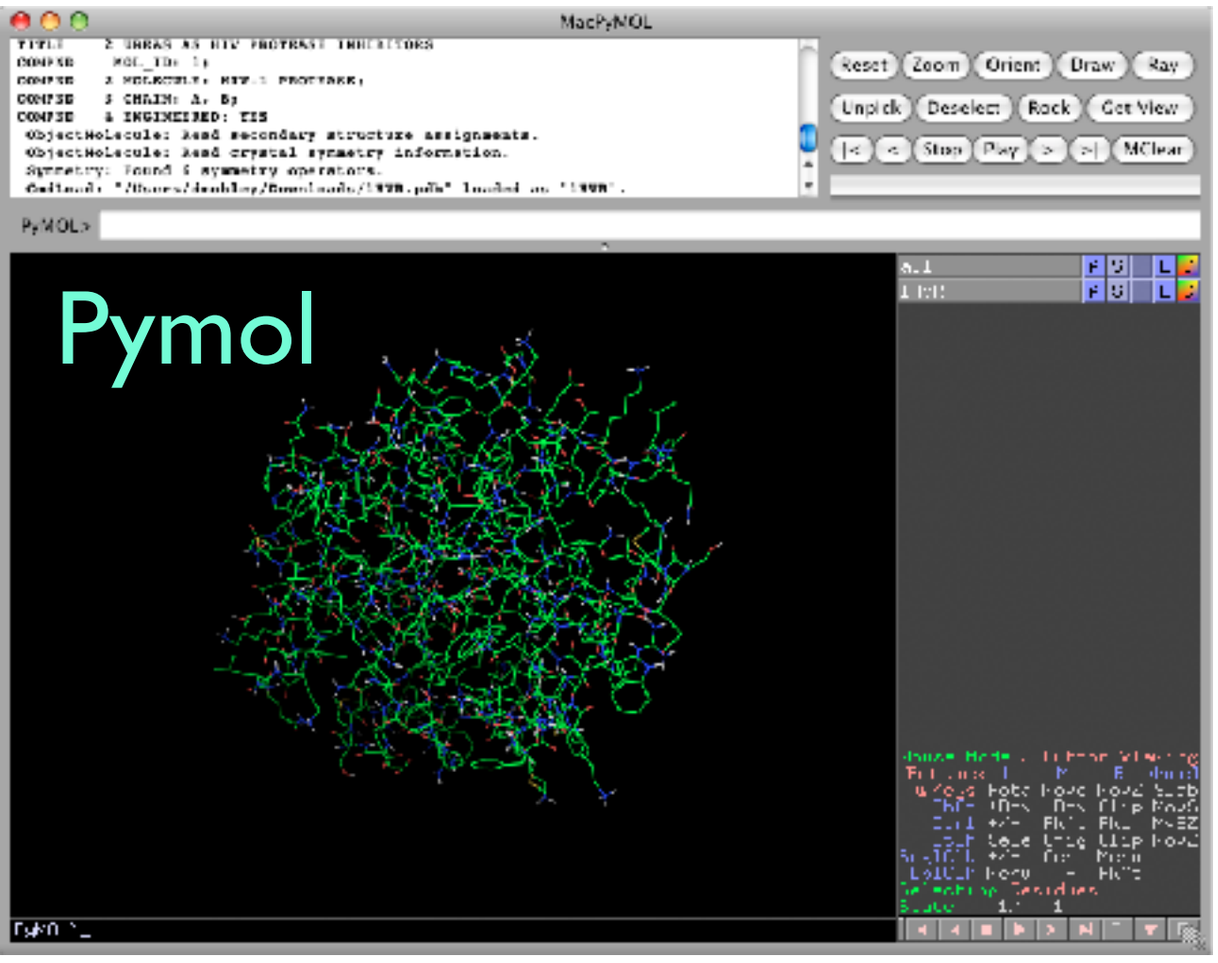
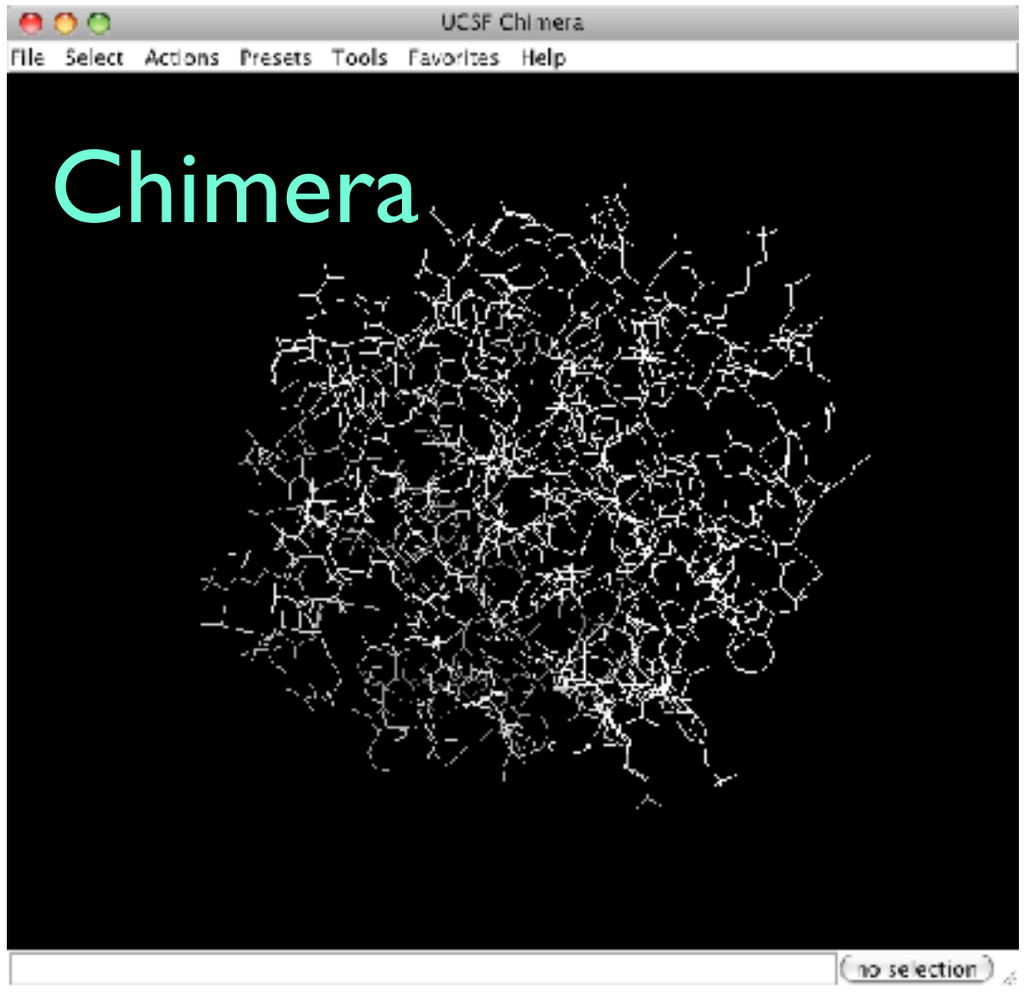
Notes/Logistics

- 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

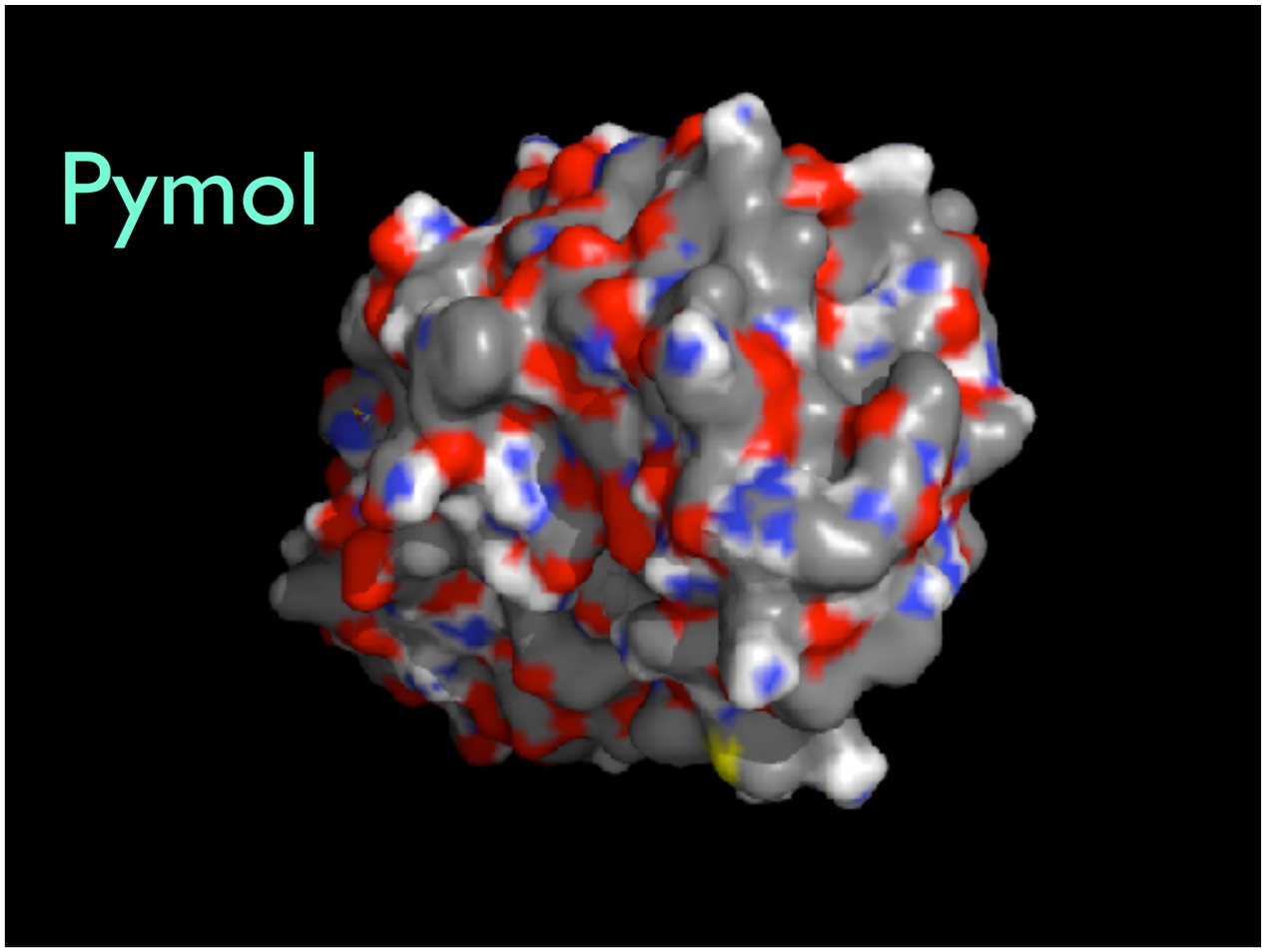
To work in 3D, you should become familiar with a general purpose viewer

