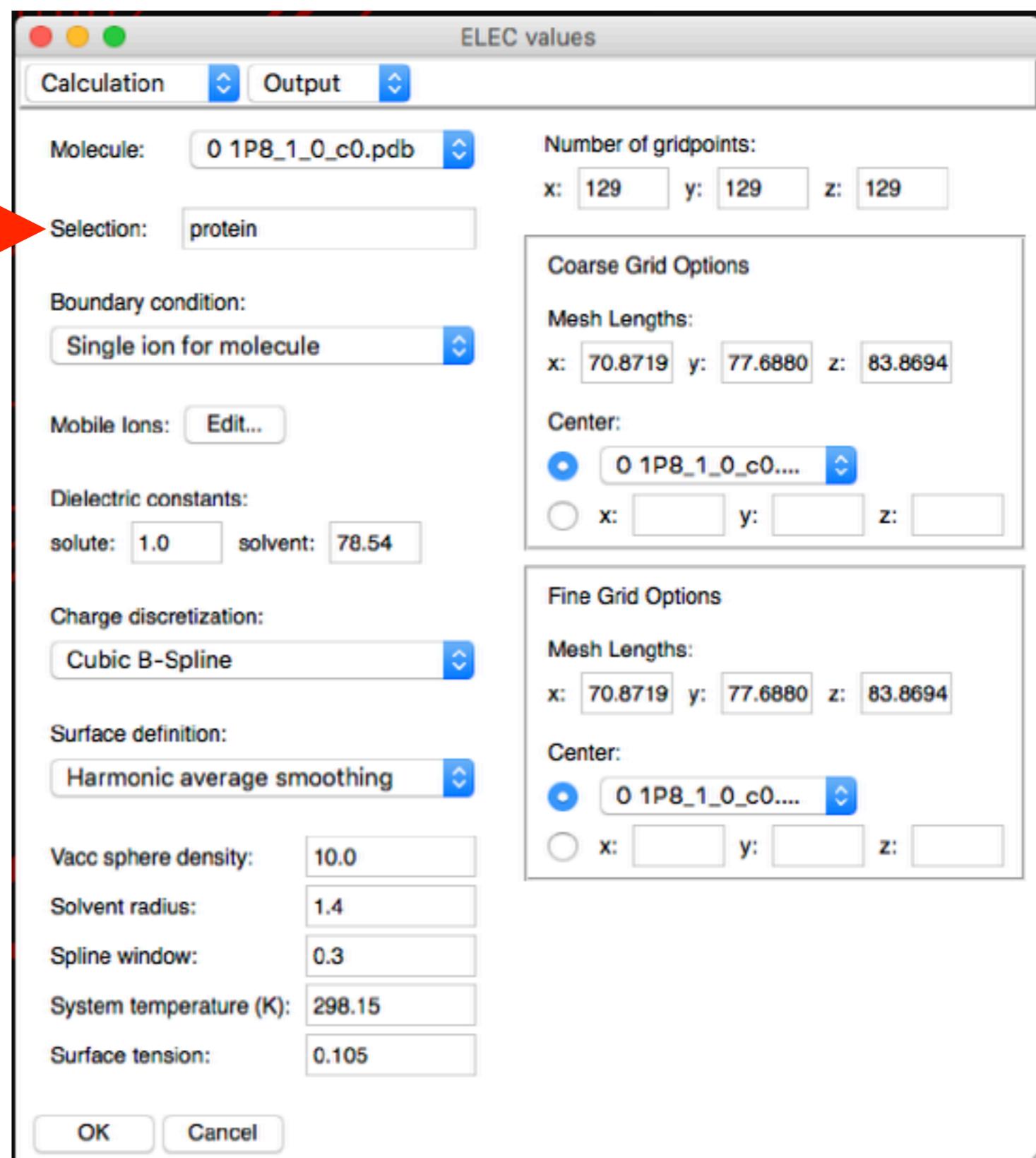
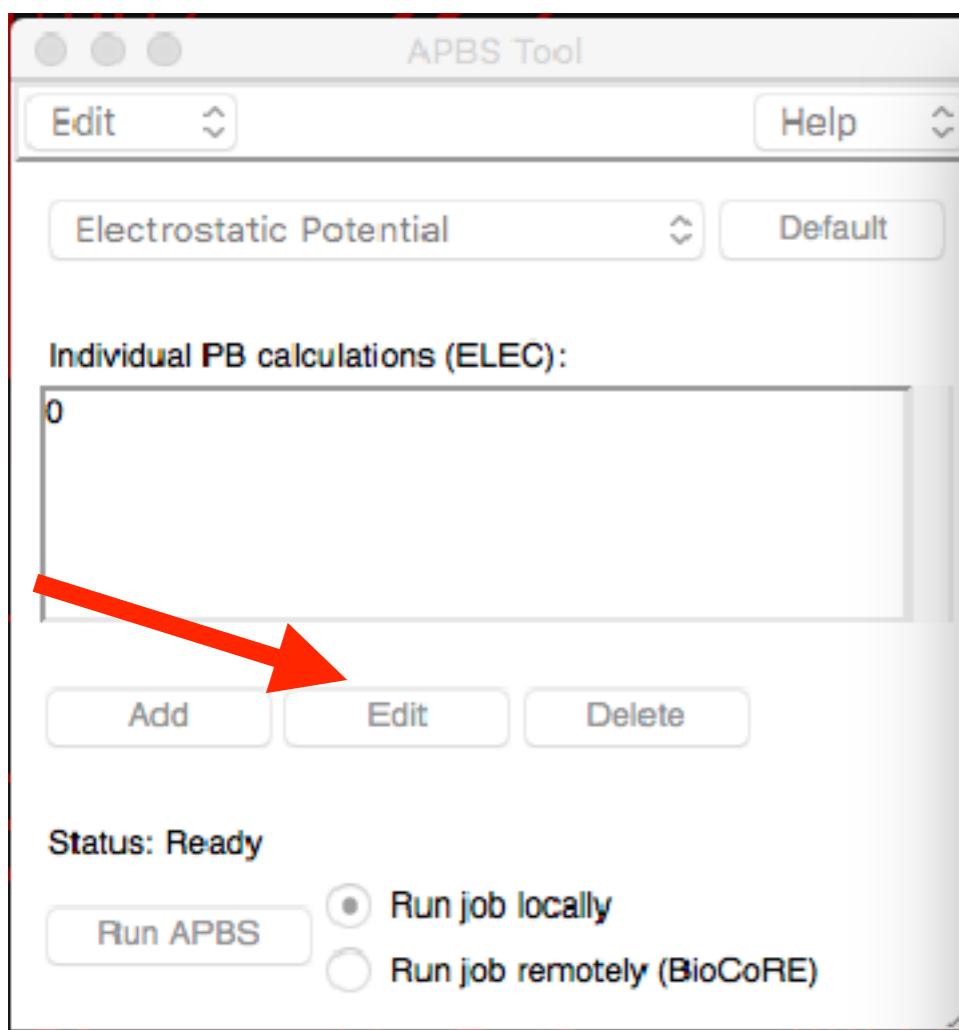
# Before we do that, we have to calculate the electrostatic potential

- Various tools can do this, including OpenEye's ZAP, **APBS**, and Delphi.
- APBS is a built-in VMD plugin!
- PDB files only contain atom coordinates.
- Need to add charge parameters into molecule.
- Load in a parameter file (PSF):
  - **[Menu]: *File* → *Load Data into molecule* → 1P8\_1.psf**

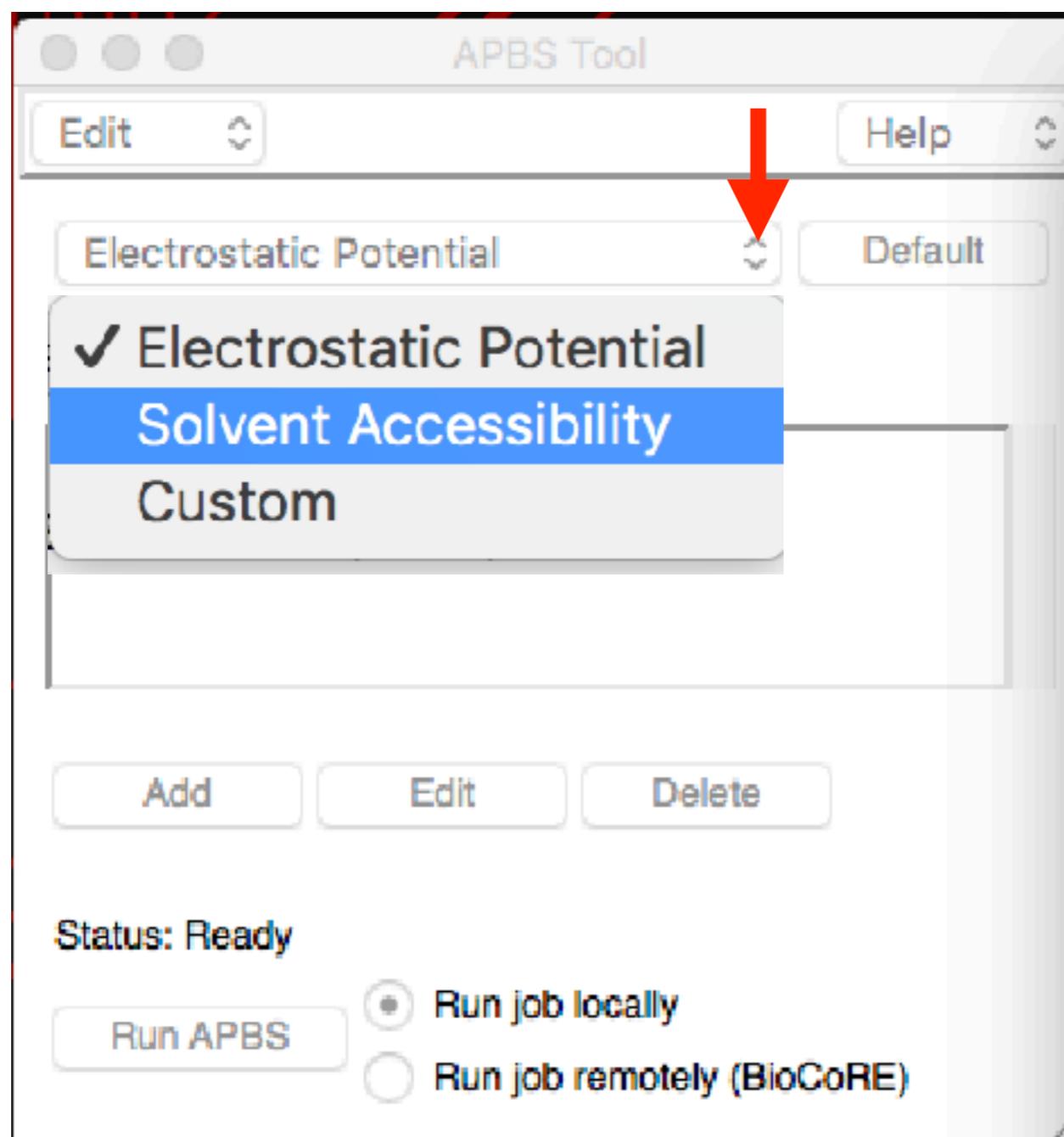
# Task: Calculating the electrostat. potential

