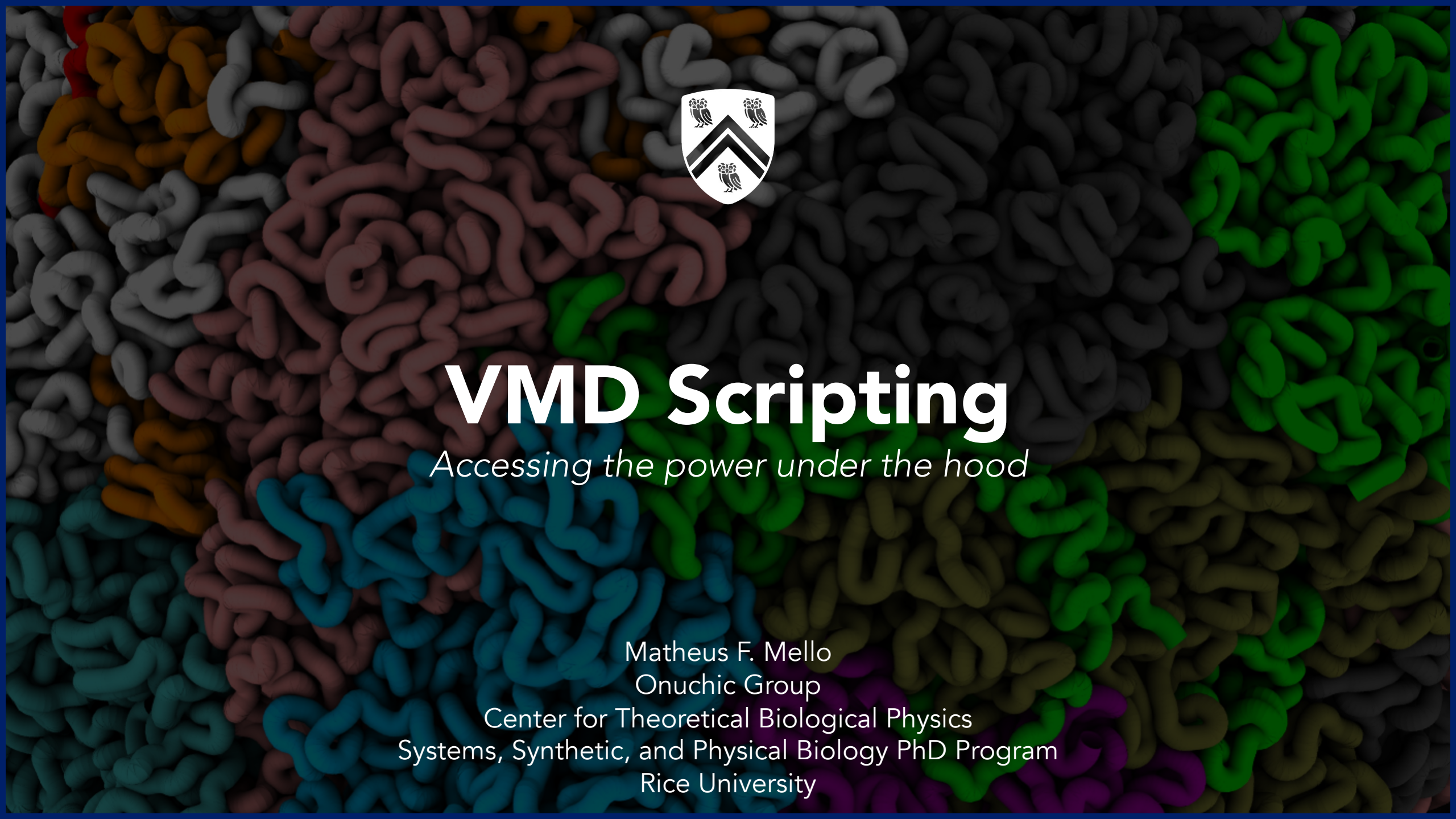


VMD Scripting

Accessing the power under the hood

Matheus F. Mello
Onuchic Group
Center for Theoretical Biological Physics
Systems, Synthetic, and Physical Biology PhD Program
Rice University

Agenda

The Tk console and basic commands

VMD scripting and managing trajectories

Movie making

Why use scripting in VMD?

Avoid performing numerous manual operations in the GUI

Explore VMD capabilities beyond the GUI

Producing advanced high-quality videos supporting your research

VMD 2.0.0 – alpha version

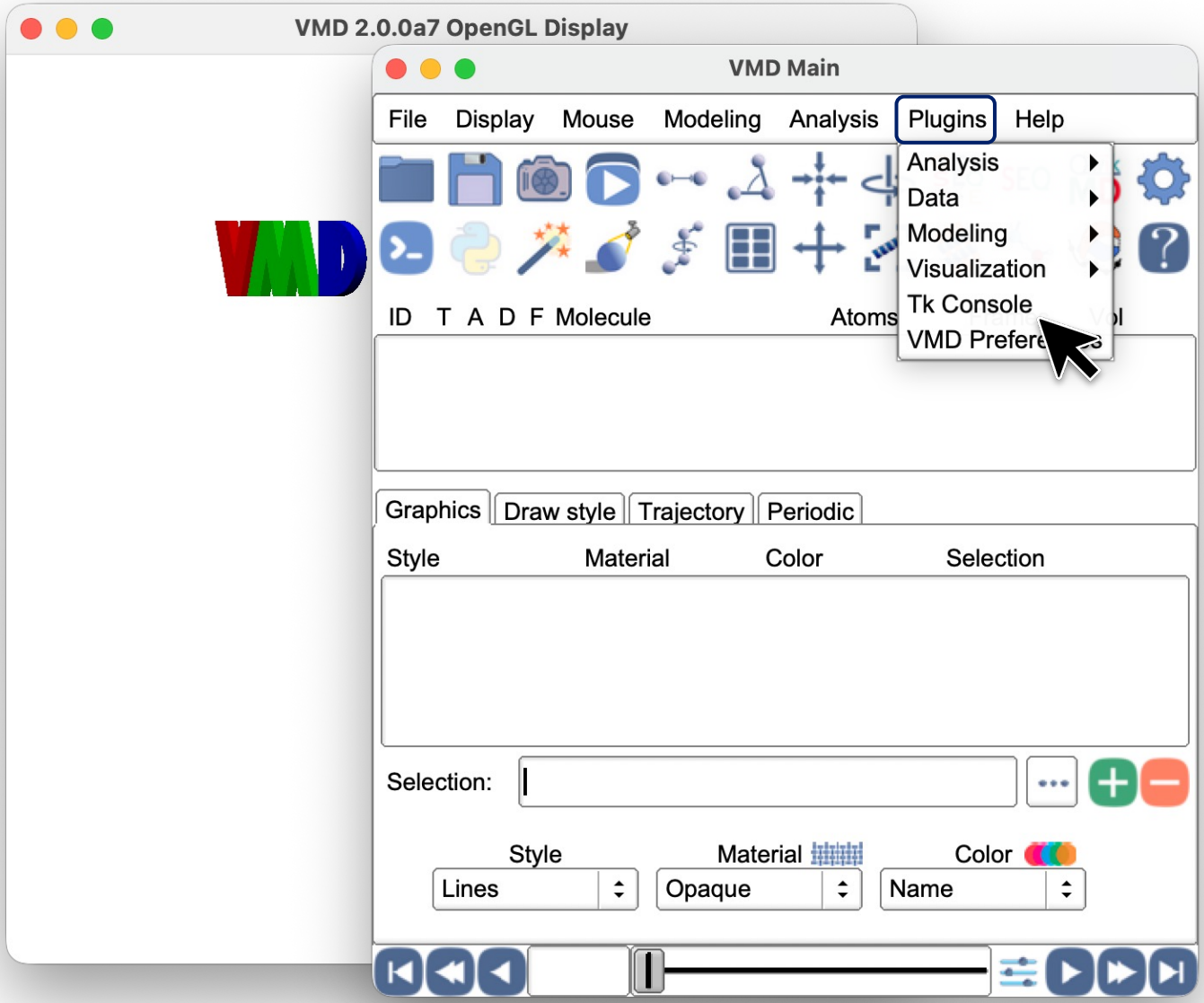
We will use the pre-release VMD 2.0.0 alpha test version

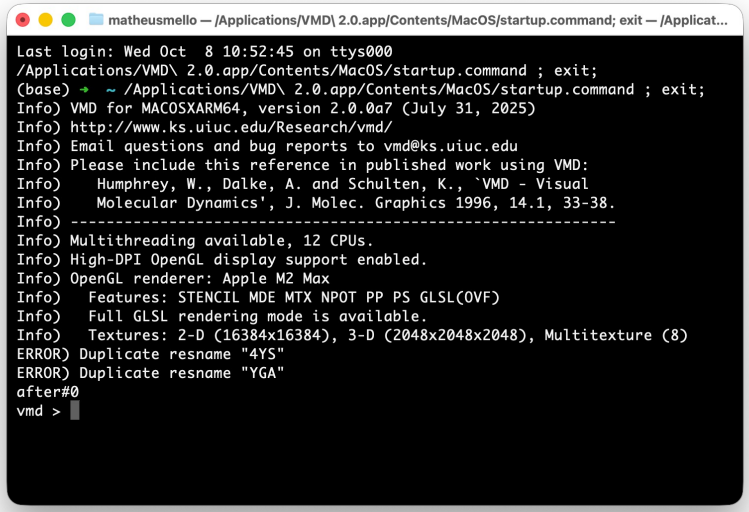
As VMD 2.0.0a7 is a recent release (August 2025), it may still contain some bugs.

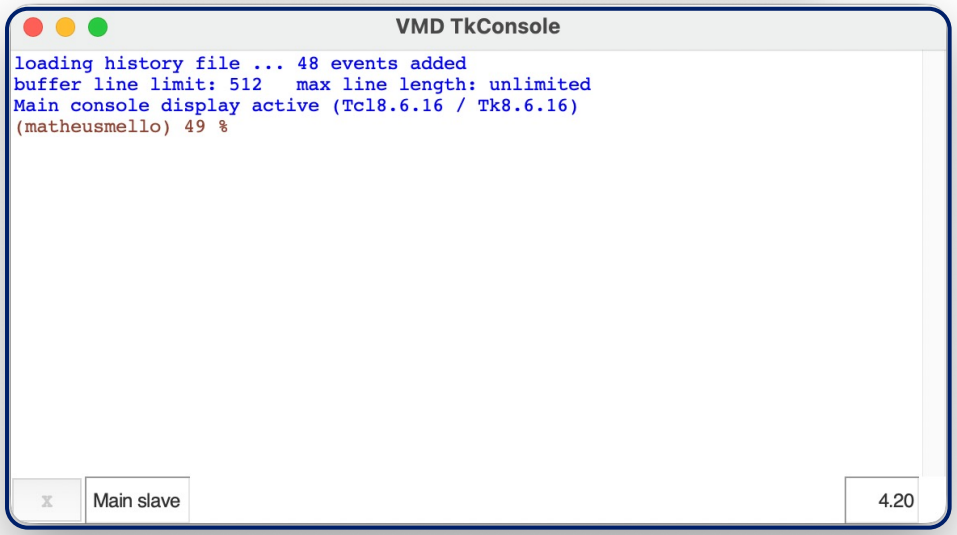
Bug reports, comments, and suggestions can be sent to `vmd@ks.uiuc.edu`

Using previous VMD versions will not affect your ability to participate

Opening VMD and the Tk console







version 2.0.0

The Tk console

The Tk console is a Tcl interpreter

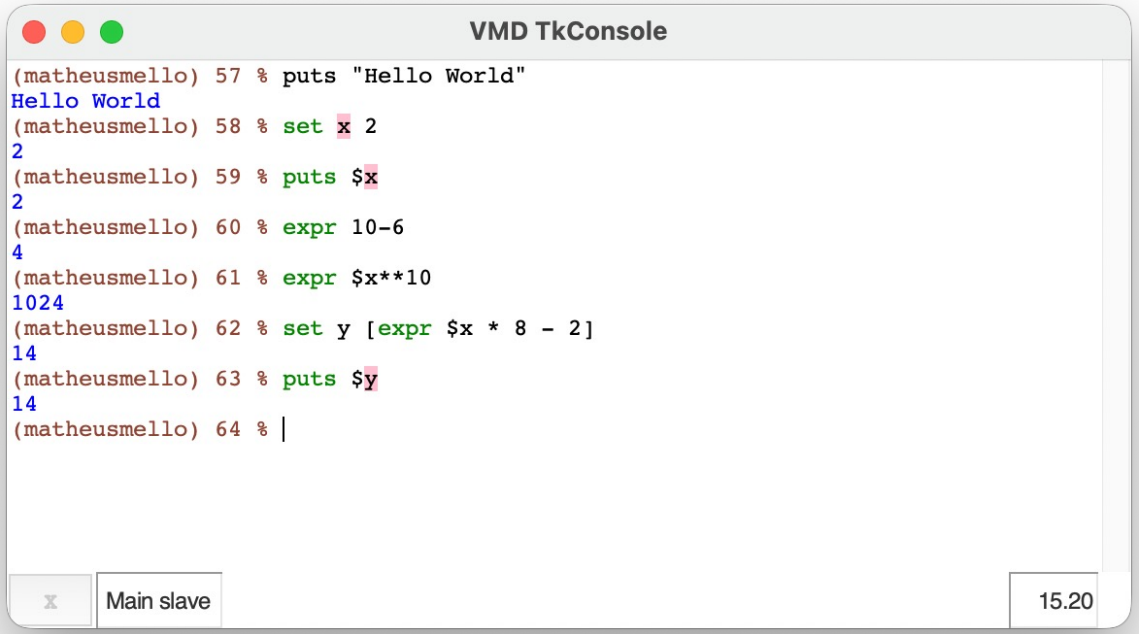
Some Bash commands that still work:

