VMD Visualization Tips

Esteban Dodero

CTBP

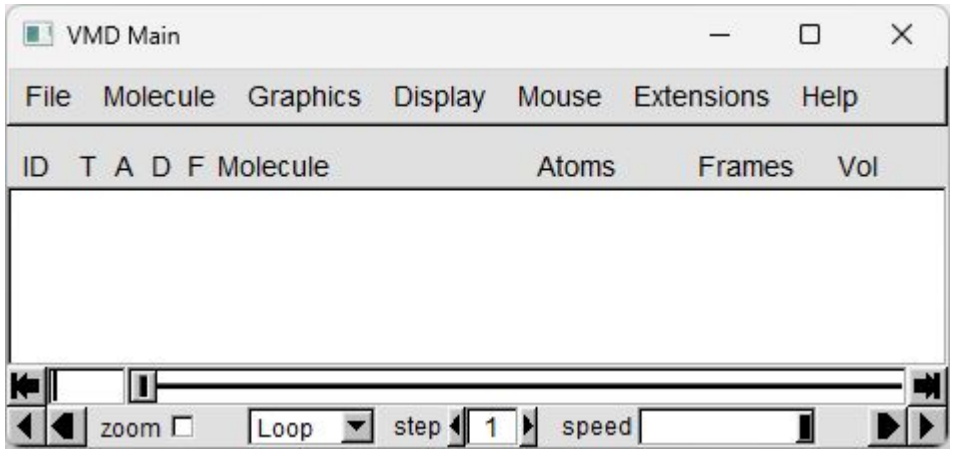
09/04/23

Topics

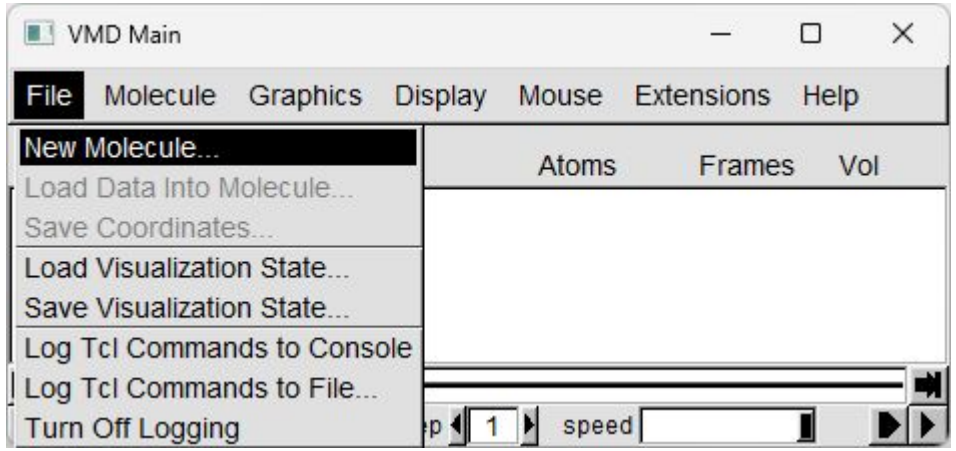
- Rendering images
  - Perspectives
  - Colors, representations, viewpoints
  - Rendering frames
  - Editing frames
- Making a movie
- Other visuzalation tips

VMD – Loading a molecule

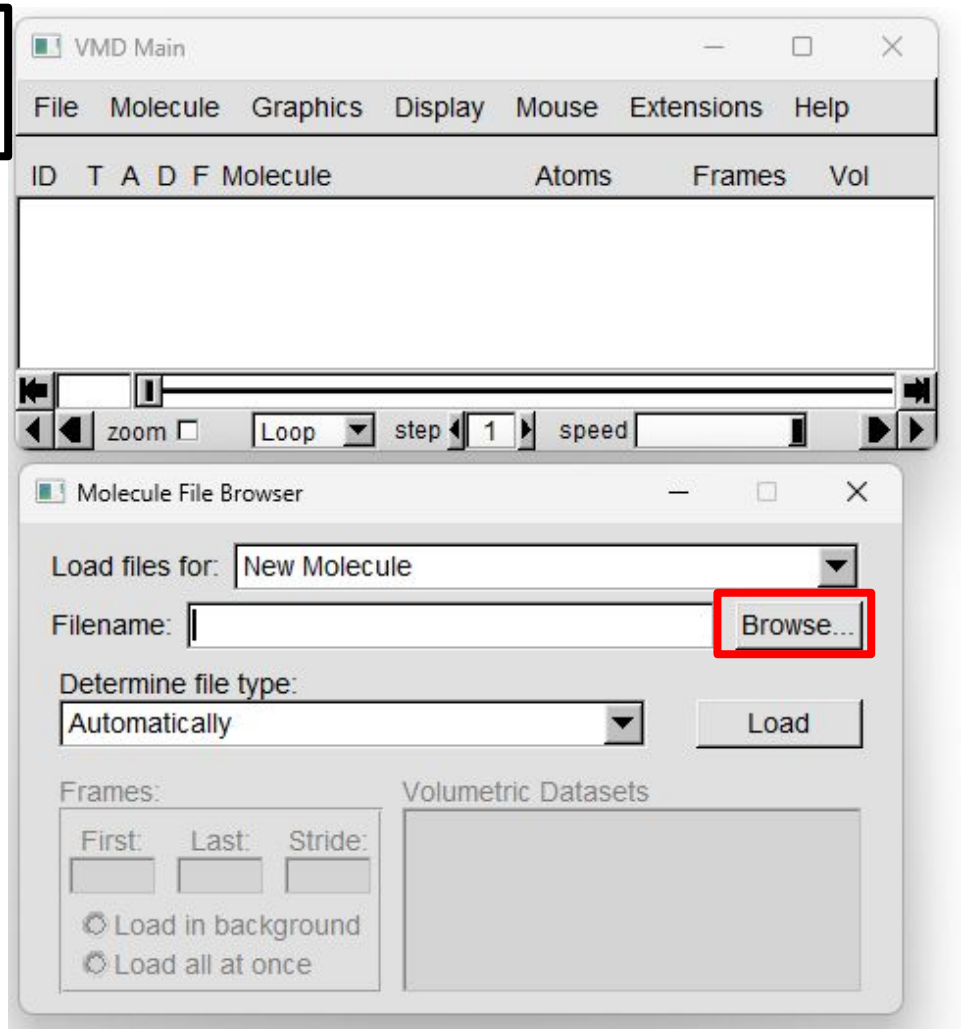
1



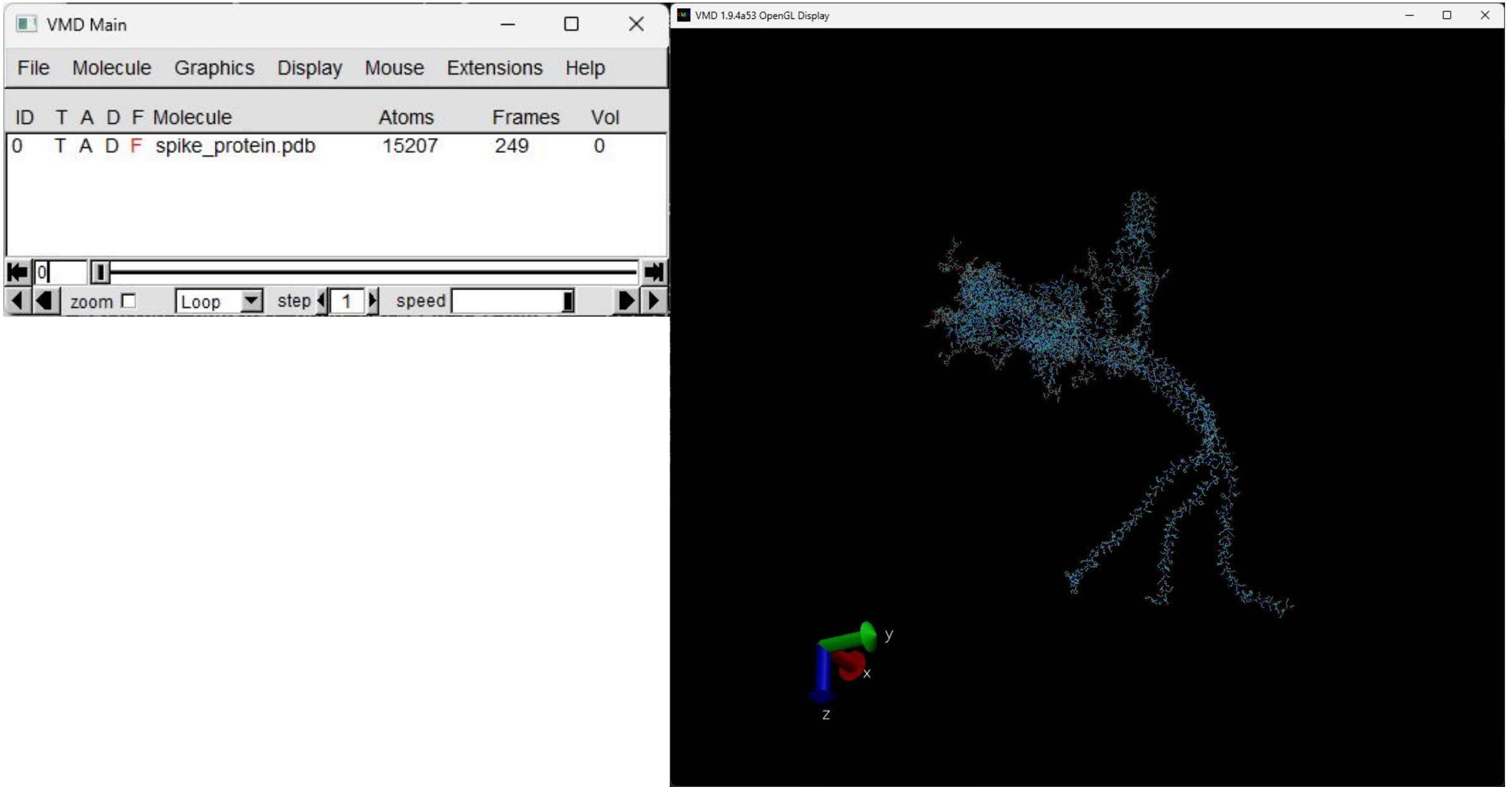
2



3



VMD – Loading a molecule

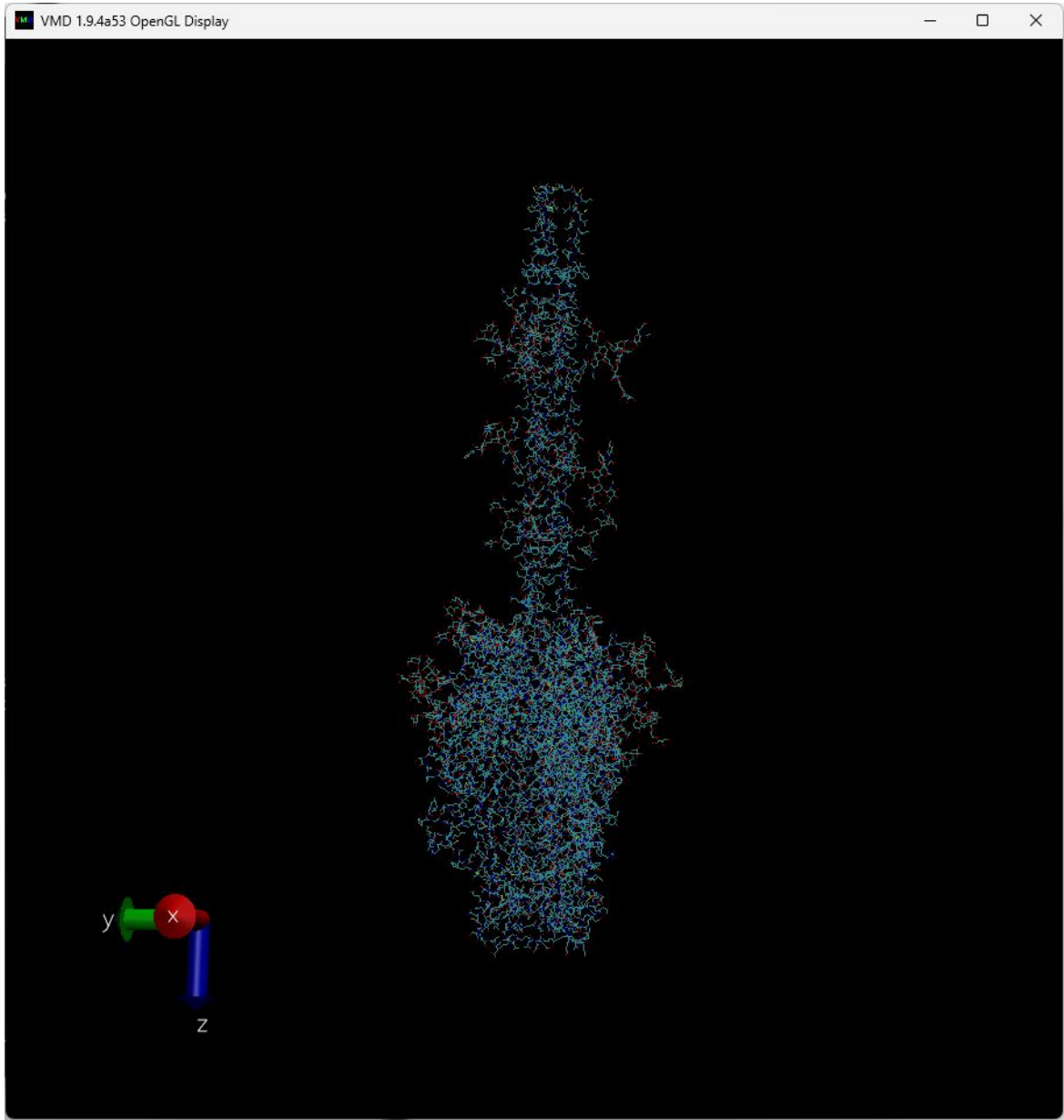


VMD – perspective

- Selecting a perspective

  Orthographic > Perspective

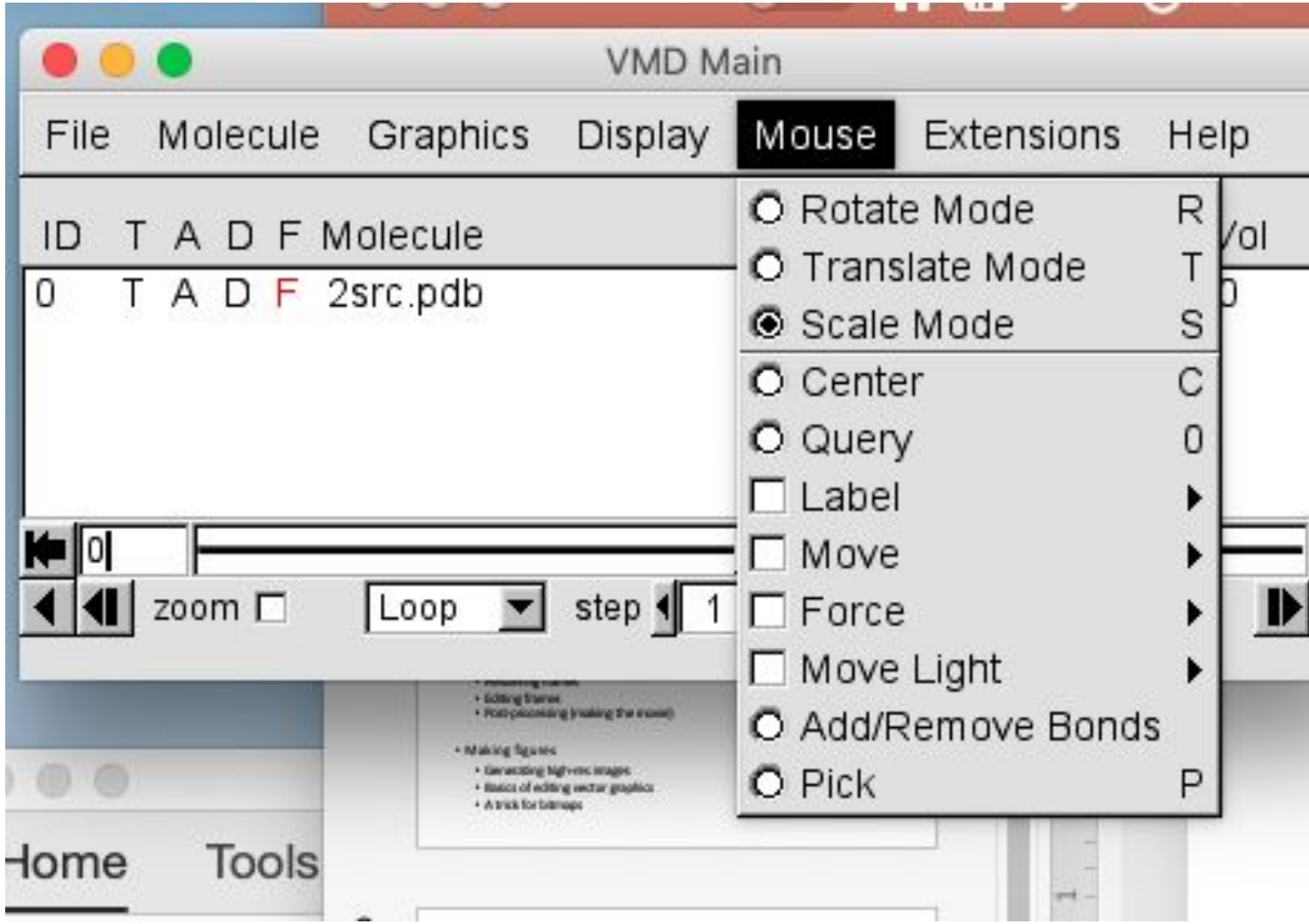
- Hide axes



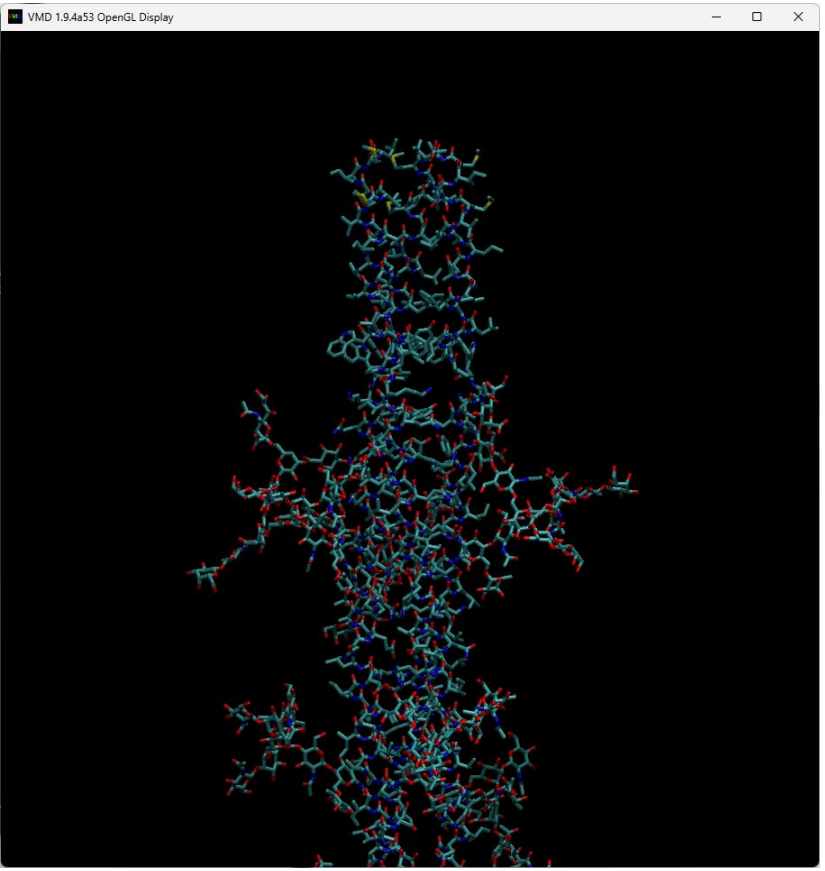
VMD – Move around

- Get used to navigating the different mouse modes
  - Rotate
  - Translate
  - Scale
  - Center

