



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



The graphics suite



Rendering videos
(like a pro)

Why use scripting in VMD?



Avoid performing numerous manual operations in the GUI



Expand VMD capabilities



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.

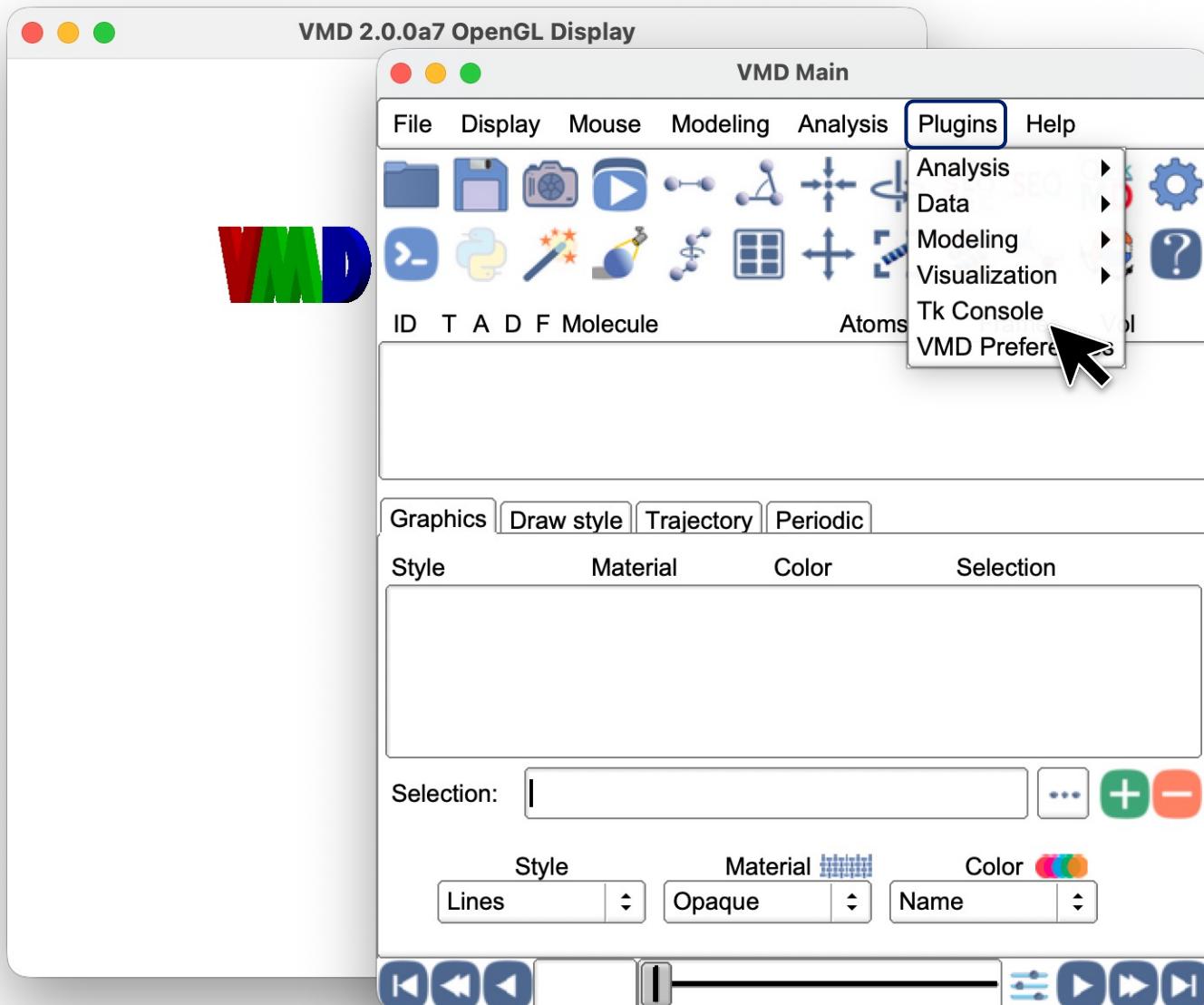


Bug reports, comments, and suggestions can be sent to
vmd@ks.uiuc.edu



Using previous VMD versions will not affect your ability to participate

Opening VMD and the Tk console



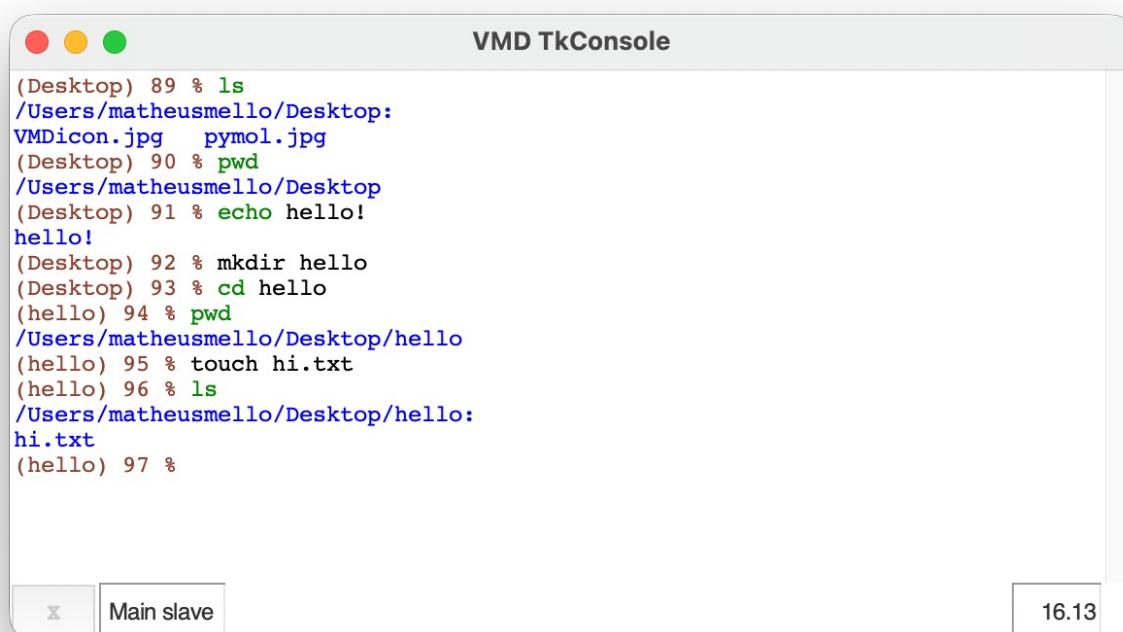
```
Last login: Wed Oct  8 10:52:45 on ttys000
/Applications/VMD\ 2.0.app/Contents/MacOS/startup.command ; exit;
(base) > ~ /Applications/VMD\ 2.0.app/Contents/MacOS/startup.command ; exit;
Info) VMD for MACOSXARM64, version 2.0.0a7 (July 31, 2025)
Info) http://www.ks.uiuc.edu/Research/vmd/
Info) Email questions and bug reports to vmd@ks.uiuc.edu
Info) Please include this reference in published work using VMD:
Info) Humphrey, W., Dalke, A. and Schulten, K., 'VMD - Visual
Info) Molecular Dynamics', J. Molec. Graphics 1996, 14.1, 33-38.
Info) -----