`ls`	`rm`
`cd`	`mkdir`
`pwd`	`touch`
`echo`	`clear`

More resources on Tcl/Tk: https://www.tcl-lang.org/doc/

The Tk console

The Tk console is a Tcl interpreter



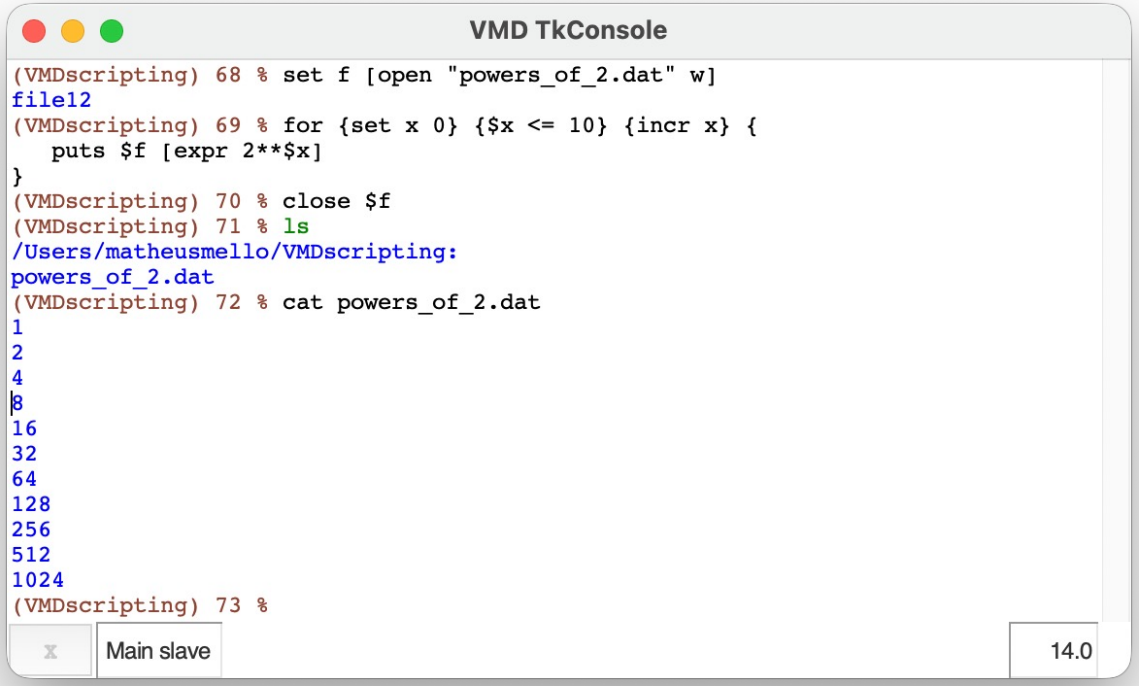
Some useful Tcl commands:

`puts`	Print message on display
`set`	Declare variables
`expr`	Perform mathematical operations
`open`	Open files for reading/writing
`for`	Executes a `for` loop
`foreach`	Iterate through a list of elements
`while`	Loop until condition is met
`list`	Declare lists
`source`	Execute commands in file

More resources on Tcl/Tk: https://www.tcl-lang.org/doc/

The Tk console

The Tk console is a Tcl interpreter



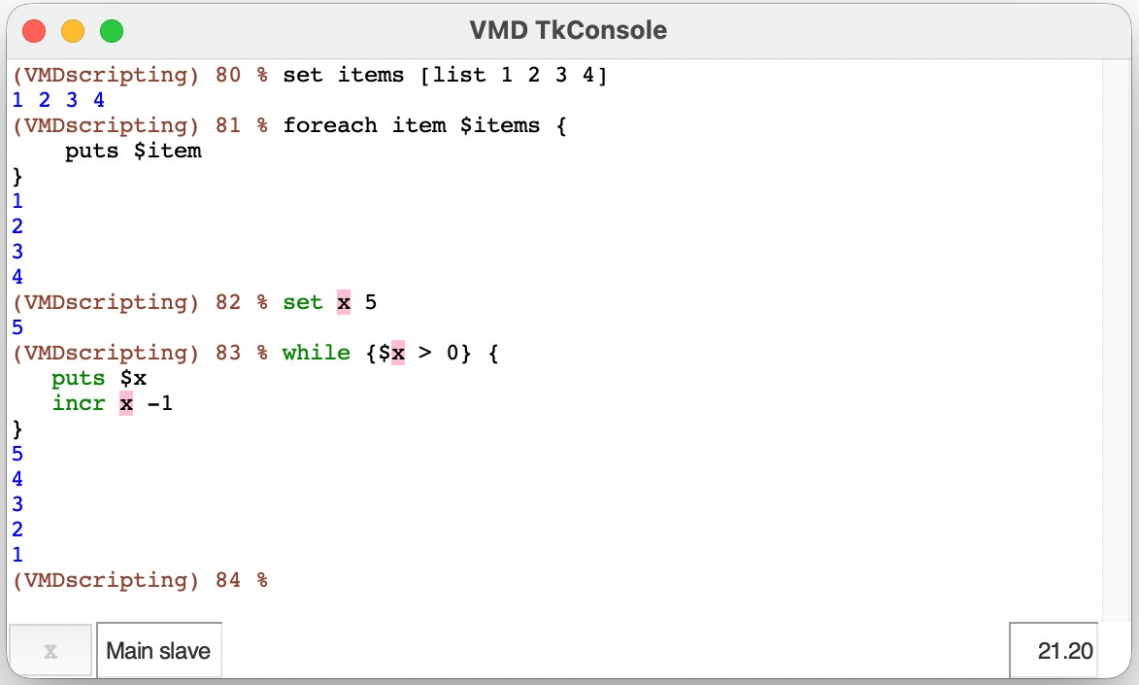
Some useful Tcl commands:

`puts`	Print message on display
`set`	Declare variables
`expr`	Perform mathematical operations
`open`	Open files for reading/writing
`for`	Executes a `for` loop
`foreach`	Iterate through a list of elements
`while`	Loop until condition is met
`list`	Declare lists
`source`	Execute commands in file

More resources on Tcl/Tk: https://www.tcl-lang.org/doc/

The Tk console

The Tk console is a Tcl interpreter



Some useful Tcl commands:

`puts`	Print message on display
`set`	Declare variables
`expr`	Perform mathematical operations
`open`	Open files for reading/writing
`for`	Executes a `for` loop
`foreach`	Iterate through a list of elements
`while`	Loop until condition is met
`list`	Declare lists
`source`	Execute commands in file

More resources on Tcl/Tk: https://www.tcl-lang.org/doc/

VMD Scripting

Almost anything that we do in the GUI can be done with text commands!

Each of these commands have many options within them. We'll talk about

`atomselect`

`animate`

`measure`

`mol`

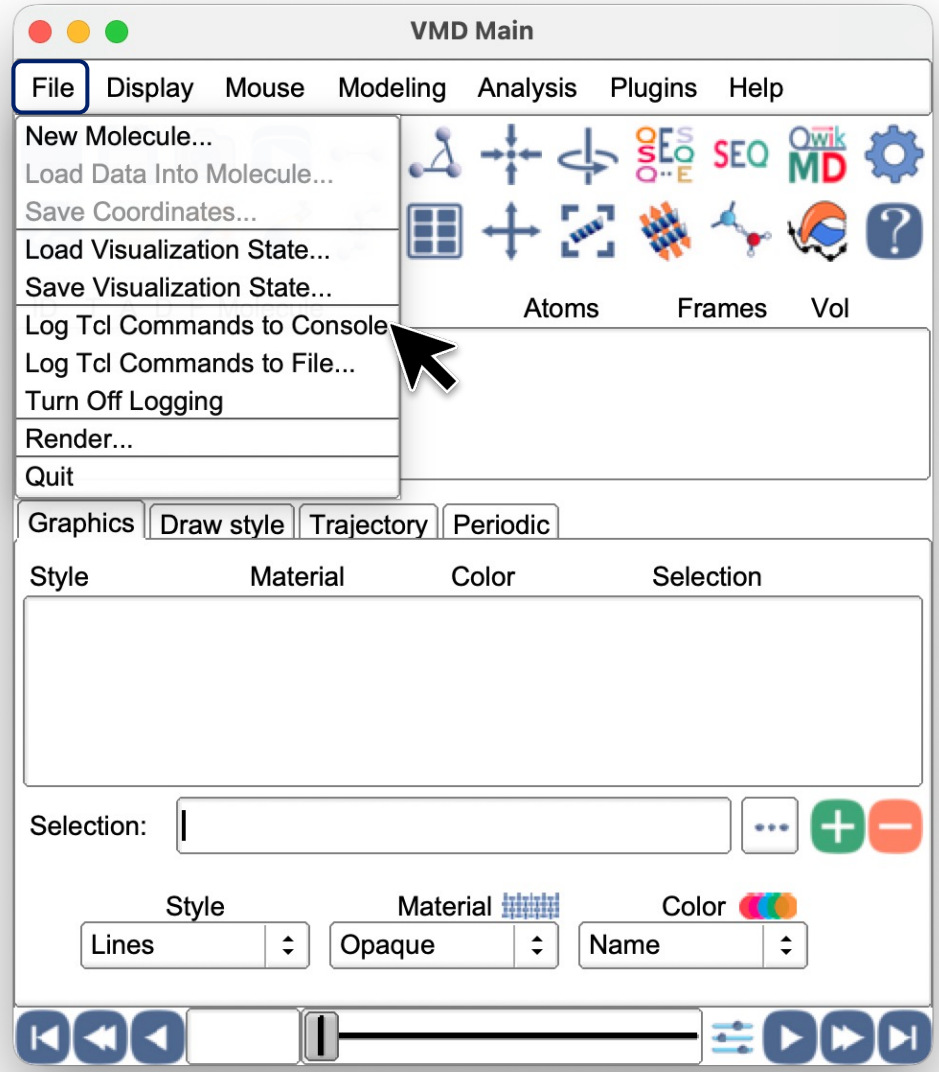
`render`

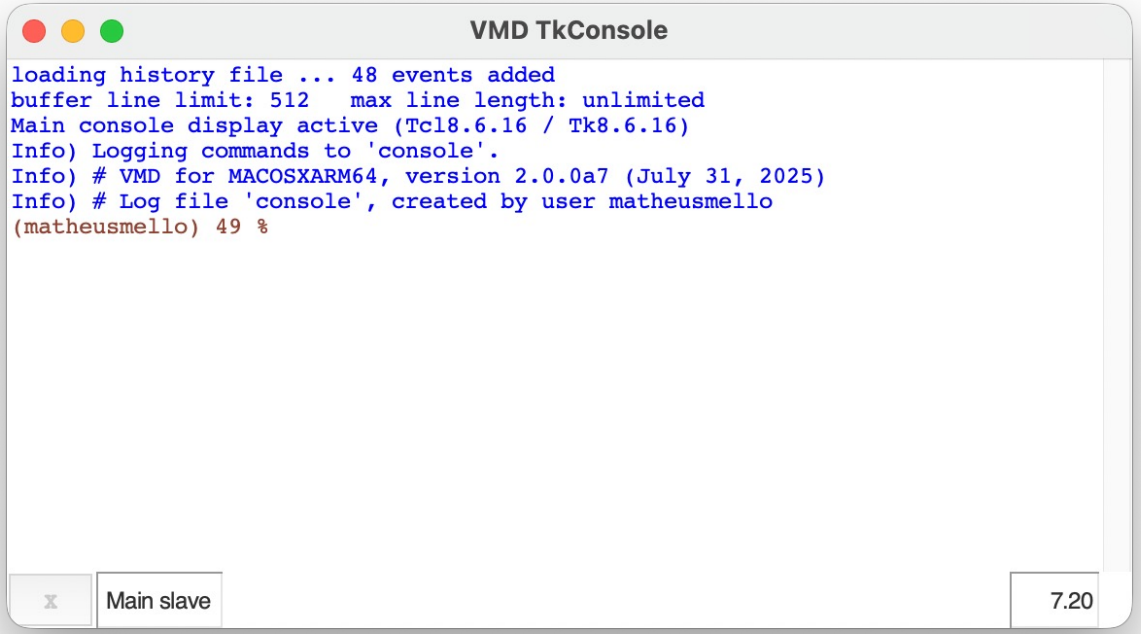
Table 8.1: Summary of core text commands in VMD.

First Word	Description
animate	Play/Pause/Rewind a molecular trajectory.
atomselect	Create atom selection objects for analysis.
axes	Position a set of XYZ axes on the screen.
color	Change the color assigned to molecules, or edit the colormap.
colorinfo	(Tcl) Obtain color properties for various objects
display	Change various aspects of the graphical display window.
exit, quit	Quit VMD.
help	Display an on-line help file with an HTML viewer.
imd	Control the connection to a remote simulation.
label	Turn on/off labels for atoms, bonds, angles, dihedral angles, or springs.
light	Control the light sources used to illuminate graphical objects.
logfile	Turn on/off logging a VMD session to a file or the console.
material	Create new material definitions and modify their settings.
measure	Measure properties of moleculear structures.
menu	Control or query the on-screen GUI menu forms.
molecule or mol	Load, modify, or delete a molecule.
molinfo	Get information about a molecule or loaded file.
mouse	Change the current state (mode) of the mouse.
play	Start executing text commands from a specified file.
render	Output the currently displayed image (scene) to a file.
rock	Rotate the current scene continually at a specified rate.
rotate	Rotate the current scene around a given axis by a certain angle.
scale	Scale the current scene up or down.
stage	Position a checkerboard stage on the screen.
tool	Initialize and control external spatial tracking devices.
translate	Translate the objects in the current scene.
user	Add new keyboard commands.
vmdinfo	(Tcl) Get information about this version of VMD
wait	Wait a number of seconds before reading another command. Animation continues.
sleep	Sleep a number of seconds before reading another command. Animation is frozen.