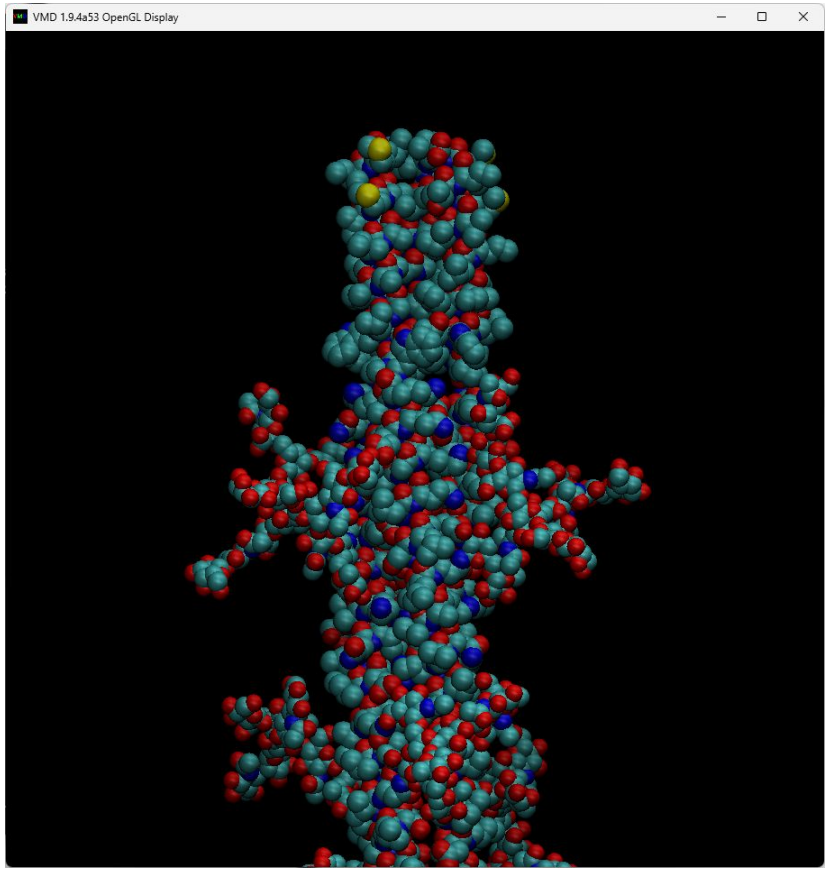
Keyboard shortcuts are on the right of each menu (e.g. T changes to translate mode)