Info) Multithreading available, 12 CPUs.
Info) High-DPI OpenGL display support enabled.
Info) OpenGL renderer: Apple M2 Max
Info) Features: STENCIL MDE MTX NPOT PP PS GLSL(OVF)
Info) Full GLSL rendering mode is available.
Info) Textures: 2-D (16384x16384), 3-D (2048x2048x2048), Multitexture (8)
ERROR) Duplicate resname "4YS"
ERROR) Duplicate resname "YGA"
after#0
vmd >
```

```
loading history file ... 48 events added
buffer line limit: 512  max line length: unlimited
Main console display active (Tcl8.6.16 / Tk8.6.16)
(matheusmello) 49 %
```

version 2.0.0

The Tk console

The Tk console is a Tcl interpreter



A screenshot of the VMD TkConsole window. The title bar says "VMD TkConsole". The window contains a command-line session:

```
(Desktop) 89 % ls  
/Users/matheusmello/Desktop:  
VMDIcon.jpg pymol.jpg  
(Desktop) 90 % pwd  
/Users/matheusmello/Desktop  
(Desktop) 91 % echo hello!  
hello!  
(Desktop) 92 % mkdir hello  
(Desktop) 93 % cd hello  
(hello) 94 % pwd  
/Users/matheusmello/Desktop/hello  
(hello) 95 % touch hi.txt  
(hello) 96 % ls  
/Users/matheusmello/Desktop/hello:  
hi.txt  
(hello) 97 %
```