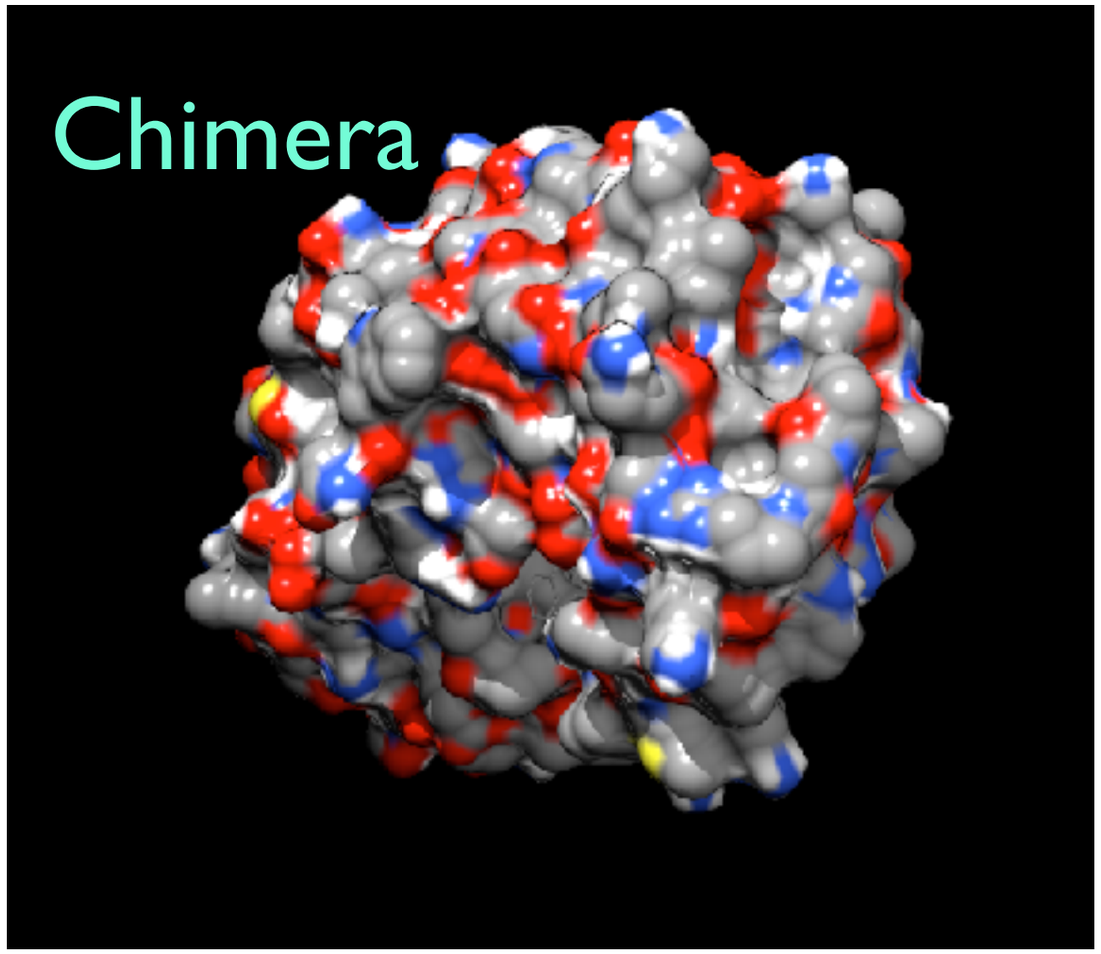




- Pymol and Chimera are good, free, general-purpose viewers
- Both Python-based:
  - Can run Python code
  - Easily extensible

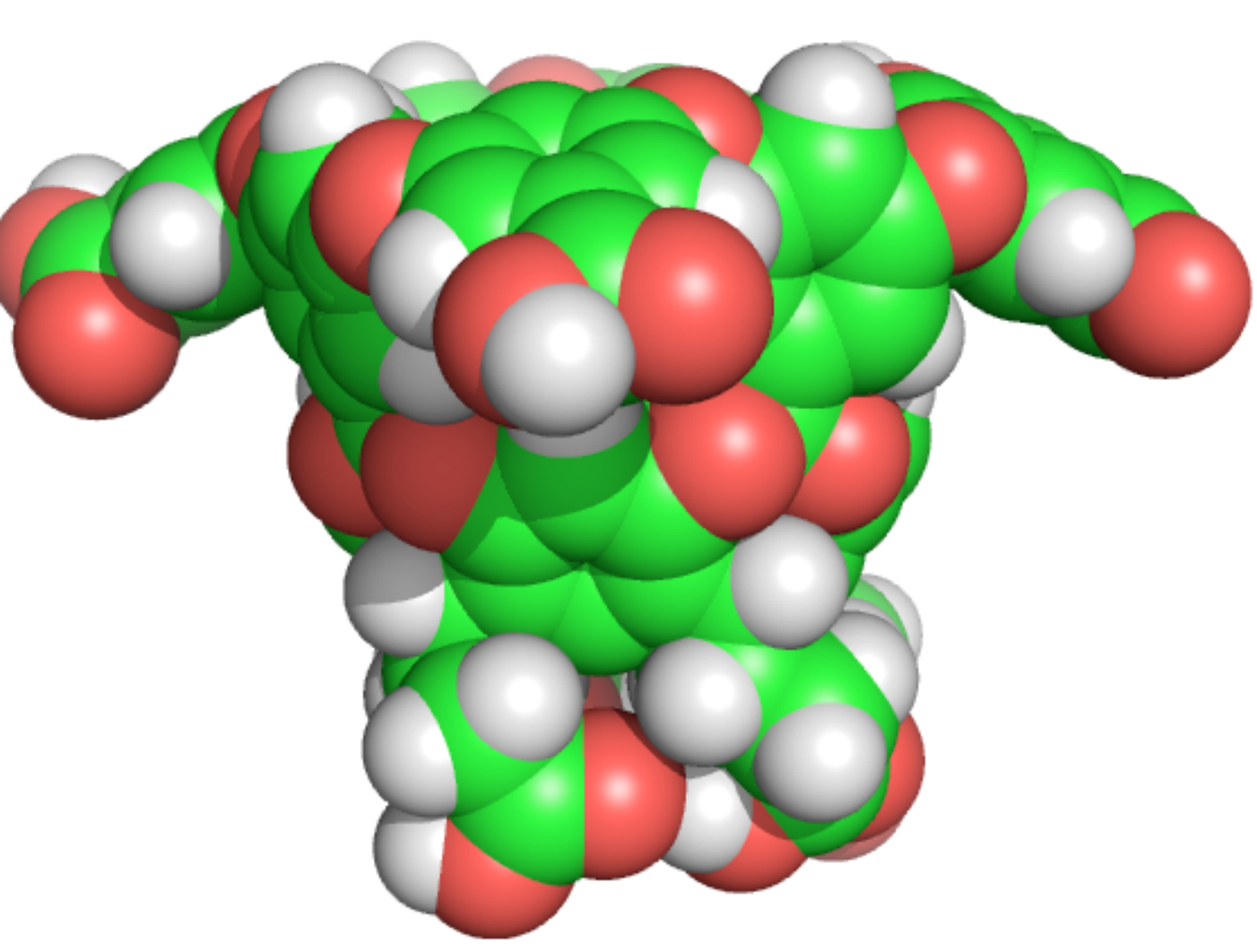
Both make high quality, ray-traced images

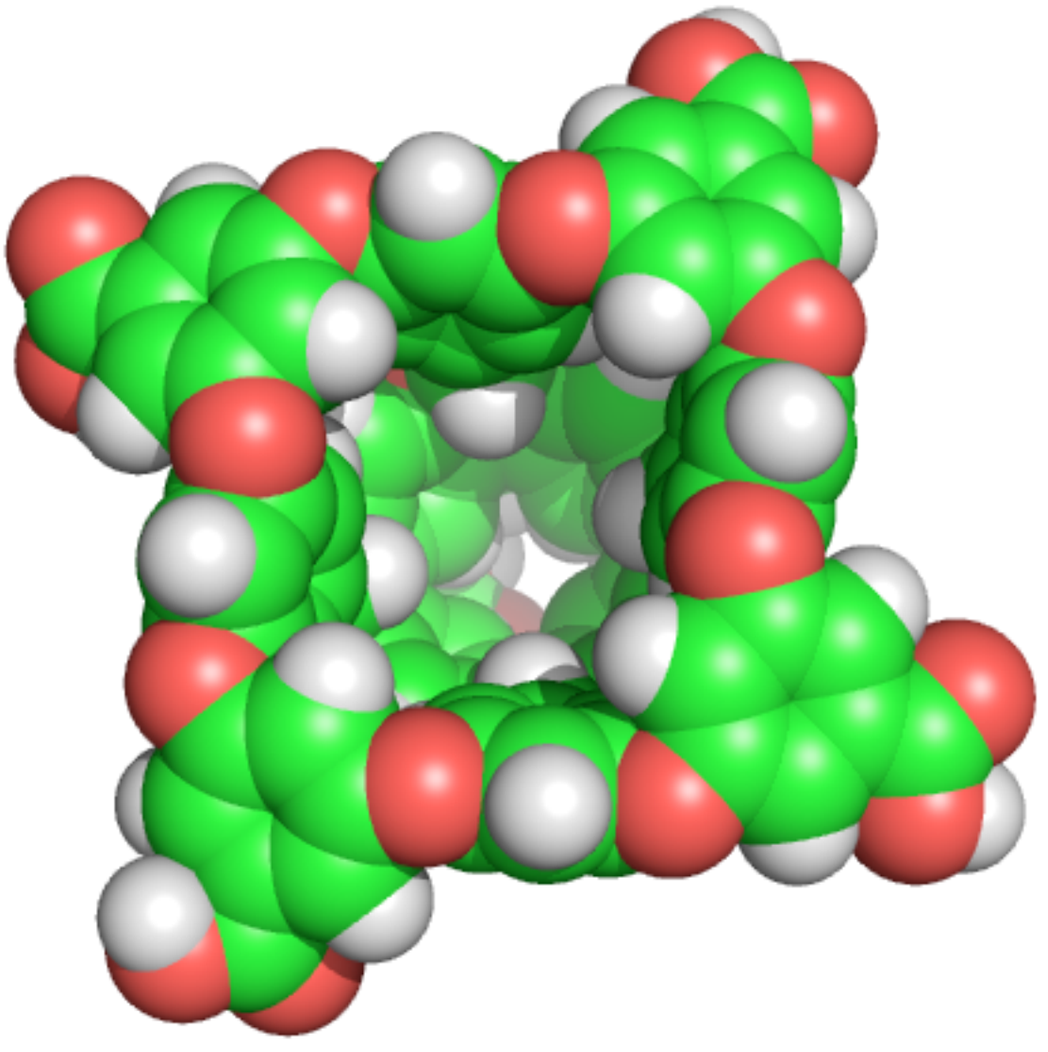




We have to pick one, so in this course when I explain how to do visualization tasks, it will be mostly with PyMol