For details on each command: https://www.ks.uiuc.edu/Research/vmd/vmd-1.8.3/ug/node105.html

Discovering commands from the VMD GUI

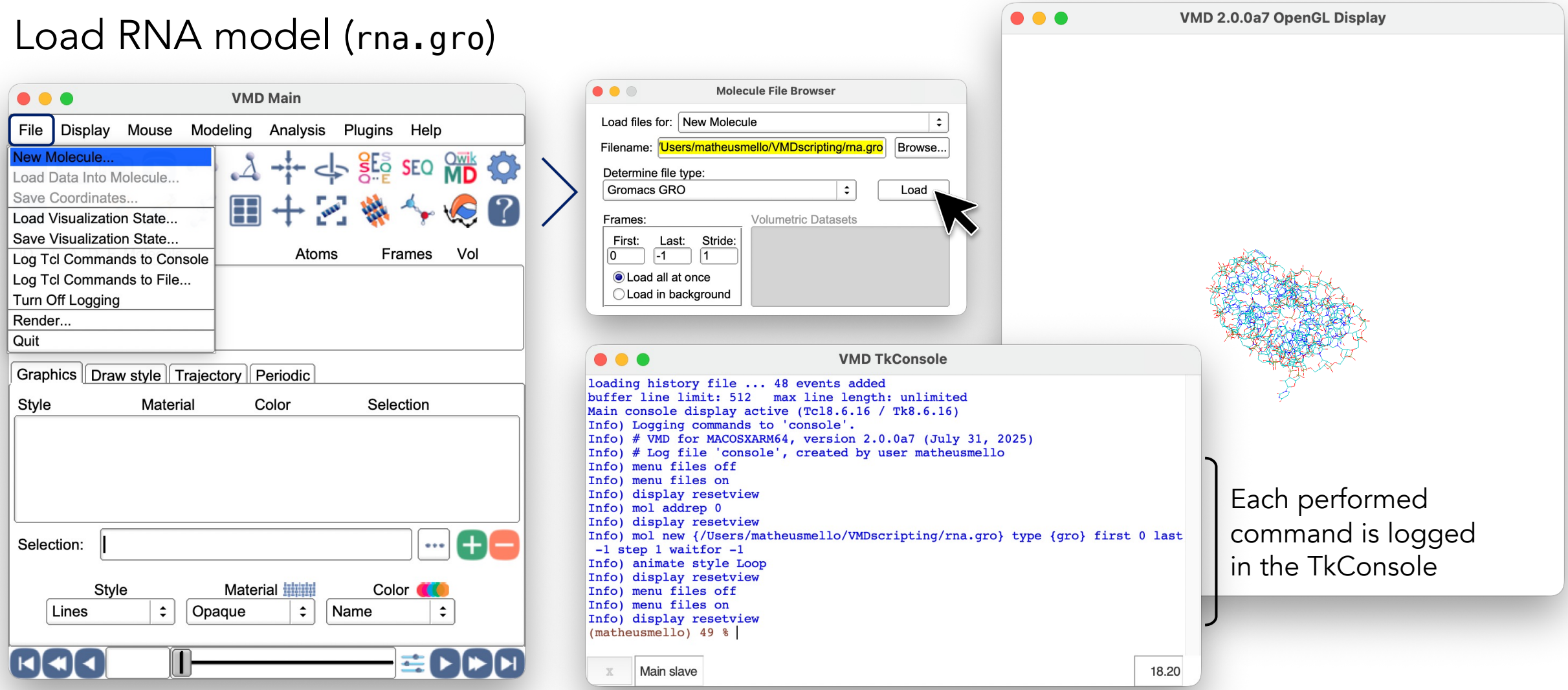




version 2.0.0

Discovering commands from the VMD GUI

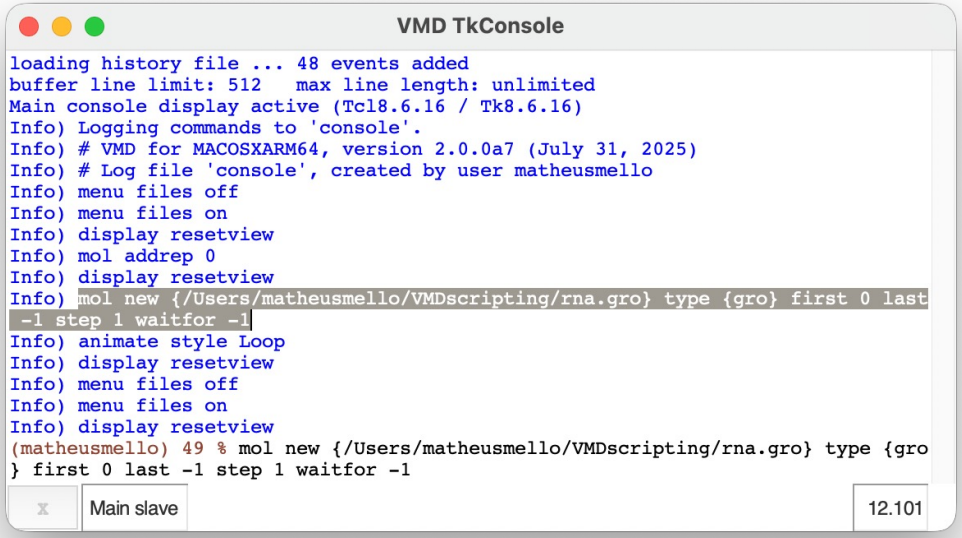
Load RNA model (`rna.gro`)



https://github.com/Whitford/ctbp-techtalks/tree/main/2025/2.VMDscripting/resources

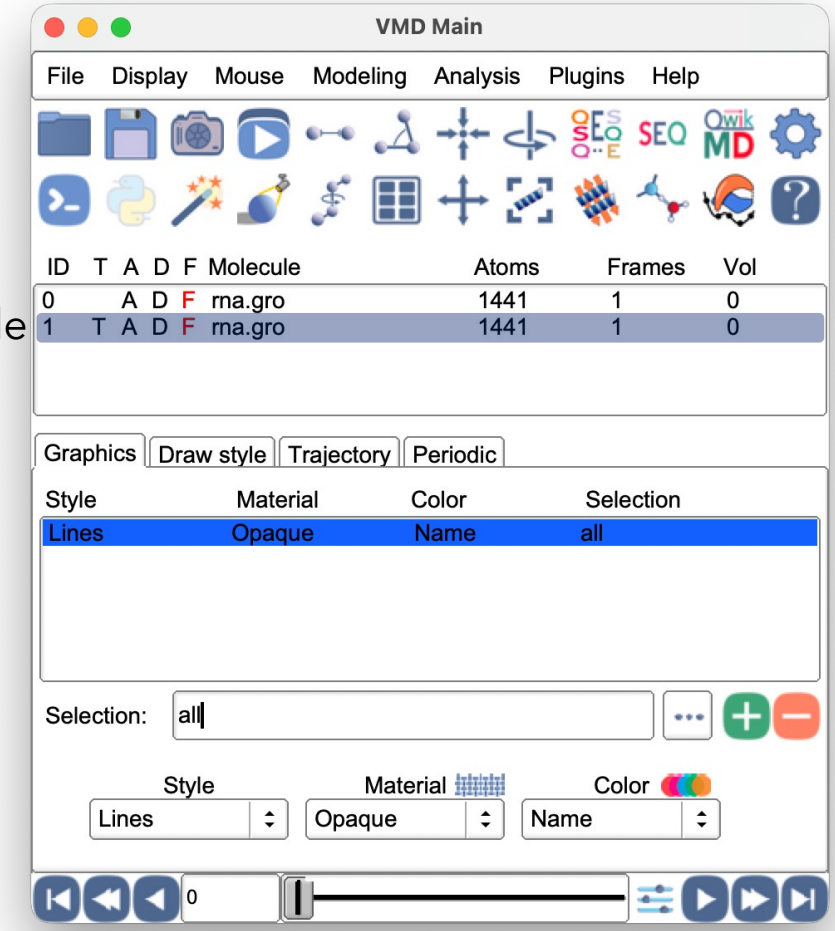
Discovering commands from the VMD GUI

Load RNA model (`rna.gro`), again, now with text commands



Copy and paste `mol` command

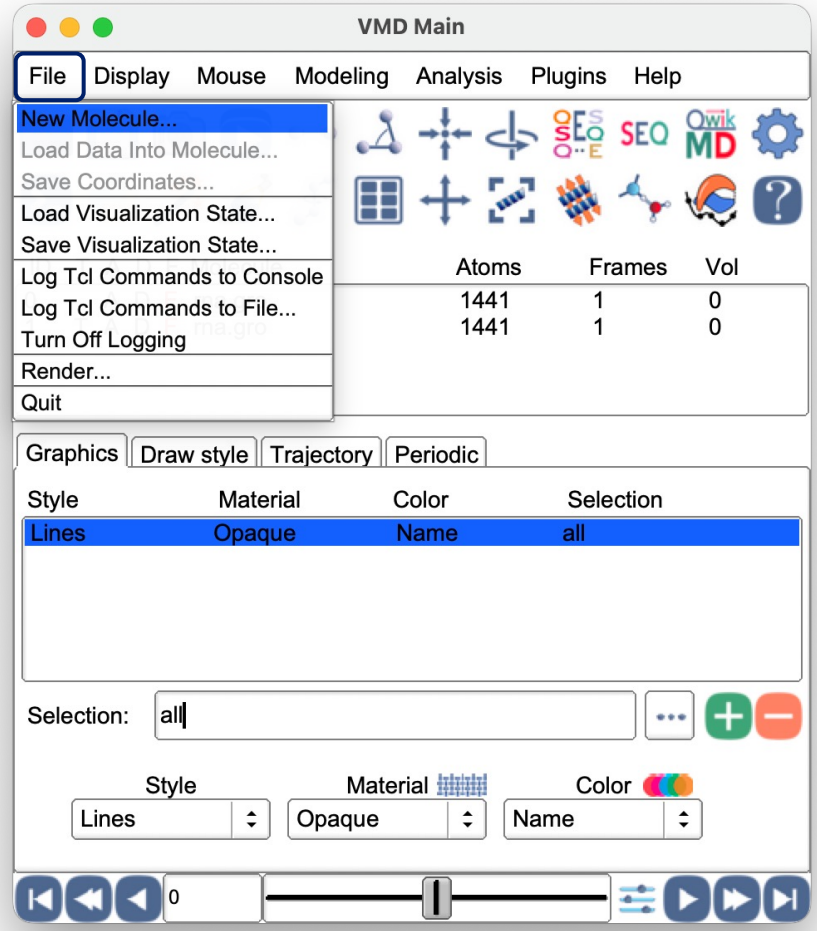
new molecule

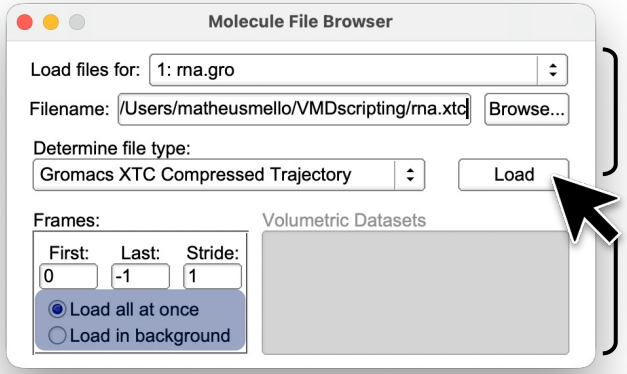


version 2.0.0

Managing trajectories

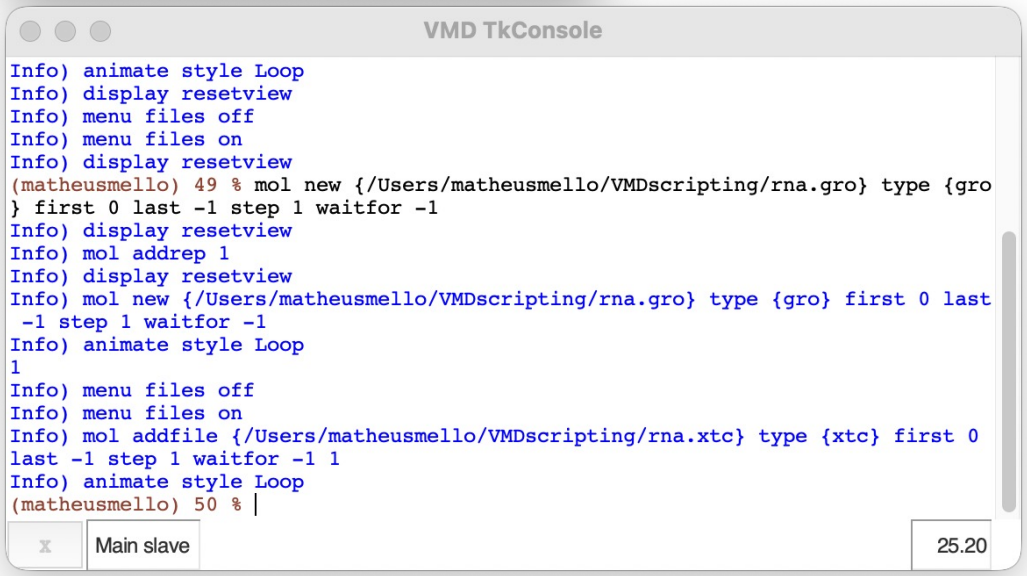
Load RNA trajectory (`rna.xtc`) on the second molecule





The trajectory file is loaded on top of the topology file

`Load all at once` loads faster, especially for larger systems

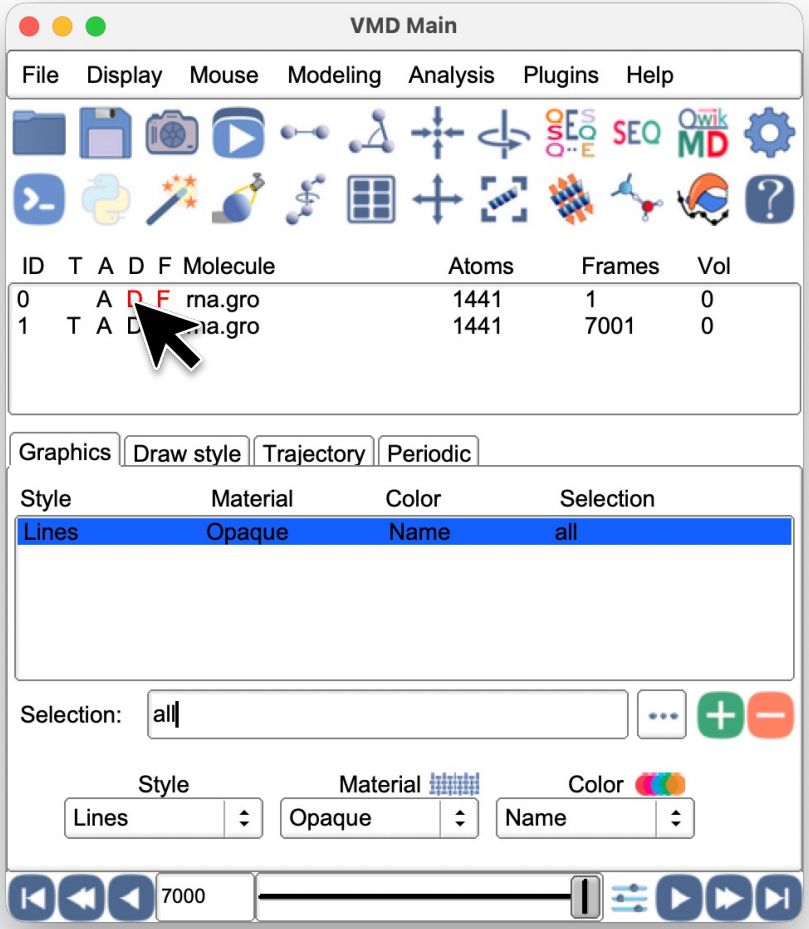


Notice the new commands

https://github.com/Whitford/ctbp-techtalks/tree/main/2025/2.VMDscripting/resources