- **[Menu]: Extensions → Analysis → APBS Electrostatics**

*Default is to get  
elec. potential for protein*

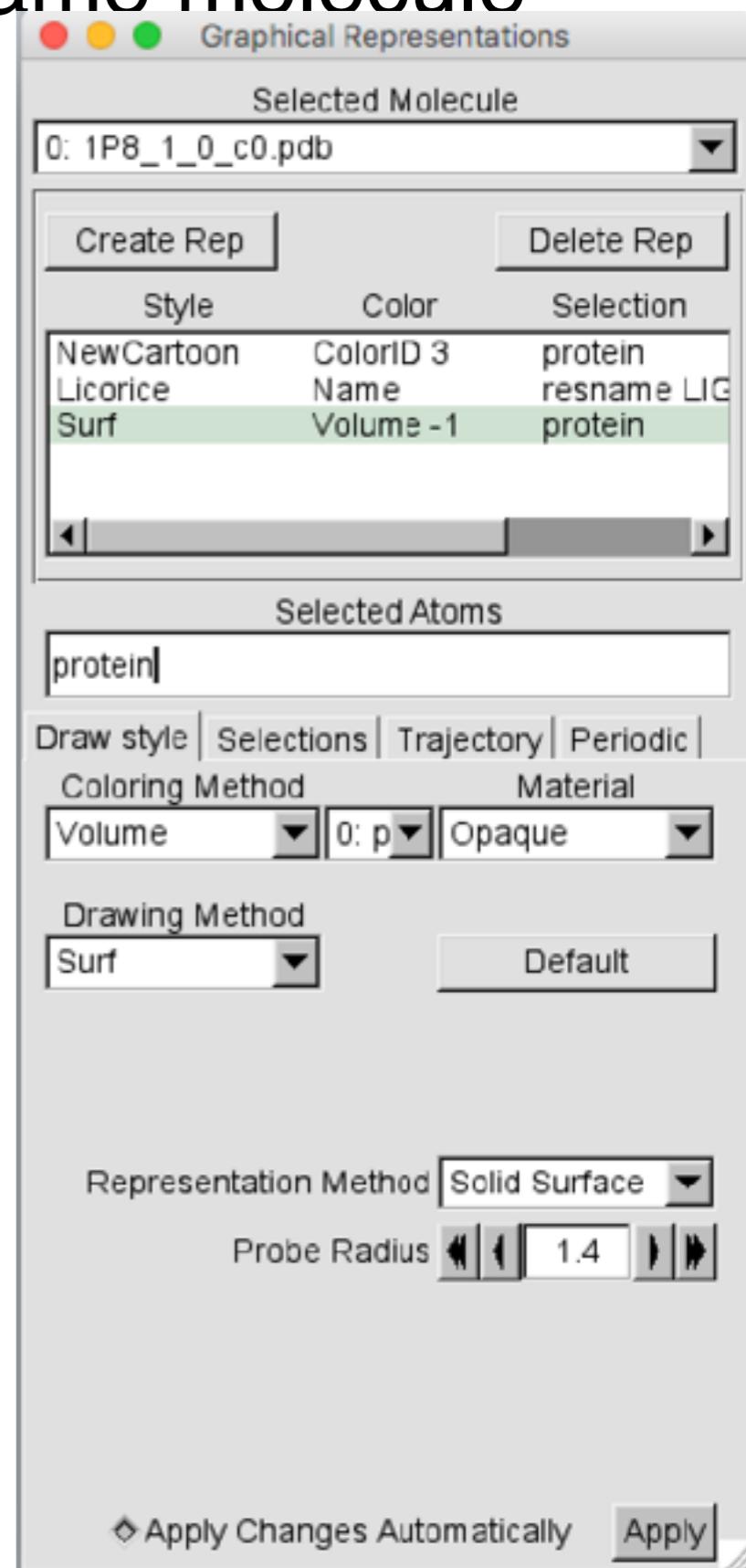
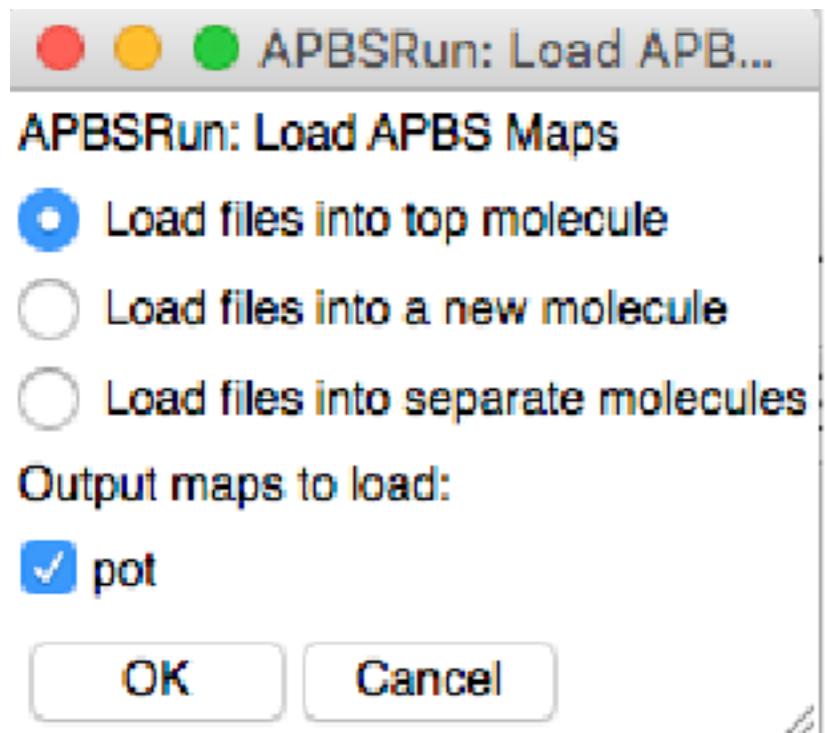


# Can use APBS to calculate Solvent Accessibility



# Task: Calculating the electrostat. potential

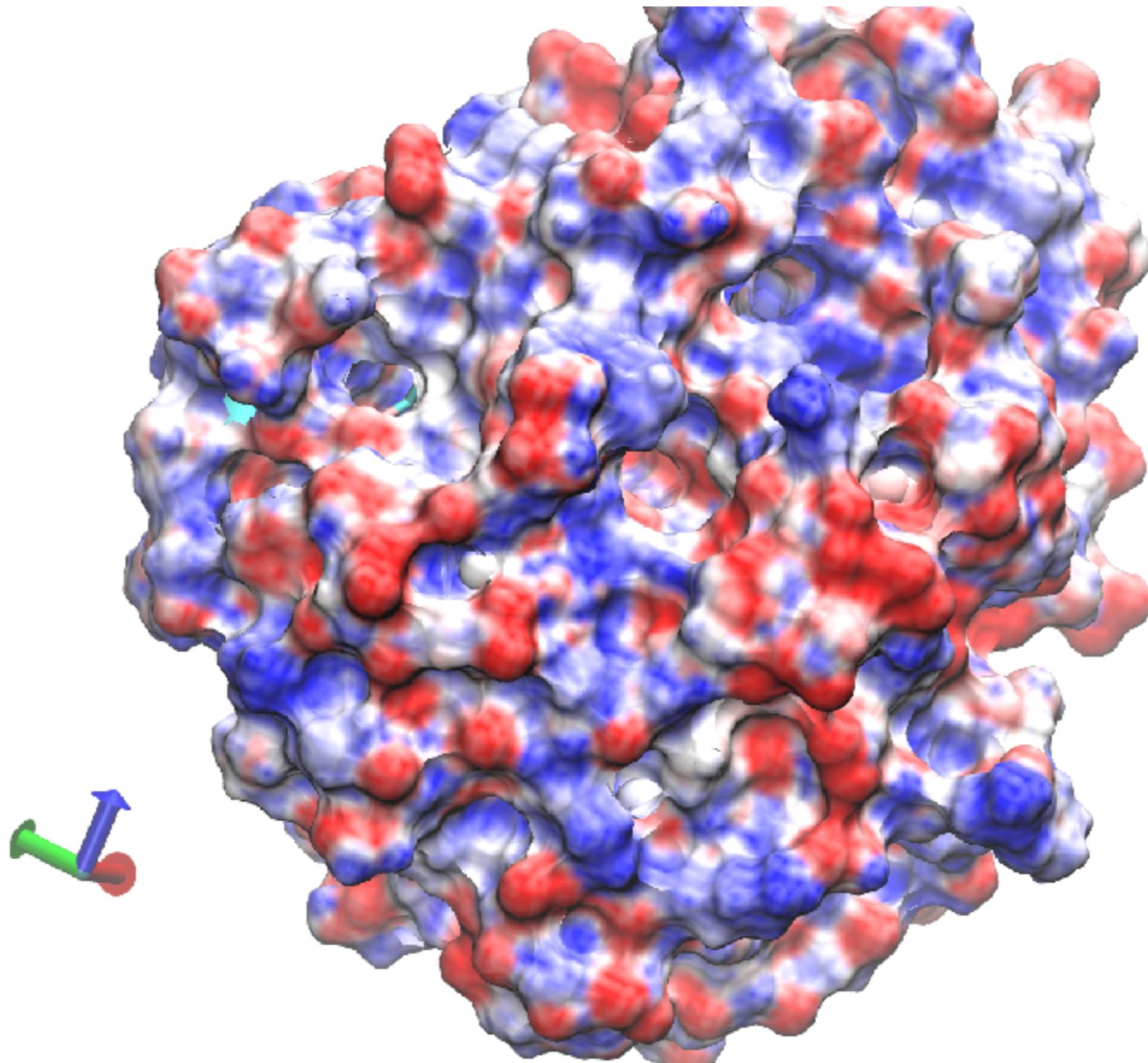
- After running APBS, load files into the same molecule