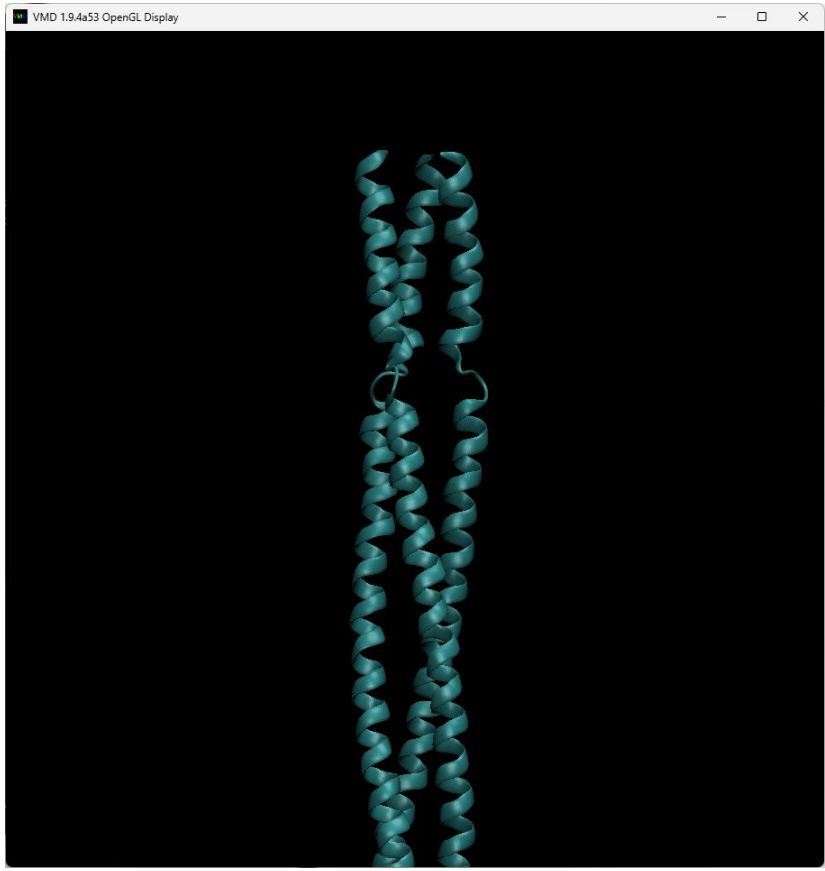
VMD – Representations



Licorice

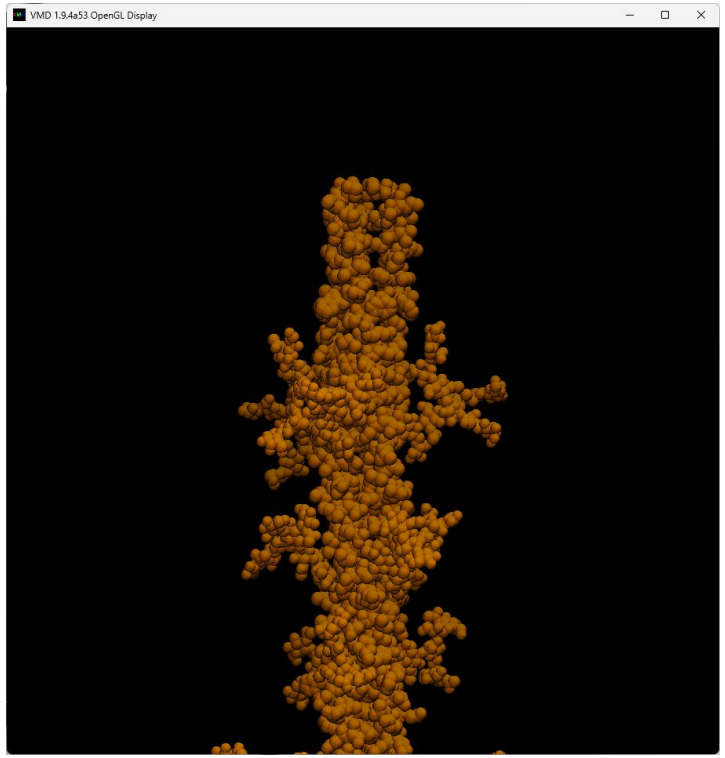


VdW

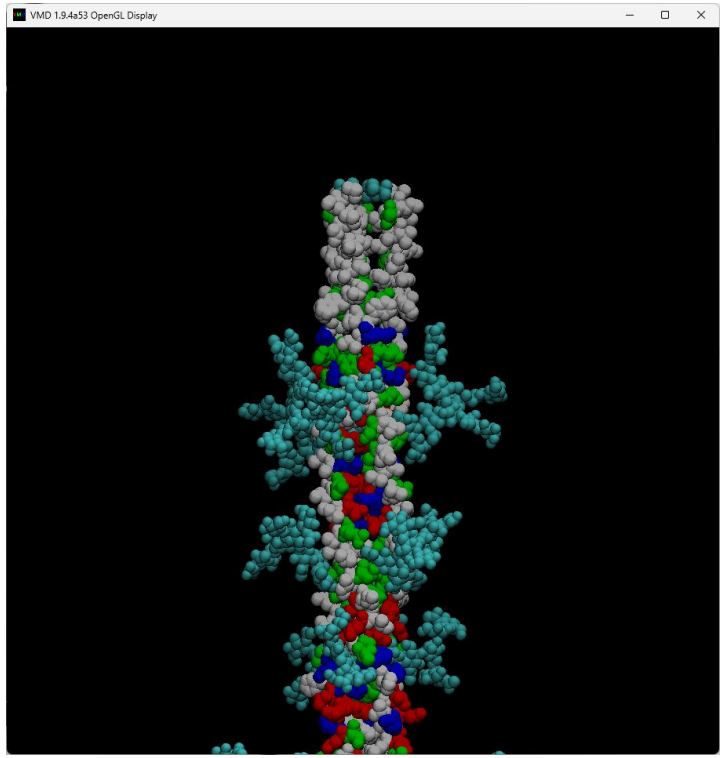


New cartoon

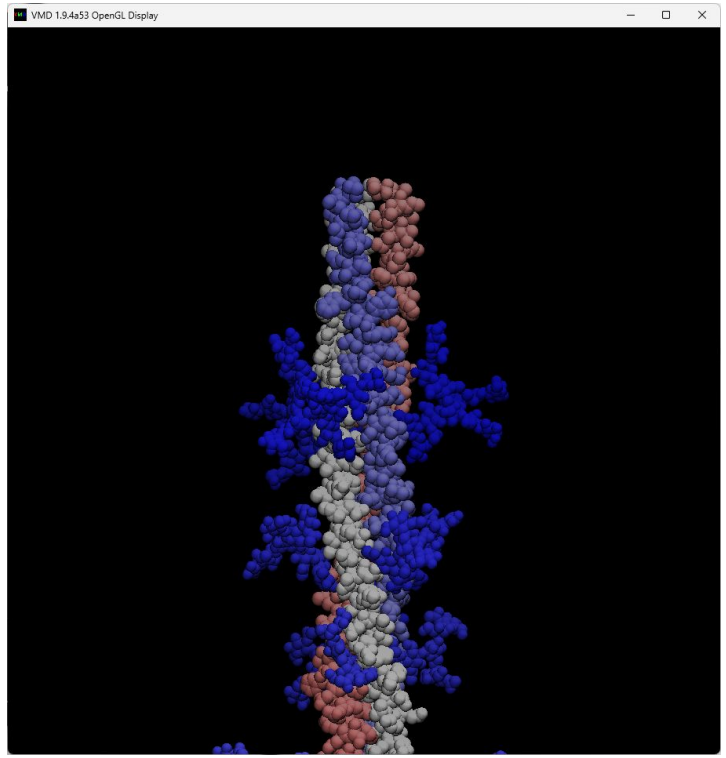
VMD – Colors



ColorID

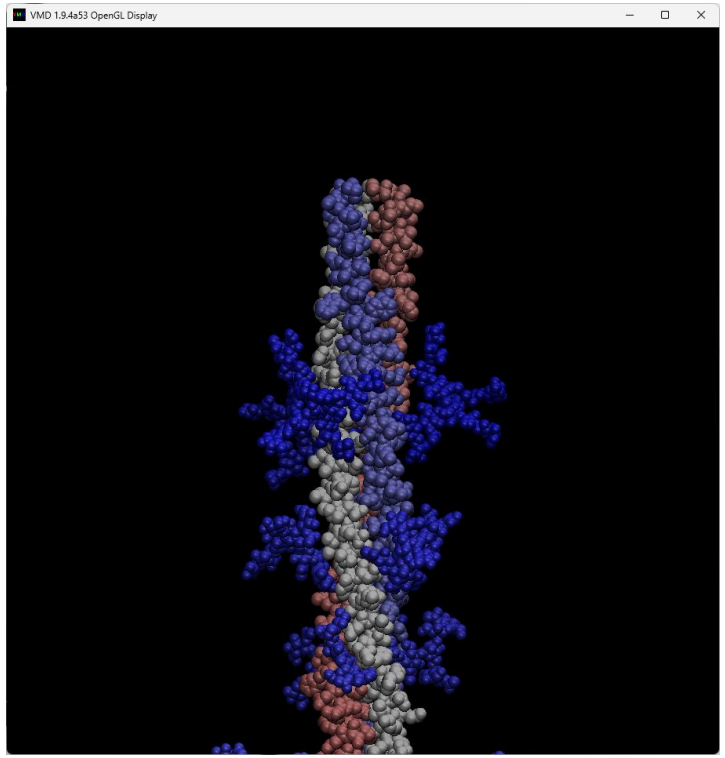


ResType

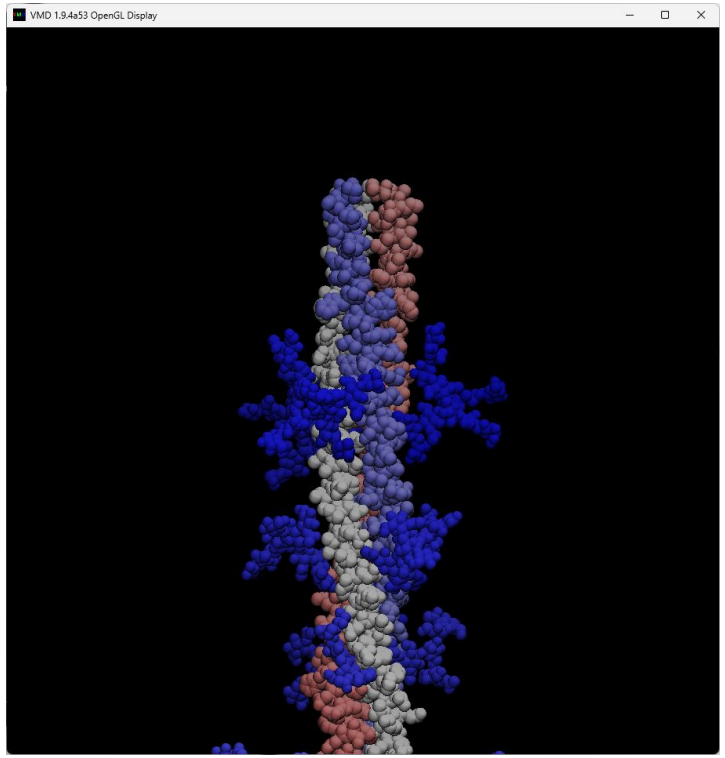


Index

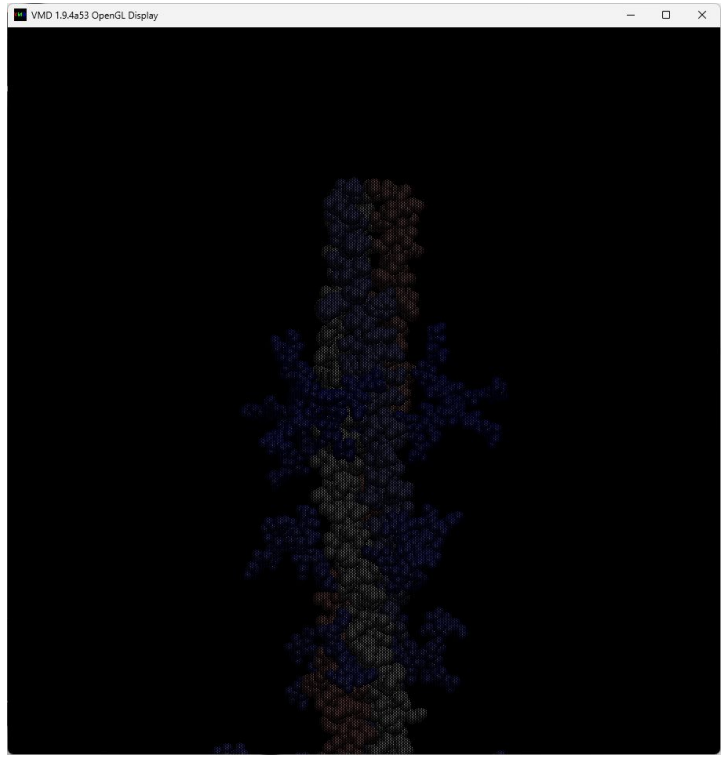
VMD – Material



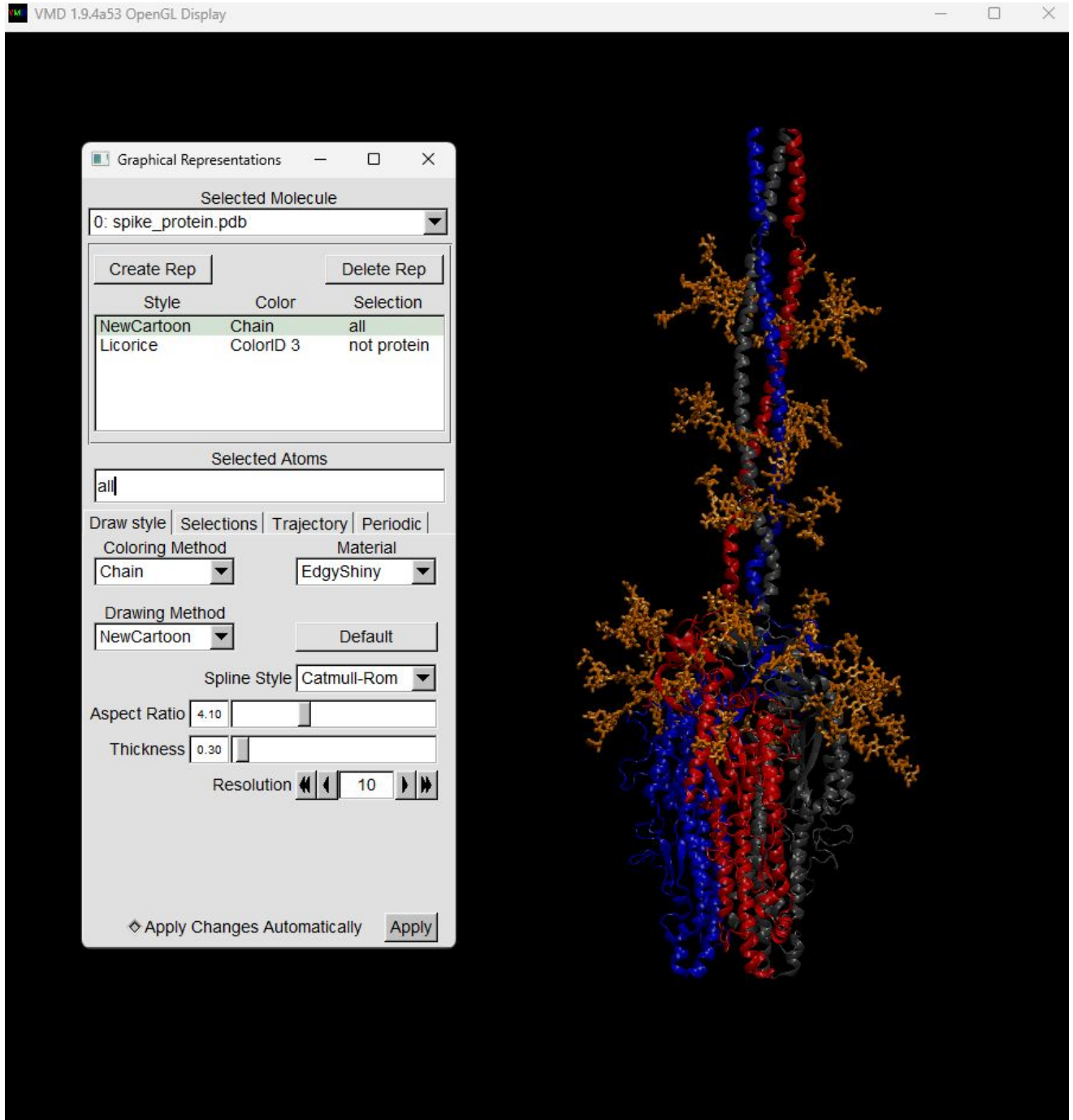
Opaque



AOChalky



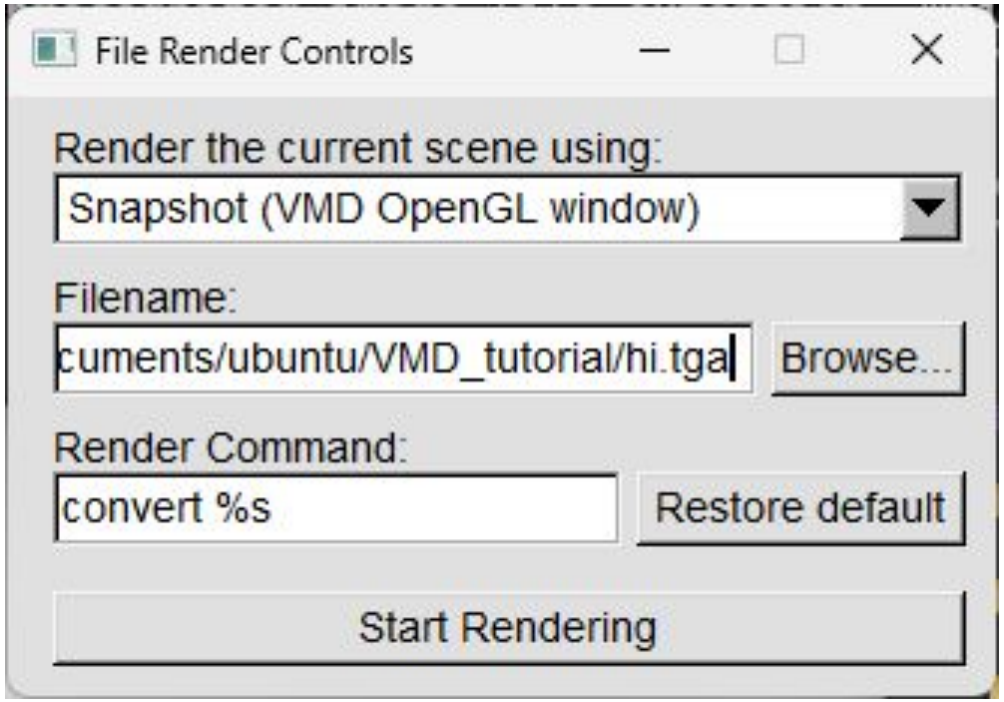
Transparent



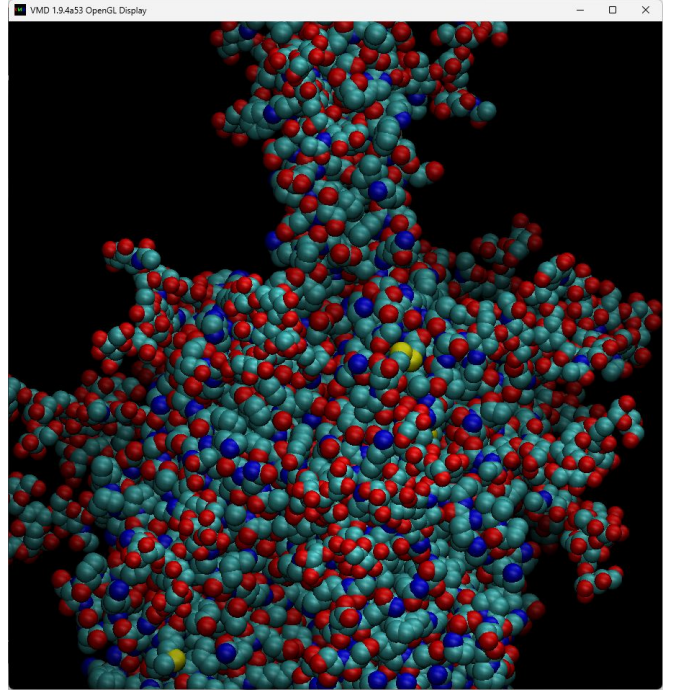
Representations again (but cooler)