Managing trajectories

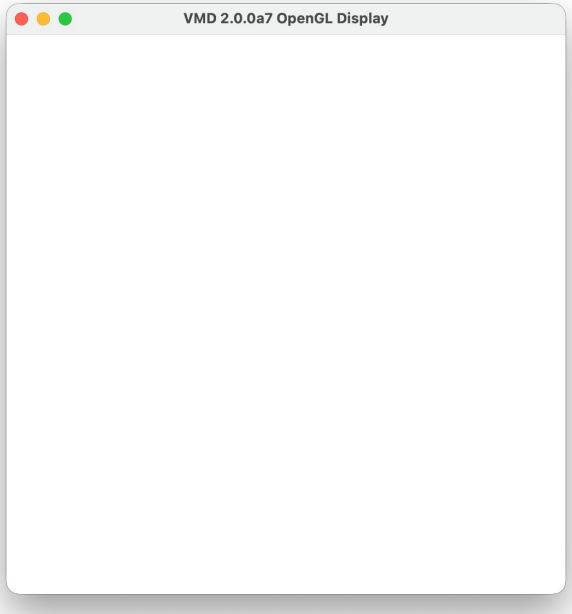
Let's hide the first molecule for now…

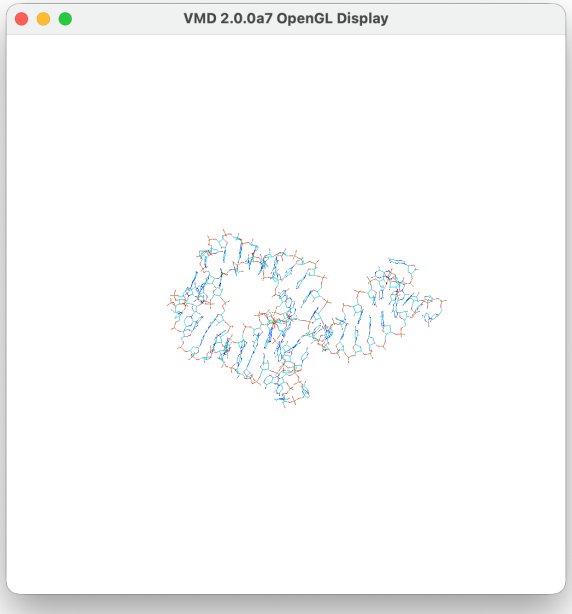


Note that the last frame is out of the field of view

Solutions:

- press '=' to reset view
- align trajectory to first frame

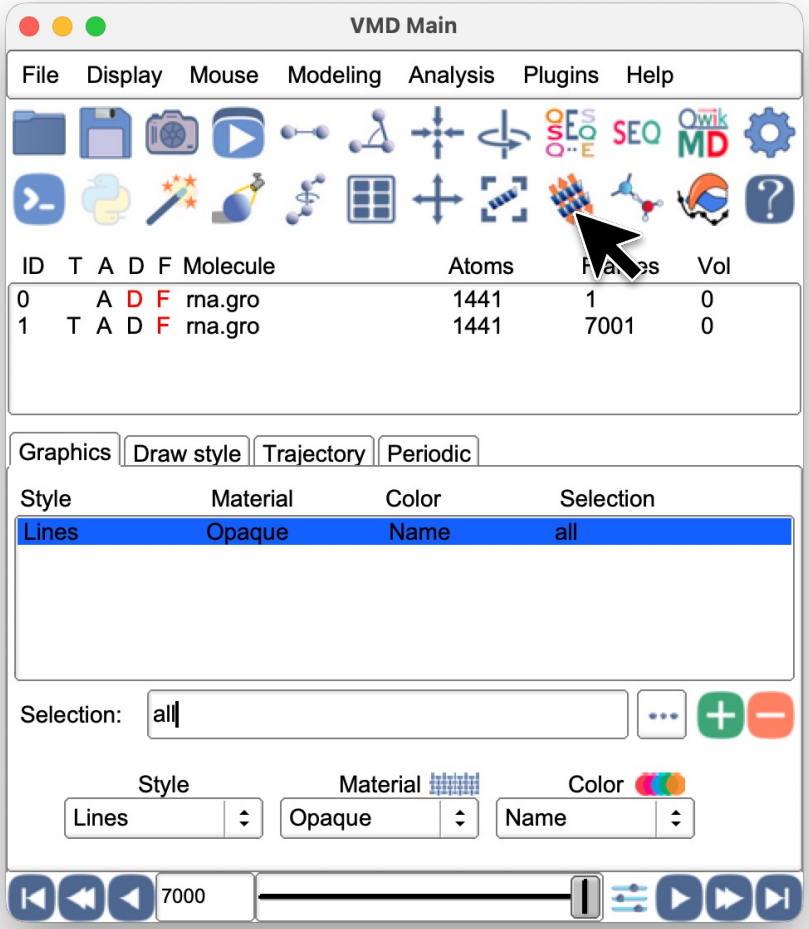




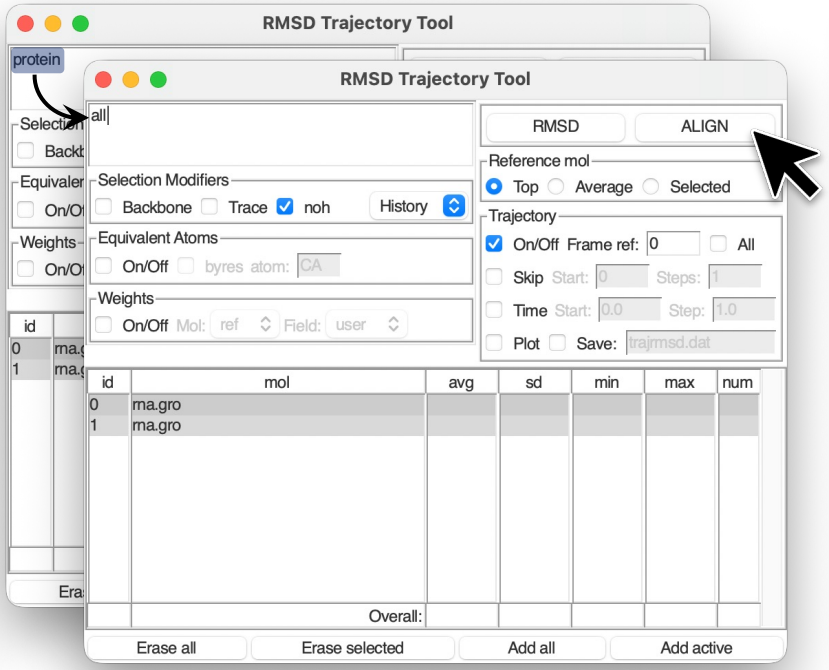
version 2.0.0

Aligning trajectories

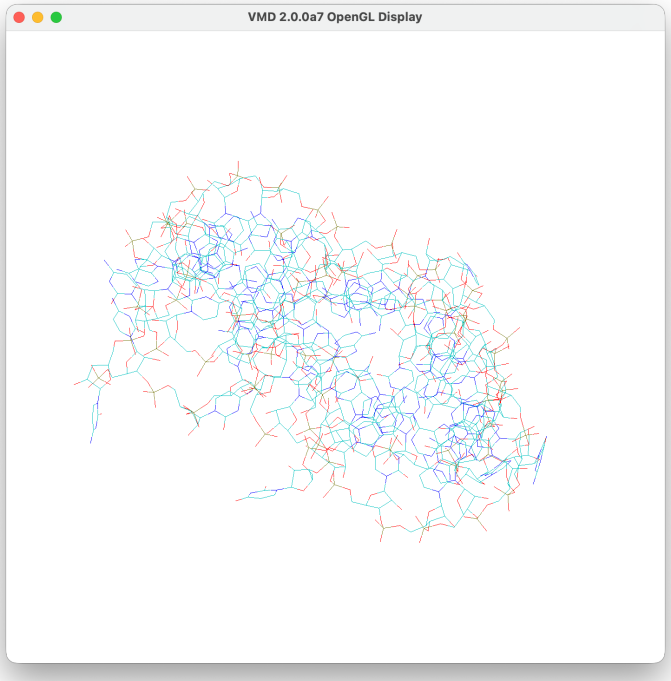
Open RMSD Trajectory tool



Change selection and align



Reset view (=)

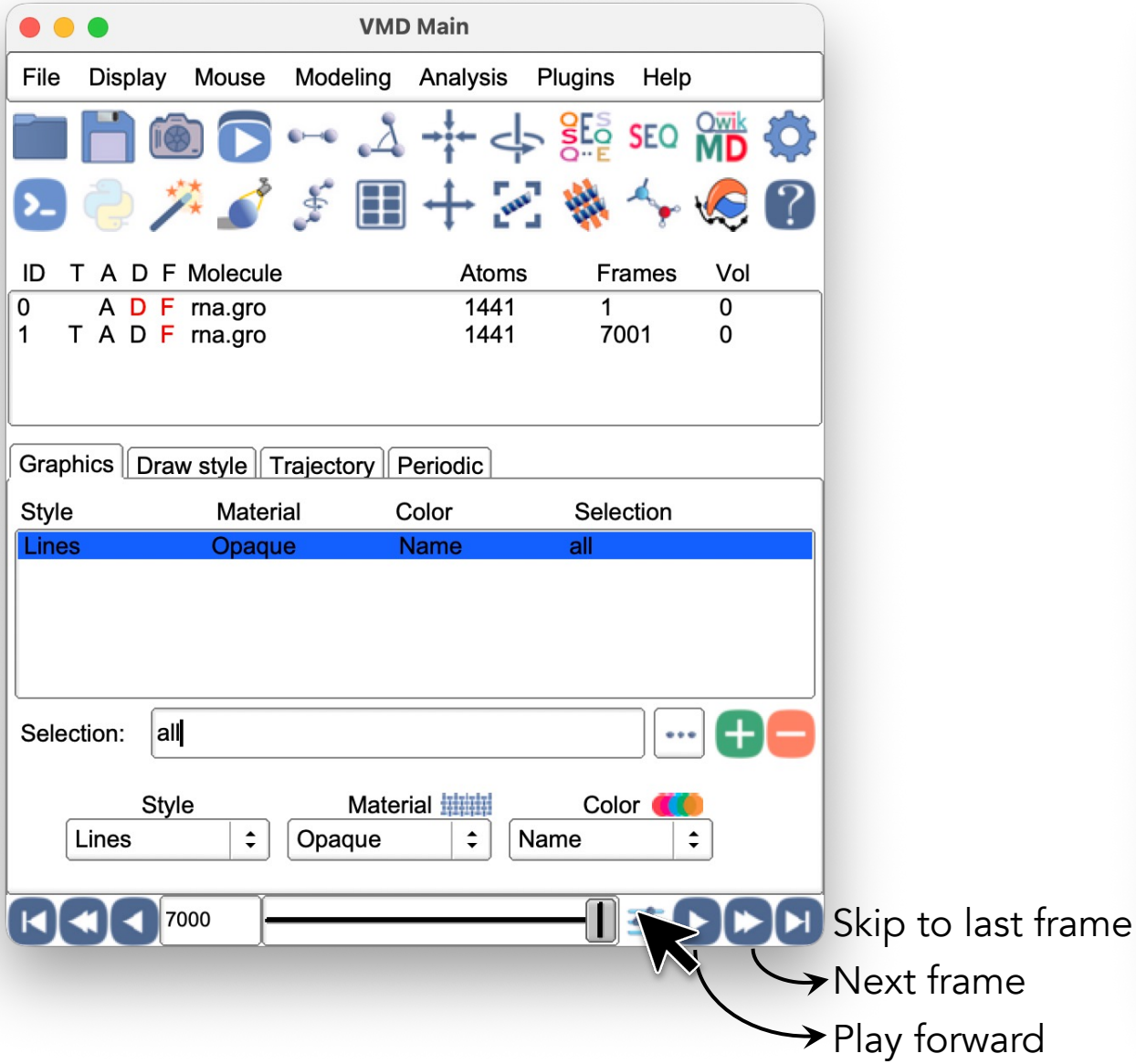


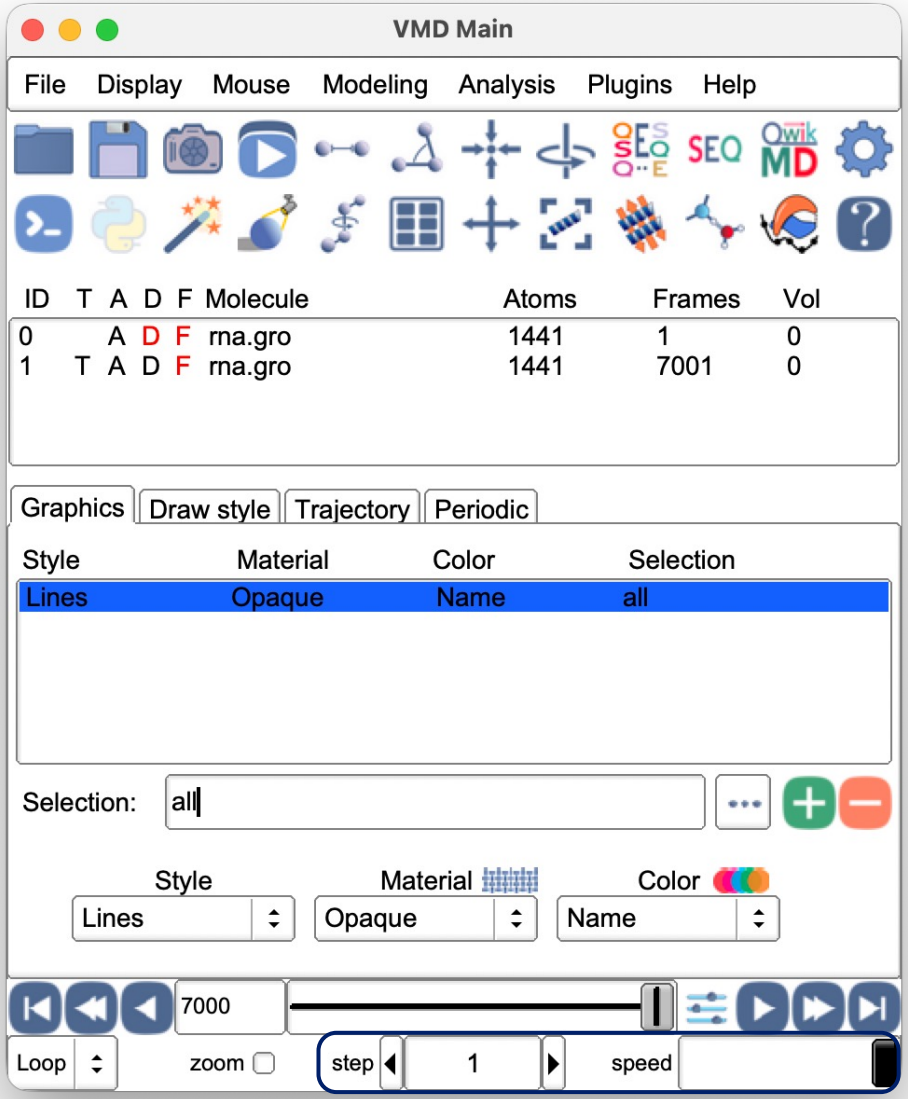
Try a custom selection and explore a representation that you like!