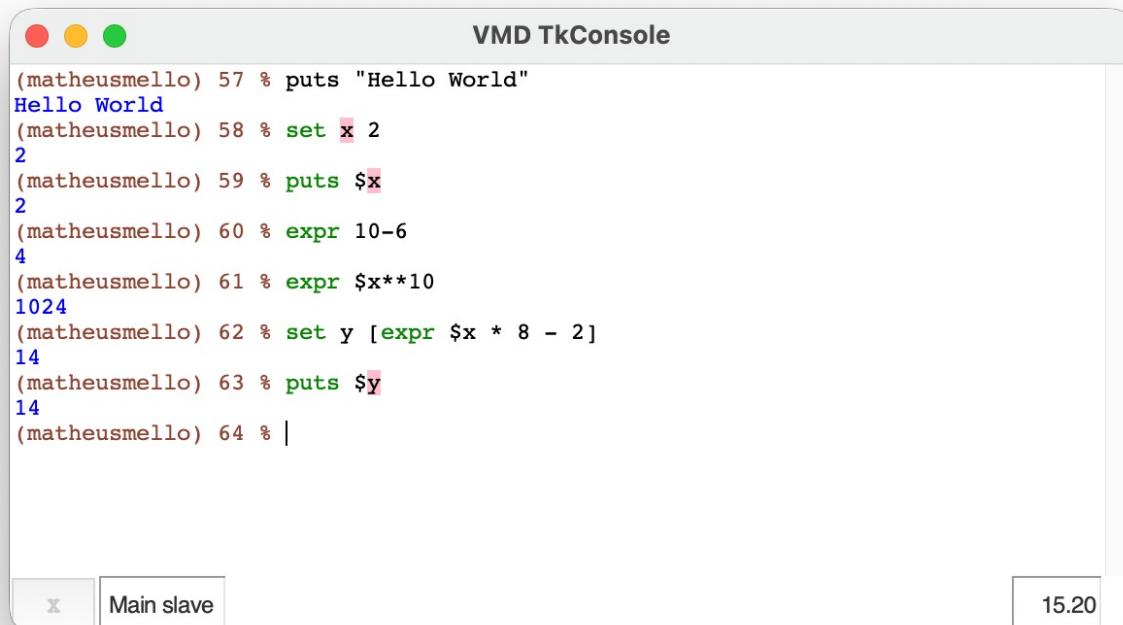
The bottom left corner shows "Main slave" and the bottom right corner shows "16.13".

Some Bash commands that still work:

ls	rm
cd	mkdir
pwd	touch
echo	clear

The Tk console

The Tk console is a Tcl interpreter



A screenshot of the VMD TkConsole window. The title bar says "VMD TkConsole". The console area contains the following Tcl session:

```
(matheusmello) 57 % puts "Hello World"
Hello World
(matheusmello) 58 % set x 2
2
(matheusmello) 59 % puts $x
2
(matheusmello) 60 % expr 10-6
4
(matheusmello) 61 % expr $x**10
1024
(matheusmello) 62 % set y [expr $x * 8 - 2]
14
(matheusmello) 63 % puts $y
14
(matheusmello) 64 % |
```