Zoom in and render

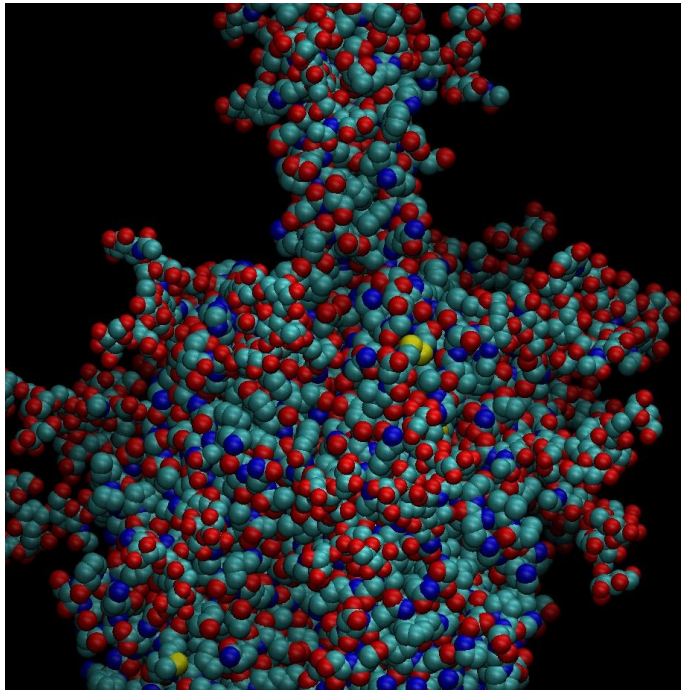
To render: File -> render



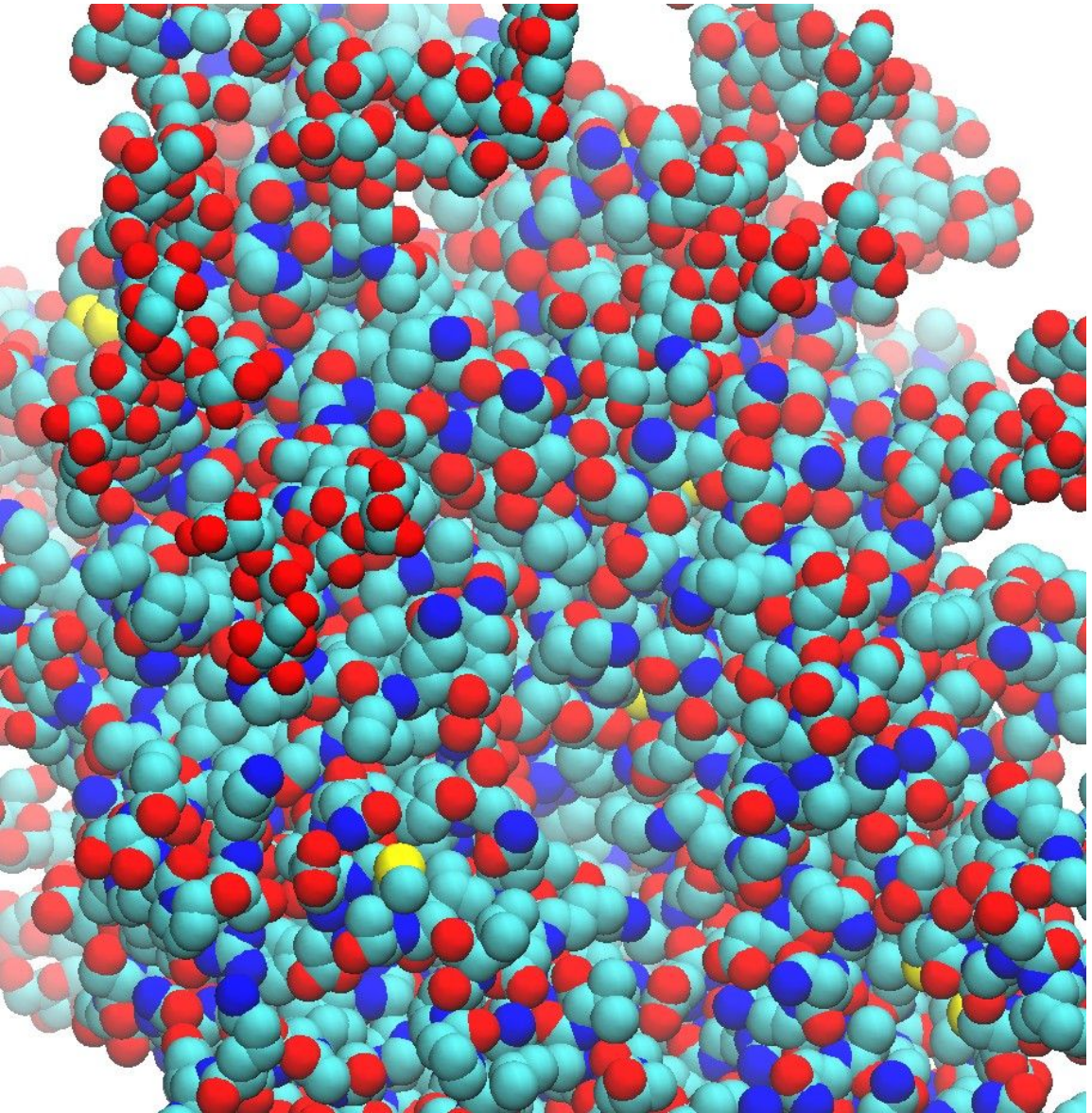
display



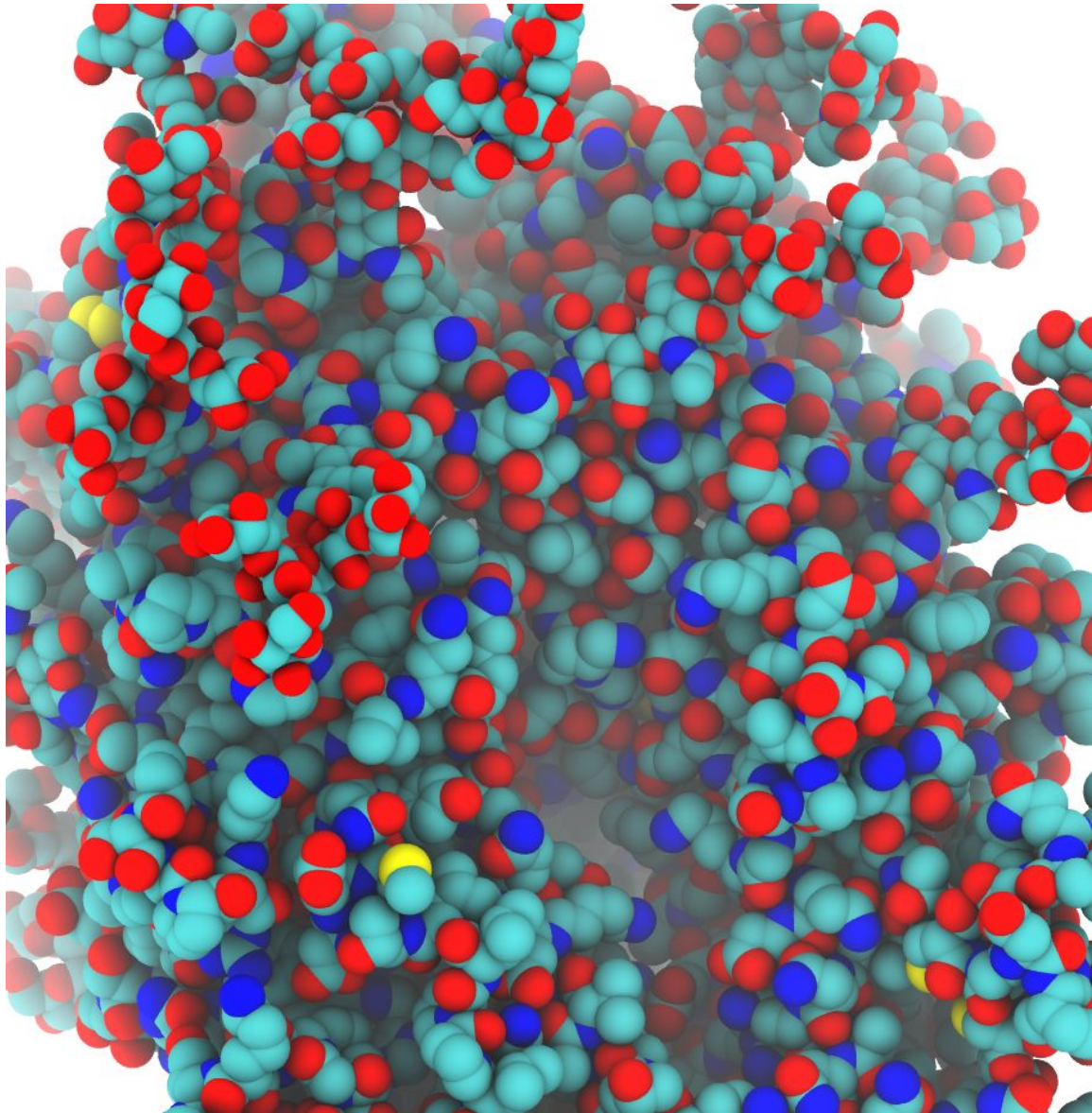
Snapshot –
What you see
is what you get



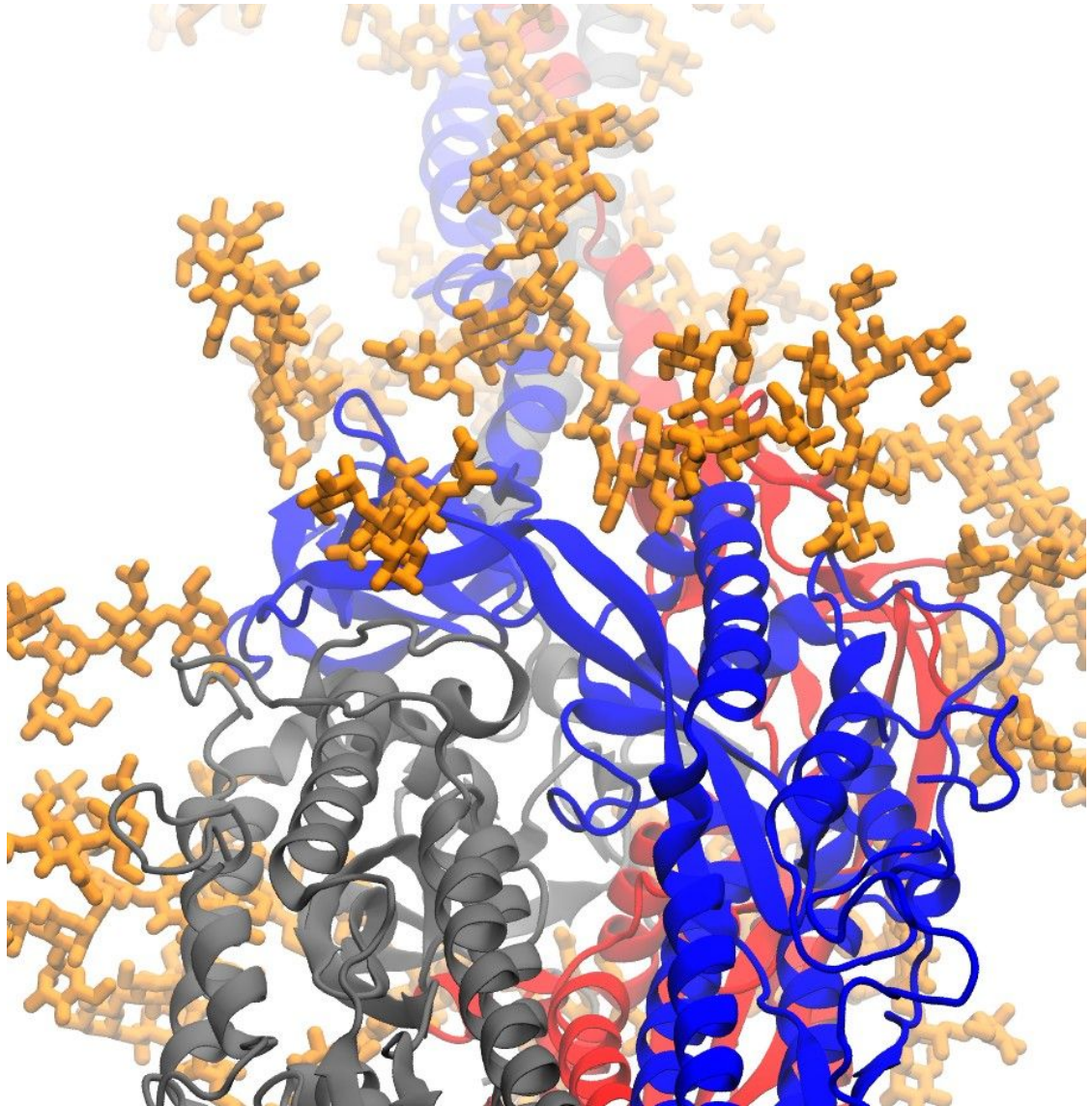
Built in rendering



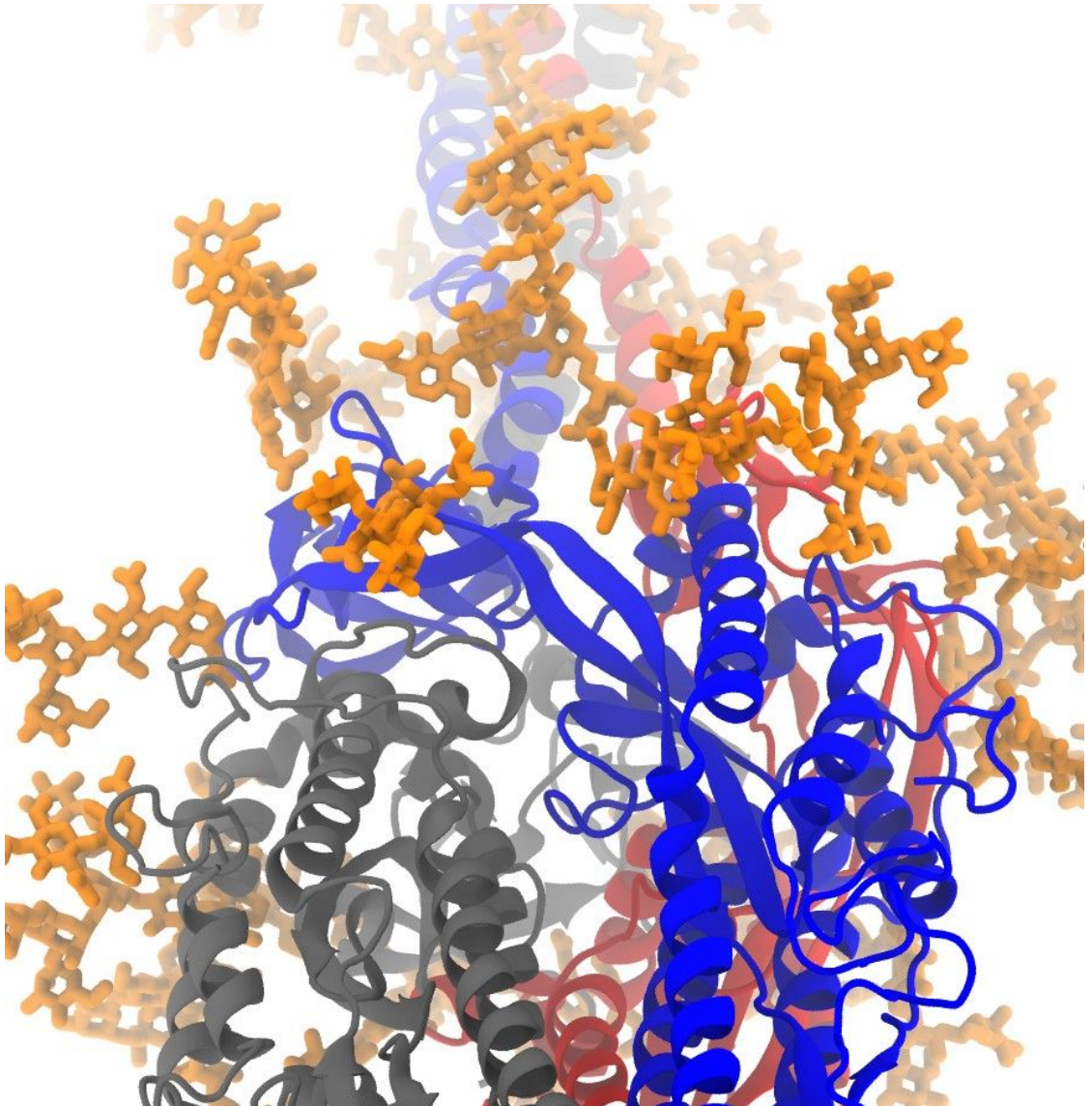
Tachyon



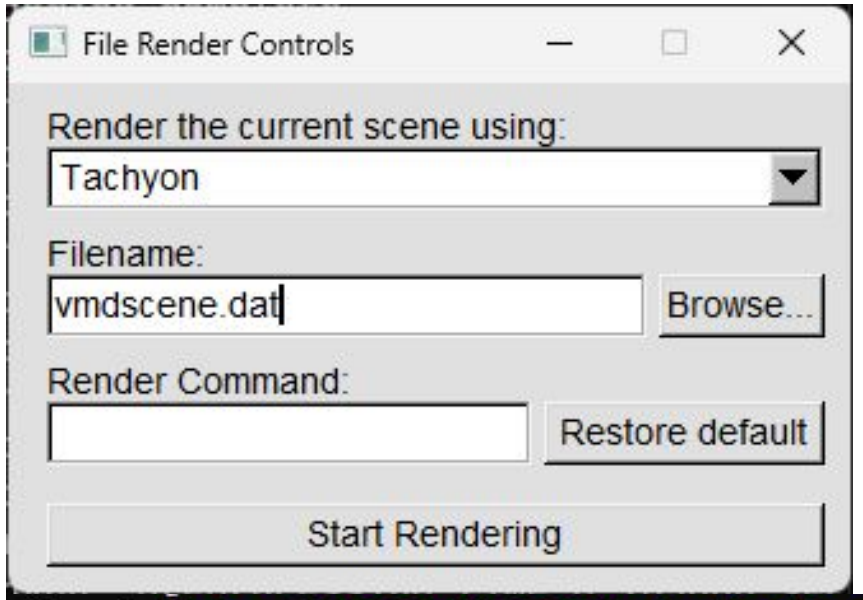
Built in rendering



Tachyon



Render with tachyon



tachyon -fullshade -auto_skylight 1.4 file.dat -o output.tga

```
(base) dodero@LAPTOP-VLIR34F9:/mnt/c/Users/doder/Documents/ubuntu/VMD_tutorial$ tachyon -fullshade
-auto_skylight 1.4 file.dat -o output.tga
Tachyon Parallel/Multiprocessor Ray Tracer   Version 0.99
Copyright 1994-2013,    John E. Stone <john.stone@gmail.com>
-------------------------------------------------------------
Scene Parsing Time:     0.9152 seconds
Scene contains 389373 objects.
Preprocessing Time:     0.2244 seconds
Rendering Progress:       100% complete
  Ray Tracing Time:   111.5722 seconds
    Image I/O Time:     0.0366 seconds
```

Let’s use the .dat file to render

- Directly call Tachyon

Use the command:

tachyon -fullshade -auto_skylight 1.4 vmdscene.dat -o vmdscene.AO.tga

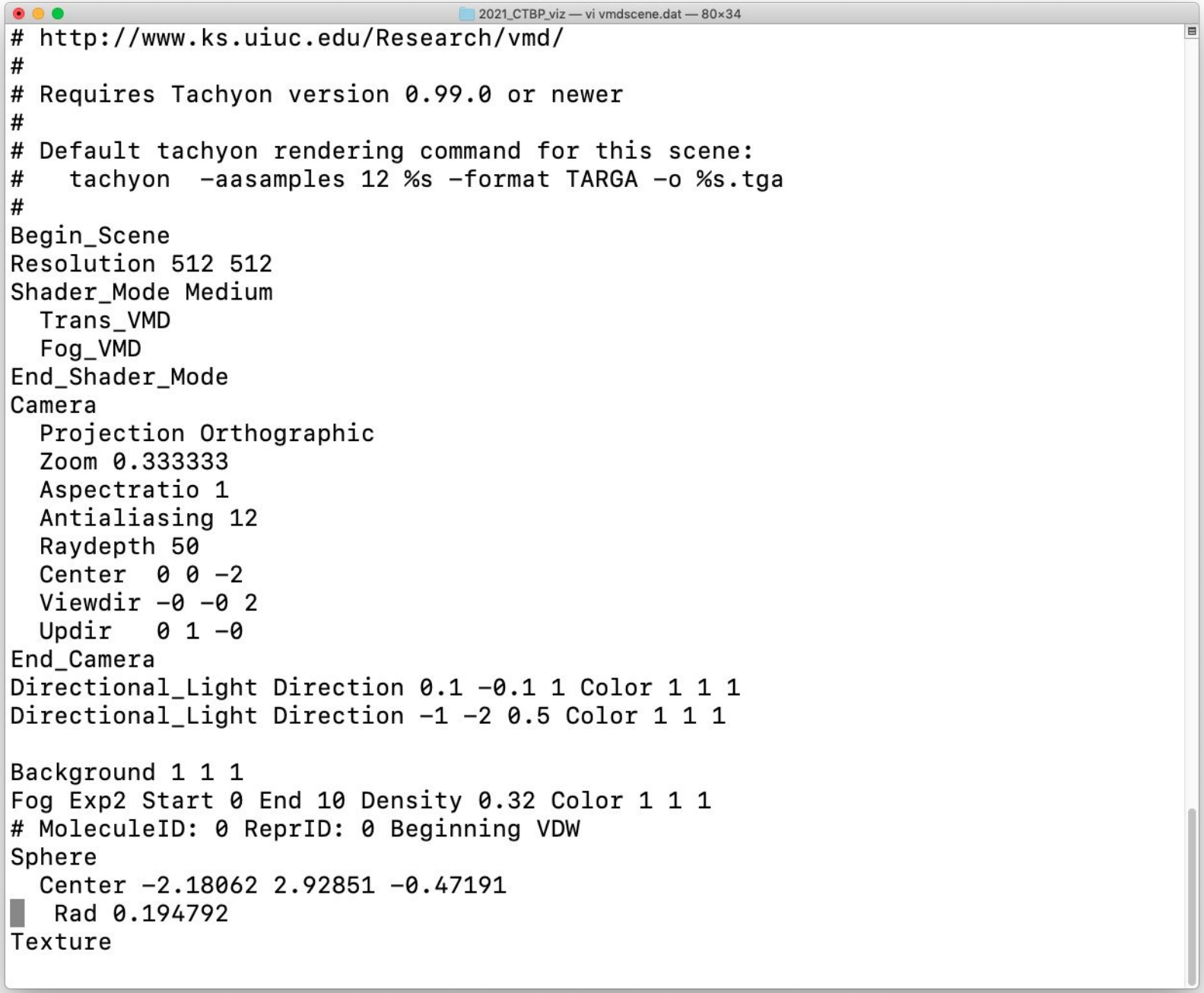
You may notice rendering is very slow. This is a good sign.

If too slow, add “–res 200 200” to your tachyon call, so make a smaller version.

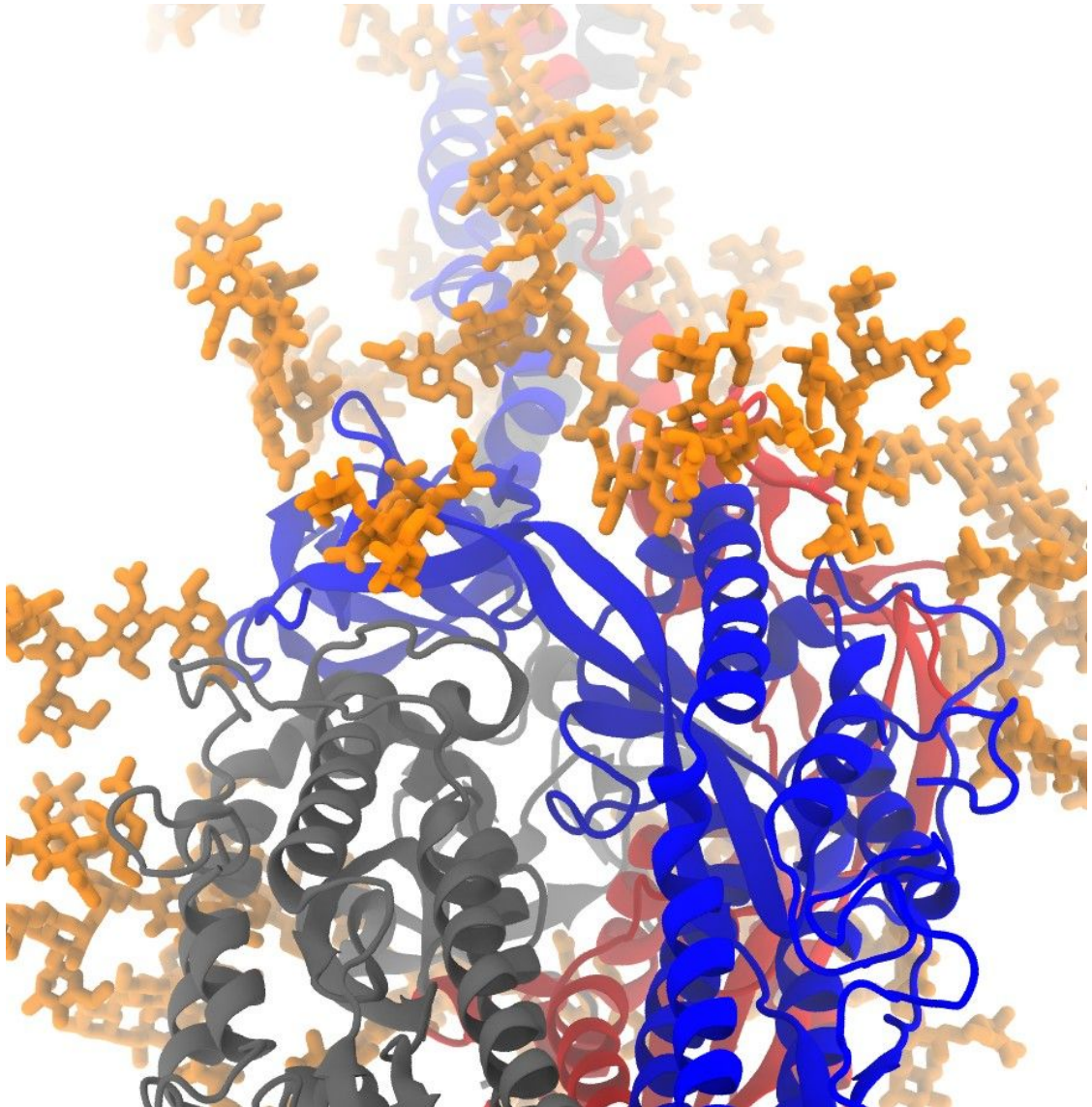
Open the .dat file

- Defines the scene
  - Perspective
  - Objects
  - Colors
  - Shading
  - Fading (fog)
  - Material
  - Everything…