Alternatively : `Extensions > Analysis > RMSD Trajectory Tool`

version 2.0.0

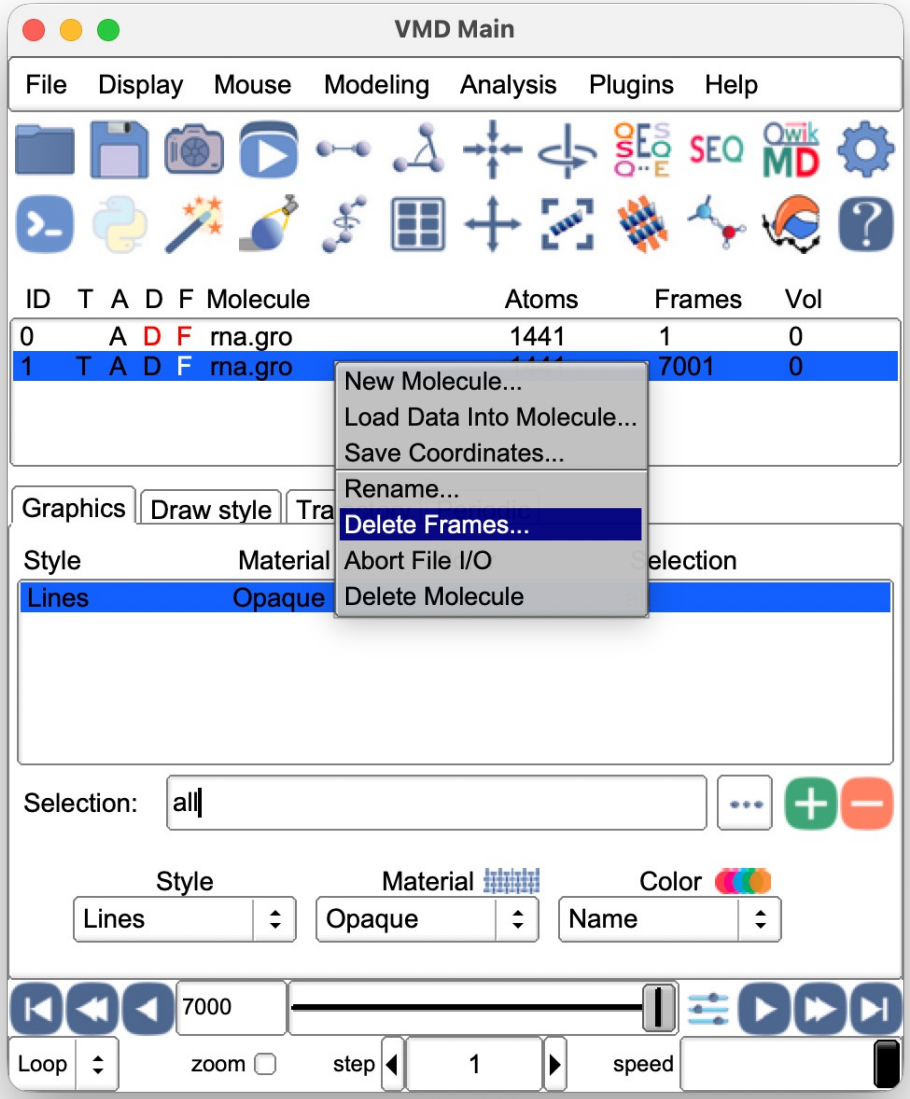
Managing trajectories



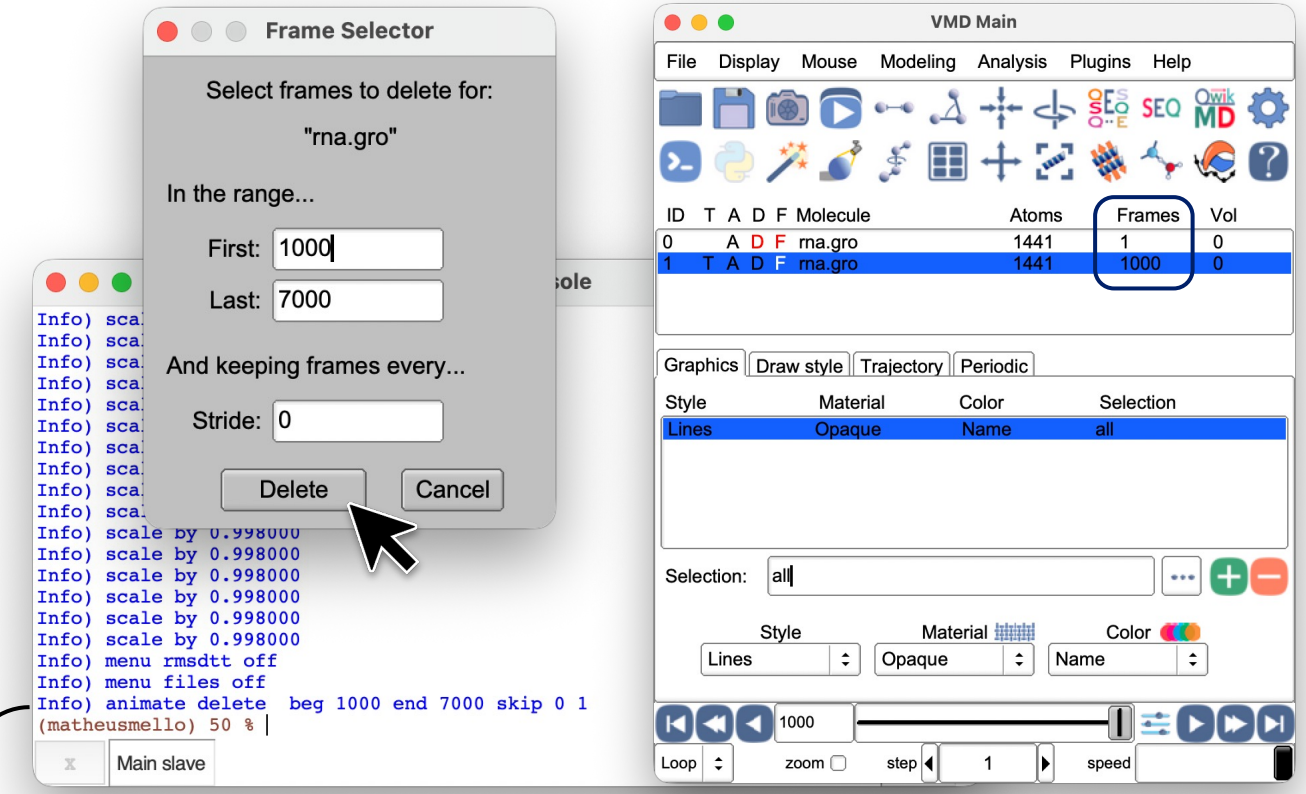


version 2.0.0

Managing trajectories



The trajectory is quite long!
Let's keep only the first 1,000 frames.



Notice the commands on the Tk console

version 2.0.0

Going back to scripting…

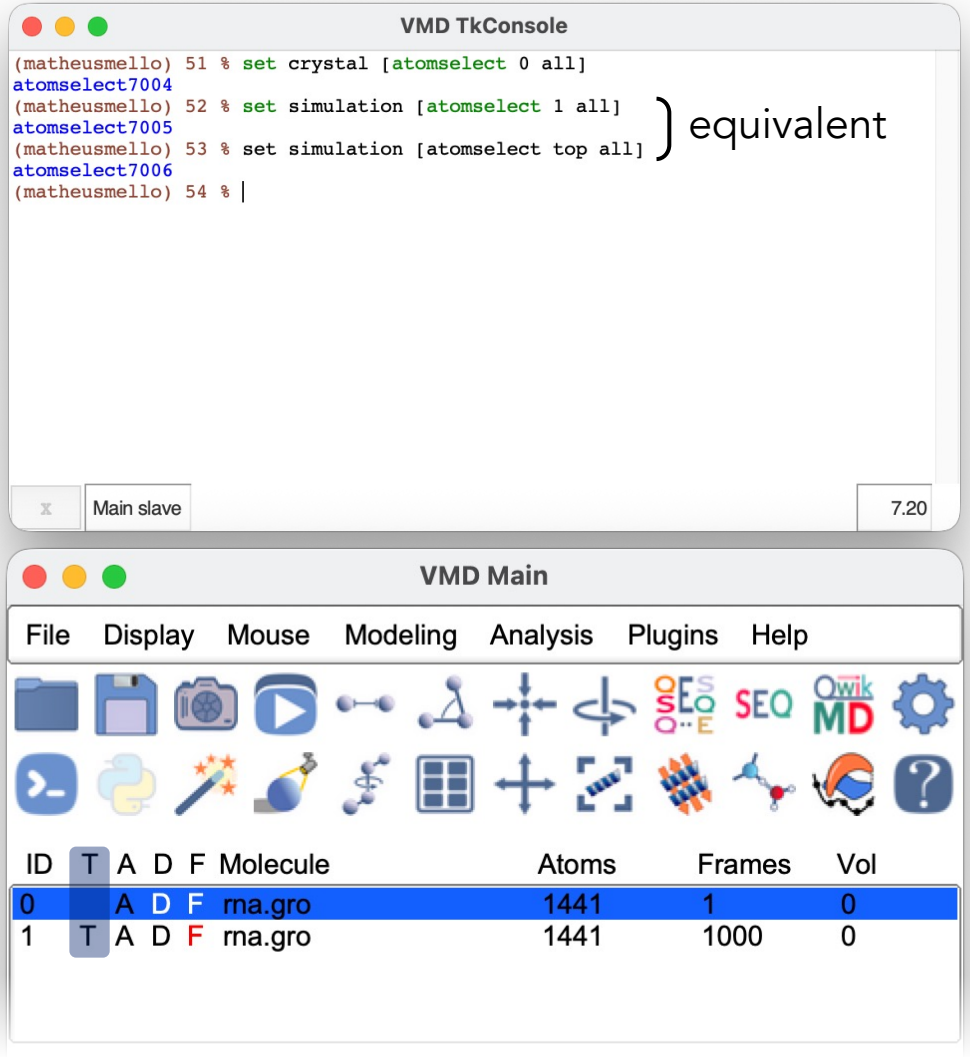
Our goal: align last frame of `mol1` to `mol0`

Our tools: `atomselect` and `measure`

```
> atomselect molid selection
```

Let's create two selections:

1. all atoms on the first molecule *(our crystal structure)*
2. all atoms on the second molecule *(our simulation)*



Aligning two different molecules

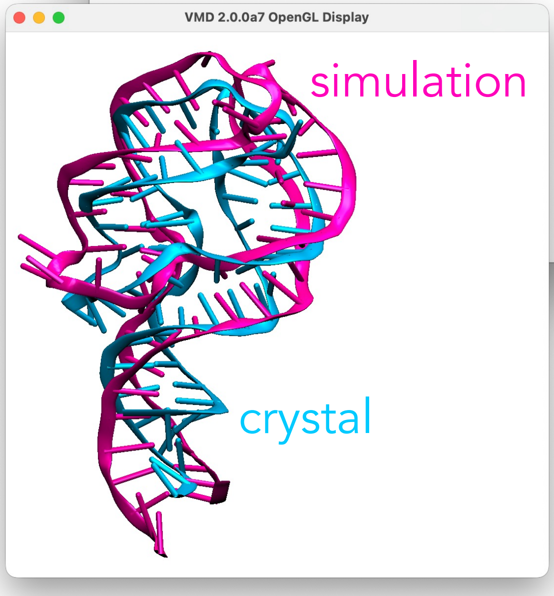
The `atomselect` object has some properties

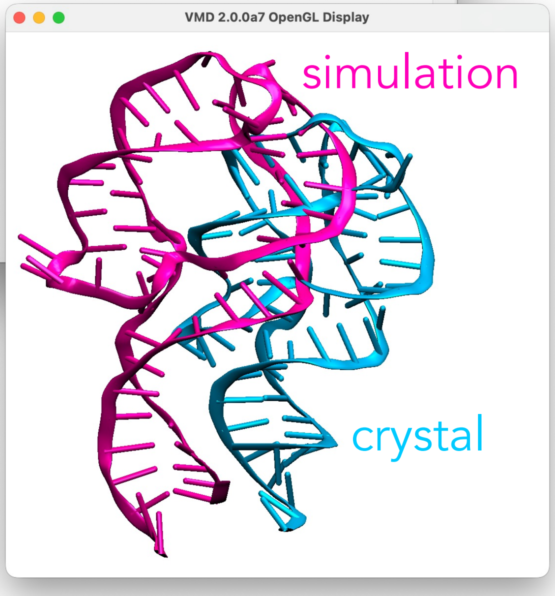
```
VMD TkConsole
(matheusmello) 51 % set crystal [atomselect 0 all]
atomselect7004
(matheusmello) 52 % set simulation [atomselect 1 all]
atomselect7005
(matheusmello) 53 % set simulation [atomselect top all]
atomselect7006
(matheusmello) 54 % $simulation num
1441
(matheusmello) 55 % $simulation get resname
G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P G0P
G G G G G G G G G G G G G G G G G G G G G G G C C C C C C C C C C C C C C C C C C
C C C C C C C C C C C C C C C C C C C C C C C U U U U U U U U U U U U U U U U U U
U U U U U U U U U U U U U U U U U U U U U U U A A A A A A A A A A A A A A A A A A
A A A A A U U U U U U
A A A A A A A C C C C
A A A A A A A A A G G
G G G G G G G G G G G
U U U U U U U U U U U
G G G G G G G G G G G
C C C A A A A A A A A
x   Main slave
```

```
VMD TkConsole
G G G G G G G G G G G G G G G G G G G G G U U U U U U U U U U U U U U U U U U U
U C C C C C C C C C C C C C C C C C C C C
(matheusmello) 56 % $simulation get {x y z}
{252.58126831054688 232.31719970703125 220.2235870361328} {253.63482666015625 23
1.818359375 221.13516235351562} {254.5537567138672 231.18777465820312 220.364654
54101562} {254.1644287109375 230.0826416015625 219.72955322265625} {254.96287536
621094 231.99610900878906 219.18508911132812} {256.00213623046875 232.8908386230
4688 219.44261169433594} {255.5416717529297 231.0959014892578 218.05799865722656
} {257.0010681152344 231.17723083496094 218.23025512695312} {254.9283905029297 2
29.86895751953125 218.4996337890625} {254.04730224609375 229.4636688232422 217.4
2898559570312} {252.6425323486328 229.70860290527344 217.19444274902344} {252.31
59942626953 228.9972381591797 216.06178283691406} {253.46109008789062 228.429443
359375 215.4509735107422} {253.62960815429688 227.56622314453125 214.31367492675
78} {252.70802307128906 227.2698211669922 213.54319763183594} {254.8783874511718
8 227.10073852539062 214.13015747070312} {255.8603973388672 227.34652709960938 2
15.07176208496094} {256.939697265625 226.86167907714844 214.87098693847656} {255
.6680450439453 228.0742645263672 216.13699340820312} {254.51266479492188 228.603
2257080078 216.4507598876953} {255.74134826660156 234.45089721679688 219.6815
948486328} {256.7967834472656 234.77171325683594 220.74513244628906} {254.367
85888671875 234.7481231689453 220.01589965820312} {256.2721862792969 235.2000
x   Main slave   12.1197
```