Here's the cavitand again: Much more clear

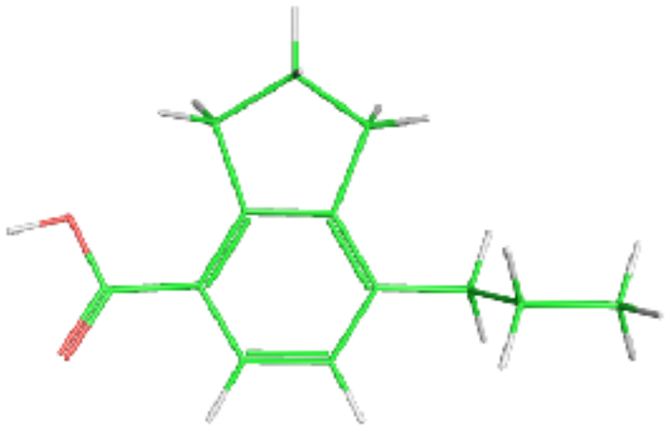


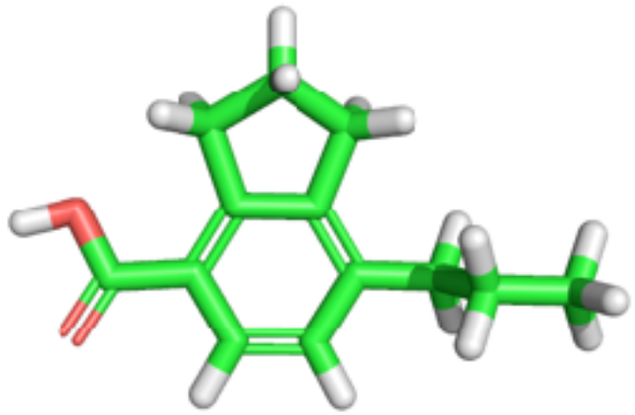


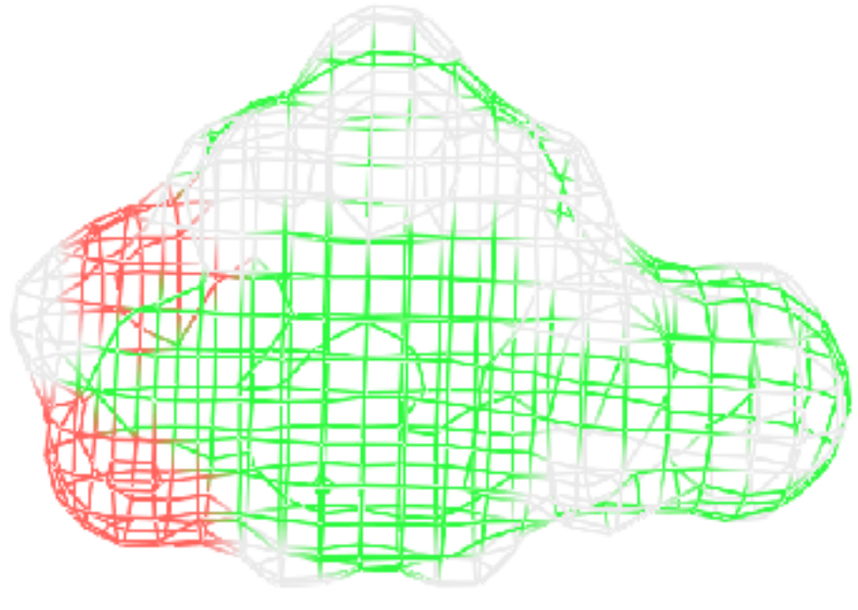
Both have a pretty full set of features; Chimera probably even more

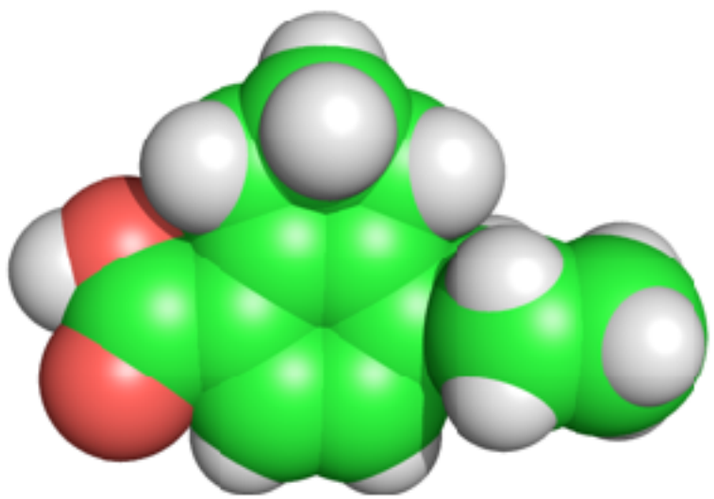
- Can build and edit molecules
- Build new proteins from protein sequence, for example
- Molecular movies
  - Morphs and different views of static molecules
  - Viewing molecular simulations
- Structure editing tools, mutagenesis tools
- Structure alignments
- Identify hydrogen bonds, add text labels, etc.
- Can work with electron density
- Chimera adds other things:
  - Working with EM density
  - Building structures of large assemblies (i.e. viruses),
  - Many others

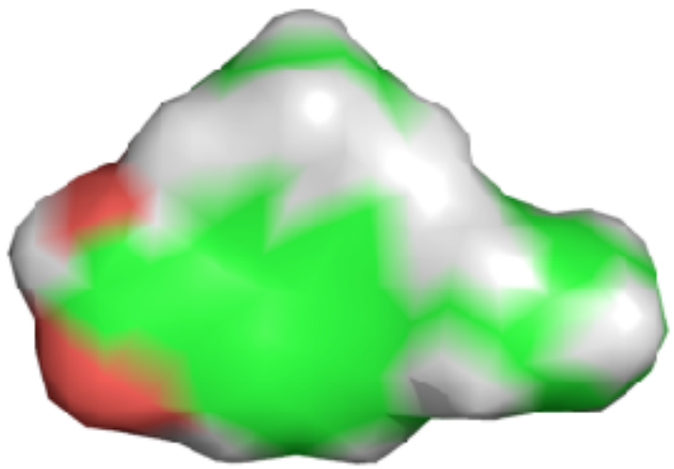
There are many common representations of molecules