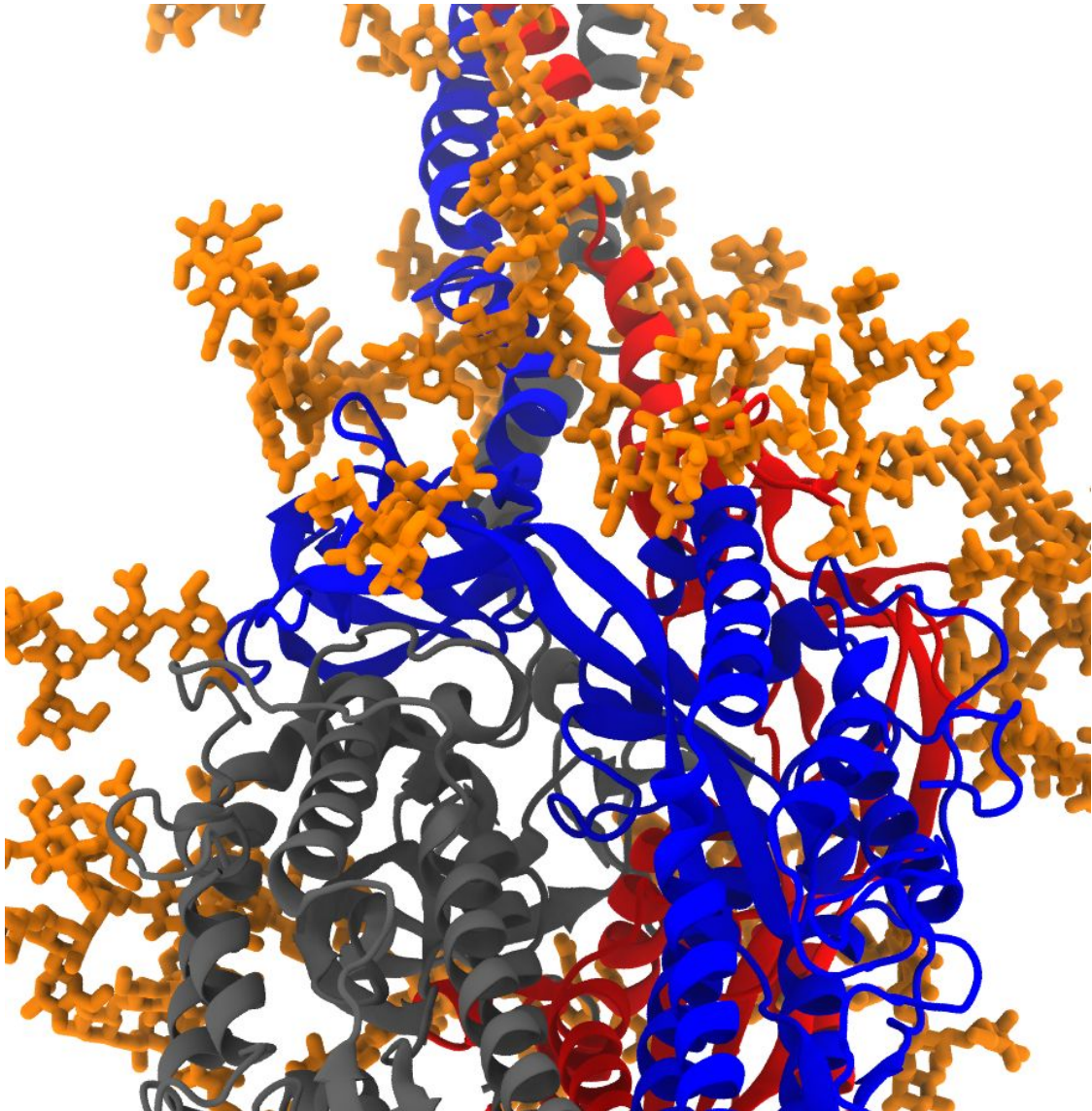
- At this point, we no longer necessarily need VMD.



Foggy

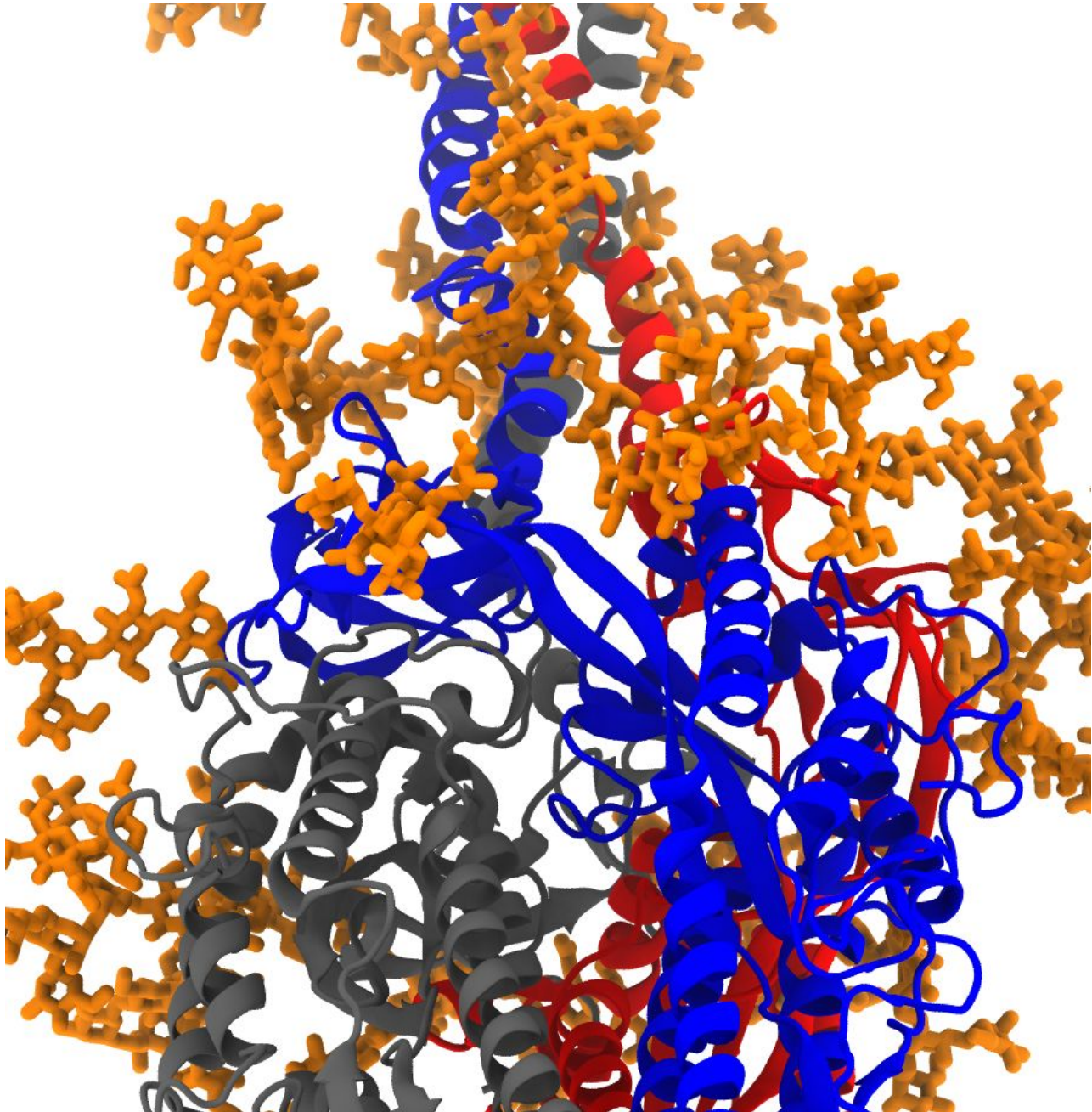


Less fog

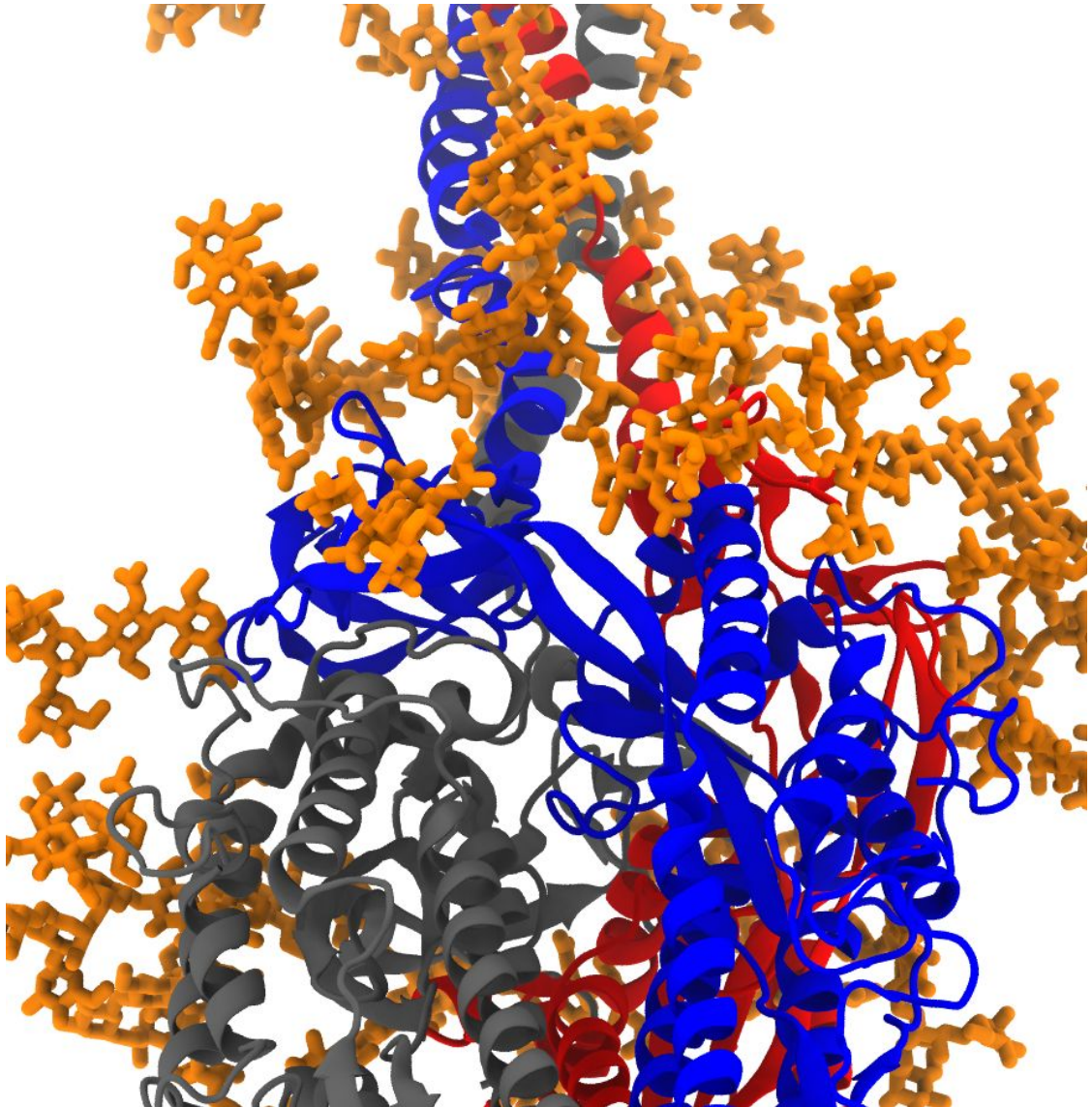


Light intensity and oversaturation

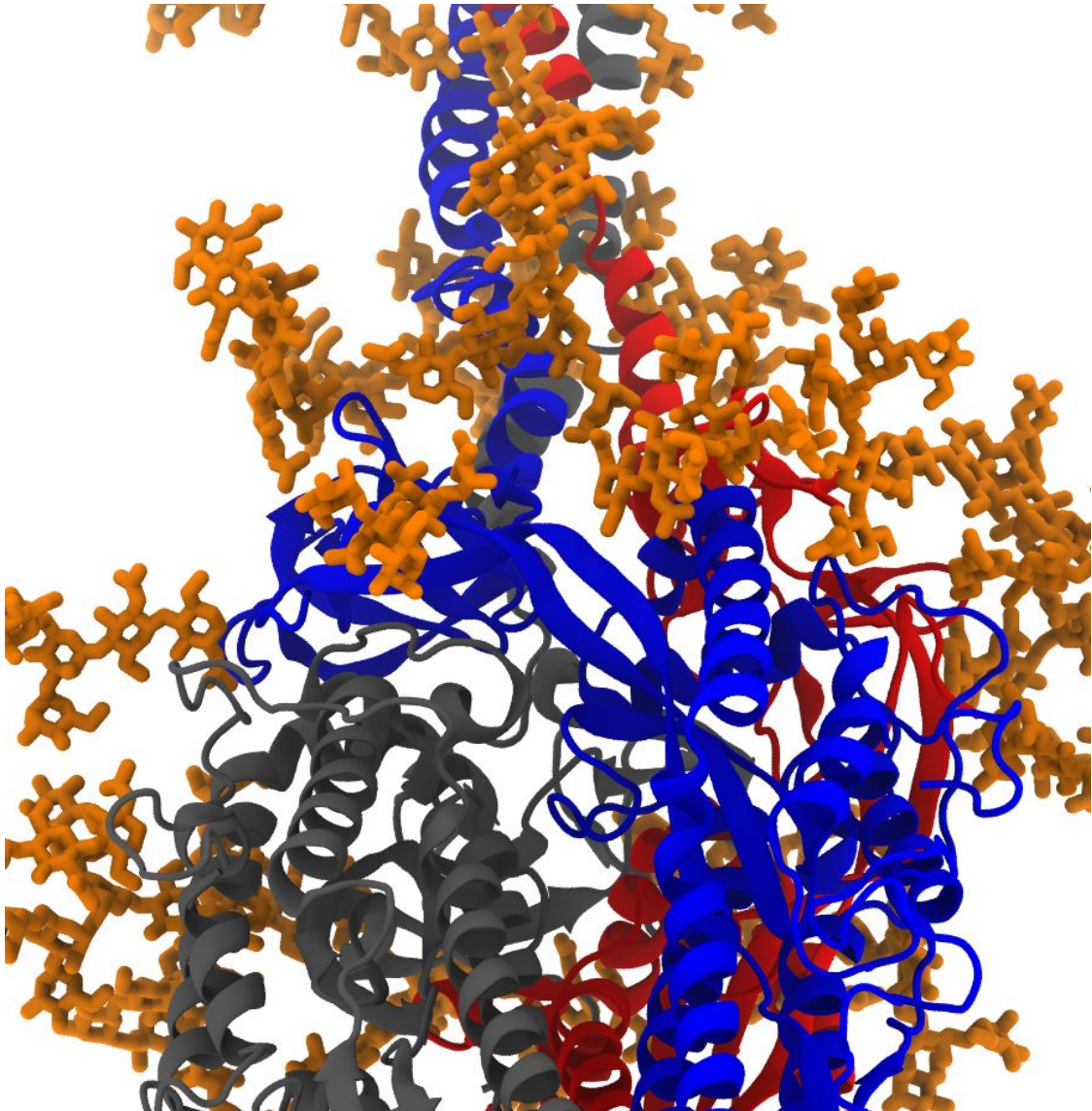
tachyon -fullshade -auto_skylight 1.2 vmdscene.dat -o vmdscene.AO.tga



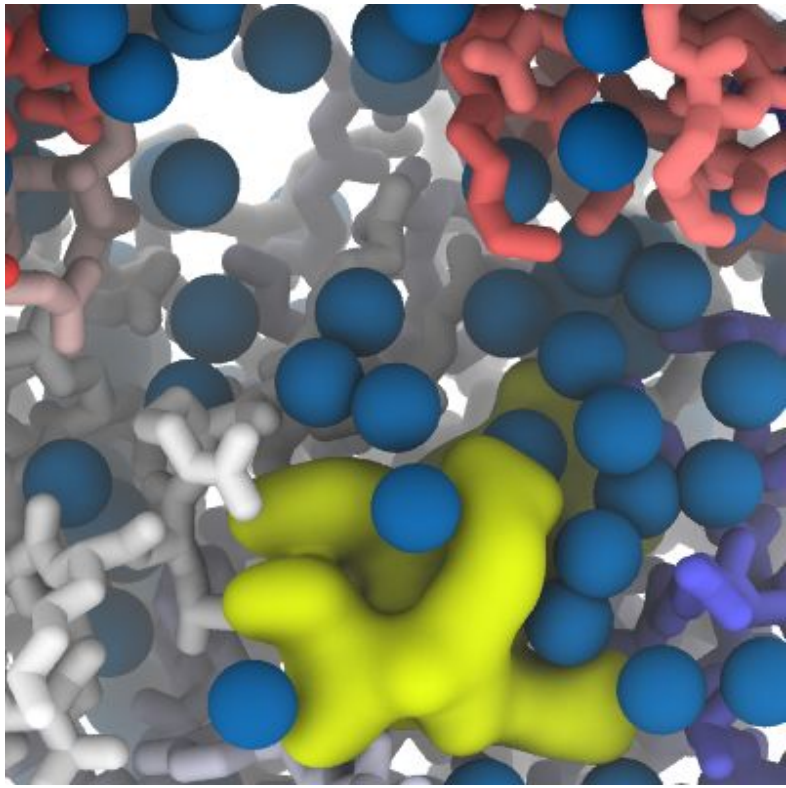
Saturated



Less saturated



General tips



- Only show atoms that are in the frame
- Even particles that are not in view will cast shadows
- Best representations
  - Vdw
  - Licorice (expensive to render, see right)
    - Use thicker bond radius than the default
    - May want higher sphere resolution the default
  - Tubes
  - Quicksurf
- AO often results in darker images than in the display. Plan accordingly
- Remember you can reassign colors in terms of RGB values in the scene file
- For a publication, expect to render, re-render and repeat…. You only get one chance to catch the reader's eye.
- Tachyon is parallel! Use an interactive cluster node with many cores. A good image can take 5-20 minutes on a multi-core node.