The `atomselect` can also alter the selection

```
VMD TkConsole
(matheusmello) 61 % $crystal moveby {20 0 0}
(matheusmello) 62 % |
```





`atomselect` documentation: https://www.ks.uiuc.edu/Research/vmd/current/ug/node122.html

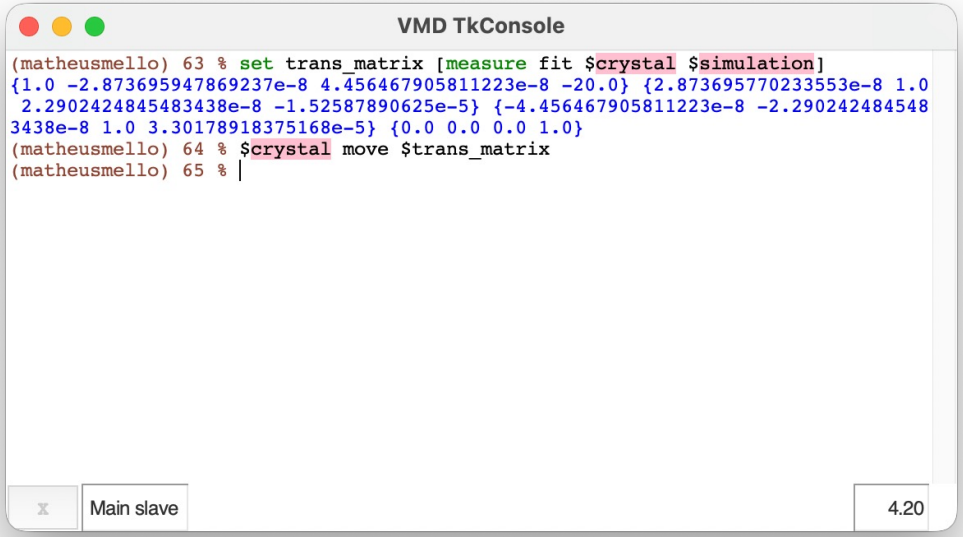
version 2.0.0

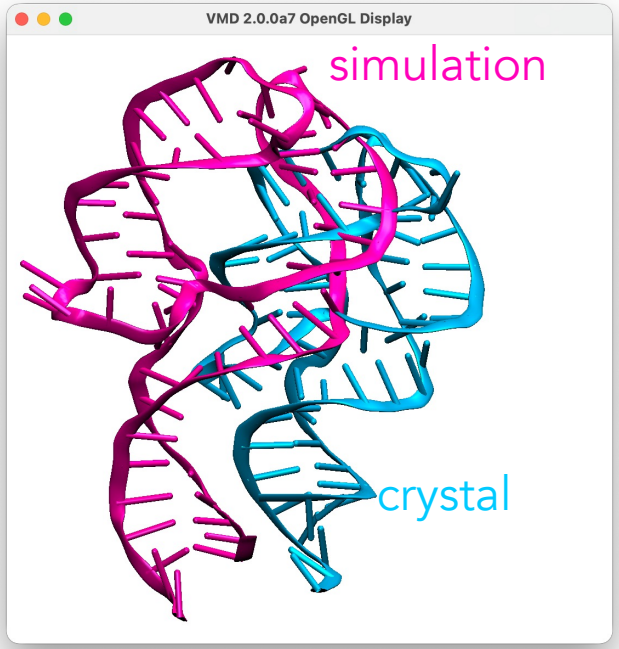
Aligning two different molecules

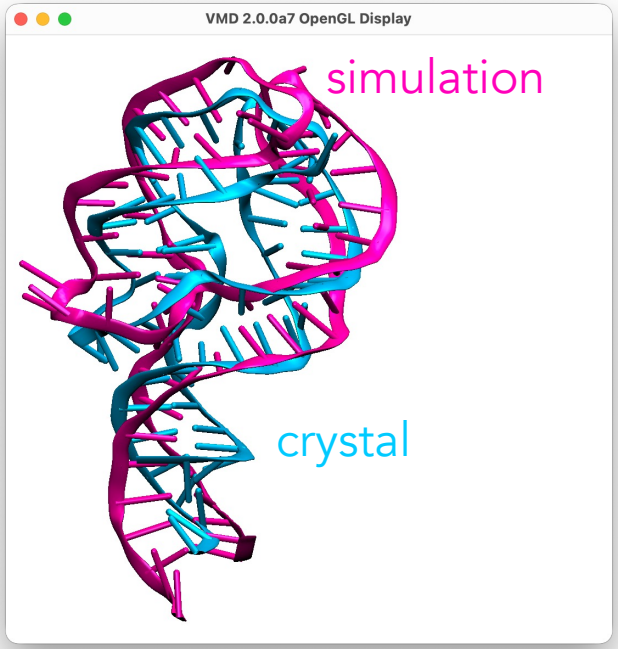
Now we compute the transformation matrix

`> measure fit atomsel1 atomsel2`

Calculates transformation matrix that best maps the *atomsel1* onto *atomsel2*







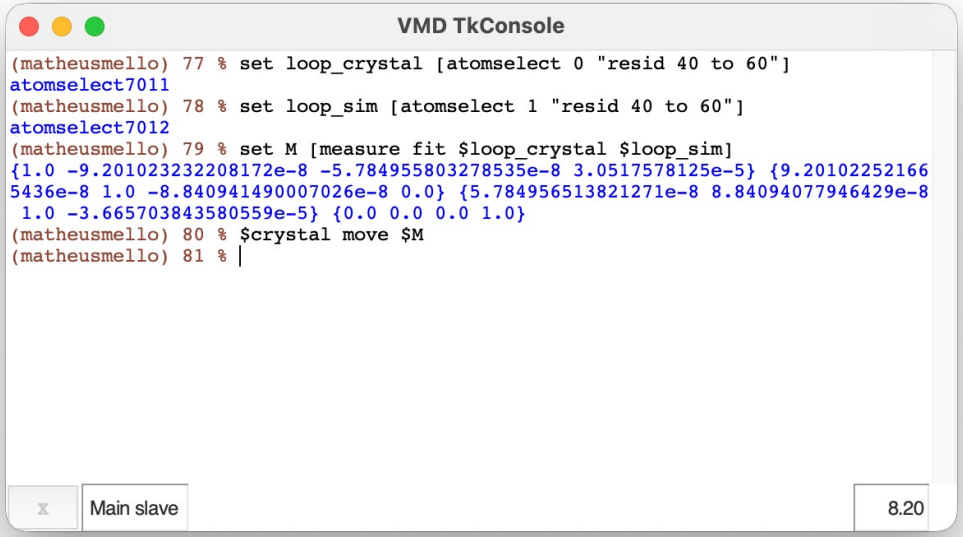
`measure` documentation: https://www.ks.uiuc.edu/Research/vmd/current/ug/node138.html

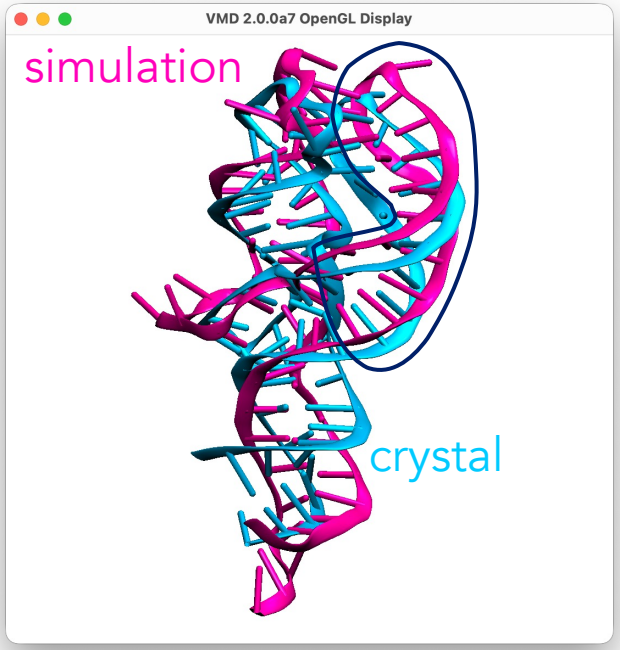
version 2.0.0

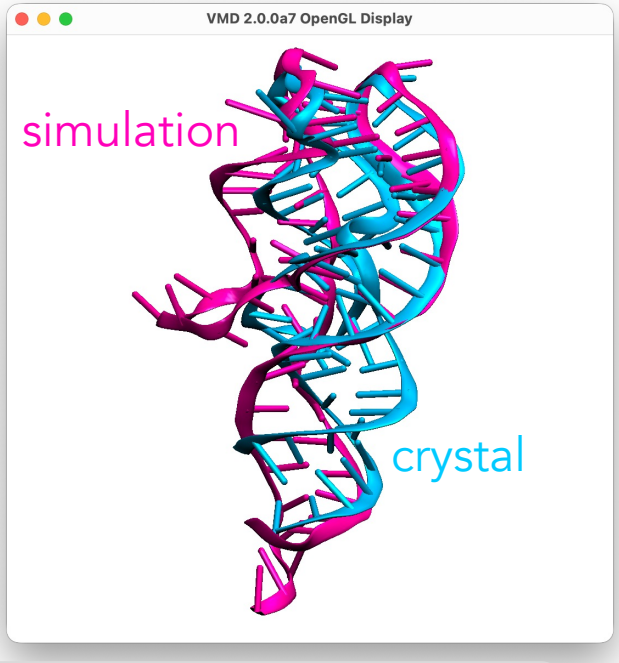
Aligning two different molecules

This procedure works for any two selections that contain the same number of atoms

Note that we move the whole crystal structure after calculating the matrix M.







This approach aligns only the current frame shown on display!

`atomselect` documentation: https://www.ks.uiuc.edu/Research/vmd/current/ug/node122.html

version 2.0.0

Aligning two different molecules

DIY

Create a script to align the loop formed by bases 40 to 60 on the simulation with the crystal structure.

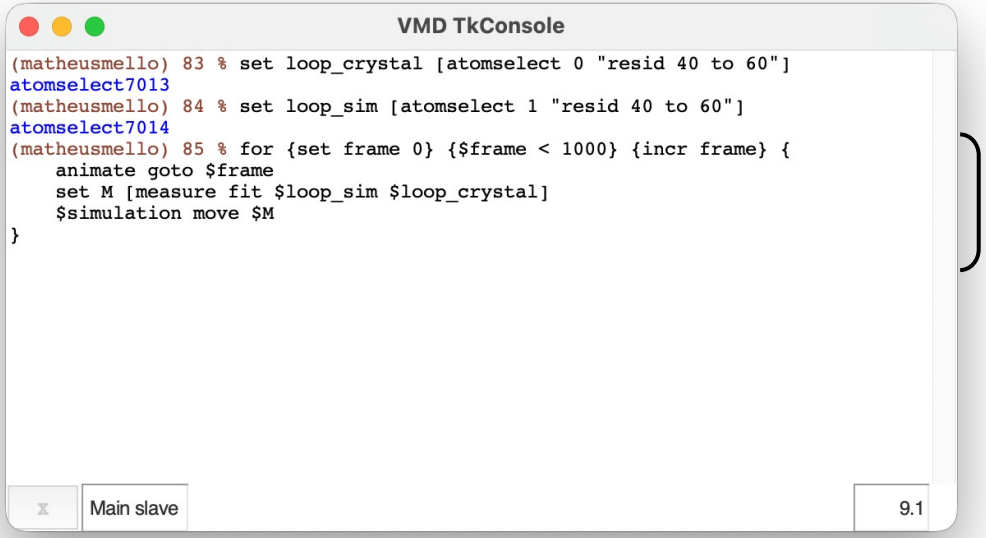
Hint: what gets logged on the Tk console when you change frames in the UI?

Aligning two different molecules

DIY

Create a script to align the loop formed by bases 40 to 60 on the simulation with the crystal structure.

Hint: what gets logged on the Tk console when you change frames in the UI?



Note that now we're moving the simulation in relation to the crystal structure

version 2.0.0

Movie making in VMD

GUI

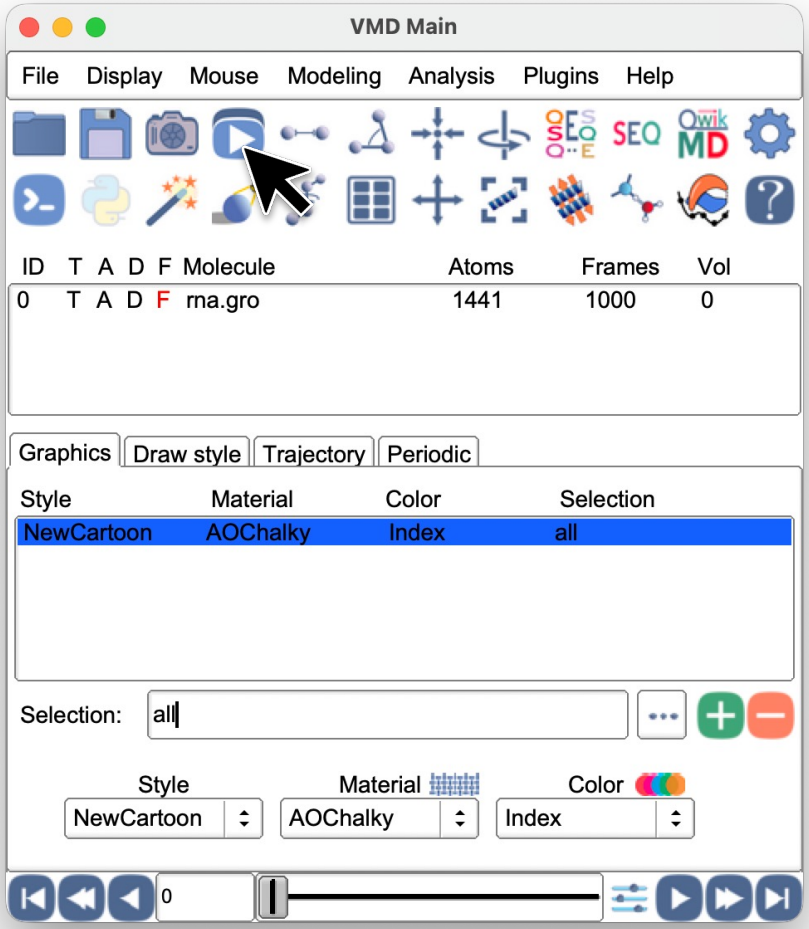
Good for quick tests but can be slow and lack quality.