- Create a representation for the protein
  - Coloring: *Volume with DX map*
  - Drawing: *Surface*
  - Change Color Scale: -10 to 10*

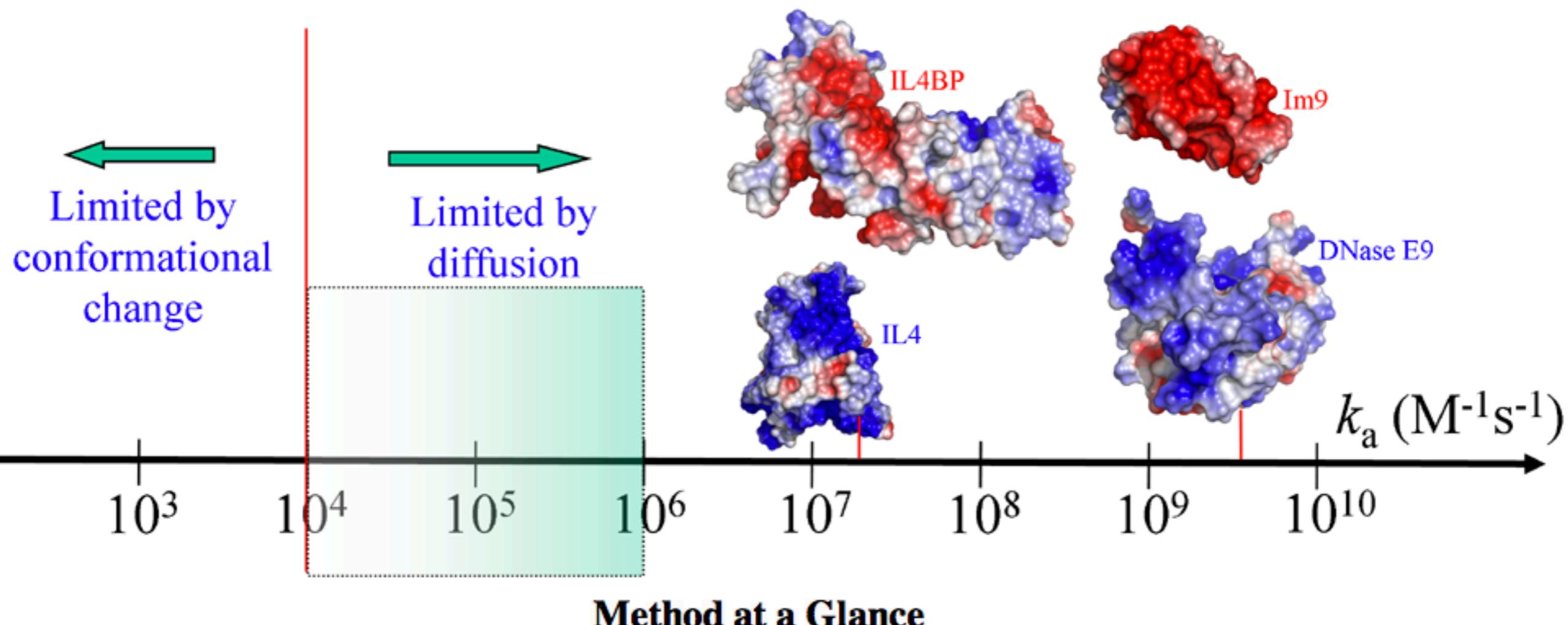


Your final result should look something like



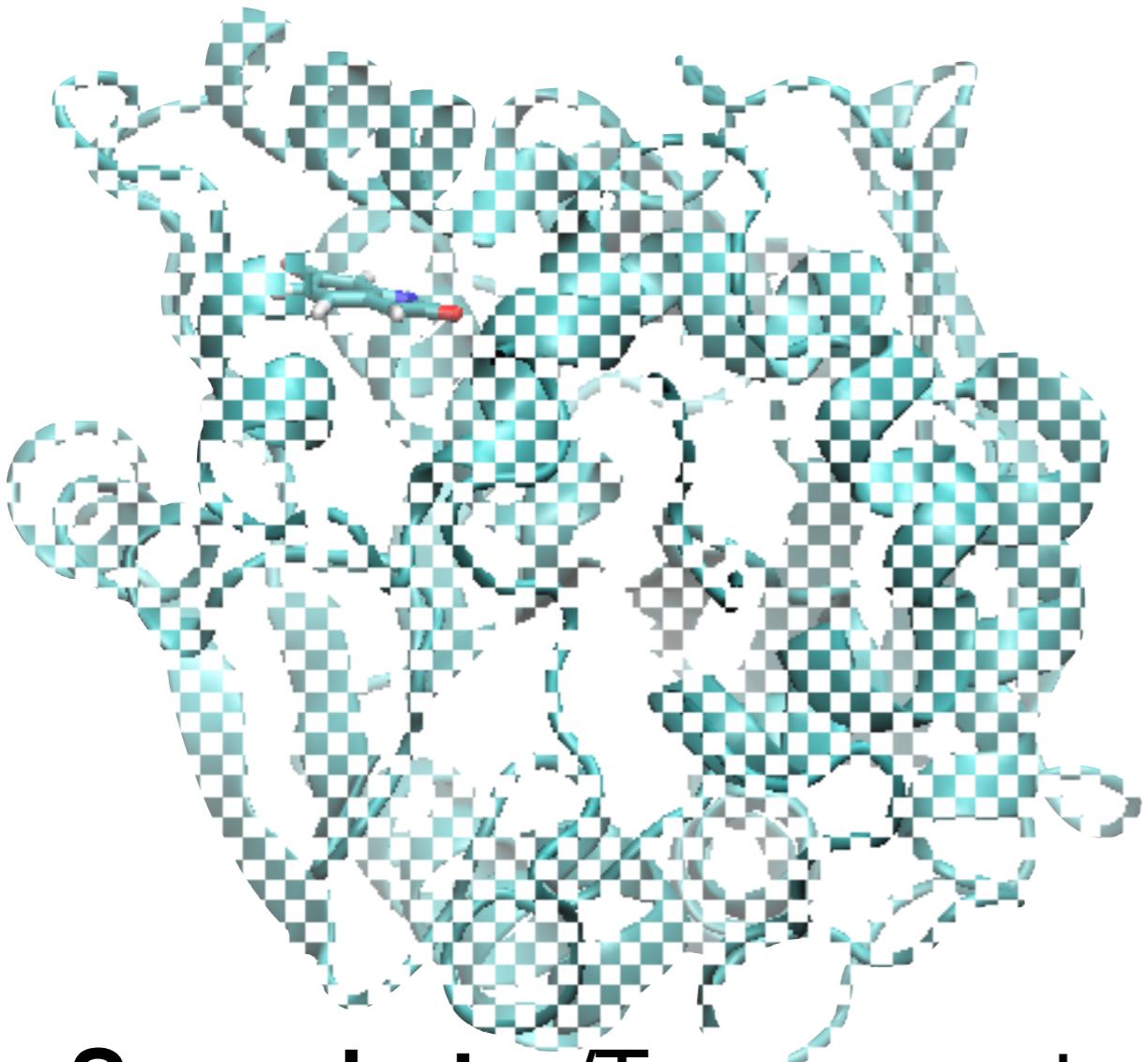
It turns out that simple electrostatics plus diffusion can go a long way towards kinetics

## TransComp: Web Server for Predicting Protein Association Rate Constants

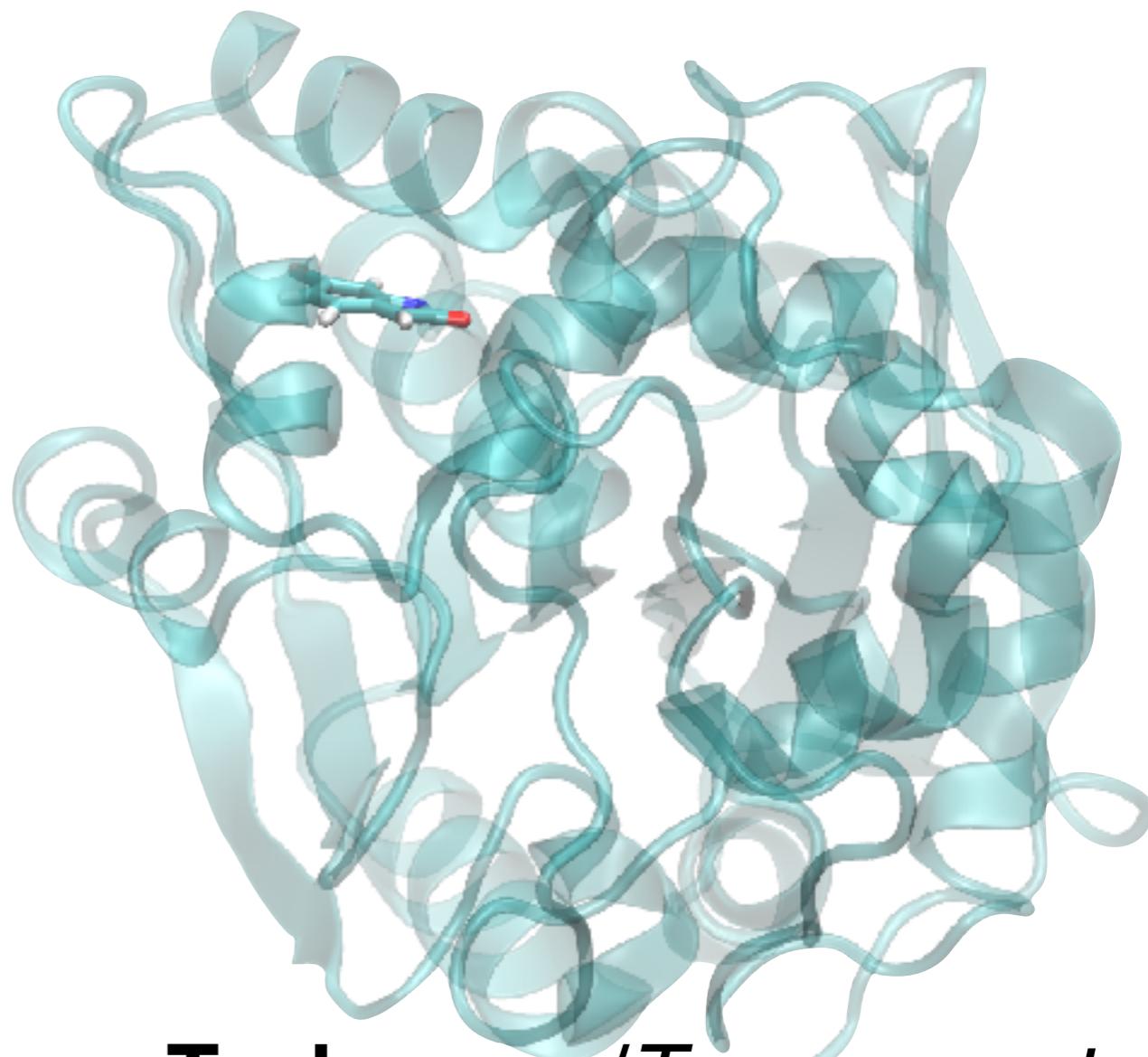


# Task: Saving a picture of the molecule

- **[Menu:]** *File* → *Render*
- Use ***Snapshot*** or ***Tachyon*** (*internal, in-memory rendering*)