Tk console

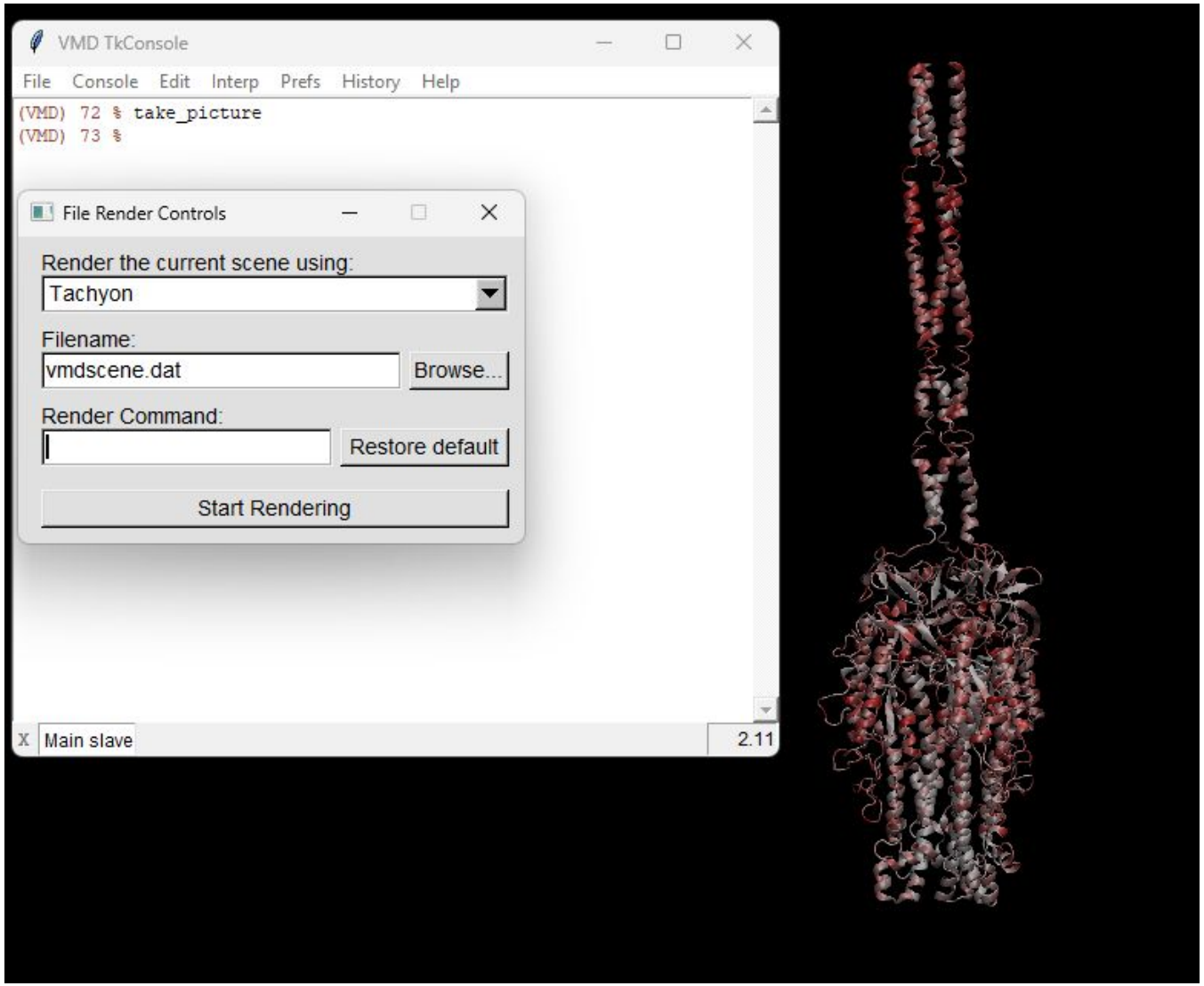
Let’s play with scripting

Open the tk console and source visualization.tcl (bag of goodies)

take_picture
colorByFile
colorByResid
tricolor_scale
bicolor_scale
align_pi
rotate_axis
increment_np
increment

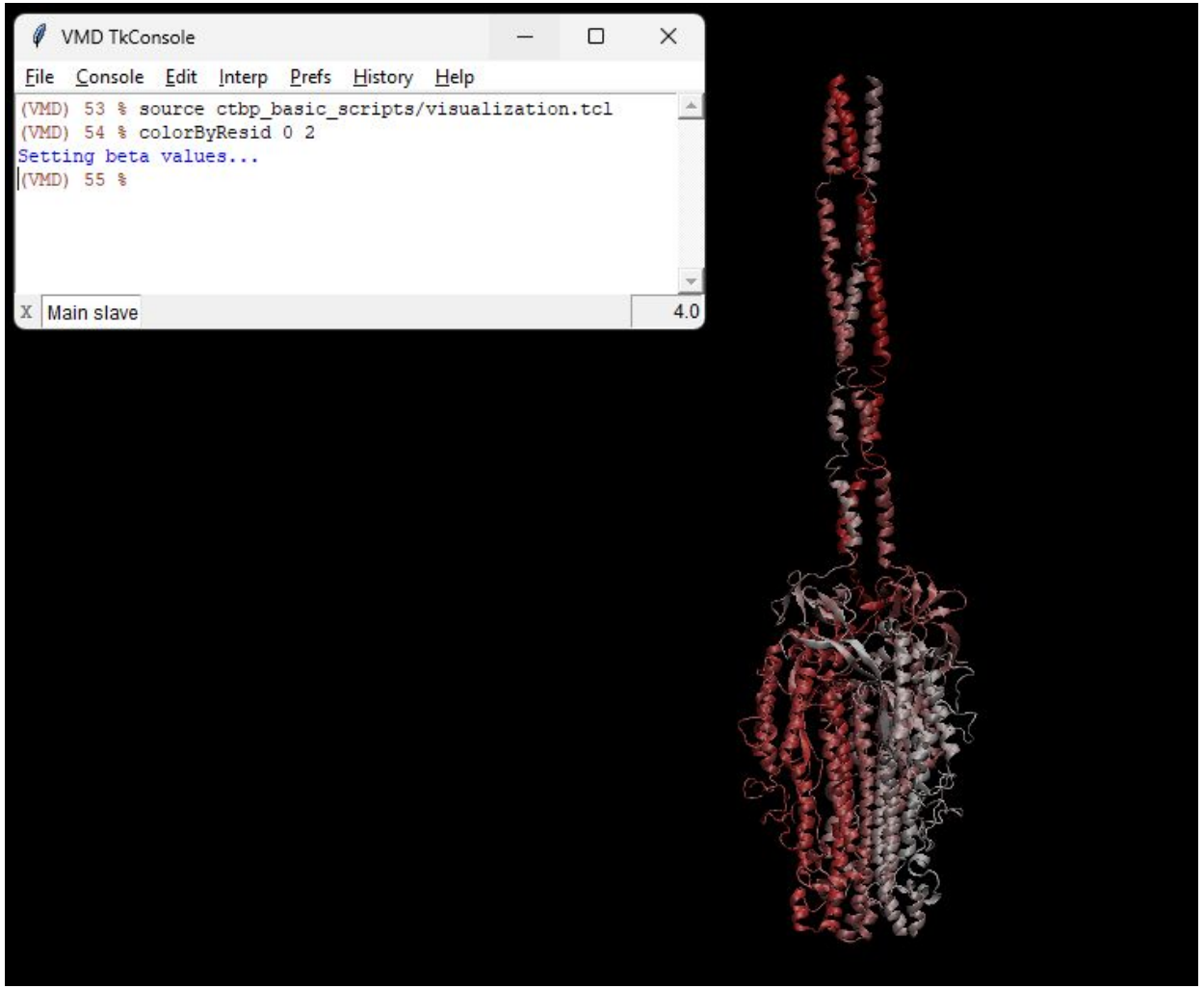
save_vp
move_vp
move_vp_render
retrieve_vp
write_vps