Scripting

May take more time to set up but can achieve more polished results.

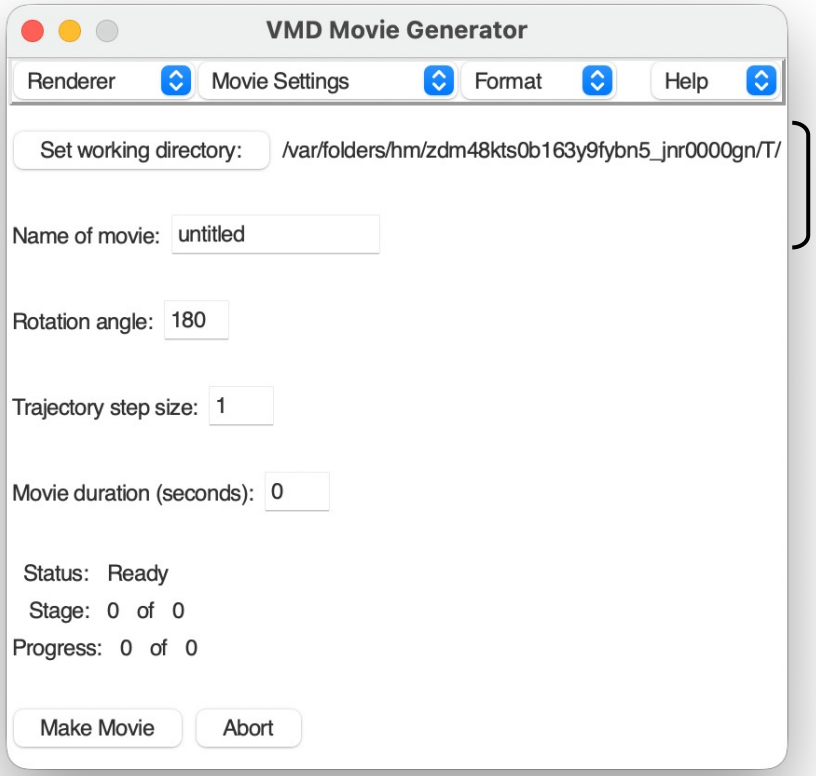
Movie making with the GUI

Let's restart and open the RNA model again



Alternatively: `Extensions > Visualization > Movie Maker`

(version 1.9.4)



Set the proper directory

Name your movie

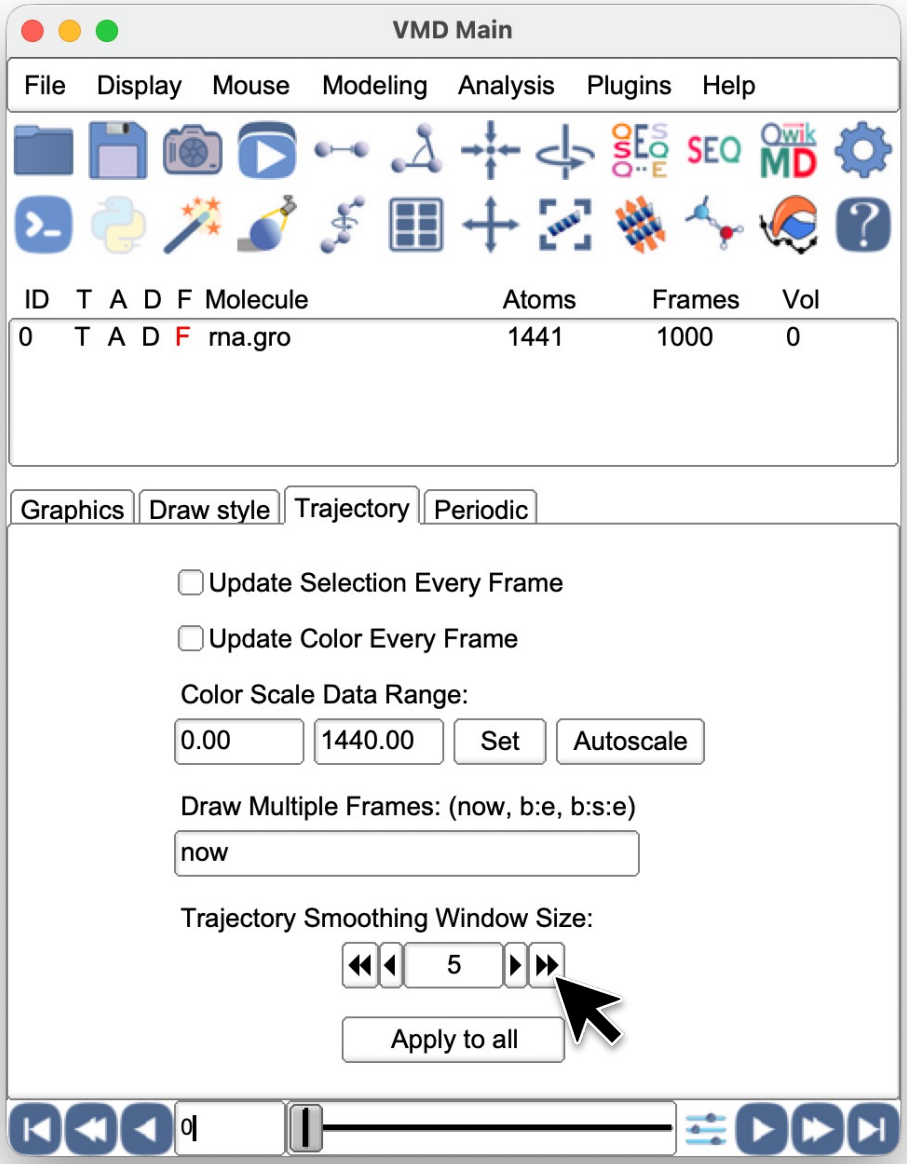
version 2.0.0

Movie making with the GUI

Our results so far…

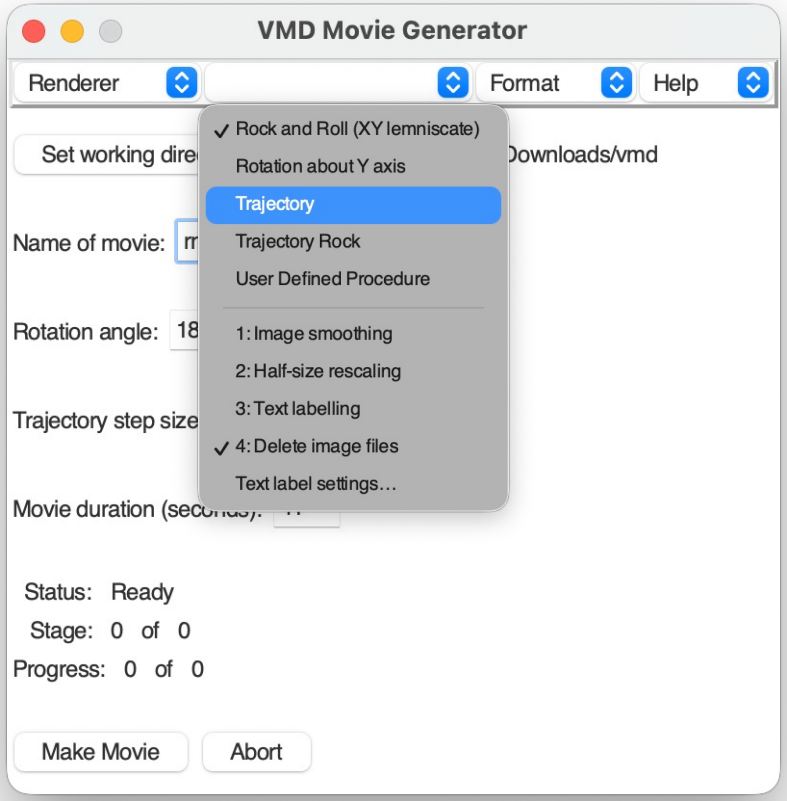
version 2.0.0

Movie making with the GUI



Change `Movie Settings` to `Trajectory`

You can also change the renderer to `Tachyon`, but the rendering time will increase significantly



version 2.0.0

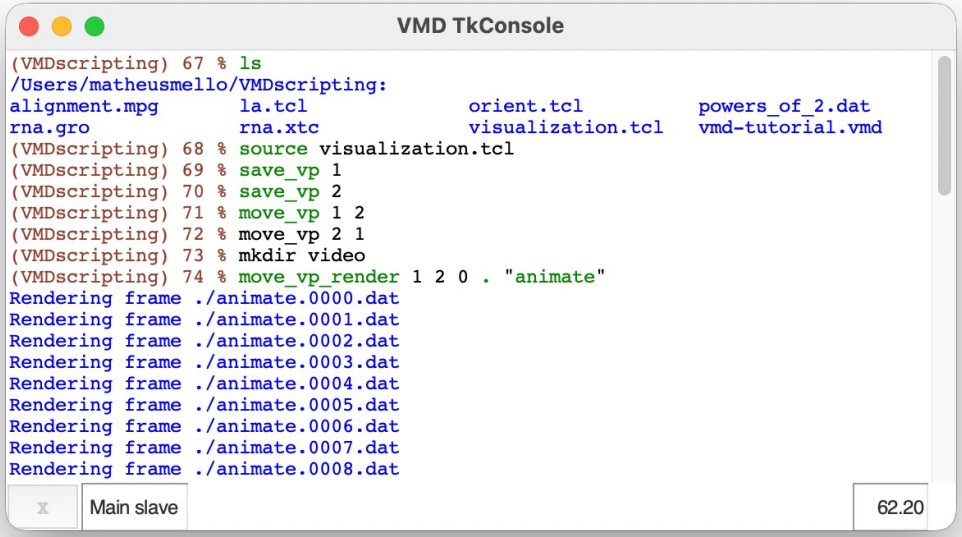
Movie making with the GUI

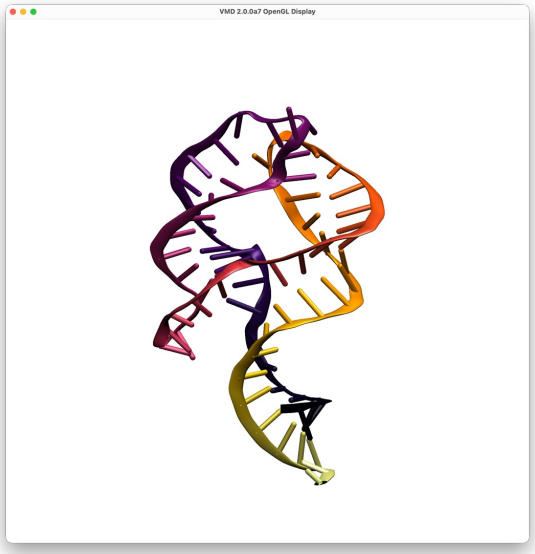
`Smoothing = 5`

`Smoothing = 0`

Movie making with the Tk console

Download the file `visualization.tcl` from the Git Repo.





Viewpoint 1

Viewpoint 2

```
> save_vp vp_num
> move_vp start end morph_frames=50 args="smooth"
> move_vp_render start end first_frame_num directory file_prefix morph_frames=50
```

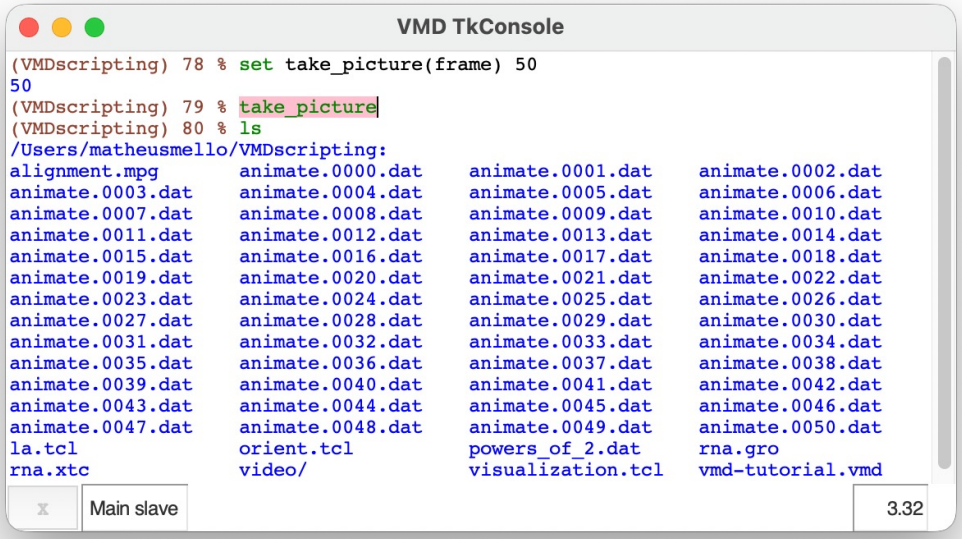
Viewpoint refers to the camera position, orientation and zoom.

https://github.com/Whitford/ctbp-techtalks/tree/main/2025/2.VMDscripting/resources

Movie making with the Tk console

We used `animate` as our prefix because that is the default for the function `take_picture`, which renders the current scene to a `.dat` file with `Tachyon`.

Before continuing our video, we need to set the current frame to 50.



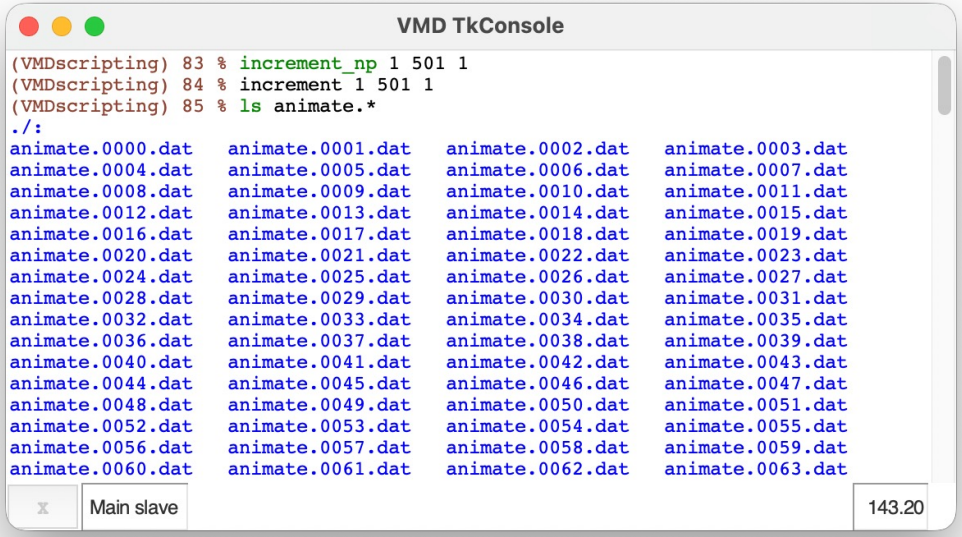
Our movie script:

- *Frames 0-49*: moving between viewpoints *1* and *2*
- *Frame 50*: snapshot of *viewpoint 2* in frame 0

The function `take_picture` automatically updates the current rendered frame when called.