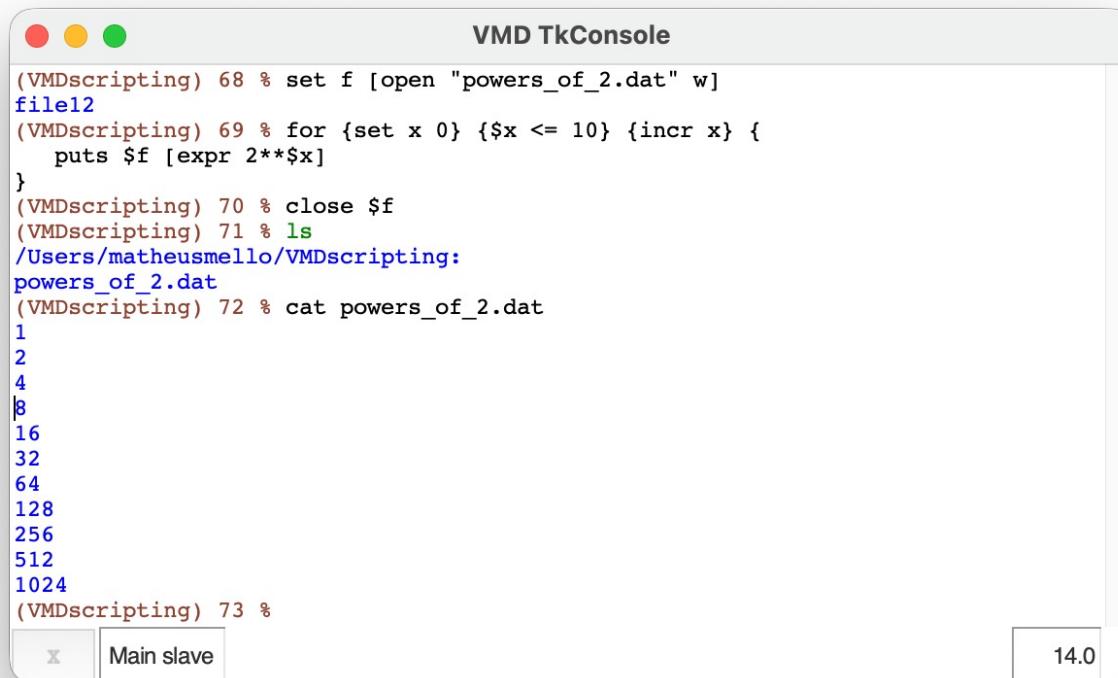
At the bottom left, there are buttons for "Main slave" and "X". At the bottom right, it says "15.20".

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

The Tk console

The Tk console is a Tcl interpreter



A screenshot of the VMD TkConsole window. The title bar says "VMD TkConsole". The console area contains the following Tcl script:

```
(VMDscripting) 68 % set f [open "powers_of_2.dat" w]
file12
(VMDscripting) 69 % for {set x 0} {$x <= 10} {incr x} {
    puts $f [expr 2**$x]
}
(VMDscripting) 70 % close $f
(VMDscripting) 71 % ls
/Users/matheusmello/VMDscripting:
powers_of_2.dat
(VMDscripting) 72 % cat powers_of_2.dat
1
2
4
8
16
32
64
128
256
512
1024
(VMDscripting) 73 %
```

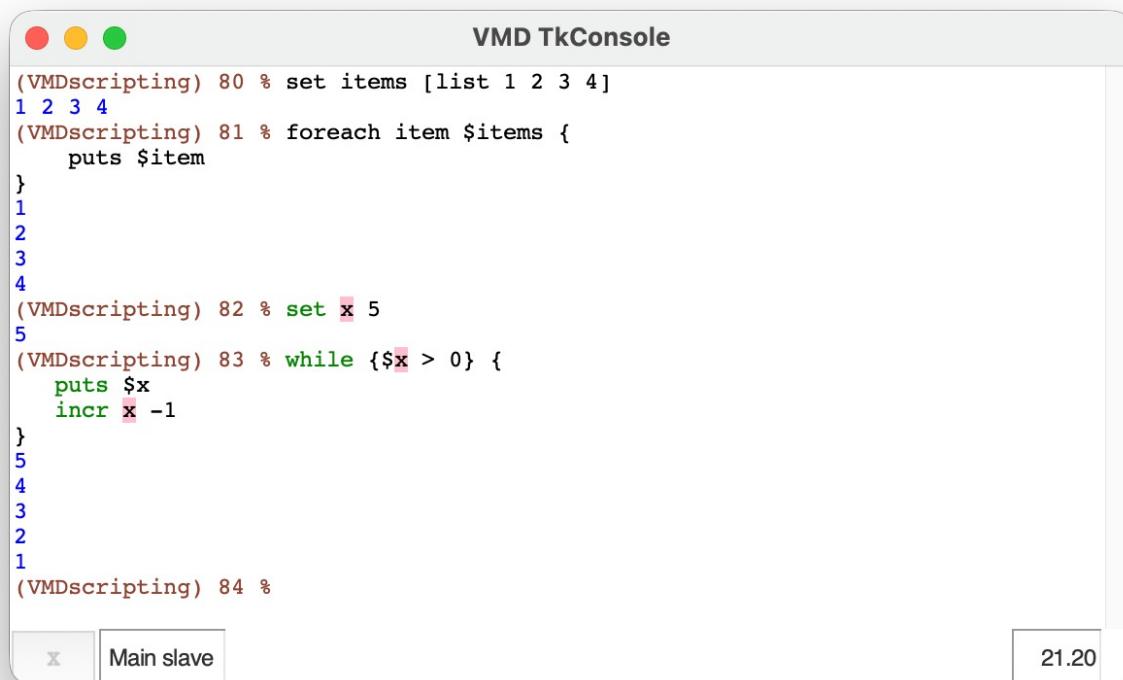
The status bar at the bottom shows "Main slave" and "14.0".