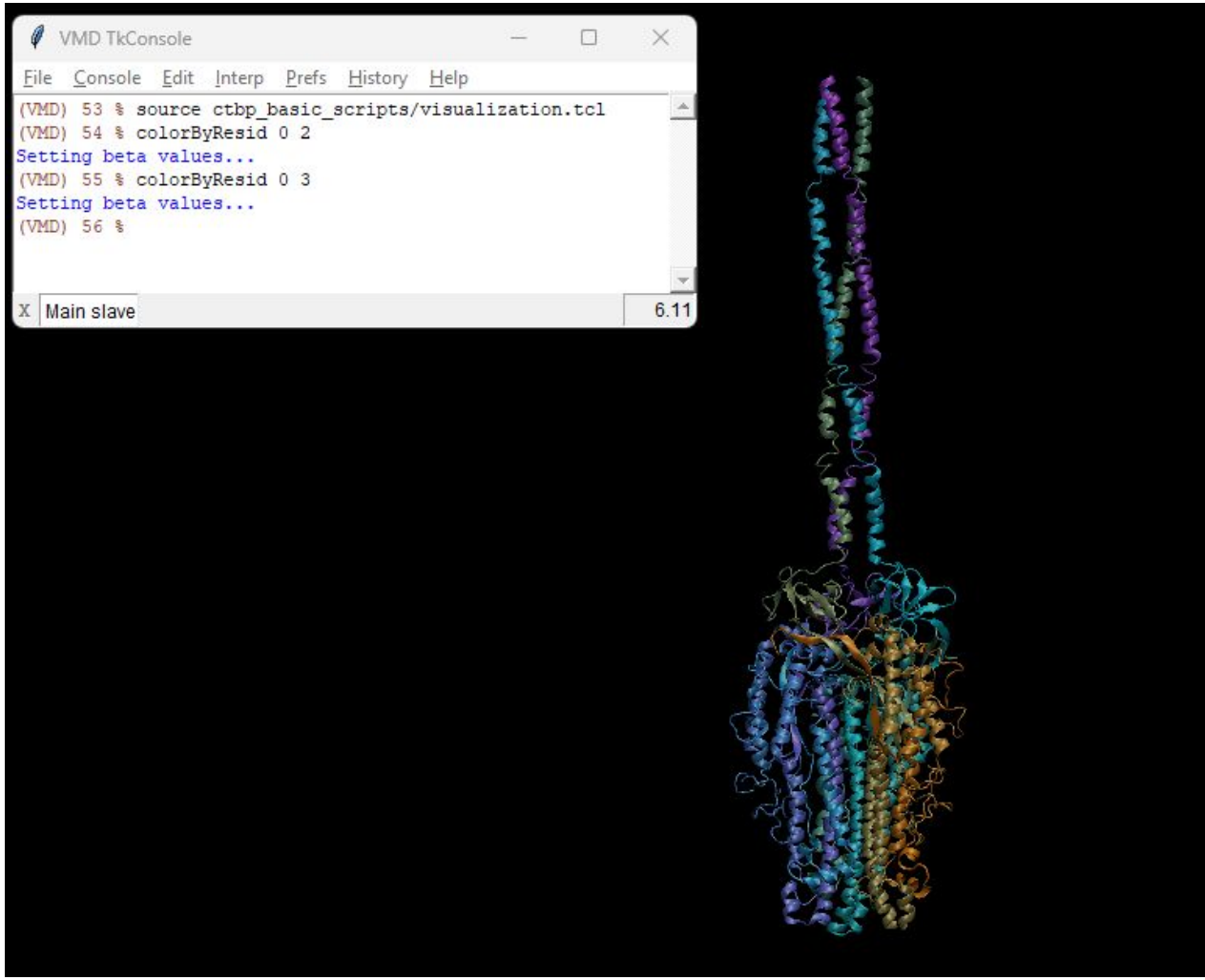
Tk console - take_picture



Renders display into .dat file

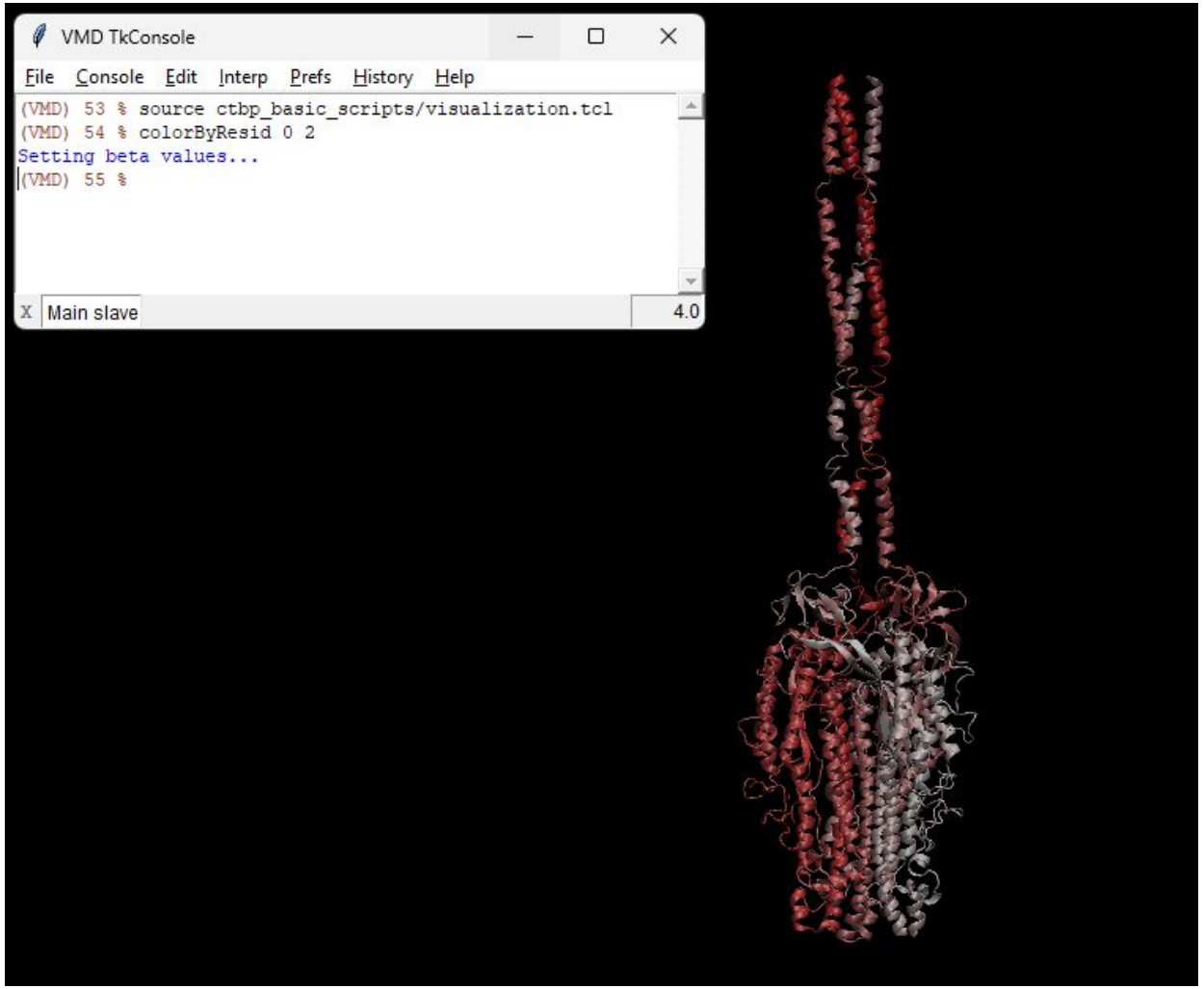
Tk console - colorByResid



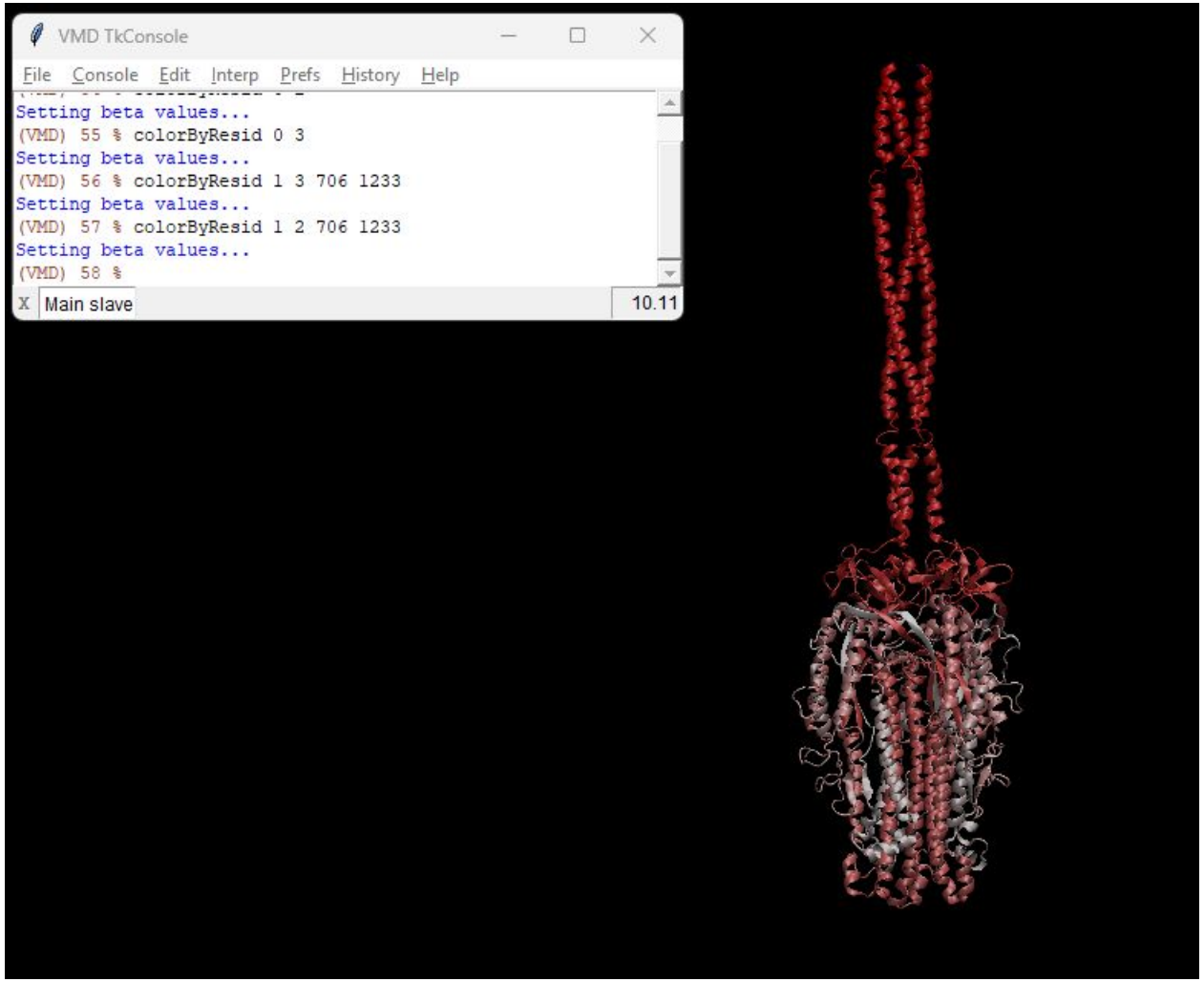


Tk console - colorByResid

GRO

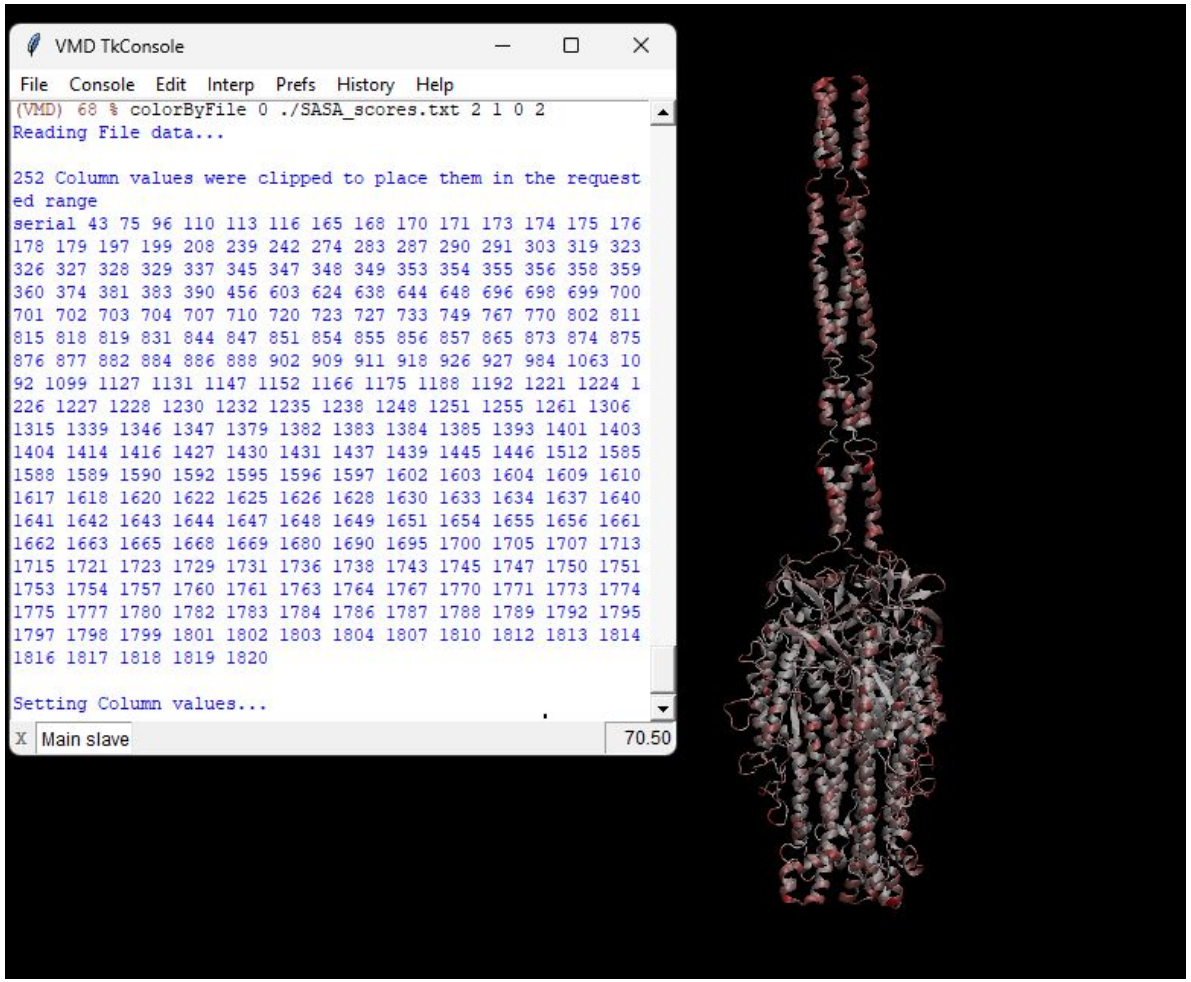


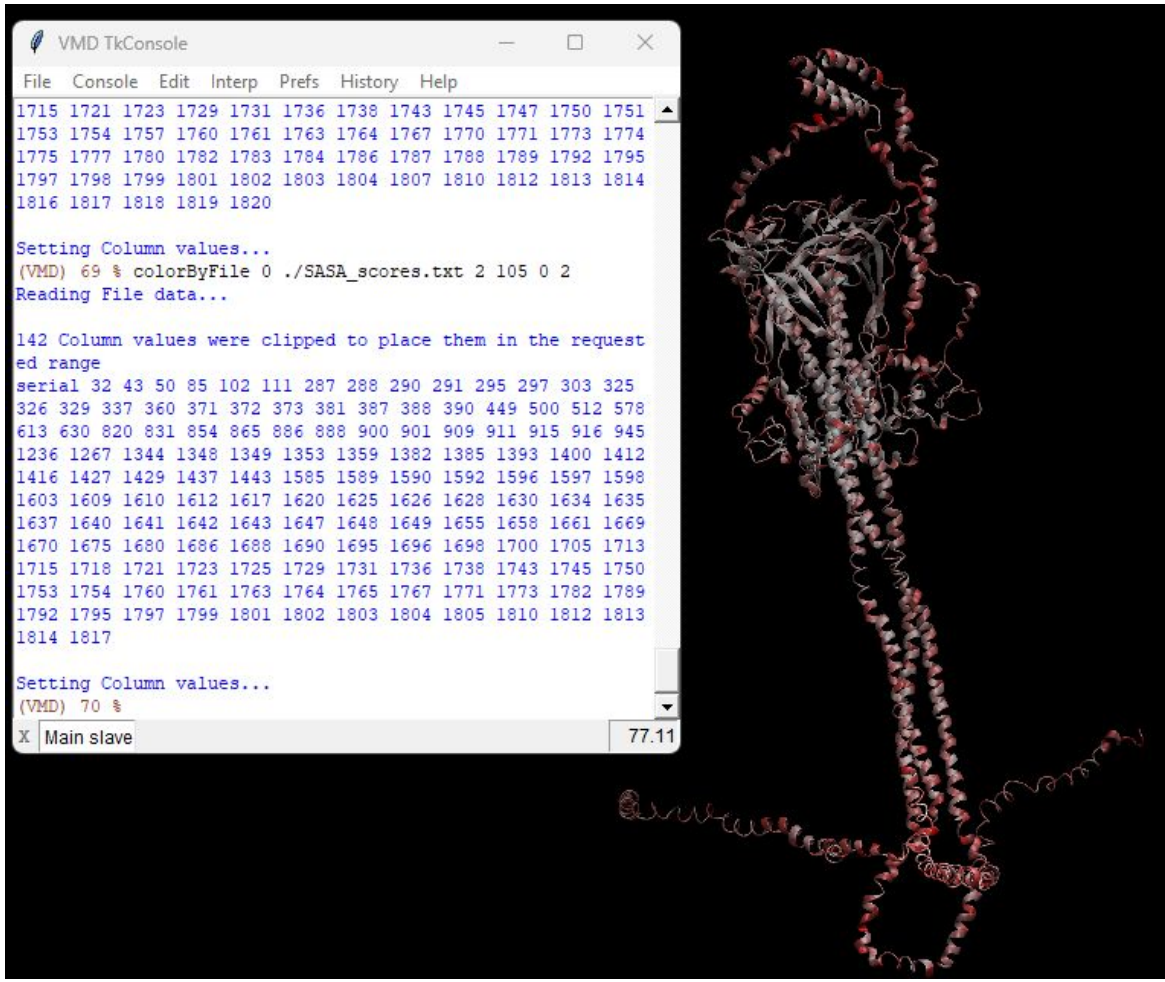
PDB



Tk console - colorByFile

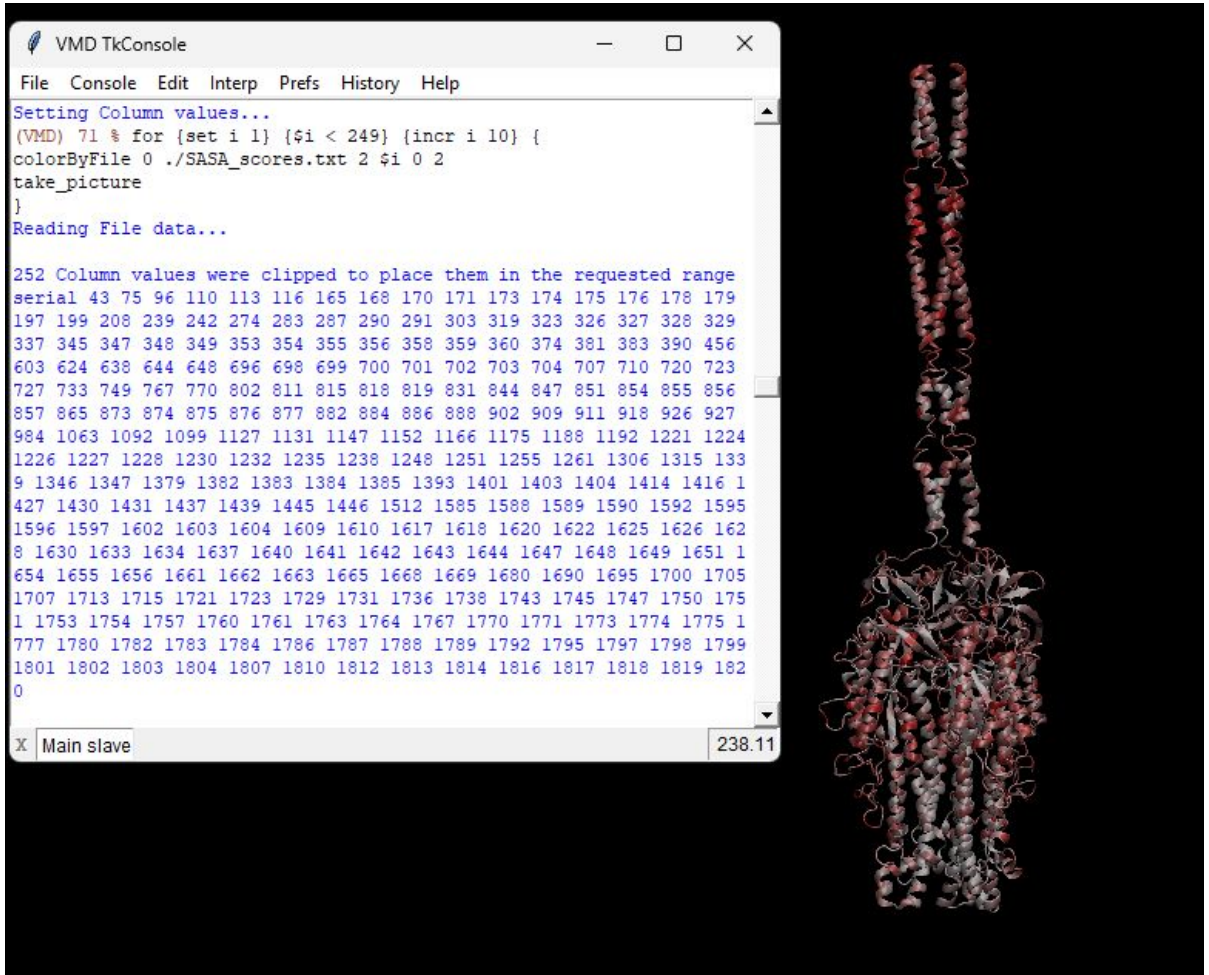
SASA: Solvent Area Surface Area

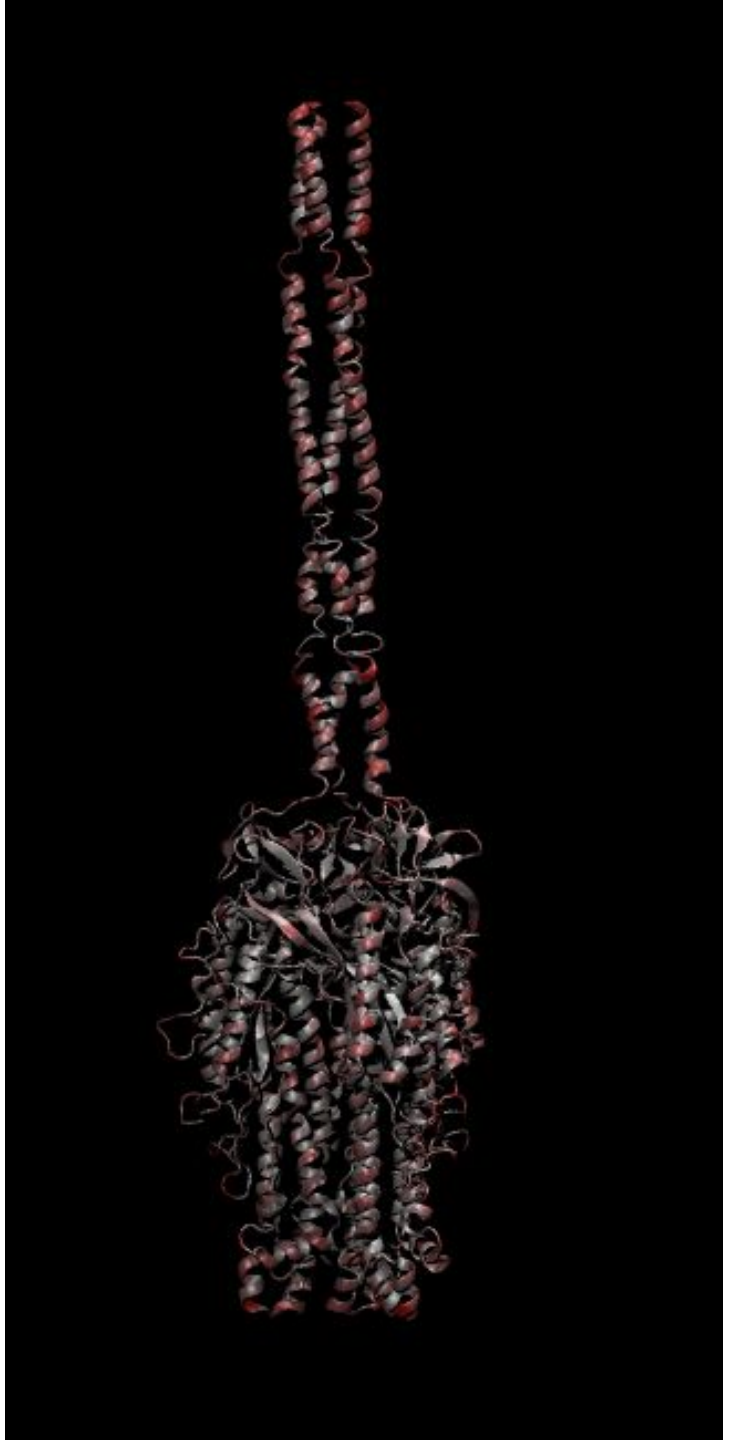




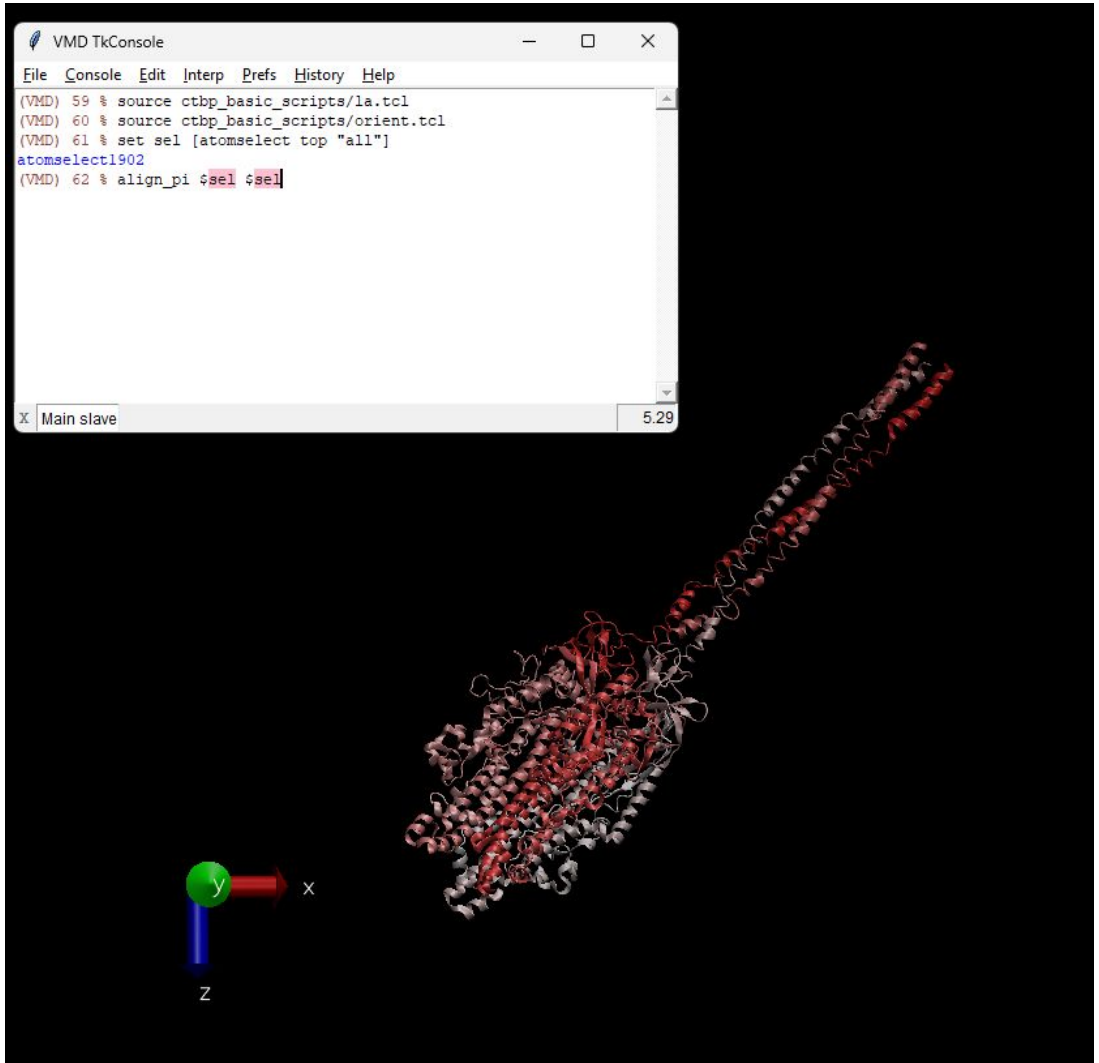
Tk console - colorByFile

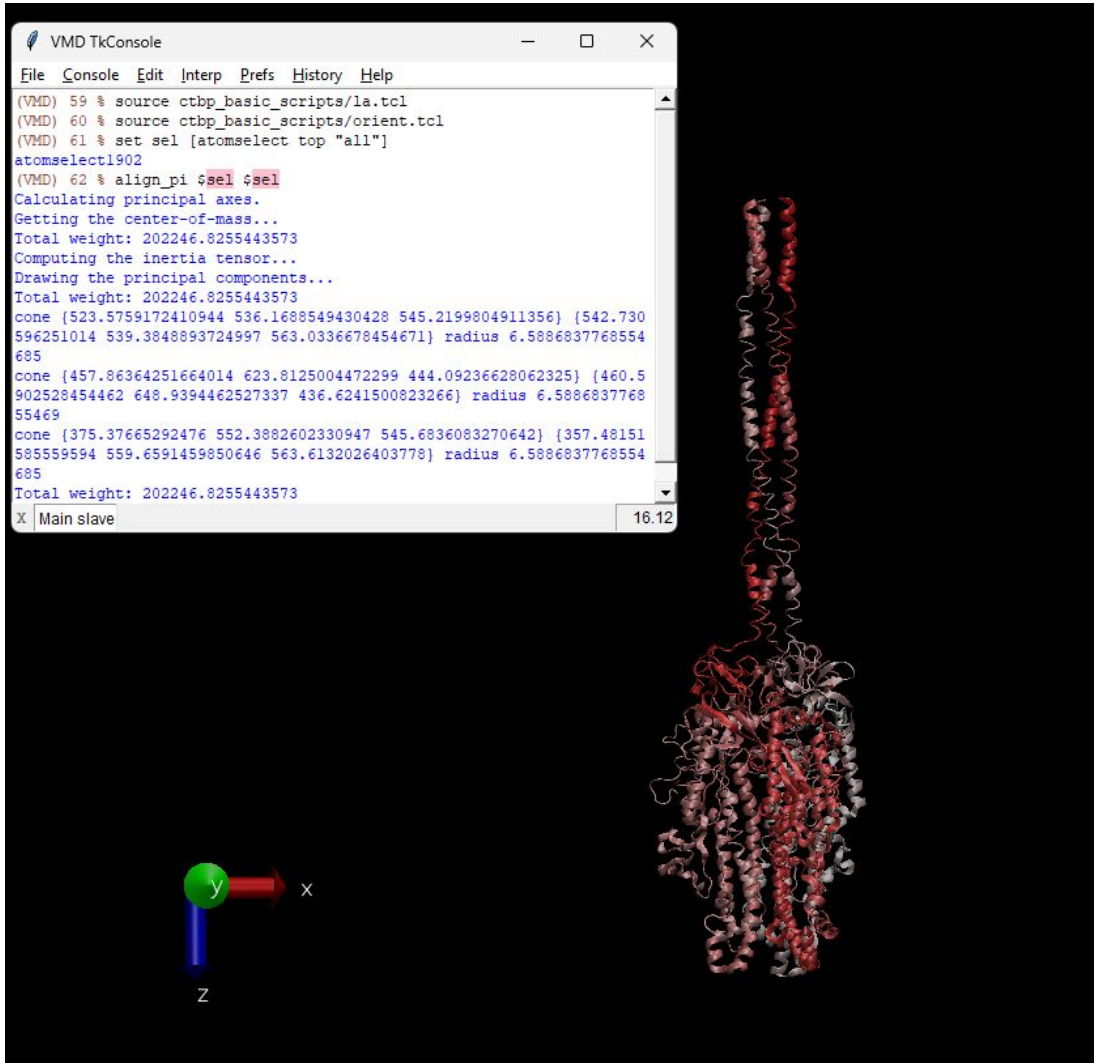
SASA: Solvent Area Surface Area



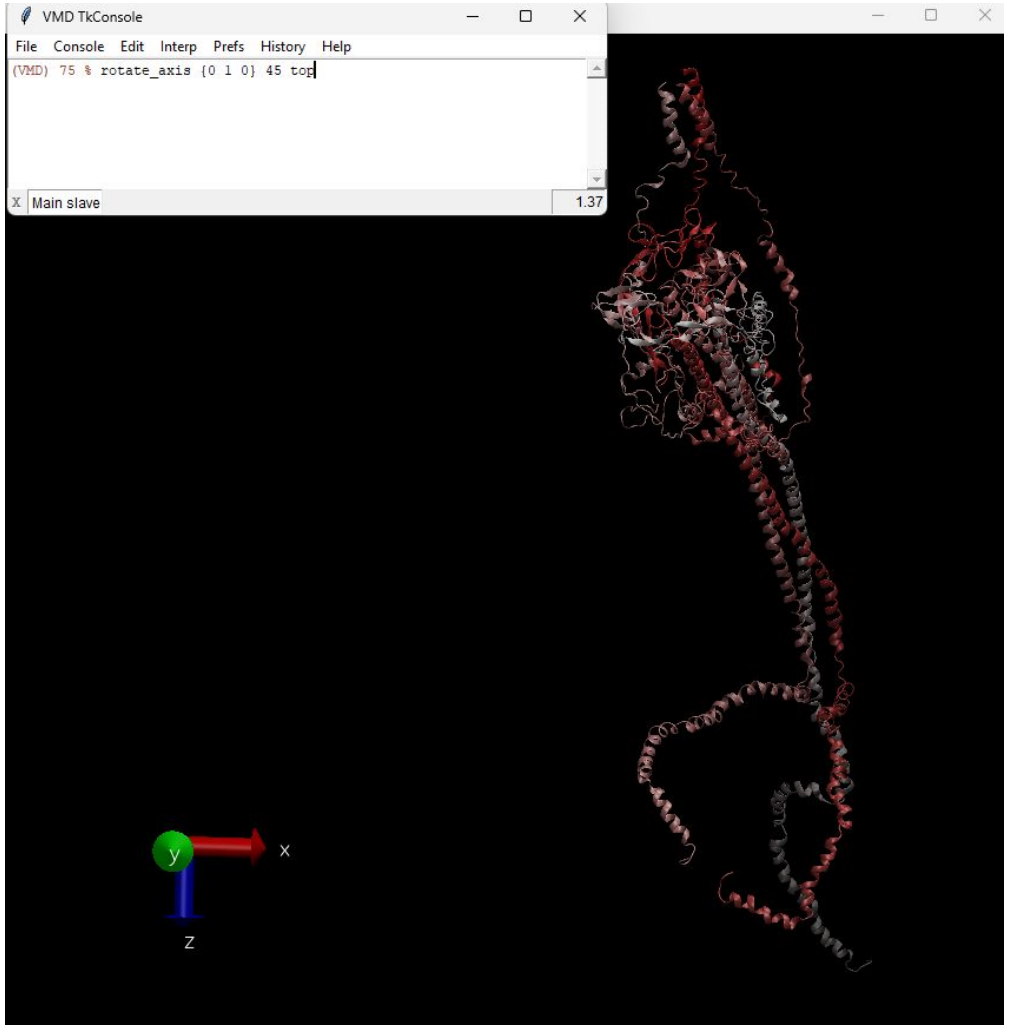


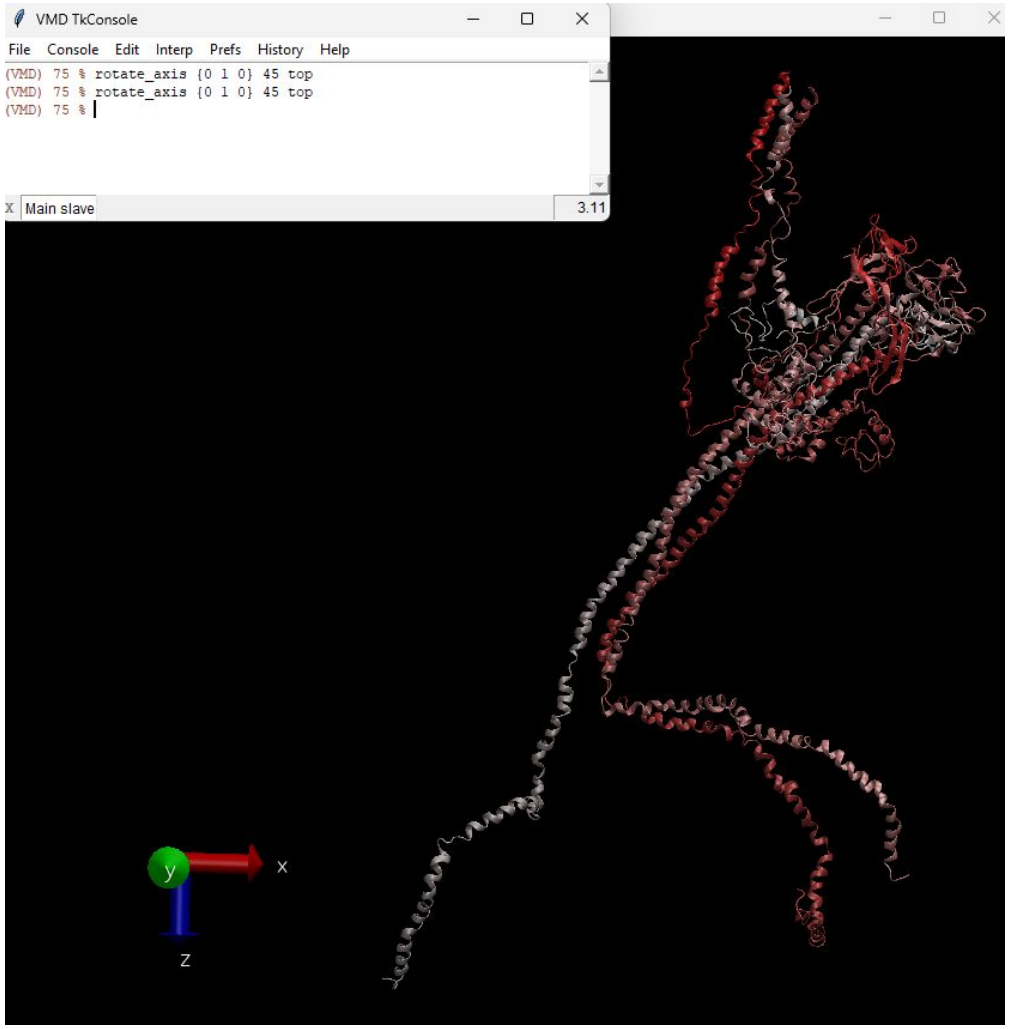
Tk console - align_pi





Tk console - rotate_axis

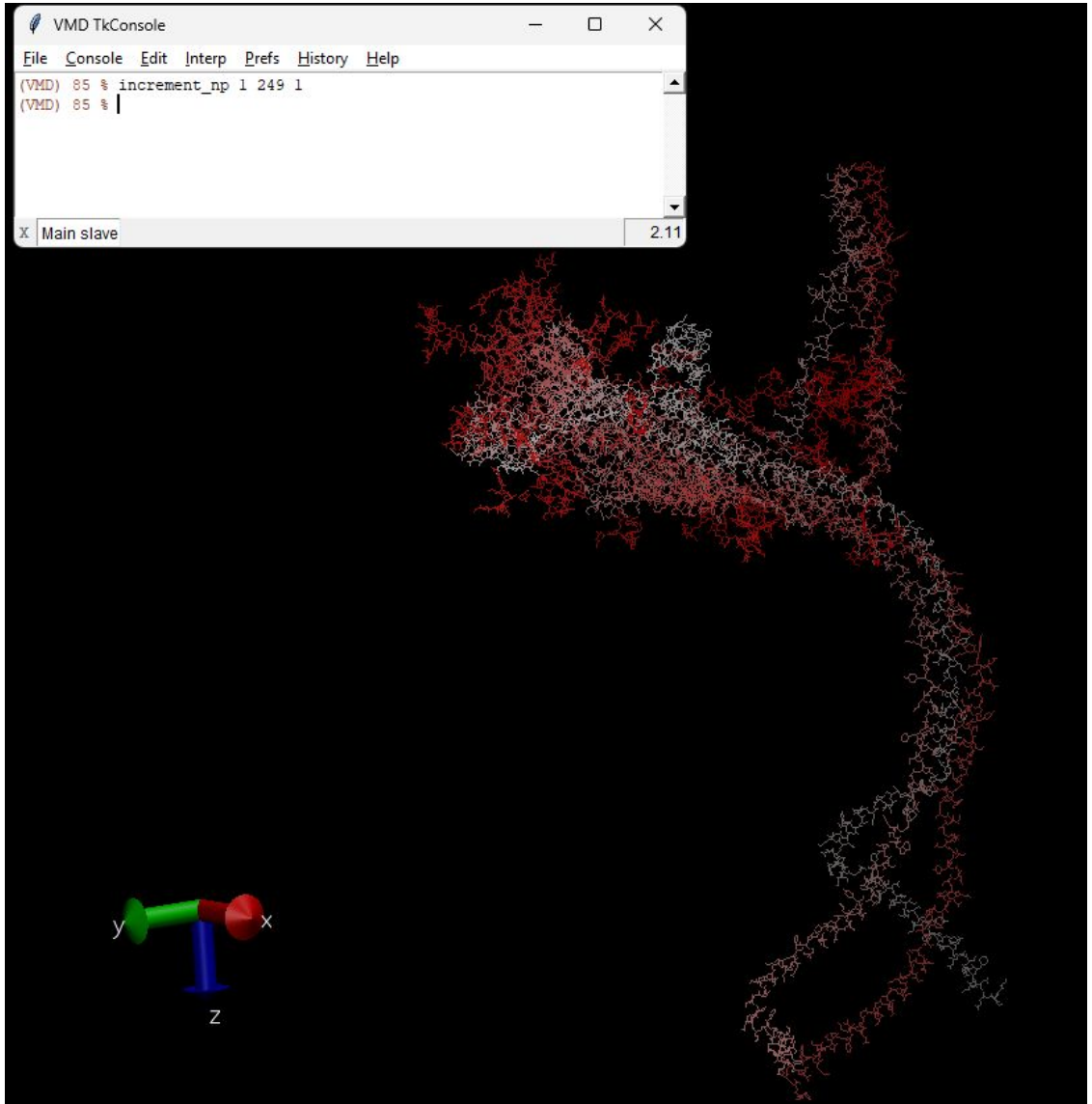




Tk console - rotate_axis

```
for {set i 1} {$i < 249} {incr i 10} {
rotate_axis {0 1 0} 10 top
display update
}
```

Tk console - increment(_np)



increment_np 1 249 1

To render
increment 1 249 1

Tk console - Viewpoints

save_vp vpID

retrieve_vp vpID

move_vp vpID_1 vpID_2

move_vp_render vpID_1 vpID_2 first_frameID dirName filePrefixName morph_frames=50

write_vps

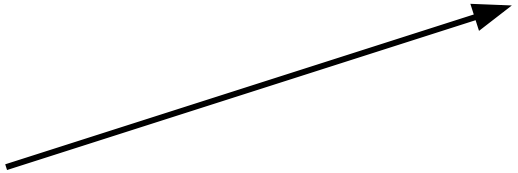
Tk console - Merging everything together

```
source ctbp_basic_scripts/la.tcl
source ctbp_basic_scripts/orient.tcl
source ctbp_basic_scripts/visualization.tcl