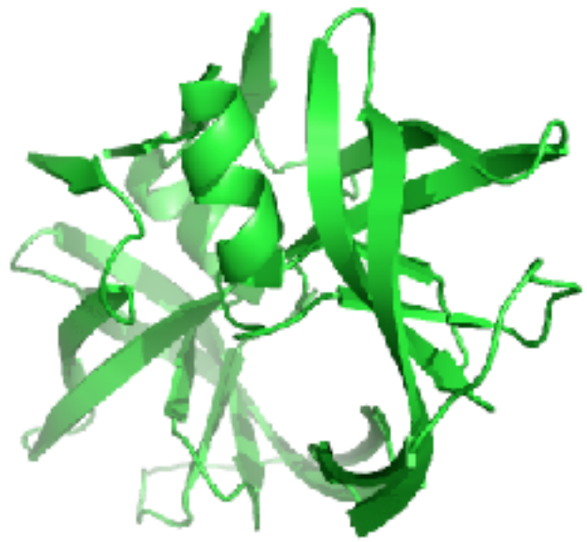


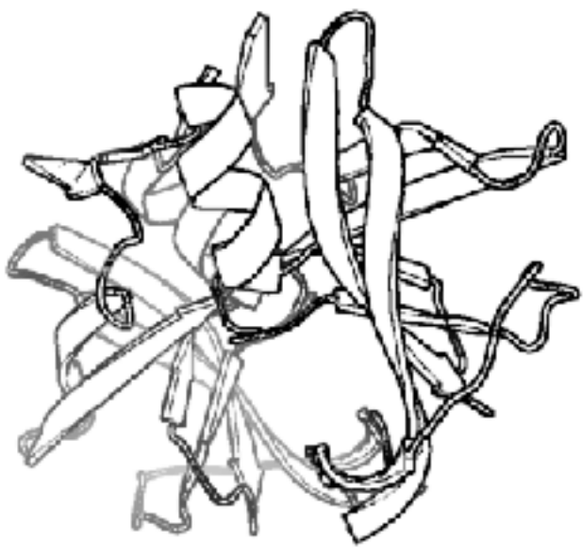


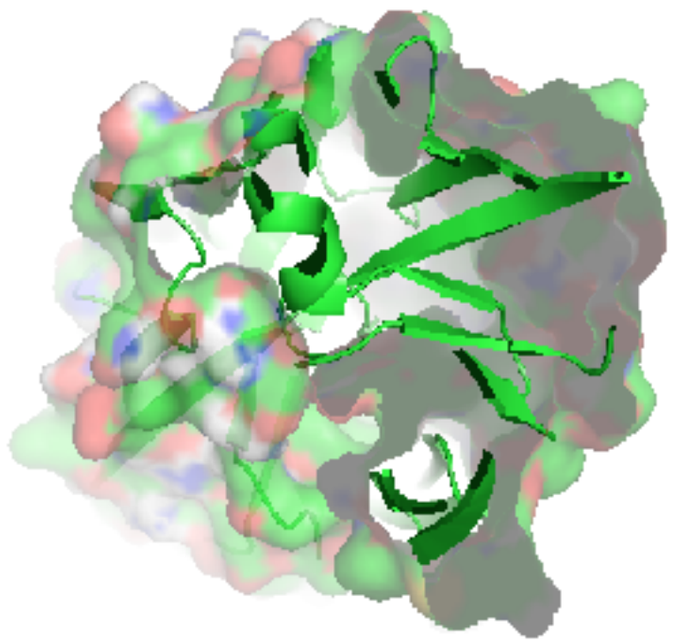




Proteins add several additional common representations







Demo time: Representations

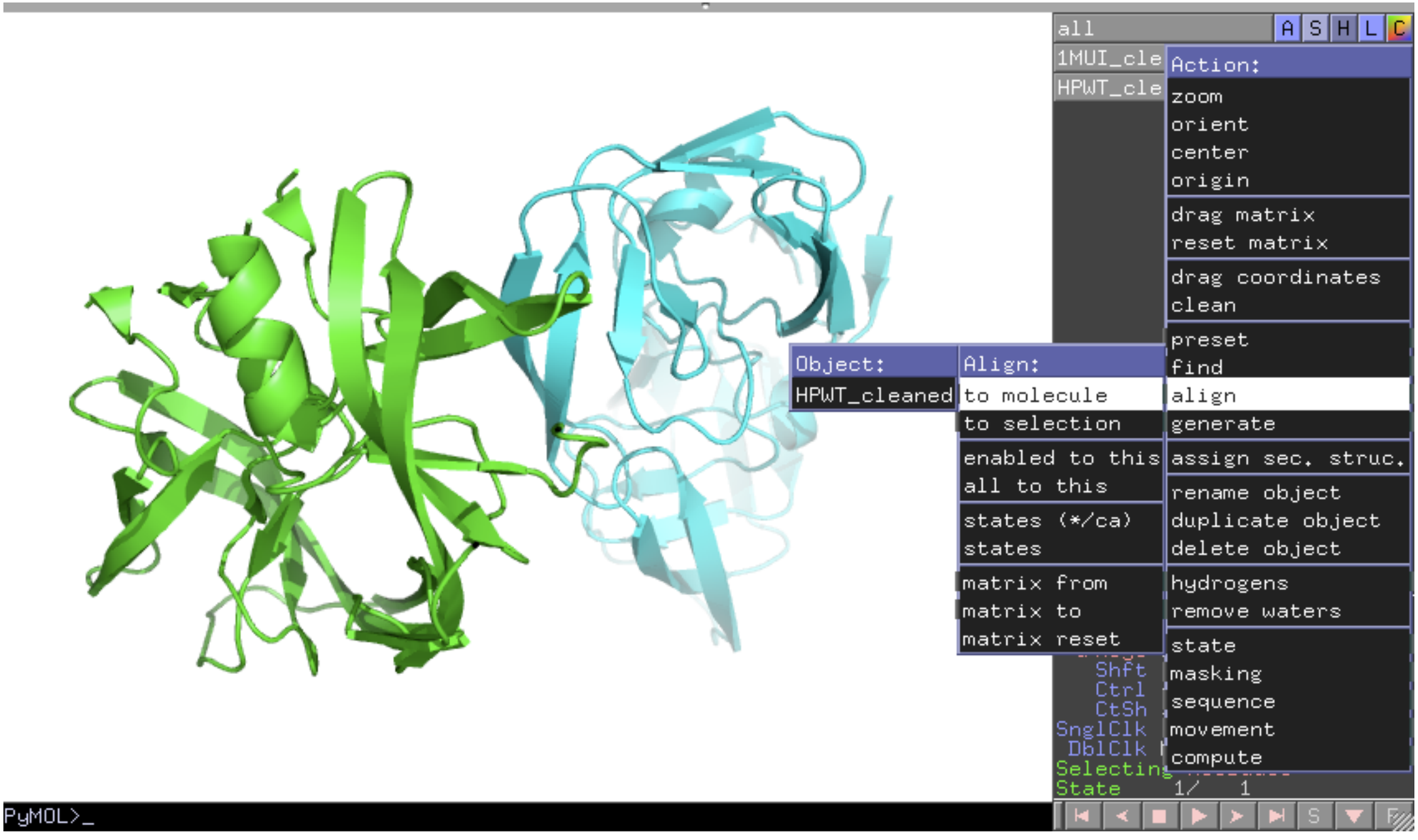
Visualizing something in 3D requires a 3D structure, which can be modeled

- 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

PyMol 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 full Python ‘interpreter’
    - Can actually run Python commands
    - Can write Python scripts to run in PyMol (we’ll come back to this)
- Important tip: escape key switches between viewer and text views

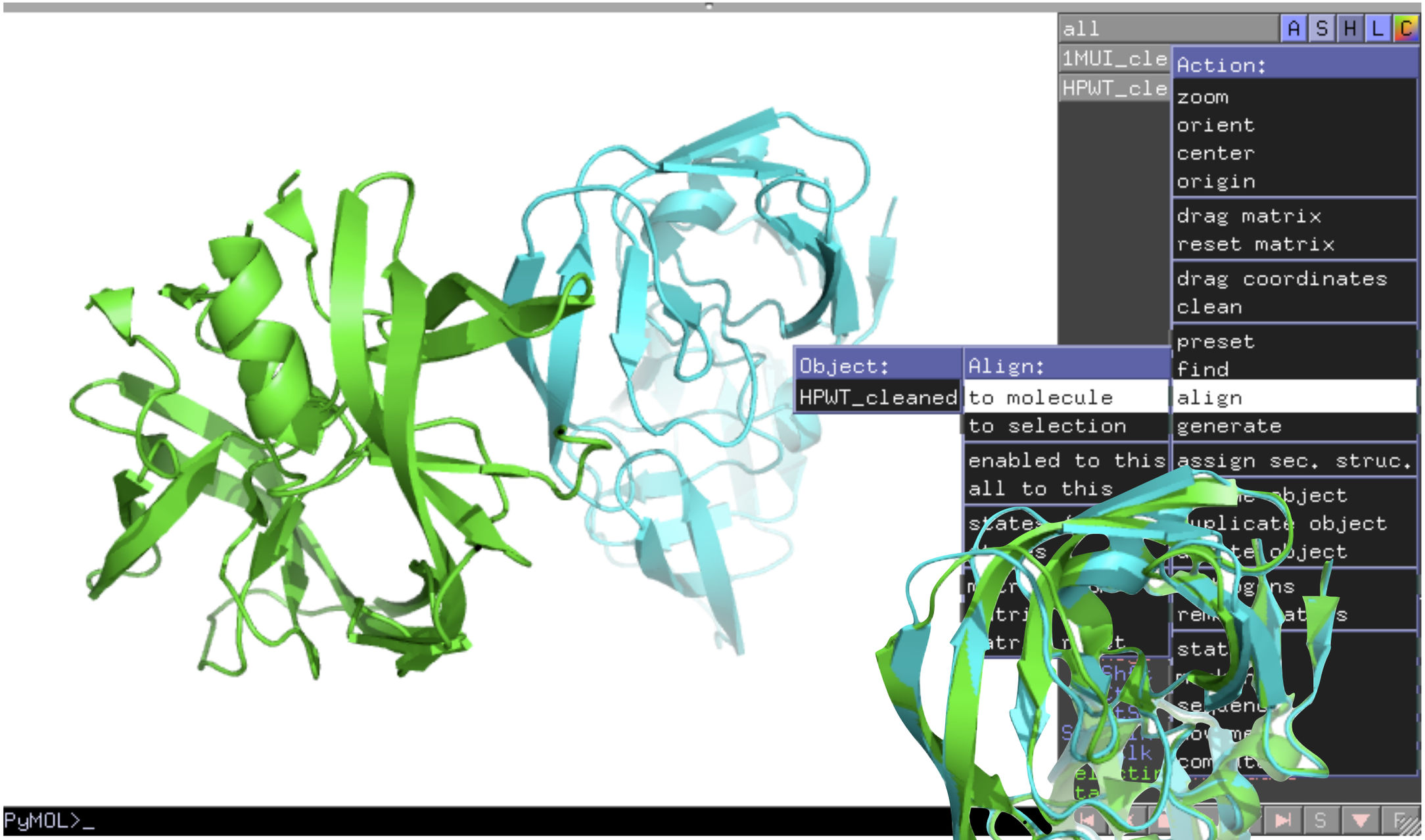
Common tasks in Pymol: Structure alignment