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

The Tk console

The Tk console is a Tcl interpreter



A screenshot of the VMD TkConsole window. The title bar says "VMD TkConsole". The main area contains the following Tcl script:

```
(VMDscripting) 80 % set items [list 1 2 3 4]
1 2 3 4
(VMDscripting) 81 % foreach item $items {
    puts $item
}
1
2
3
4
(VMDscripting) 82 % set x 5
5
(VMDscripting) 83 % while {$x > 0} {
    puts $x
    incr x -1
}
5
4
3
2
1
(VMDscripting) 84 %
```

The status bar at the bottom right shows "21.20".

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

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



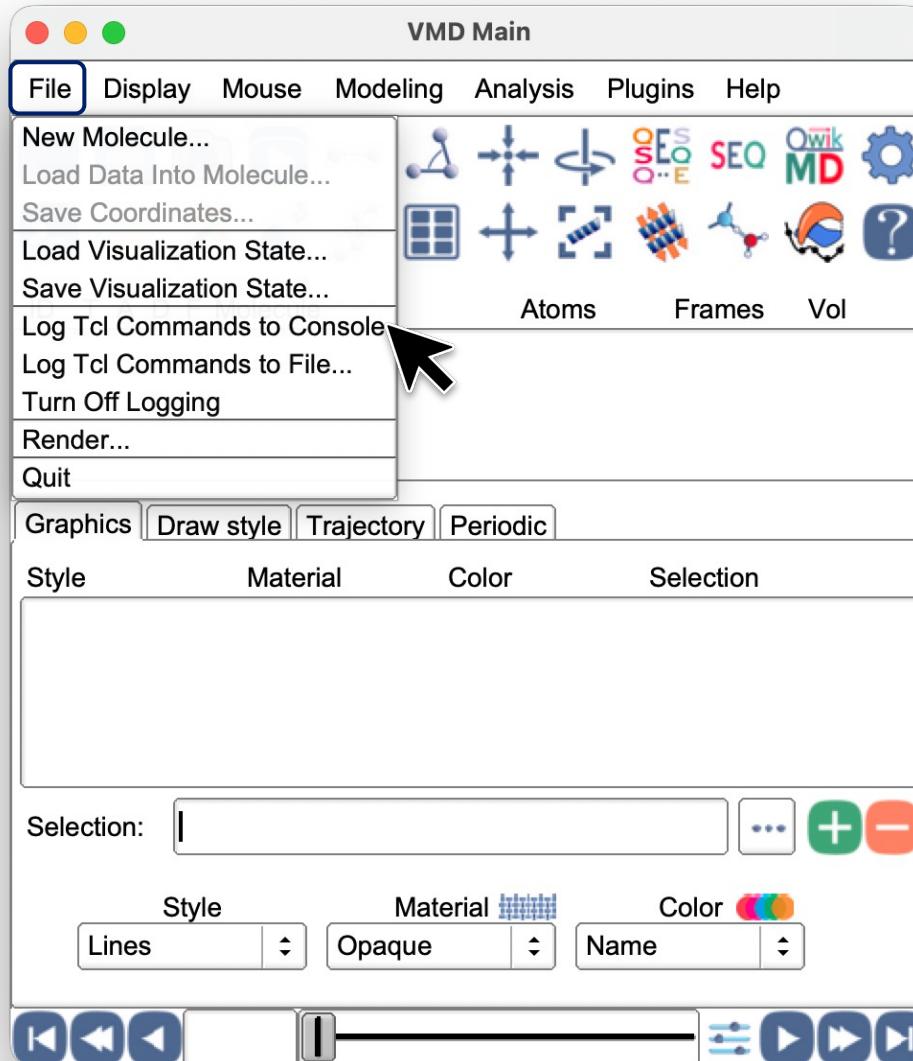
render

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 molecular 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