**Snapshot w/Transparent**  
(Low quality)



**Tachyon w/Transparent**  
(High quality)

Note: choice of *render* and *material* impacts image

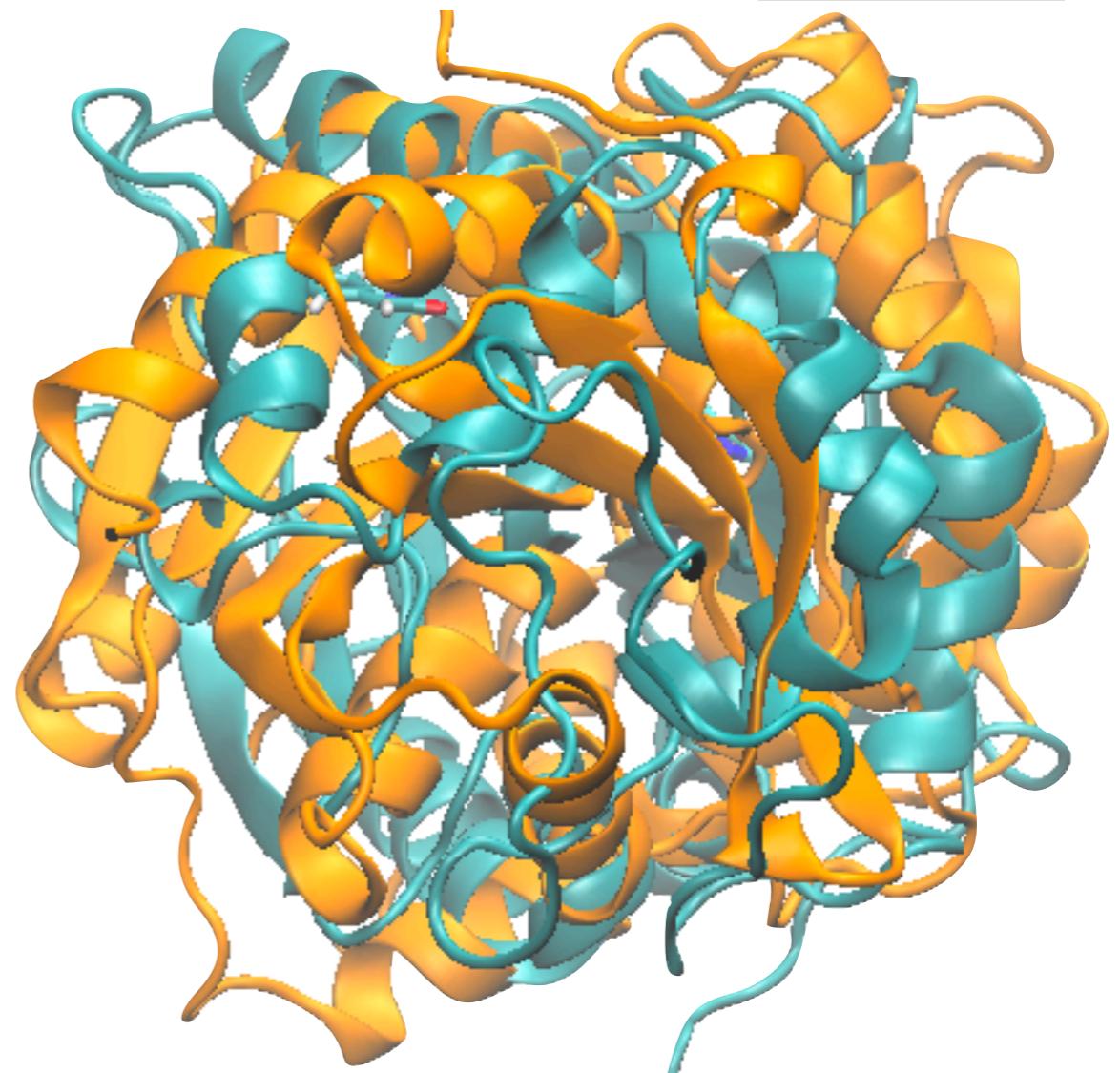
# Task: Visualizing a molecular dynamics simulation

- Load a new molecule with a trajectory (DCD) file:
  - **[Menu]:** *File* → *New Molecule* → *Browse* → *Load file:* `1P8_1_0_c1-blues.dcd`
  - **[Cmd]:** `mol new 1P8_1_0_c1-blues.dcd`
- Add the parameters (PSF) for the trajectory:
  - **[Menu]:** *File* → *Load Data Into Molecule* → *Browse* → *Load file:* `1P8_1.psf`
  - **[Cmd]:** `mol addfile 1P8_1.psf`

You may see something like this:

Cyan (reference PDB)

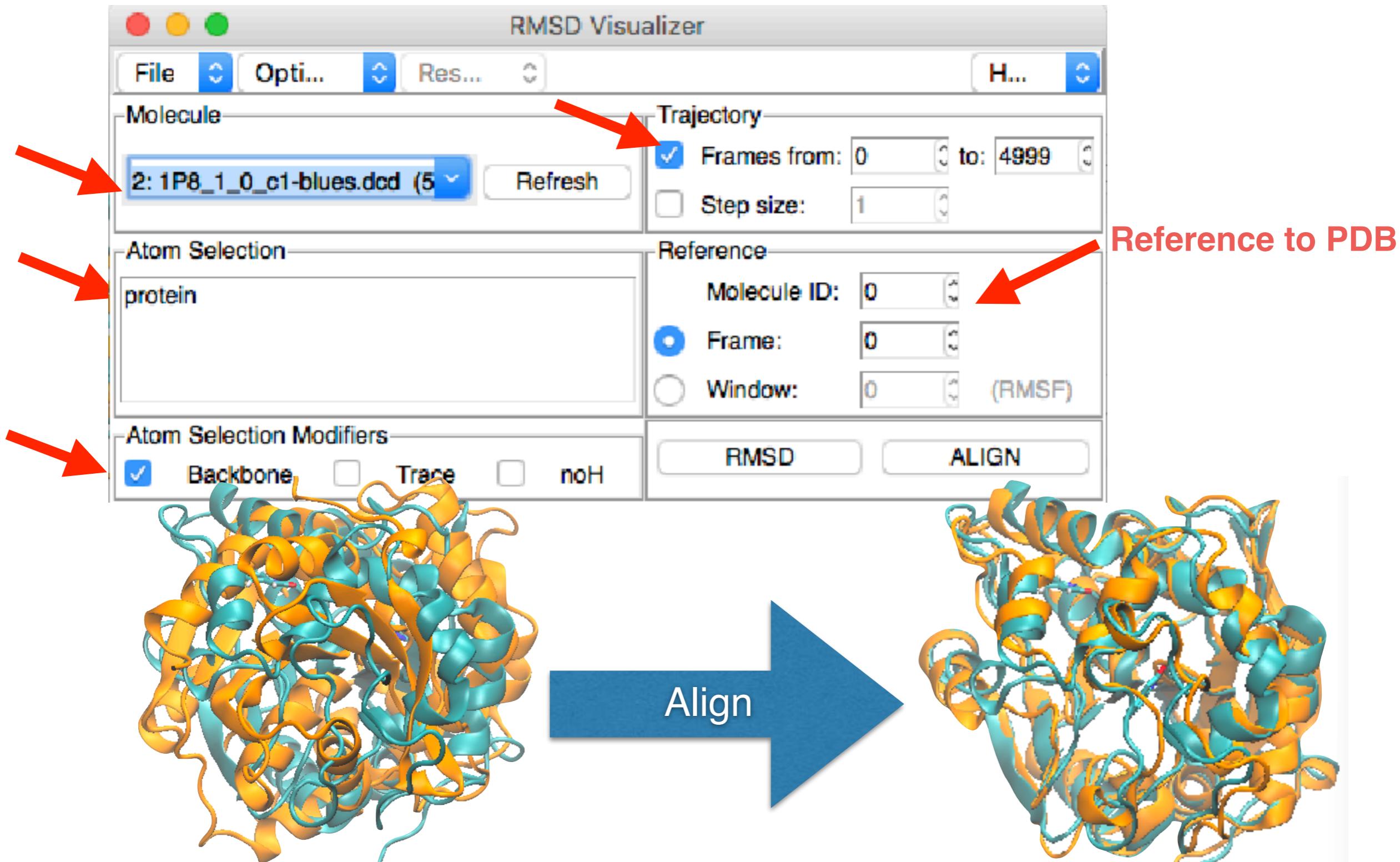
Orange (trajectory)