- Or:

```
PyMOL>align 1MUI_cleaned, HPWT_cleaned_
```

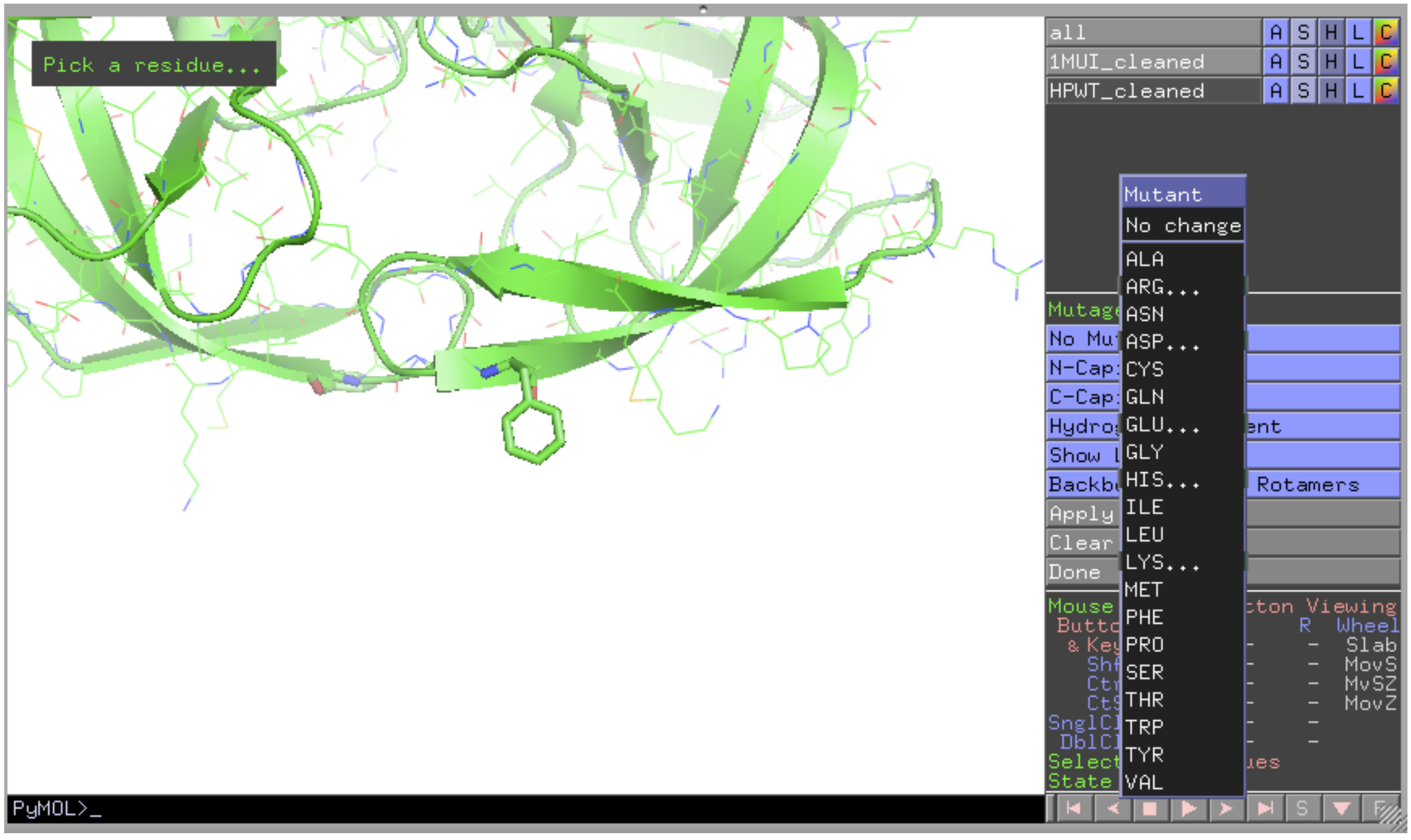
Common tasks in Pymol: Structure alignment



- Or:

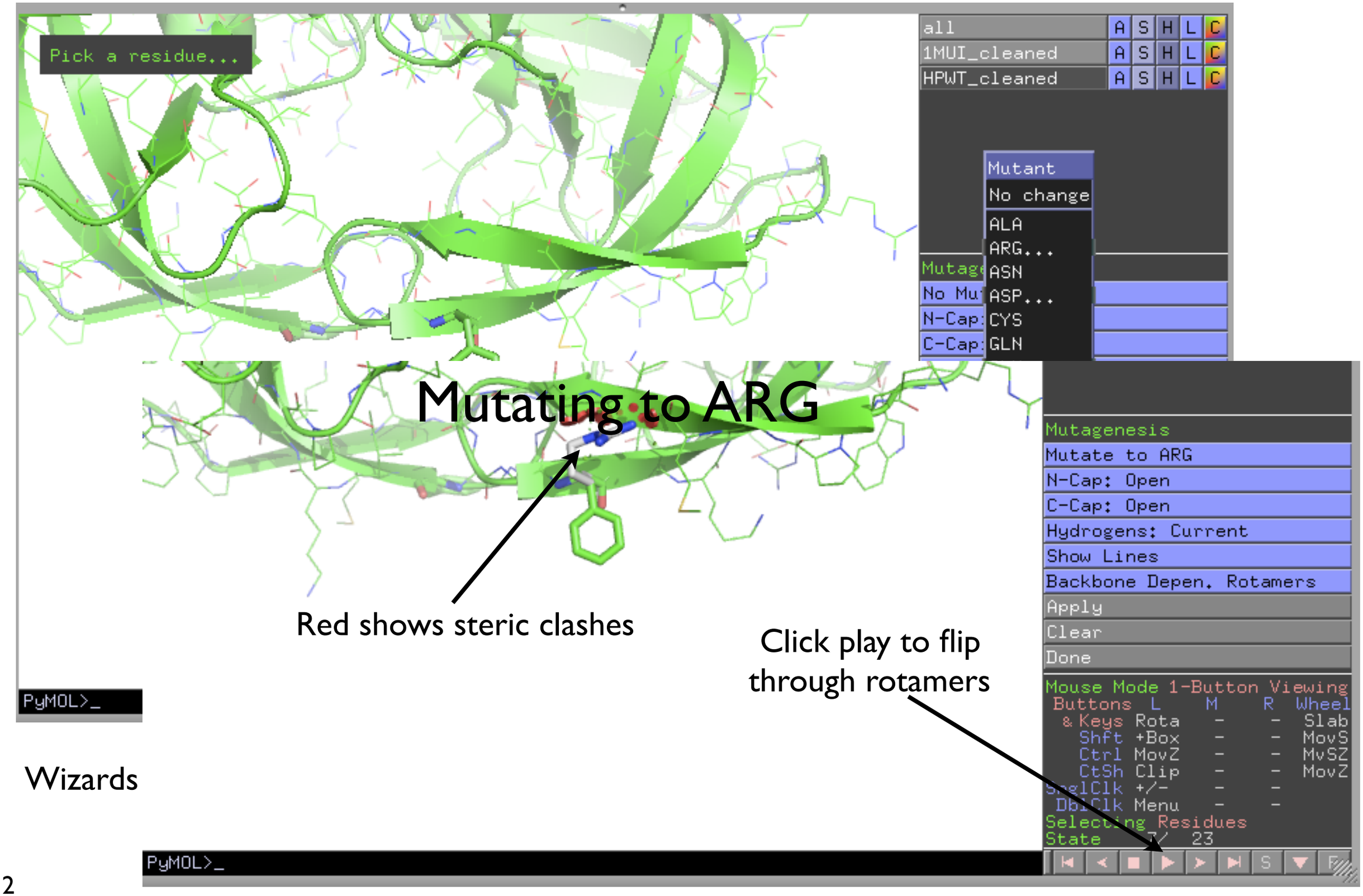
```
PyMOL>align 1MUI_cleaned, HPWT_cleaned_
```

Common tasks in PyMol: Mutagenesis



Wizards -> Mutagenesis wizard

Common tasks in PyMol: Mutagenesis



Mutating to ARG

Red shows steric clashes

Click play to flip through rotamers

Wizards

Common tasks in PyMol: Making a simple movie

- Just open (load) a structure file (i.e. PDB or mol2) containing multiple structures and use controls
- PyMol can ‘render’ these into formats you can play from within presentations, with ray tracing if you like

PyMol and Python work together

- PyMol can be extended using Python
- As noted, Python scripts run in PyMol
- Python is great for using PyMol to do repetitive tasks using command-line interface
  - See command-line 'help' command to figure out how to do this
- Example script for loading, aligning some strs:

```
cmd.set('bg_rgb','1,1,1') #Set background color to white


for prot in ['1HVR', '1MUI', '1HXW']:
    cmd.load( prot+'.pdb', prot ) #Load each protein as a new object
    cmd.align( prot, '1HVR') #Align each protein

cmd.show('cartoon') #Show cartoons for protein
cmd.hide('lines') #Hide lines
cmd.center('1HVR') #Center on one protein (really all, since aligned)
cmd.remove('SOLVENT') #Remove solvent
cmd.show('sticks', 'HETATM') #Show sticks for ligands bound
```

Python in PyMol makes it easy to build sophisticated movies that will blow people away

Python in PyMol makes it easy to build sophisticated movies that will blow people away

Part of the script

```
cmd.set('bg_rgb','1,1,1') #Set background color to white


for prot in ['1HVR', '1MUI', '1HXW']:
    cmd.load( prot+'.pdb', prot ) #Load each protein as a new object
    cmd.align( prot, '1HVR') #Align each protein

cmd.show('cartoon') #Show cartoons for protein
cmd.hide('lines') #Hide lines
cmd.center('1HVR') #Center on one protein (really all, since aligned)
cmd.remove('SOLVENT') #Remove solvent
cmd.show('sticks', 'HETATM') #Show sticks for ligands bound
preset.publication()

cmd.create('lig', 'resn RIT') #Create selection of a particular ligand
cmd.remove('1HXW and resn RIT')
cmd.show('sticks', 'lig')
cmd.zoom()

#Do some initial orientation stuff
cmd.turn('y', '45')
cmd.turn('z', '-17')
cmd.turn('x', '-20')

#Now do some movie making stuff
cmd.mset('1 x360') #Create a 360 frame movie with all frames copies of first
#Store our initial view to frame 1
cmd.mview('store', '1')
#Turn 180 degrees around y axis
cmd.turn('y', '180')
#Store this view in frame 36; intermediate frames will be interpolated
cmd.mview('store', '36')
#Turn another 180 degrees
cmd.turn('y', '180')
#Store this view to frame 72
cmd.mview('store', '72')

#We still have nothing after frame 72 and we created a 360 frame movie.
```

Useful tip: Use log_open to figure out commands for things

- log_open mysession.py, for example
- Records sequence of commands, including menu options
- Can be used as start of a simple movie
- Also, use the ‘help’ and ‘help topic’ options frequently both to get basic help and API help