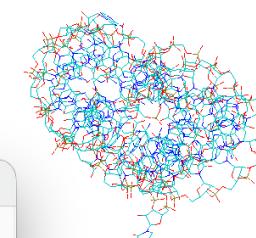
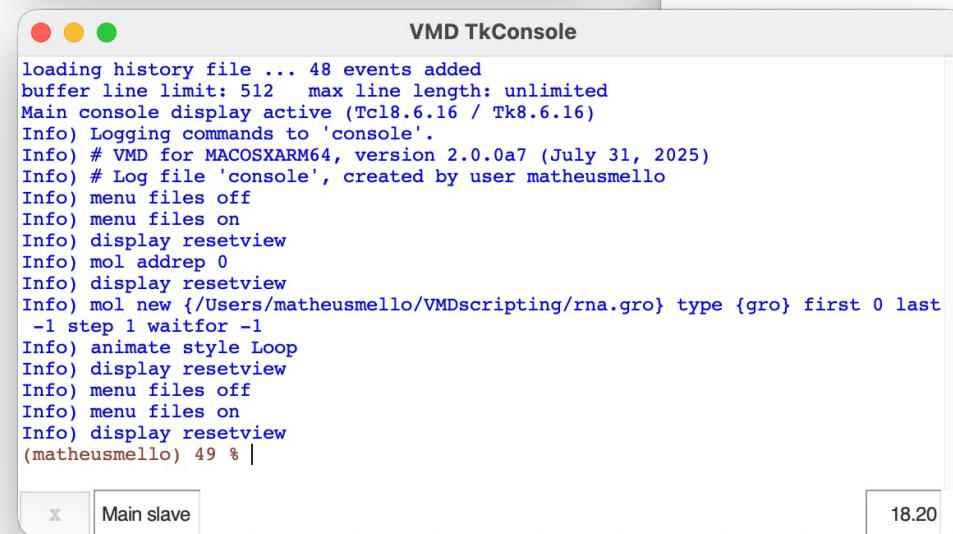
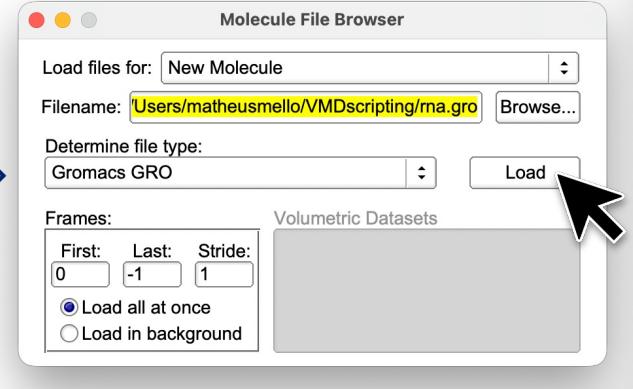
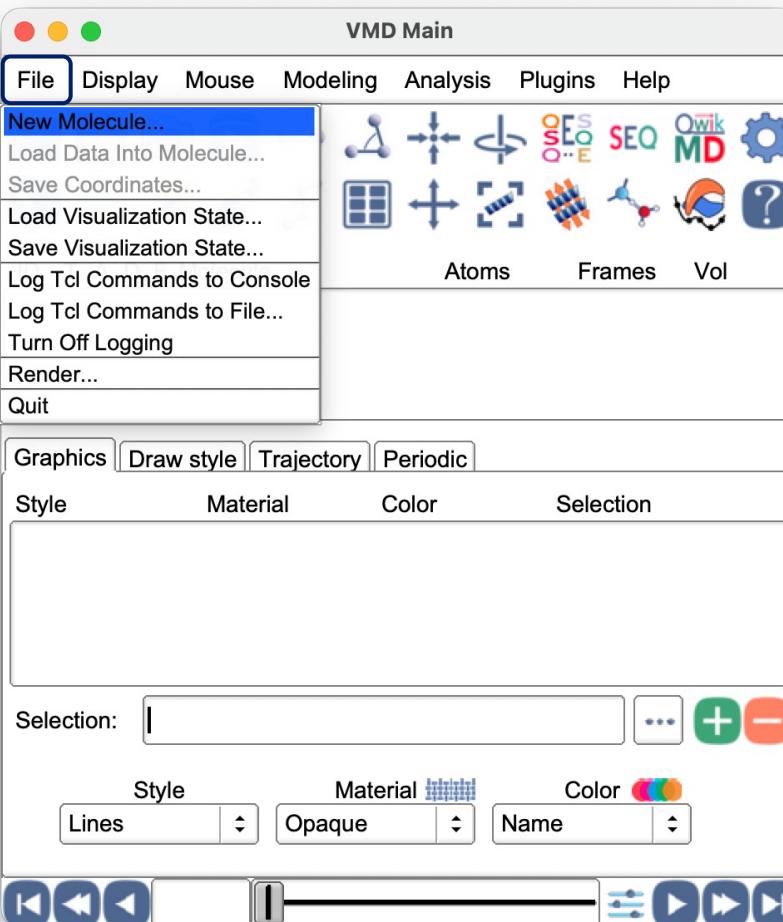
The VMD TkConsole window shows the following log output:

```
loading history file ... 48 events added
buffer line limit: 512 max line length: unlimited
Main console display active (Tcl8.6.16 / Tk8.6.16)
Info) Logging commands to 'console'.
Info) # VMD for MACOSXARM64, version 2.0.0a7 (July 31, 2025)
Info) # Log file 'console', created by user matheusmello
(matheusmello) 49 %
```