# Task: Aligning two protein structures

- **[Menu]: Extensions → Analysis → RMSD Visualizer**

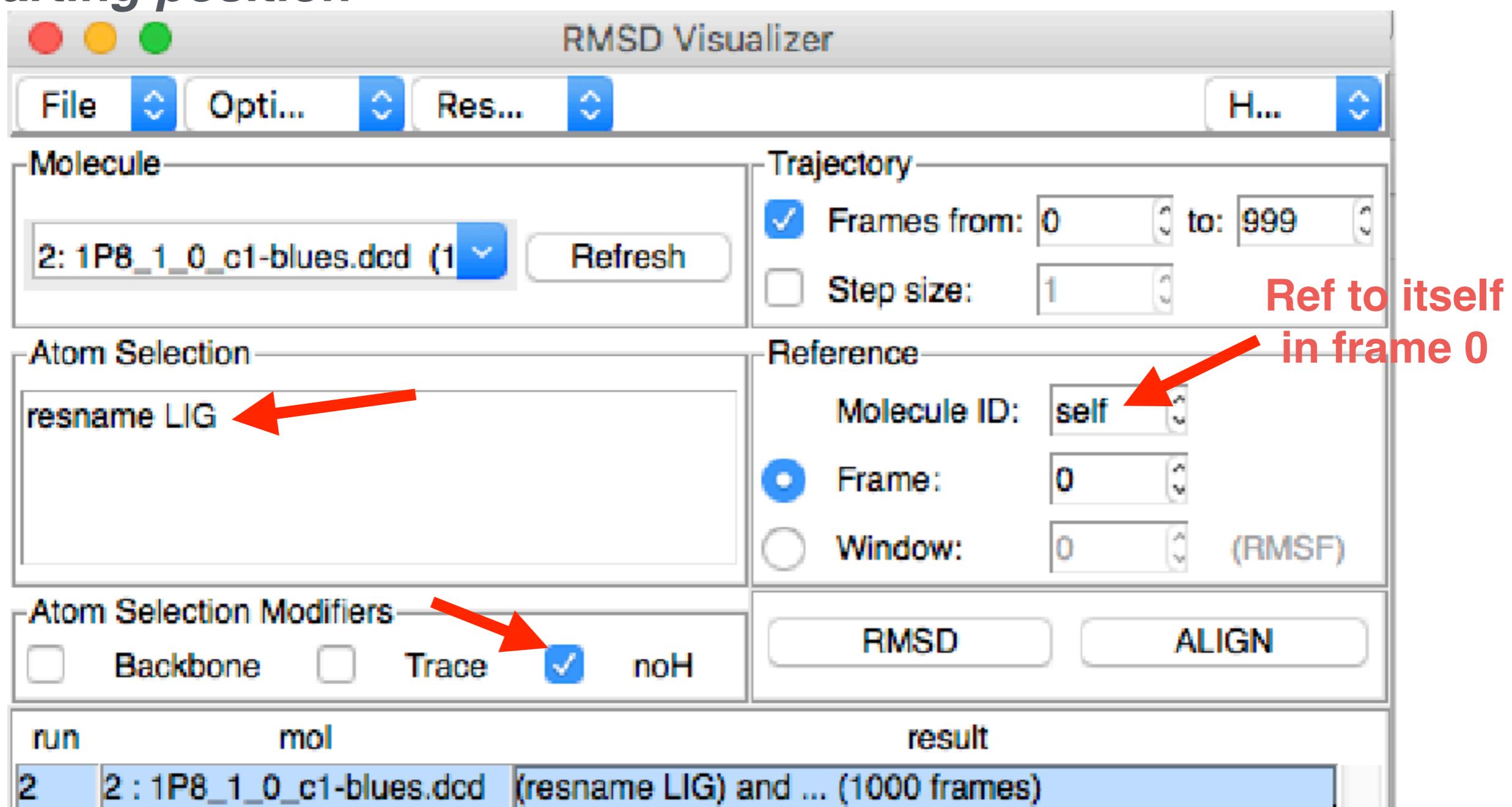
Align trajectory to the reference PDB by protein backbone



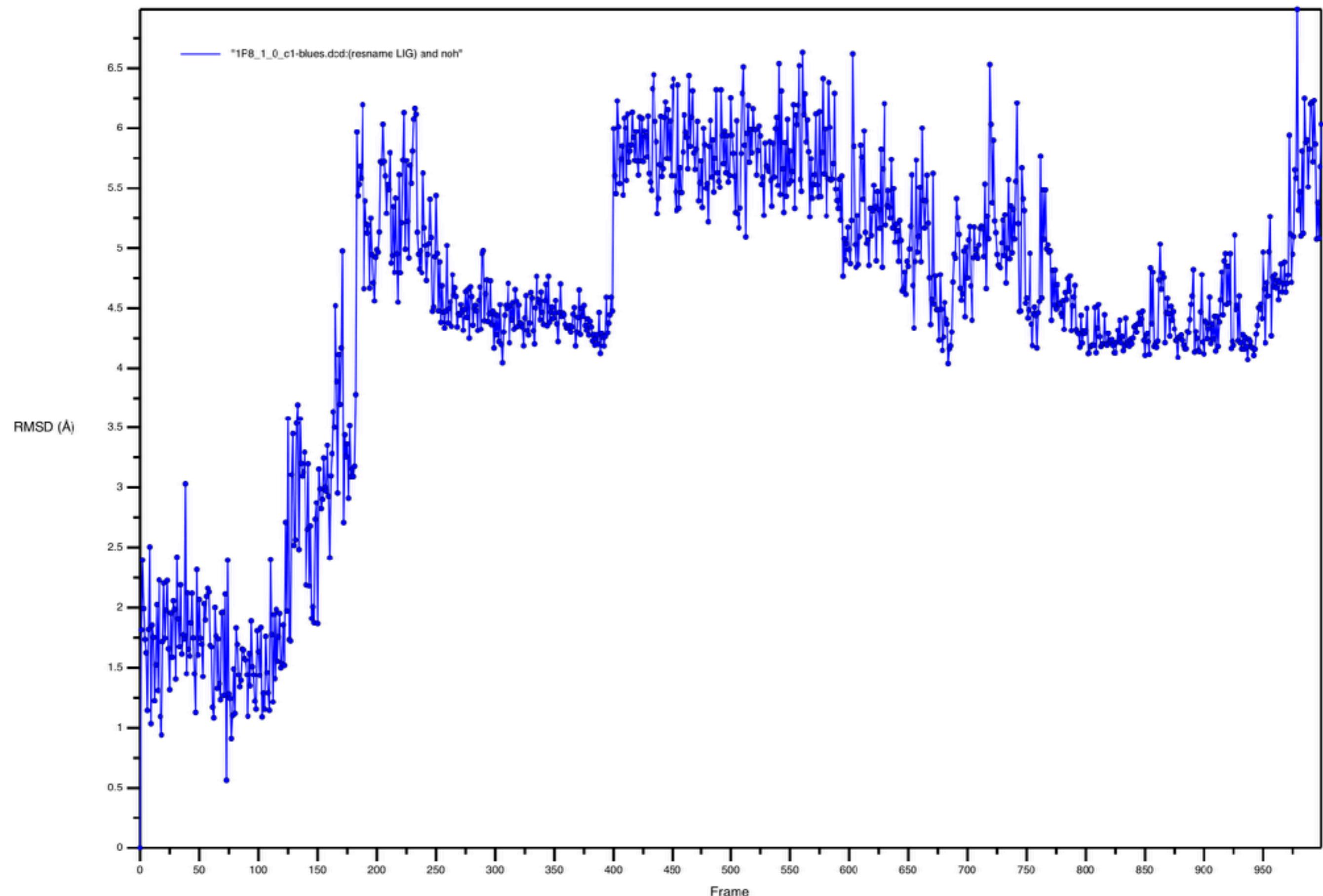
# Task: Computing the ligand RMSD

- **RMSD: Root-Mean-Square Deviation**
  - *Measurement of the average distance between the atoms*