source vps.tcl

retrieve_vp 1
animate goto 0
#mol representation NewCartoon 0.3 10 4.1
colorByFile 0 ./SASA_scores.txt 2 1 0 2

move_vp_render 1 2 0 . move 20
move_vp 1 2 0
############################
#RENDER WITH TACHYON LATER
############################

for {set i 1} {$i < 360} {incr i 2} {
rotate_axis {0 1 0} 2 top
display update
#take_picture
}
############################
#RENDER WITH TACHYON LATER
############################
```

```
move_vp_render 2 1 0 . move 20

increment_np 1 249 1
```

Tk console - Merging everything together

```
mol new ./spike_protein.gro
mol addfile ./sample_trajectory.xtc 0

source ctbp_basic_scripts/la.tcl
source ctbp_basic_scripts/orient.tcl
source ctbp_basic_scripts/visualization.tcl
source vps.tcl

retrieve_vp 1

mol representation NewCartoon 0.3 10 4.1
mol material AOChalky
color Display Background white
colorByFile 0 ./SASA_scores.txt 2 1 0 2
```

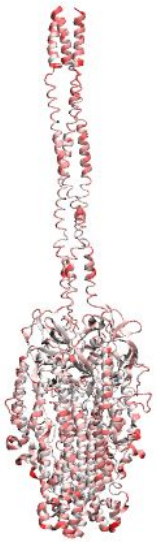


```
animate goto 1
move_vp_render 1 2 0 . move 20

for {set i 1} {$i < 360} {incr i 10} {
rotate_axis {0 1 0} 10 top
take_picture
}

move_vp_render 2 1 20 . move 20
increment 1 249 5

exit
```

VMD without looking - Command line

vmd -e sample_script.tcl -dispdev text

vmd -e sample_script_2.tcl -dispdev text



VMD without looking - Command line

vmd -e sample_script.tcl -dispdev text

vmd -e sample_script_2.tcl -dispdev text