The window also shows 'Main slave' and '7.20' in the bottom right corner.

Discovering commands from the VMD GUI

Load RNA model (rna.gro)



Each performed command is logged in the TkConsole

Discovering commands from the VMD GUI

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

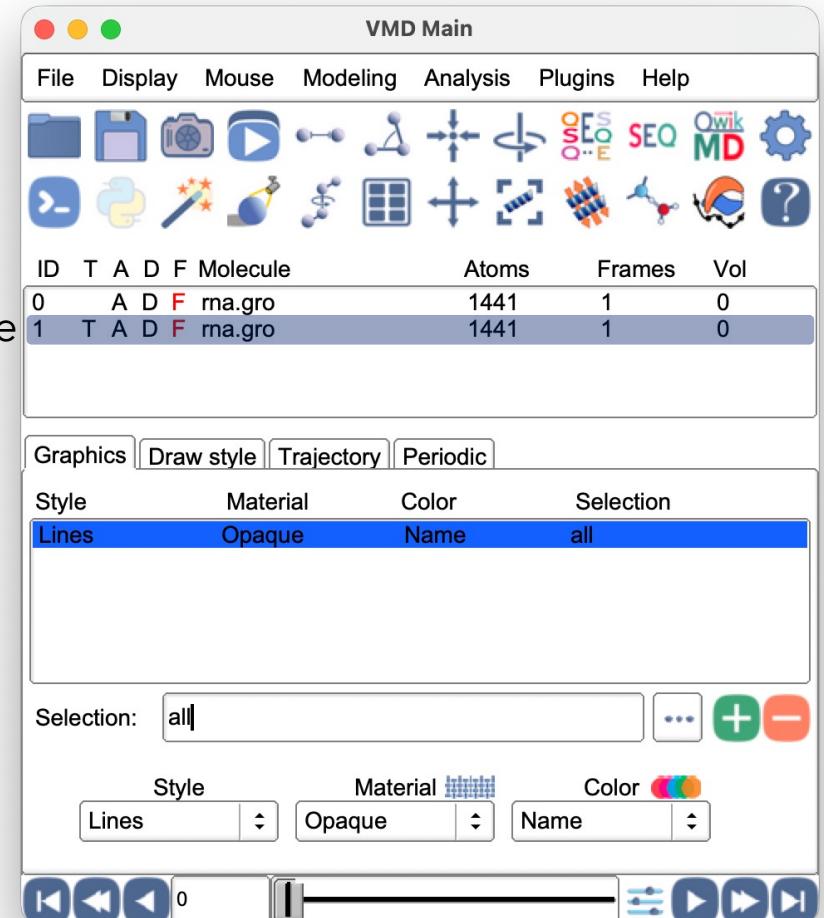
VMD TkConsole

```
loading history file ... 48 events added
buffer line limit: 512  max line length: unlimited
Main console display active (Tcl8.6.16 / Tk8.6.16)
Info) Logging commands to 'console'.
Info) # VMD for MACOSXARM64, version 2.0.0a7 (July 31, 2025)
Info) # Log file 'console', created by user matheusmello
Info) menu files off
Info) menu files on
Info) display resetview
Info) mol addrep 0
Info) display resetview
Info) mol new{/Users/matheusmello/VMDScripting/rna.gro} type {gro} first 0 last
-1 step 1 waitfor -1
Info) animate style Loop
Info) display resetview
Info) menu files off
Info) menu files on
Info) display resetview
(matheusmello) 49 % mol new{/Users/matheusmello/VMDScripting/rna.gro} type {gro}
} first 0 last -1 step 1 waitfor -1
```