**Compute the RMSD of the ligand heavy atoms, relative to it's starting position**

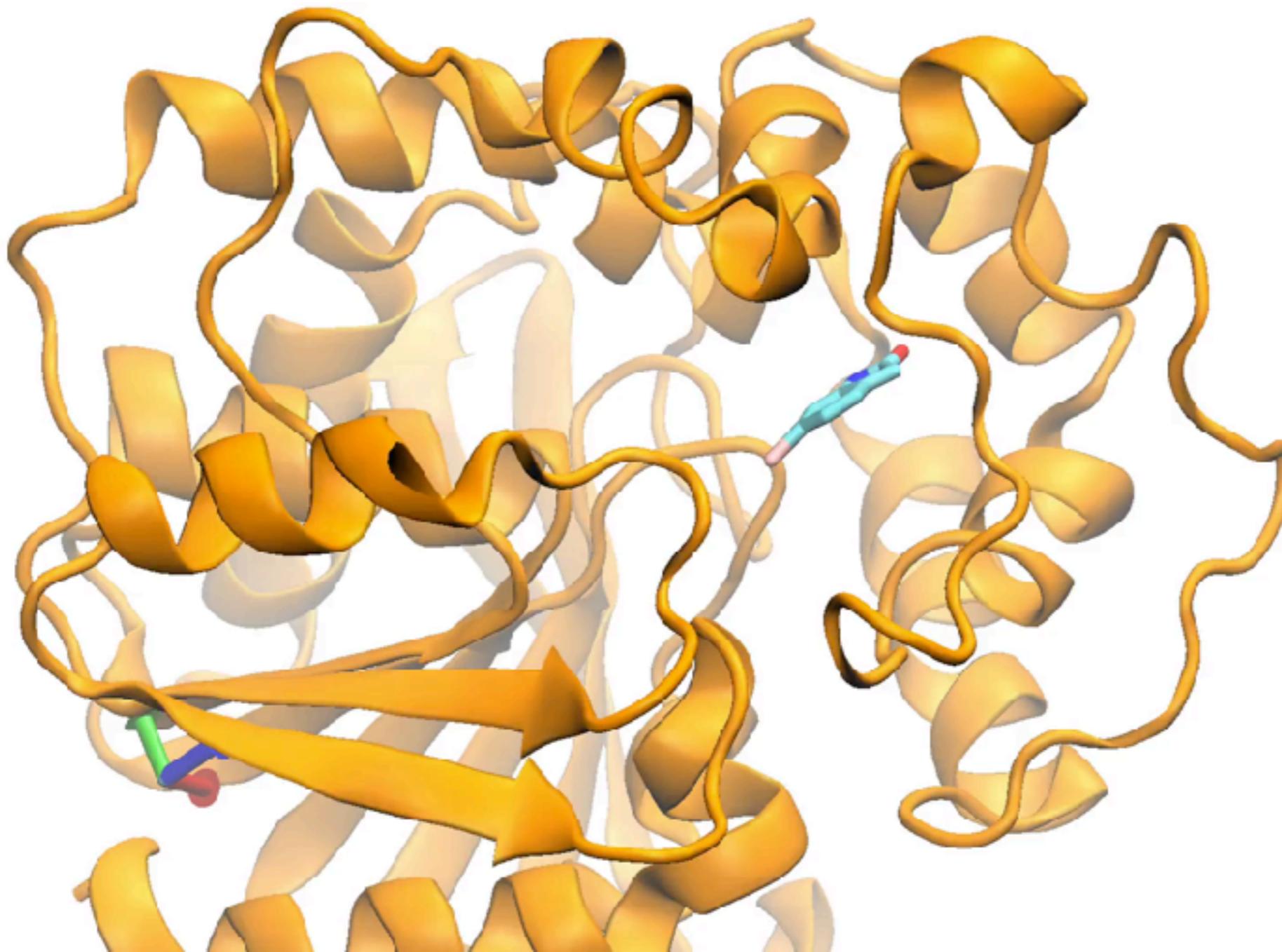


RMSD vs Frame: 1P8\_1\_0\_c1-blues.dcd



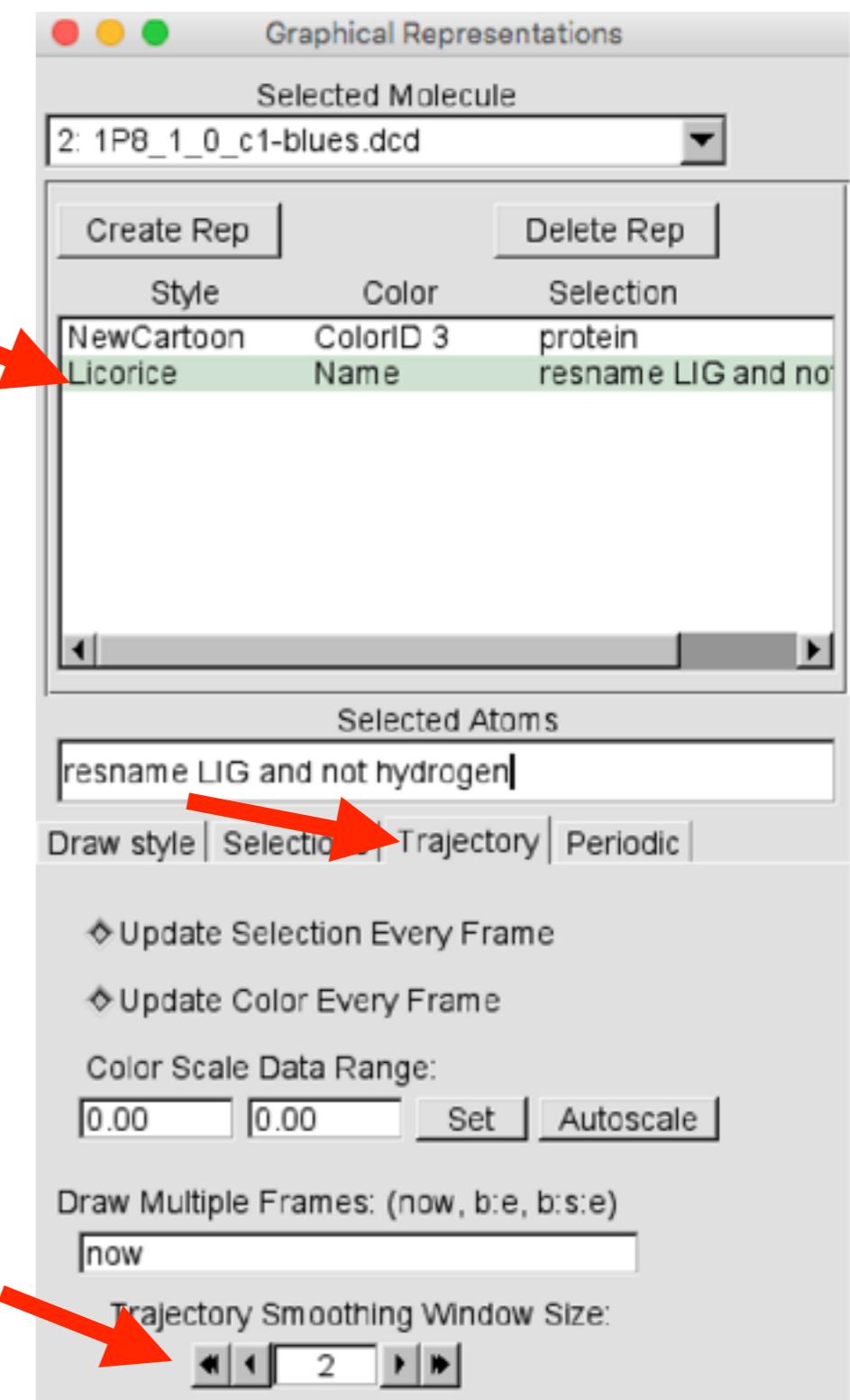
# Task: Smoothing the traj. animation

- If you play the trajectory, notice it wiggles very rapidly. (thermal noise)
- This doesn't make for very good movies, we will want to add **trajectory smoothing** before making a movie.

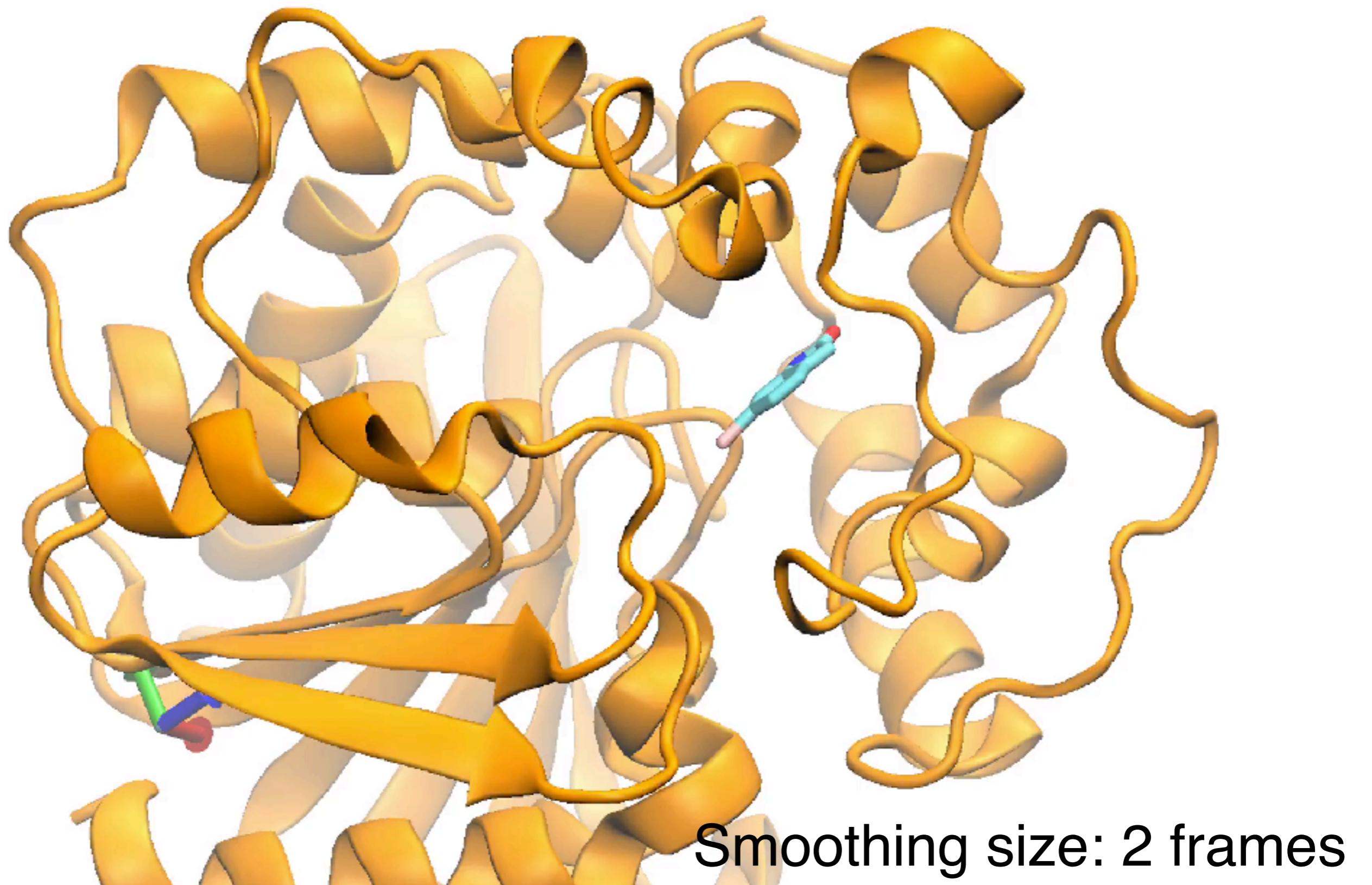


# Task: Smoothing the traj. animation

- **[Menu]: Graphics → Representations → Trajectory**
- **Smoothing takes the *structural average* from X frames**
- Do this for both the ligand and protein representation