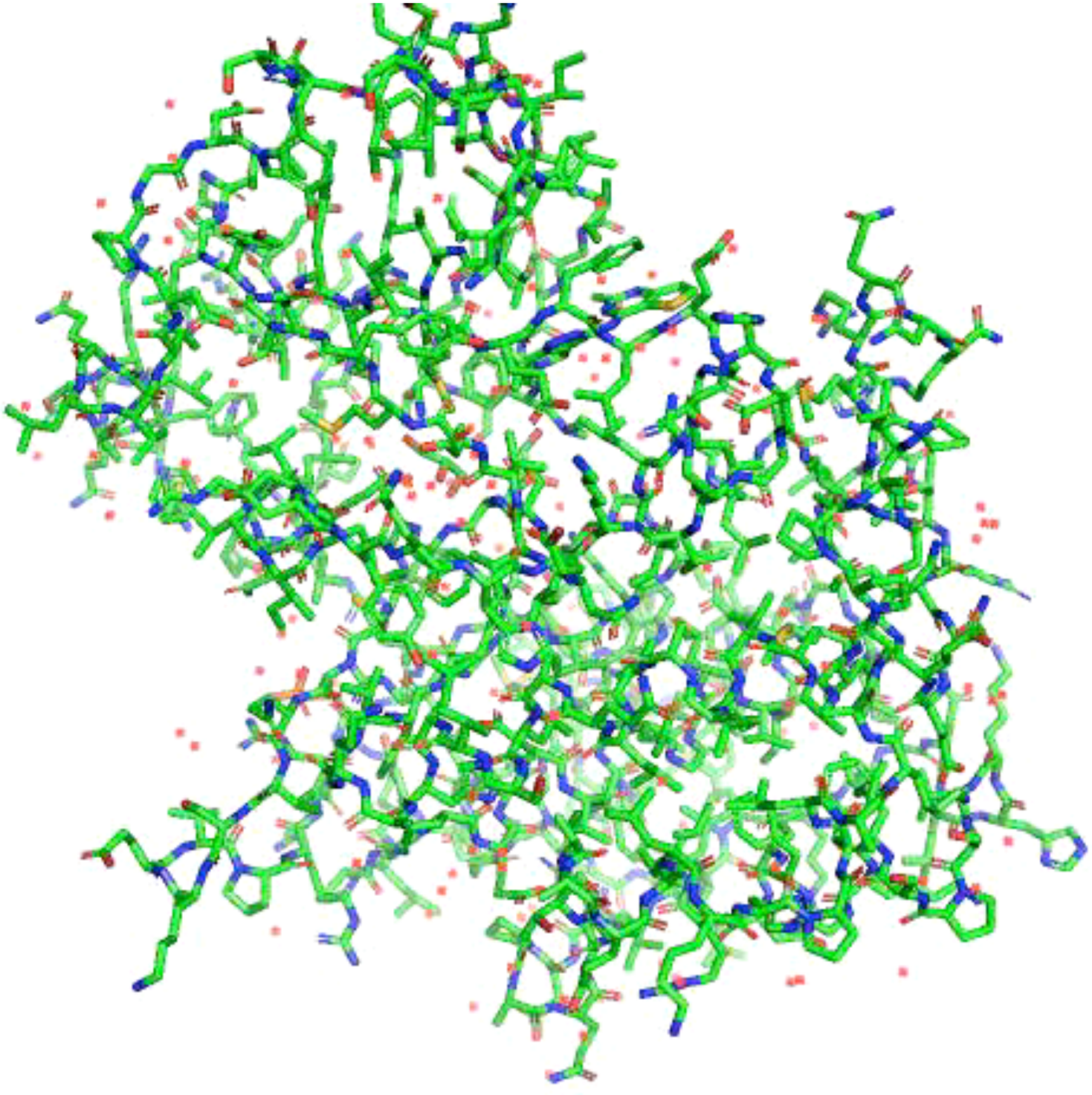
Some summary

- Lots of good, free tools are available for visualization and common molecular editing and structure building tasks
- PyMol shares a lot of features with most
  - Don’t reinvent the wheel when you don’t have to
- PyMol is great for making movies relating to structural analysis

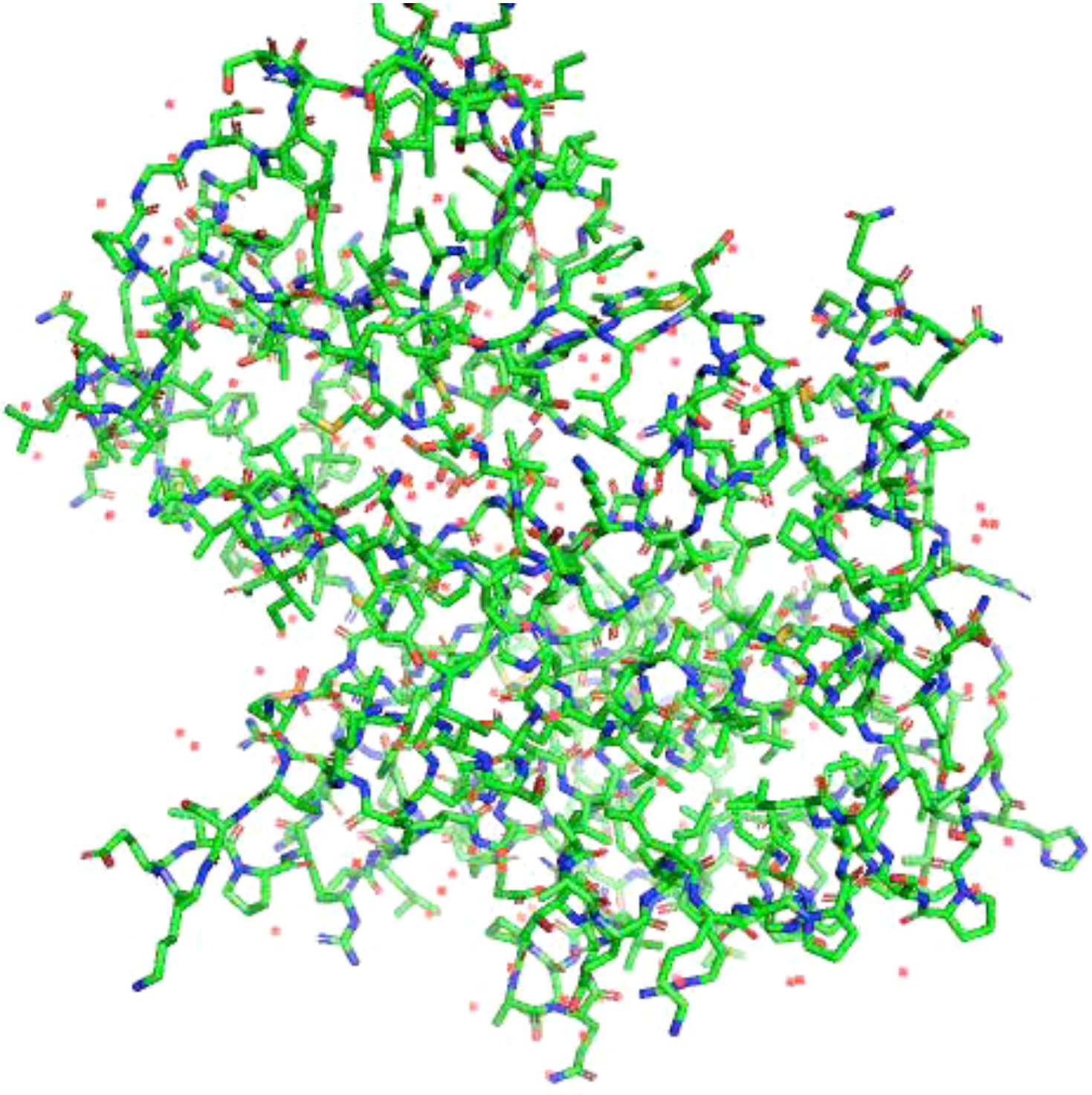
PyMol assignment: Make a movie and a short report which narrates your movie

- Assignment, tutorial material posted
- Also posted example 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

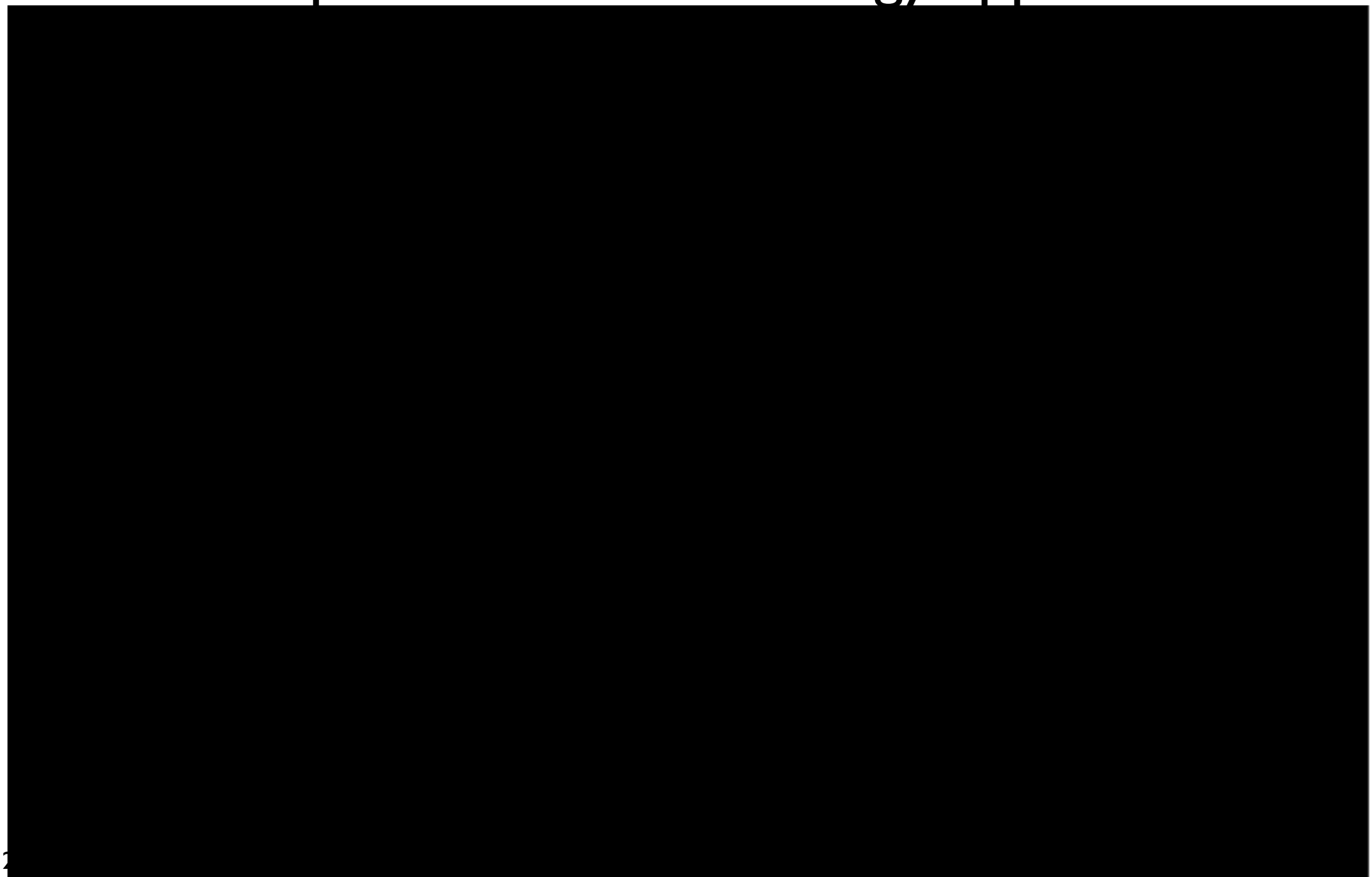
Example from another course: p21-activated kinase (PAK4) and small-molecule inhibitor PF-3758309