Movie making with the Tk console

Let's observe *viewpoint 2* until frame 500. The function `increment` allows us to move along the trajectory, rendering each step with `take_picture`. The function `increment_np` (`np`=no picture), does the same, without rendering the frames.



Our movie script:

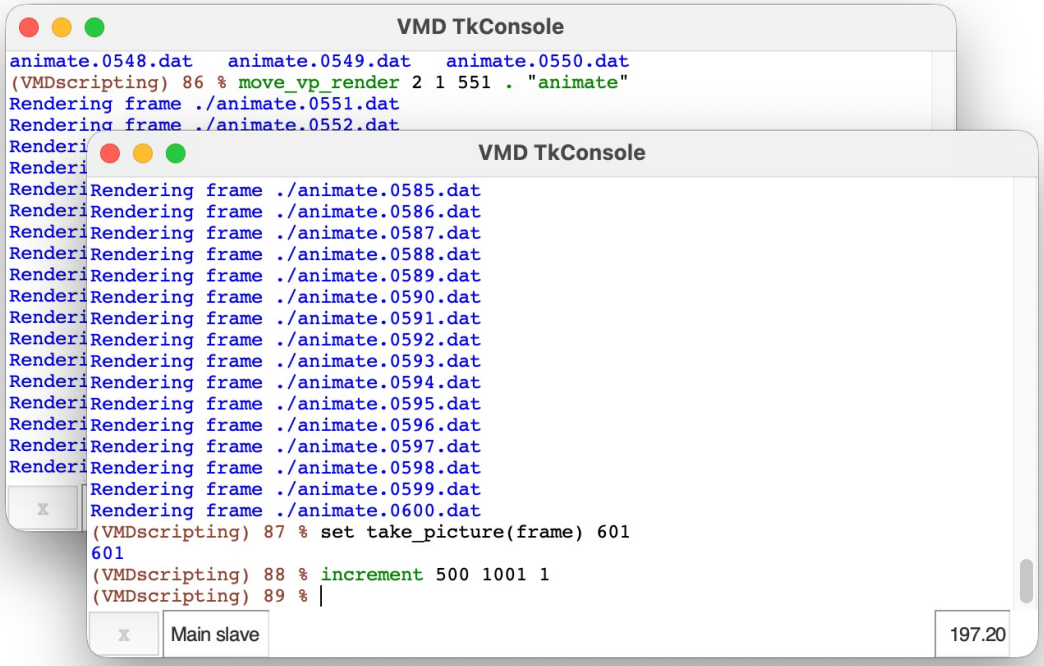
- *Frames 0-49*: moving between viewpoints *1* and *2*
- *Frame 50*: snapshot of *viewpoint 2* in frame 0
- *Frame 51-550*: dynamics of the RNA in *viewpoint 2* until frame 500

```
> increment_np start end step
> increment start end step
```

↳ end frame is not included!

Movie making with the Tk console

To finish our movie, let's return to *viewpoint 1* and evolve the system until the end of the trajectory.



Our movie script:

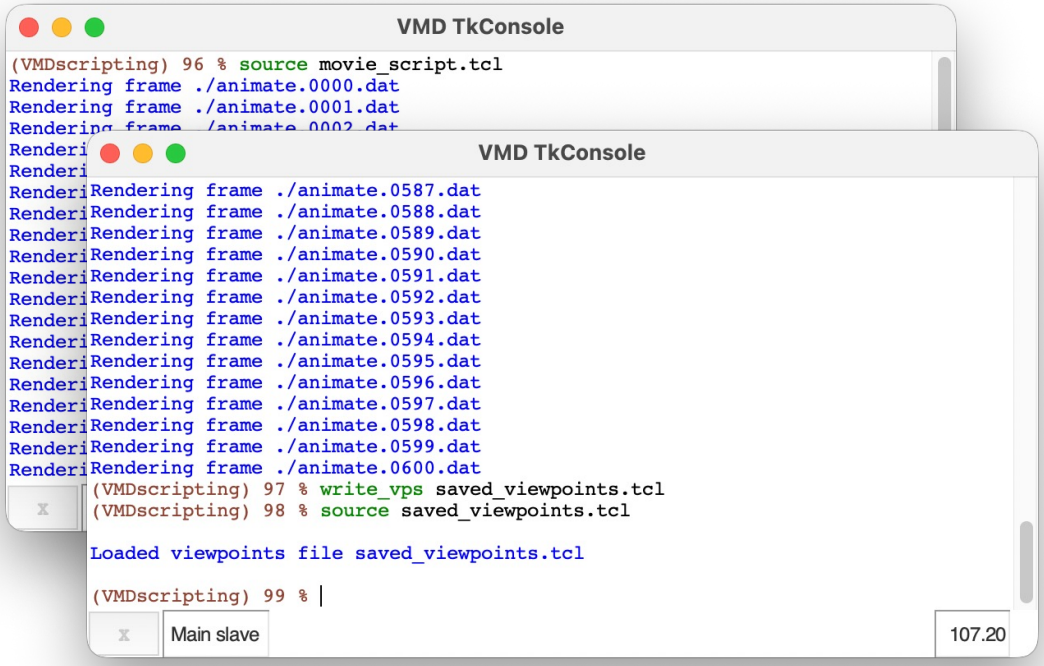
- *Frames 0-49*: moving between viewpoints *1* and *2*
- *Frame 50*: snapshot of *viewpoint 2* in frame 0
- *Frame 51-550*: dynamics of the RNA in *viewpoint 2* until frame 500
- *Frames 551-600*: moving between viewpoints *2* and *1*
- *Frames 601-1101*: dynamics of the RNA in *viewpoint 1* from frame 500 to 1000

```
> increment_np start end step
> increment start end step
```

↳ end frame is not included!

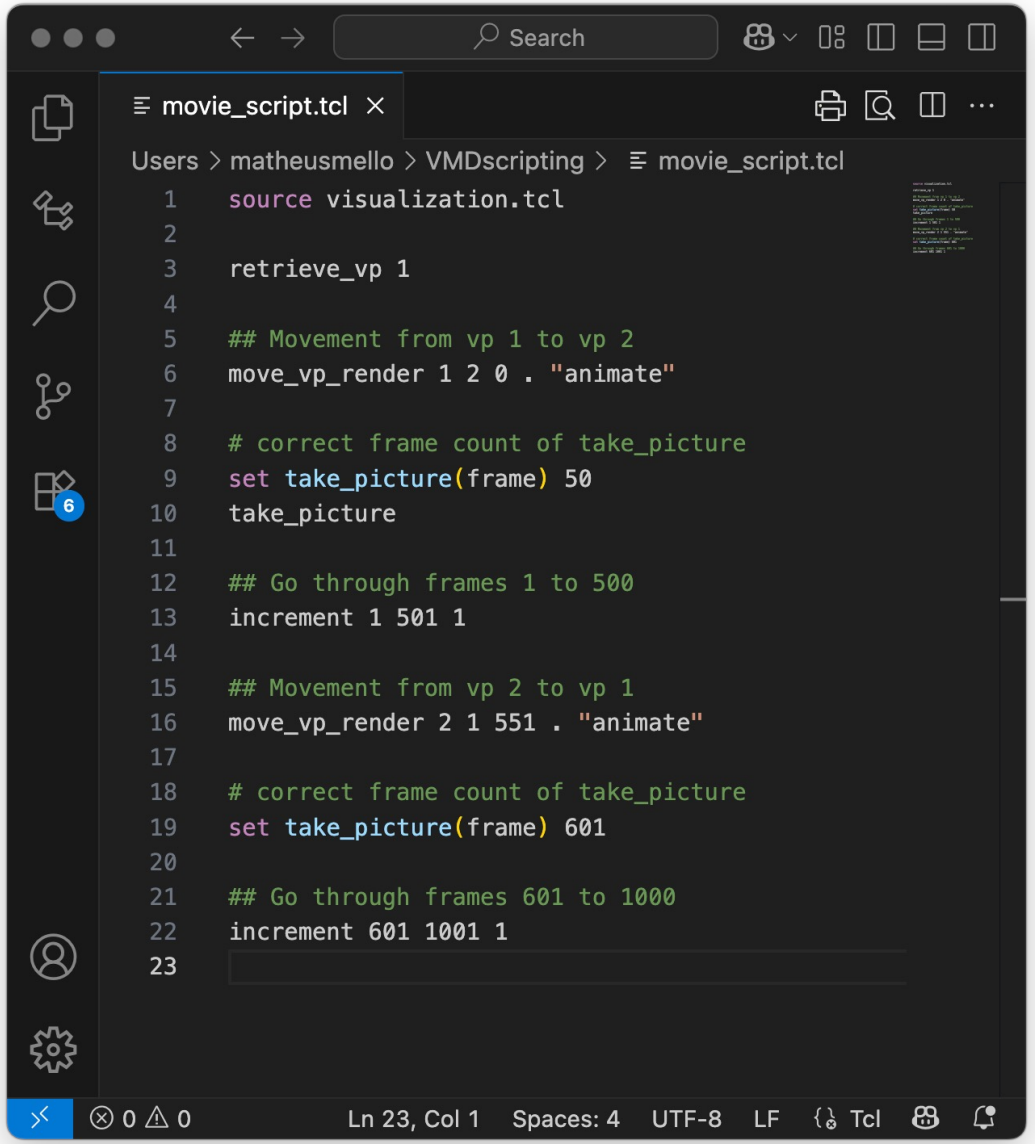
Movie making with the Tk console

We could also have everything in a .tcl script and just source it on the Tk console.



To save and load viewpoints

```
> write_vps filename
> source filename
```



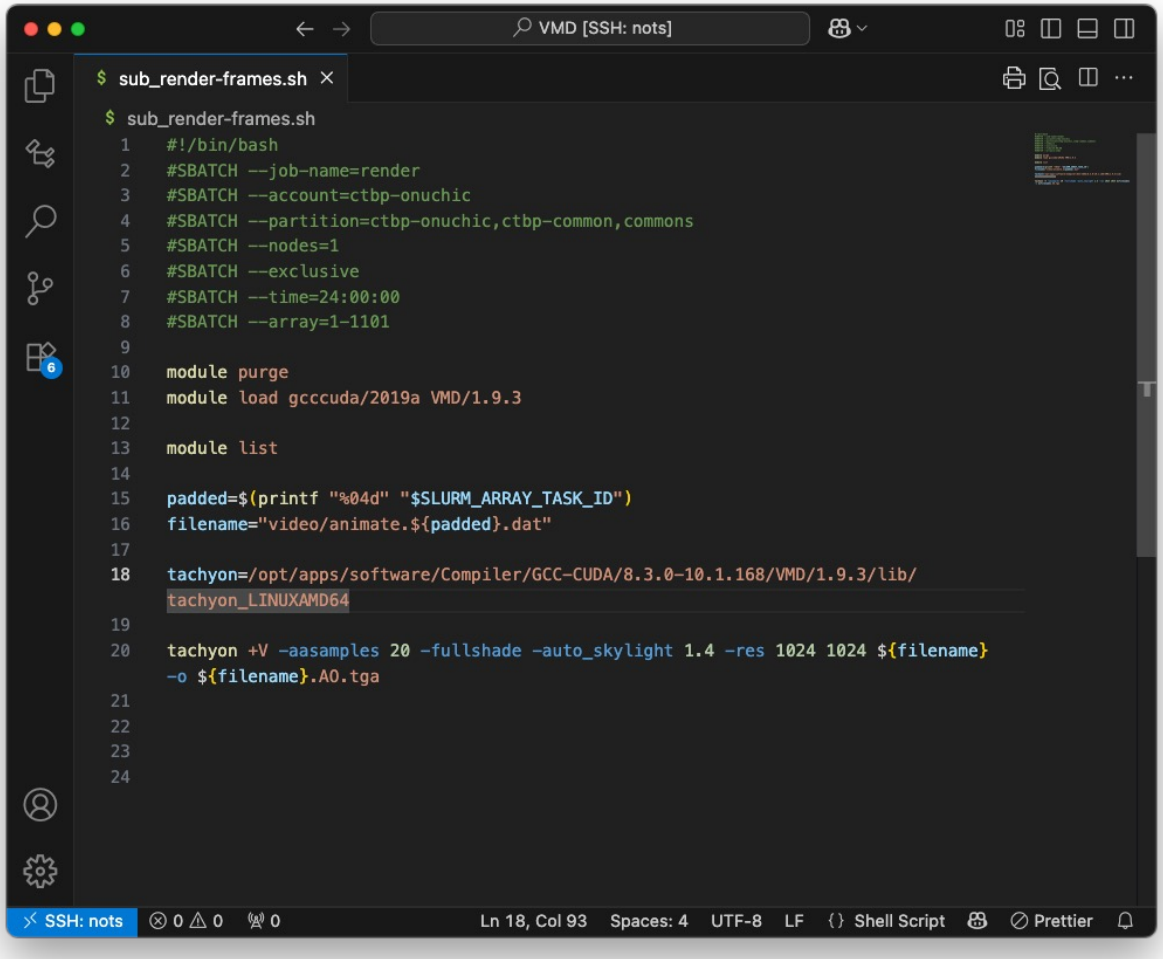
Movie making with the Tk console

Now we have to render the `.dat` files with Tachyon and then assemble the frames together into a video.

```
> mkdir video
> mv *.dat video
> rsync -avP video remote_loc
```

On remote:

```
> sbatch sub_render-frames.sh
```



To assemble the video:

```
> ffmpeg -framerate 60 -i video/animate.%04d.dat.AO.tga -c:v libx264
-crf 10 -pix_fmt yuv420p rna_dynamics.mp4
```

Additional resources

Drawing commands: spheres, cones, cylinders, etc.

`graphics:` https://www.ks.uiuc.edu/Research/vmd/current/ug/node129.html
`draw:` https://www.ks.uiuc.edu/Research/vmd/current/ug/node127.html

Advanced coloring methods on `visualization.tcl`.

Functions: `colorByResid`, `colorByFile`
think coloring each frame by contacts per residue

Function `enabletrace` in `update_per_frame.tcl`

Performs calculations on current frame as display changes
think drawing a dipole moment as the frames change

VMD efficient functions for vector and matrix manipulations

https://www.ks.uiuc.edu/Research/vmd/current/ug/node192.html

https://github.com/Whitford/ctbp-techtalks/tree/main/2025/2.VMDscripting/resources