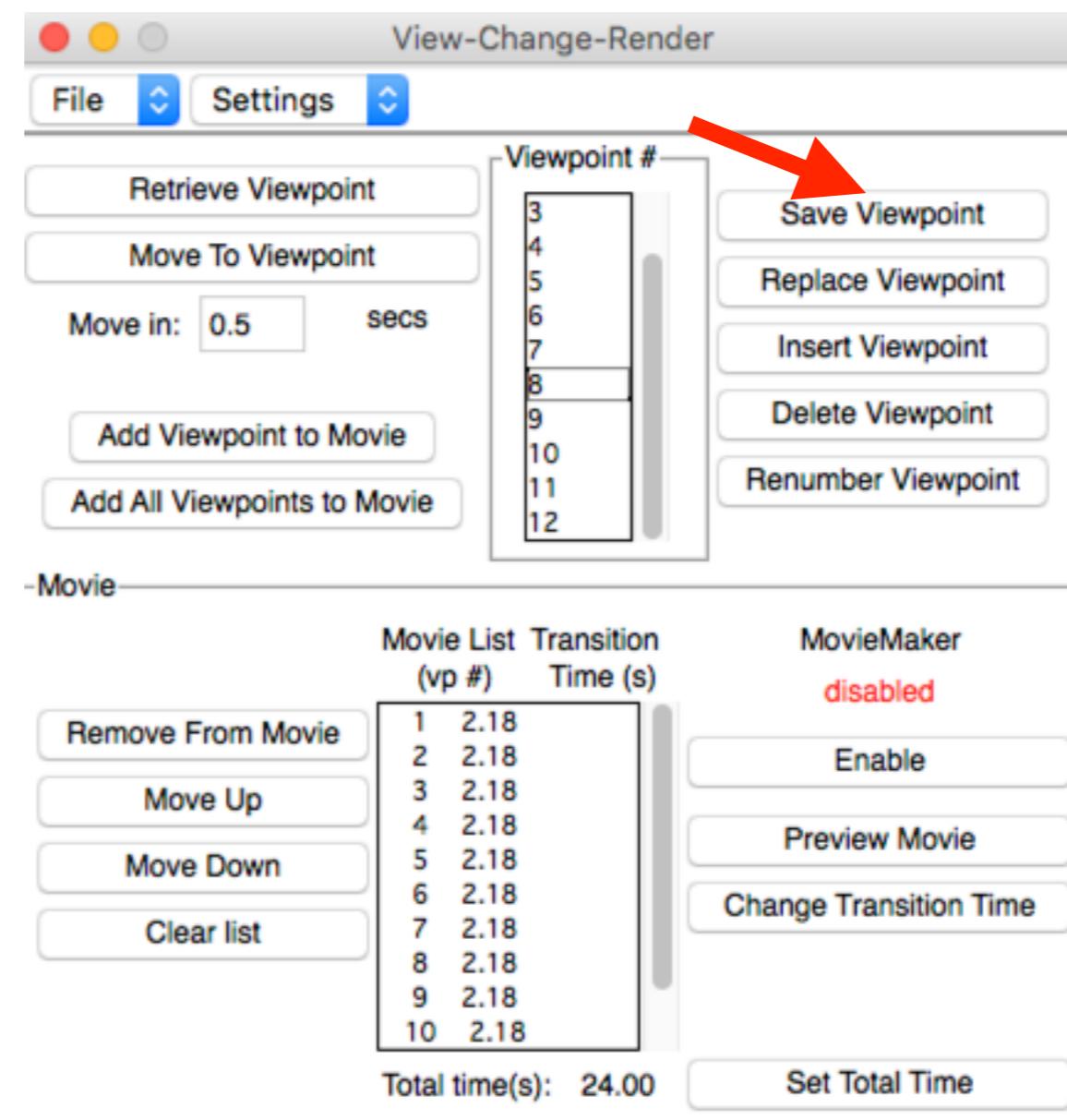


# Task: Smoothing the traj. animation



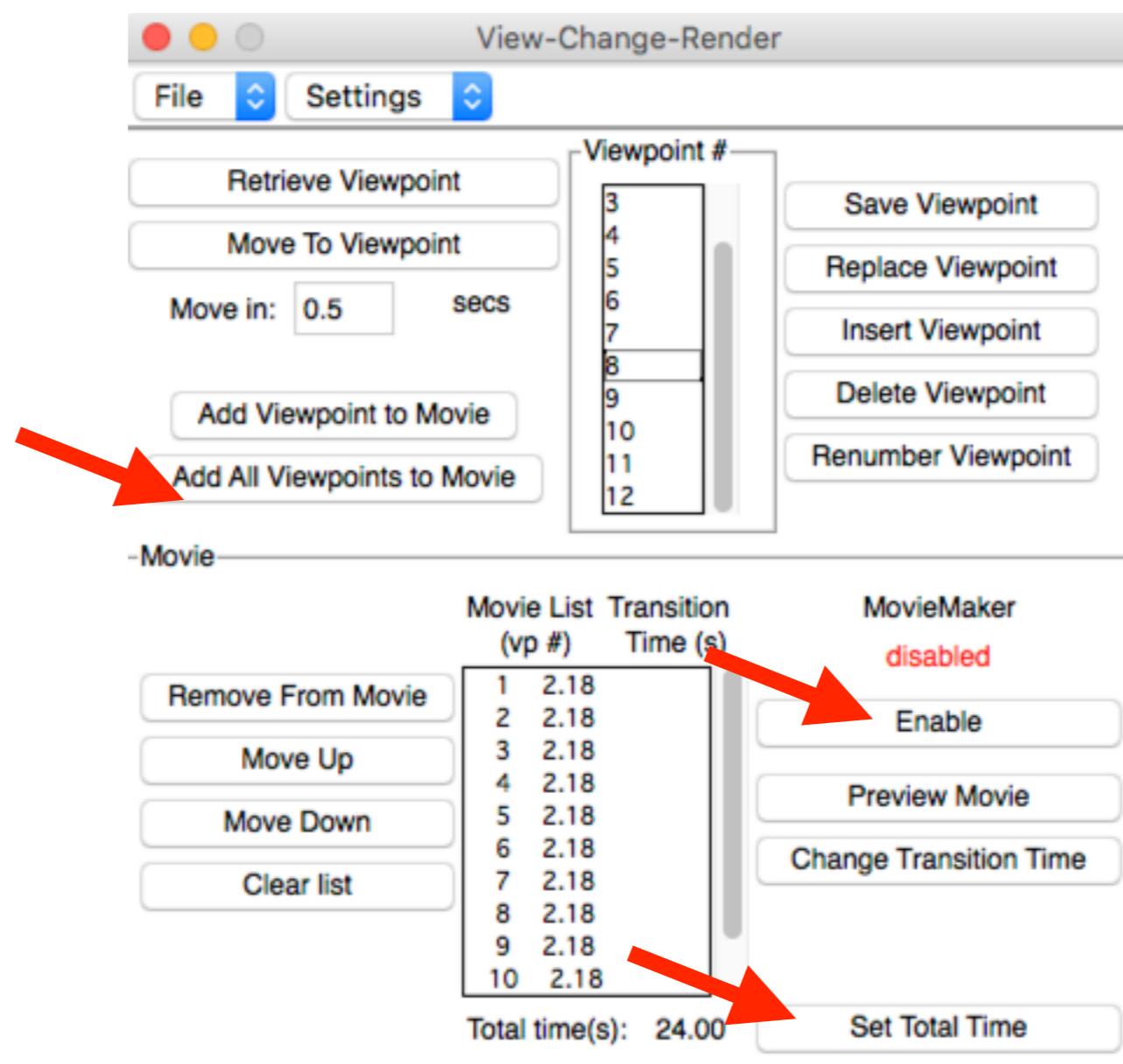
# Using ViewChangeRender to store views

- **[Menu]: Extensions → Visualization → ViewChangeRender**
- Simply alter the display to your liking and `Save Viewpoint`
- Each saved point will store the camera angle and molecular representations



# Piecing together the viewpoints

- After saving all viewpoints into ViewChangeRender, you can hit `Add All Viewpoints to Movie`.
- Set the total time and then **Enable MovieMaker** to preview the movie (Try 2s per viewpoint)

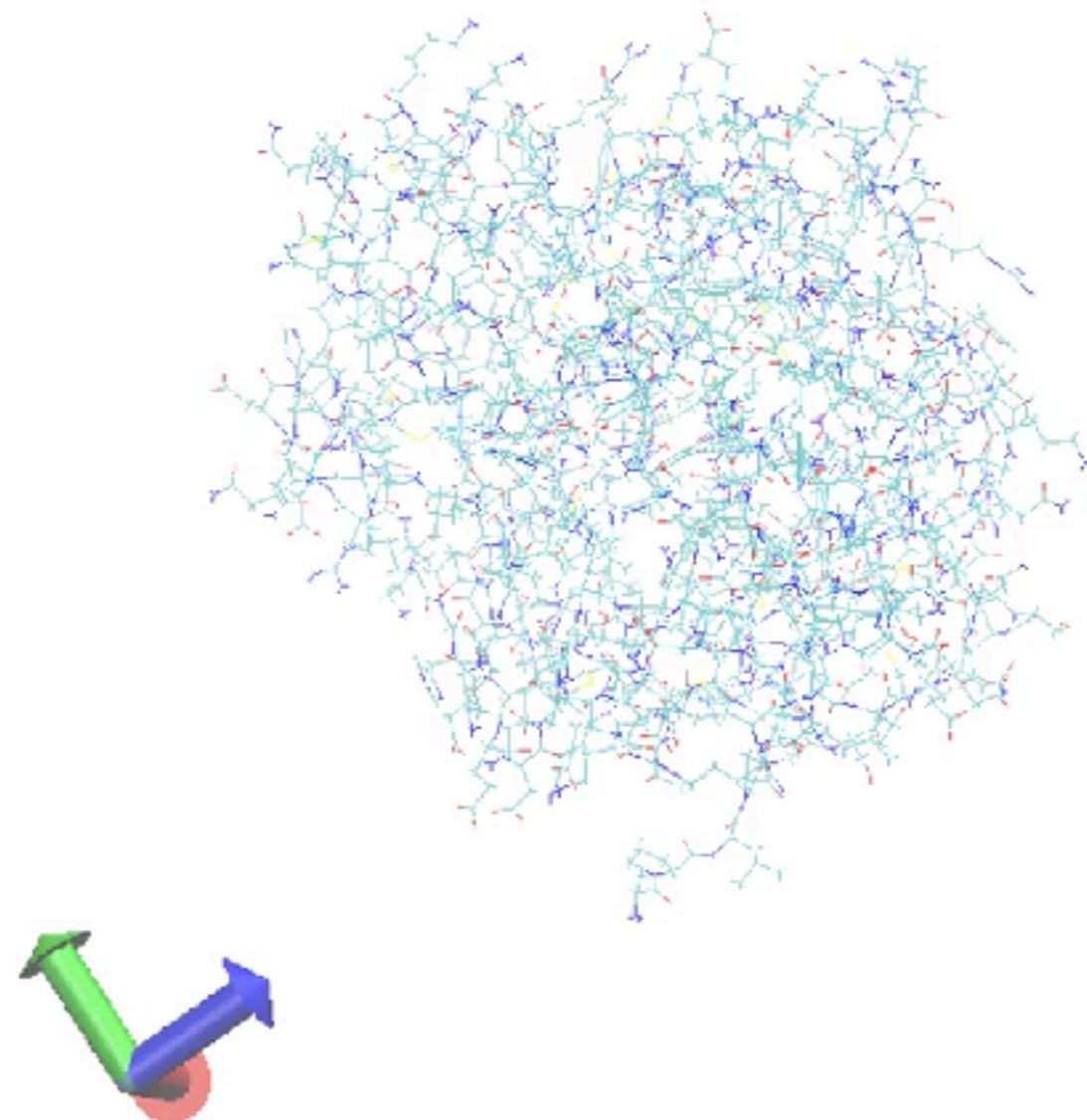


# Creating a movie from viewpoints

- **[Menu]: *Extensions* → *Visualization* → *Movie Maker***
- Set the working directory to an empty folder
- Name your movie: Ex. `1P8\_1-scene0`.`  
• **Note:** Naming scheme will be important here (you'll see later).
- Settings:
  - Renderer: Snapshot
  - Movie Settings: **User Defined Procedure**
    - *UNCHECK: Delete Image Files*
  - Format: MPEG-2 (ffmpeg)
  - Compression Settings: 30 fps (NTSC Video)