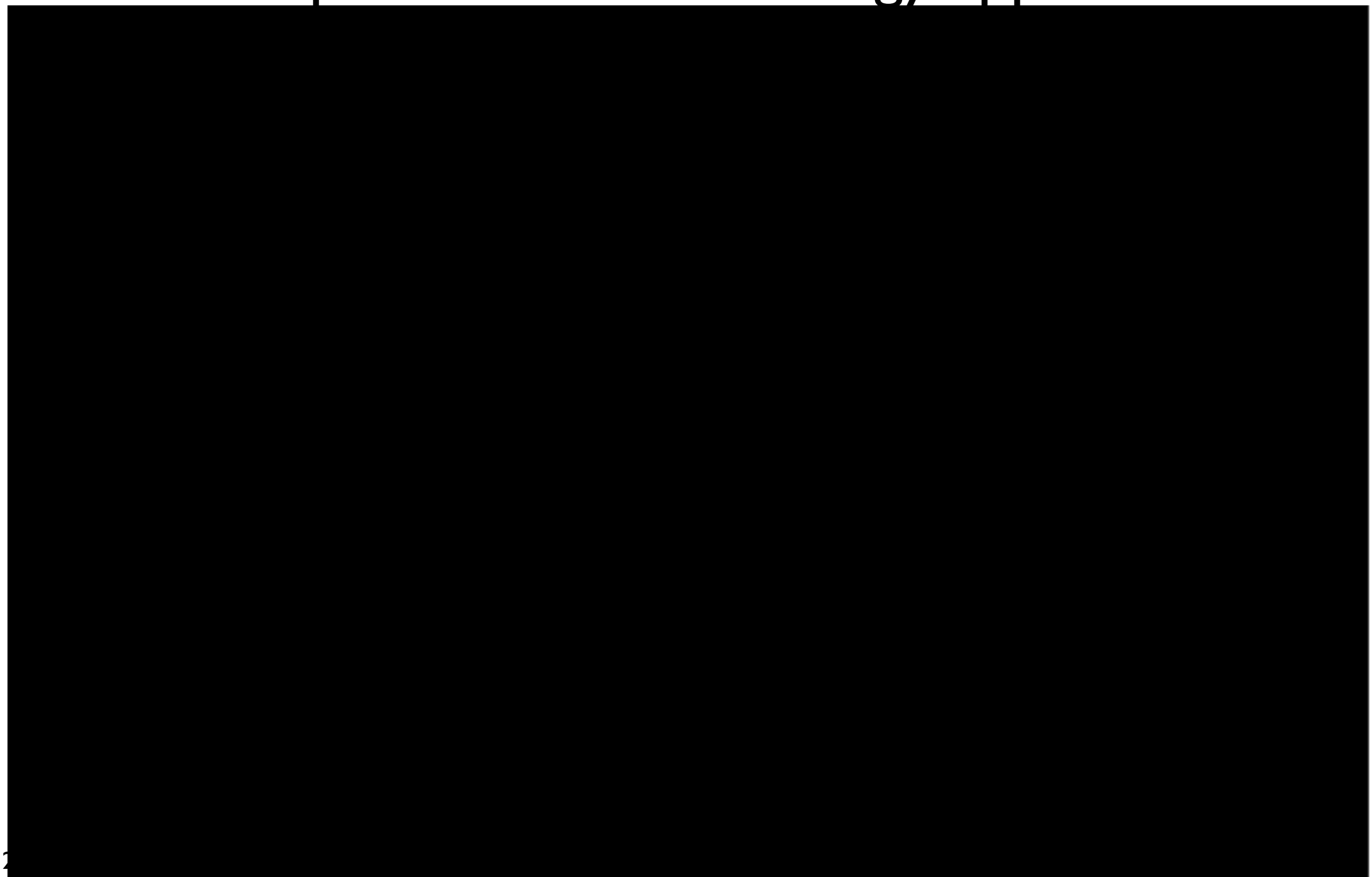
Example from another course: p21-activated kinase (PAK4) and small-molecule inhibitor PF-3758309



Second example from prior year: Designed protein nanocapsule for biotechnology applications



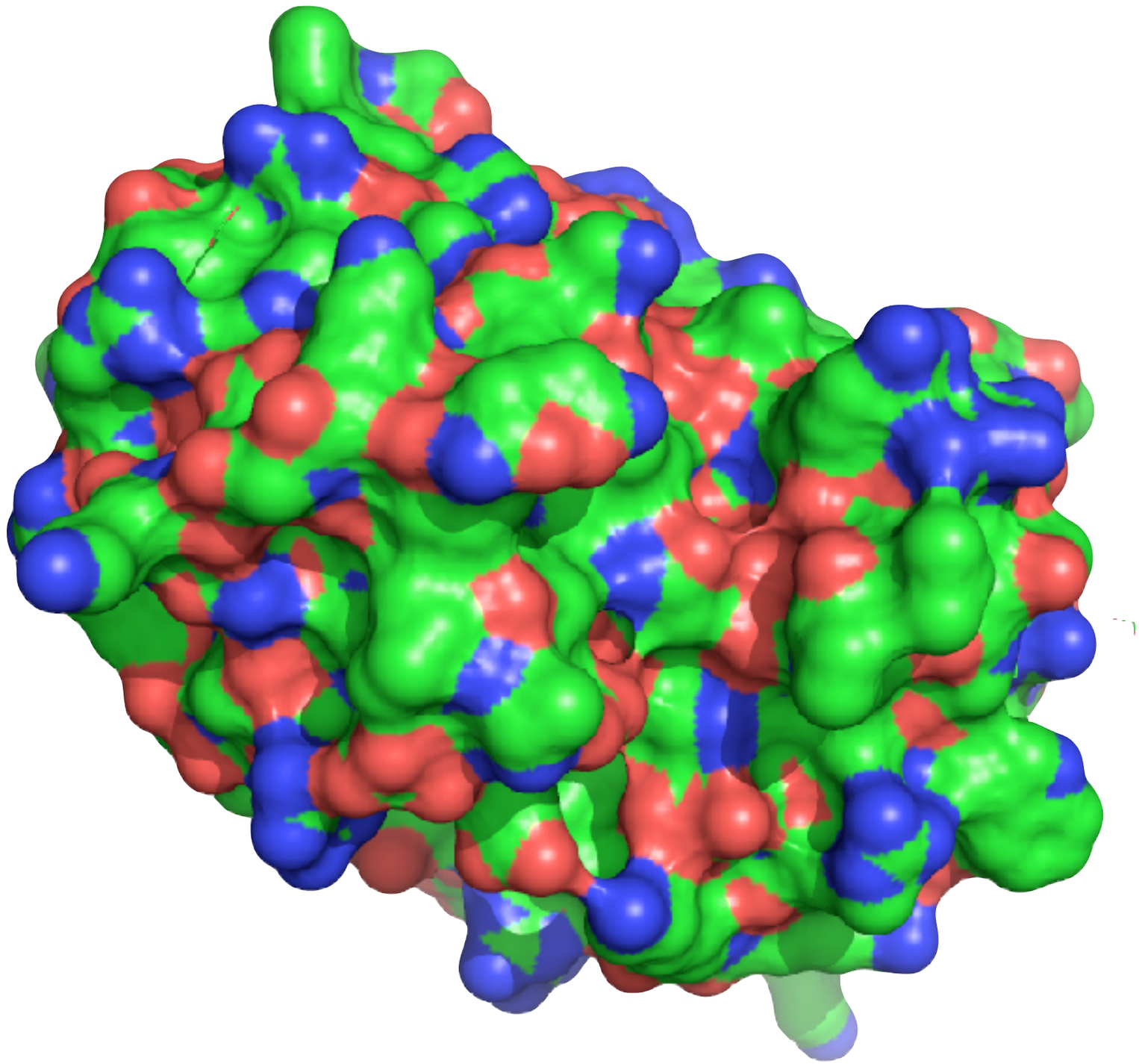
Second example from prior year: Designed protein nanocapsule for biotechnology applications



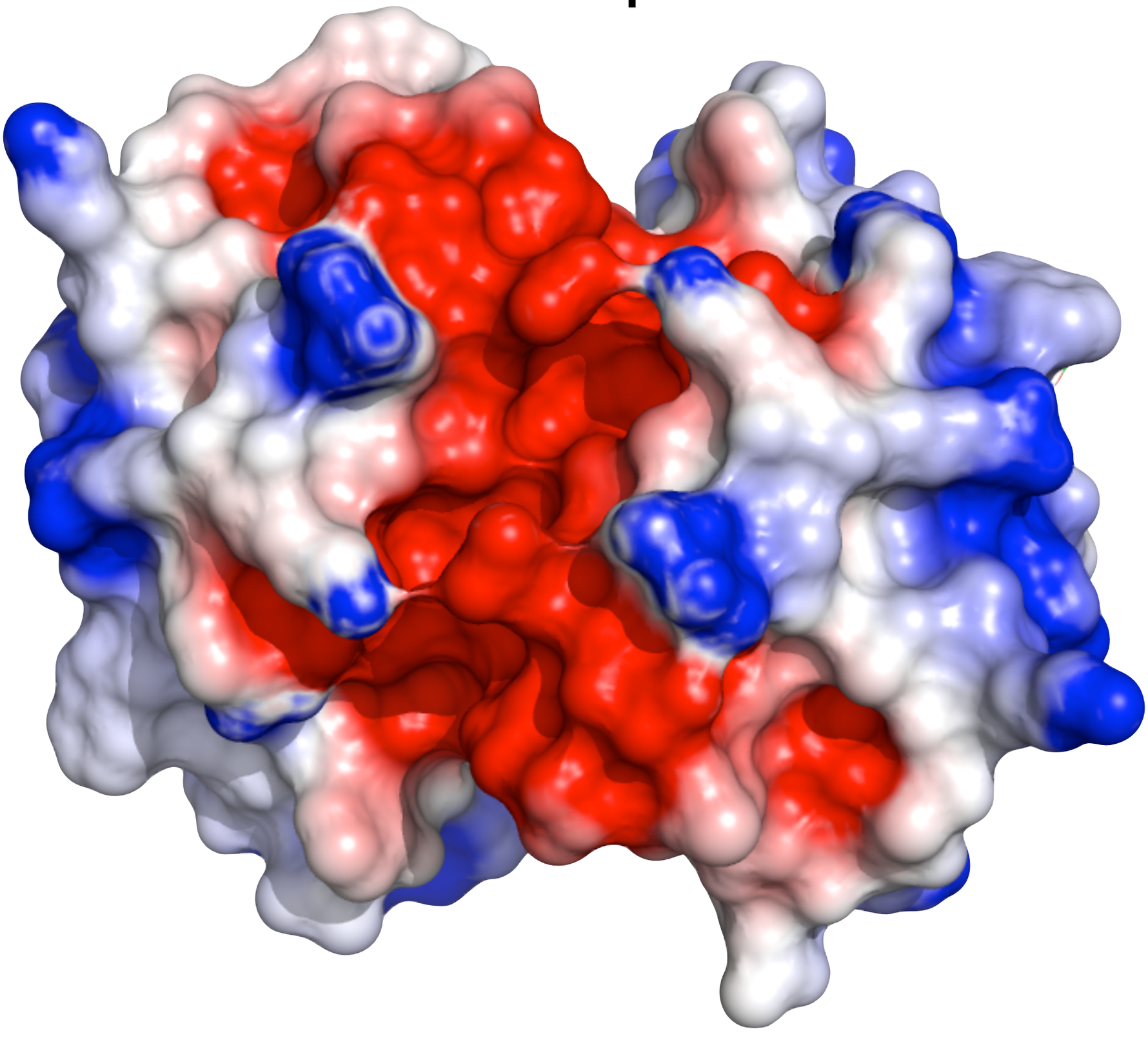
We are also interested in the electrostatic potential on the surface of molecules

- 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 change
- Can impact binding rates, strengths, ...

PyMol by default colors things by their element type



We can instead use colors to show electrostatic potential



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

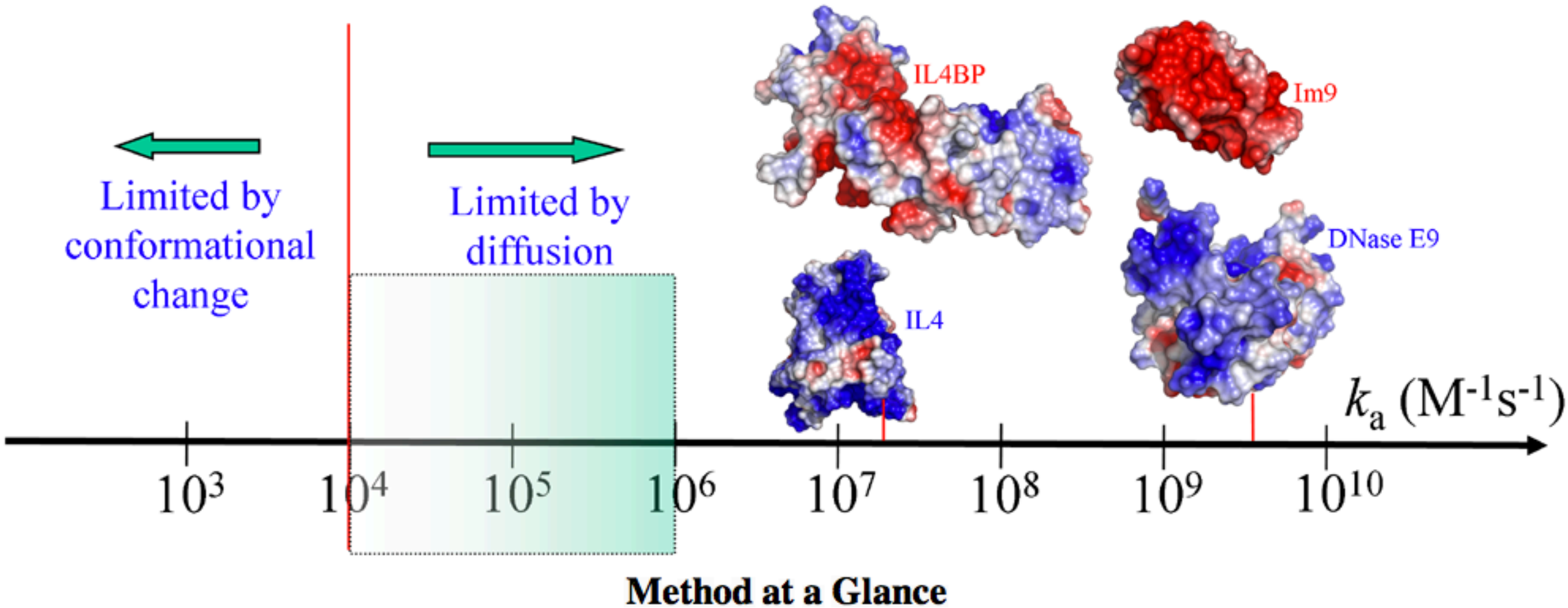
- Various tools can do this, including OpenEye's ZAP, APBS, and Delphi. Here, Delphi:
  - I can provide access to the software if this is something you want to do
  - See http://structure.usc.edu/howto/delphi-surface-pymol.html for details
  - Basically, run 'delphi delphi.param' on the command line, where delphi.param is a file specifying your target protein
  - This also uses input files specifying the size of protein atoms, and the typical amino acid charges
  - Then download the large fort.20 file as "mymap.phi"

Loading and coloring the potential in PyMol is easy

```
load 1W2I.pdb #Load protein structure
show surf #Show the surface
load mymap.phi, e_map #Load our map
ramp_new e_lvl, e_map, [-7, 0, 7] #Create new color ramp named e_lvl
set surface_color, e_lvl, 1W2I #Color surface based on the color ramp

ramp_new e_lvl, e_map, [-3, 0, 3] #Switch the color scheme range

set surface_quality, 2 #Make a nicer surface -- may take a few minutes

ray #Ray trace the graphic
```

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

TransComp: Web Server for Predicting Protein Association Rate Constants



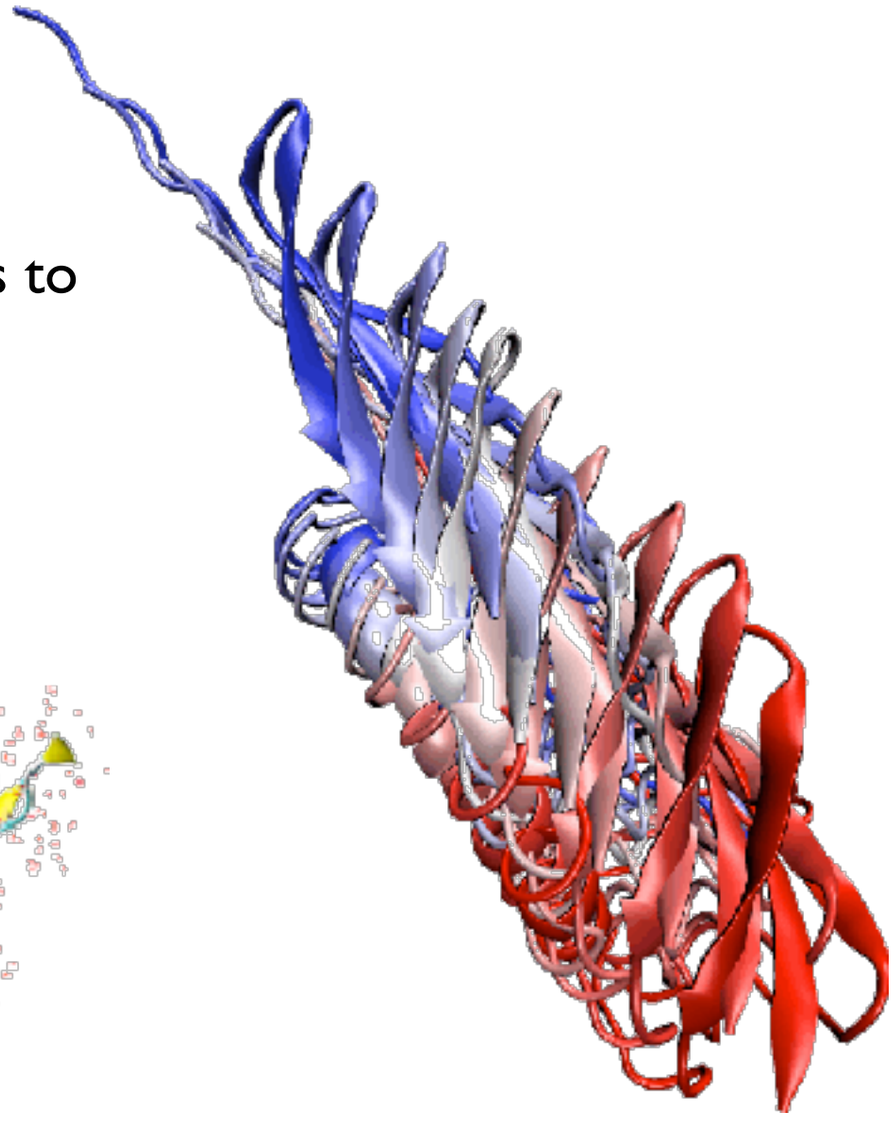
Method at a Glance

VMD is probably more popular than PyMol for movies of simulations; better equipped for this

- Memory management
- Better trajectory format handling
- Understands the idea of a ‘simulation’ or trajectory
- See for ex. http://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/

Follow through the tutorial - especially loading and working with trajectories - for basic usage

- Some things worth highlighting:
  - Trajectory smoothing: Averaging some number of frames to remove rapid oscillations
  - Displaying multiple frames at once, such as color coded



- Updating selections on the fly