Download FFmpeg: <https://www.ffmpeg.org/download.html#build-linux>

**Our first scene is meant to just show the protein structure and how the ligand fits**



# Making a movie of the MD simulation

- Now we will want to store the trajectory frames for the movie
- Name your movie: Ex. `1P8\_1-scene1`
- Settings:
  - Renderer: Snapshot
  - Movie Settings: **Trajectory**
    - *UNCHECK: Delete Image Files*
  - Format: MPEG-2 (ffmpeg)
  - Compression Settings: 30 fps (NTSC Video)

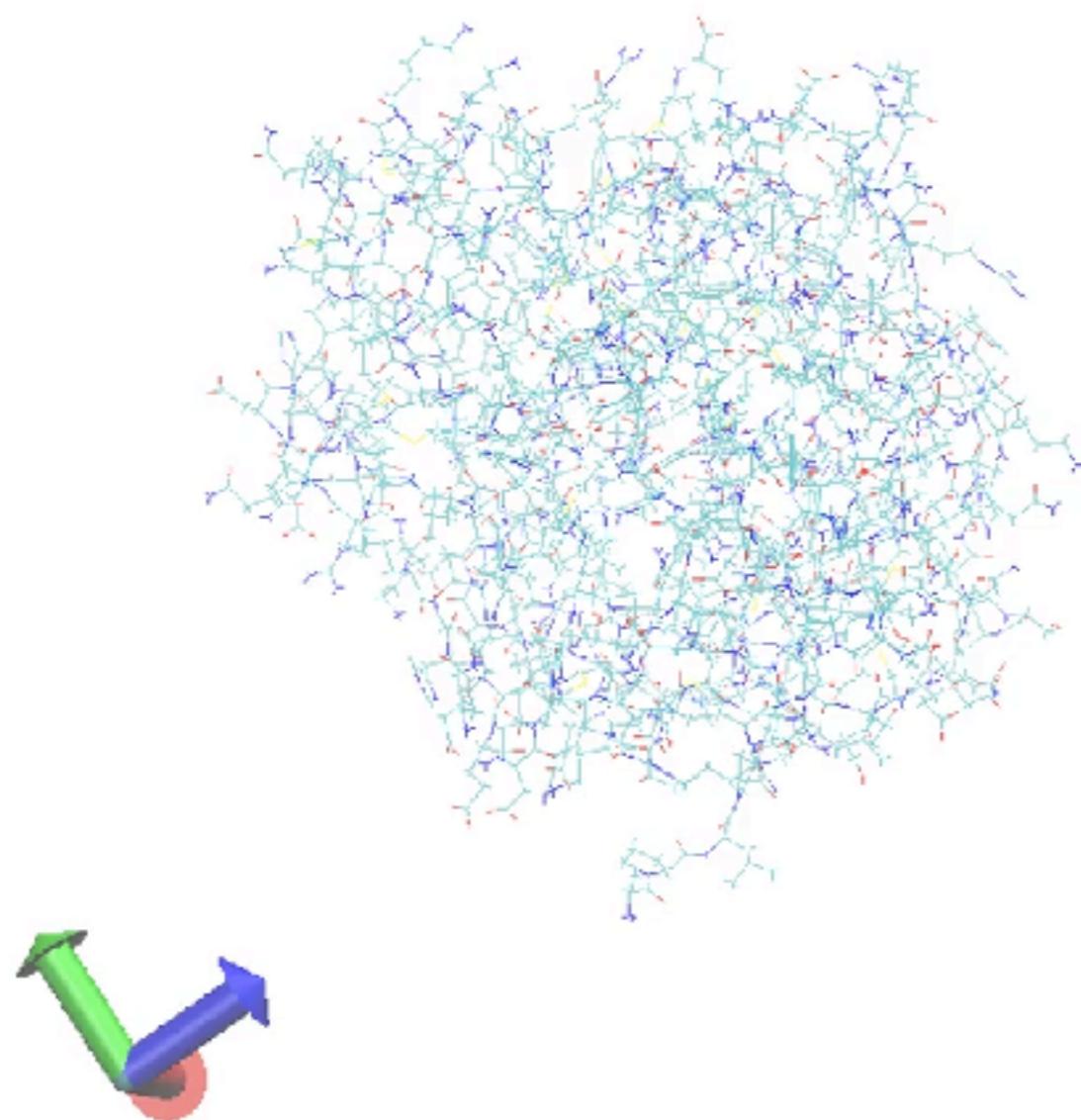
Download FFmpeg: <https://www.ffmpeg.org/download.html#build-linux>

# Making a movie of the MD simulation

- In a terminal window, navigate to the folder you've stored the movie frames in.
- Run the command:
  - `ffmpeg -r 30 -pattern_type glob -i “1P8_1-scene*.ppm” -vcodec libx264 -vf scale=-2:1080,format=yuv420p “1P8_1-movie.mp4”`
  - Glob pattern should load images from *scene0* and then *scene1*
  - The output file will be **“1P8\_1-movie.mp4”**

Download FFmpeg: <https://www.ffmpeg.org/download.html#build-linux>

# Final movie with MD simulation frames



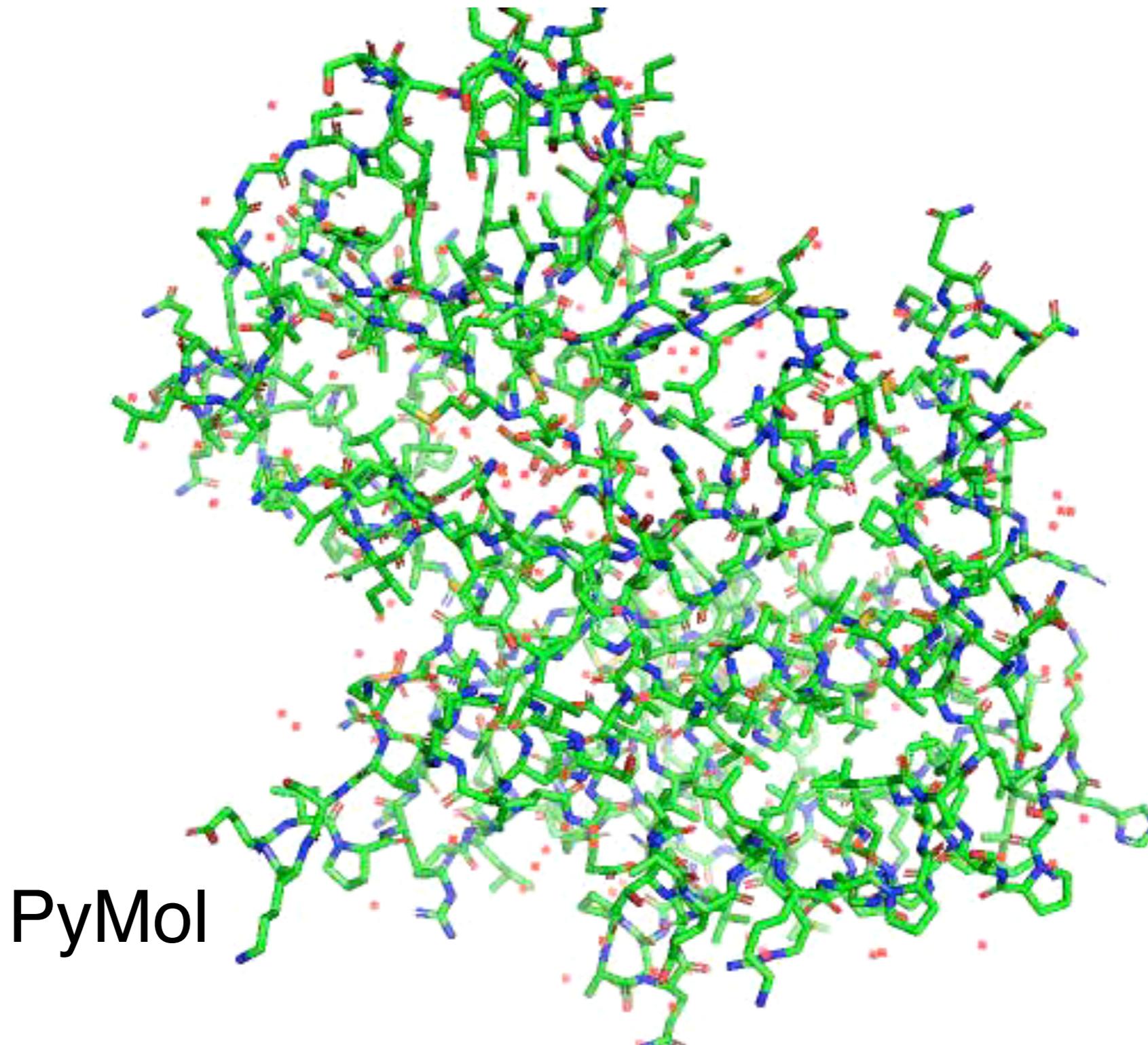
# Reference for VMD extensions and plugins

- VMD plugins: <http://www.ks.uiuc.edu/Research/vmd/plugins/>
- VMD scripts: <http://www.ks.uiuc.edu/Research/vmd/>

# VMD/PyMol assignment: Make a movie and a short report which narrates your movie

- Assignment, tutorial material posted
- Also posted example PyMol movie scripts
- Goal: Make a movie like you would use in a talk, and write a report which teaches something about the system and narrates your movie
- ***You do NOT have to make a VMD Tcl script.***

# PyMol Example from another course: p21-activated kinase (PAK4) and small- molecule inhibitor PF-3758309



# PyMol example from prior year: Designed protein nanocapsule for biotechnology

PyMol

# VMD: Soluble Epoxide Hydrolase ligand unbinding and rebinding