Main slave 12.101

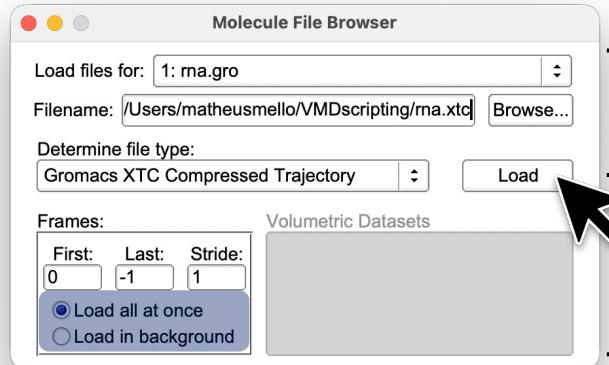
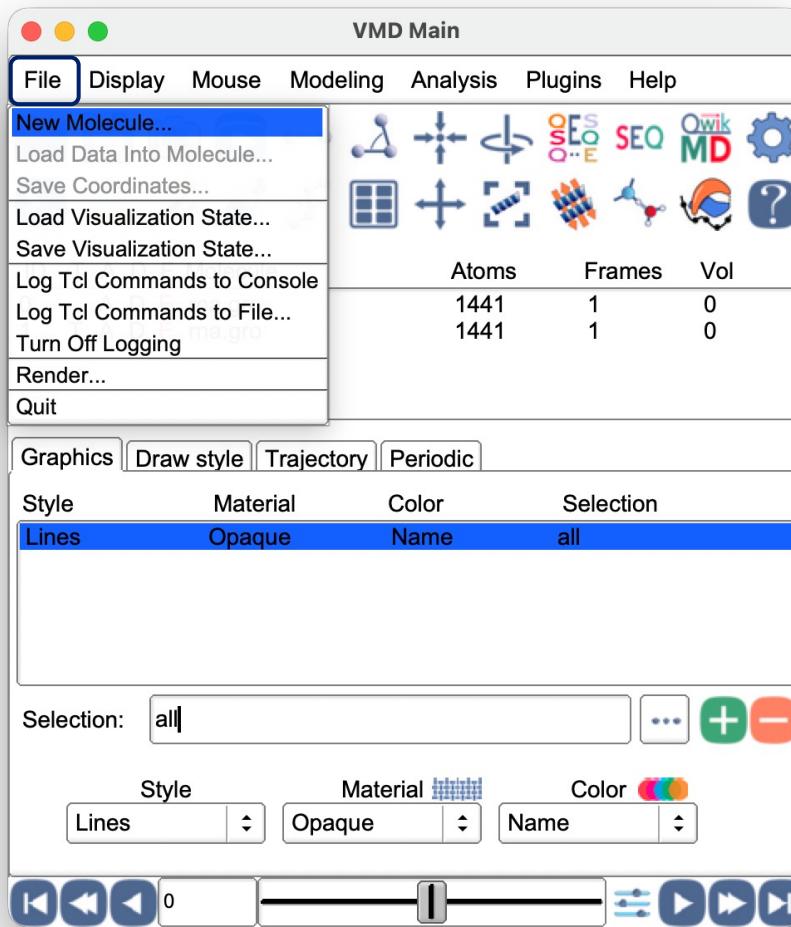
Copy and paste mol command

new molecule



Managing trajectories

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



} The trajectory file is loaded on top of the topology file

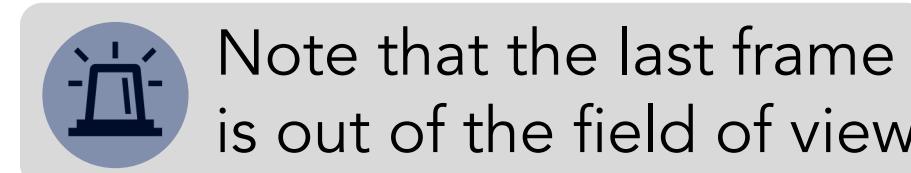
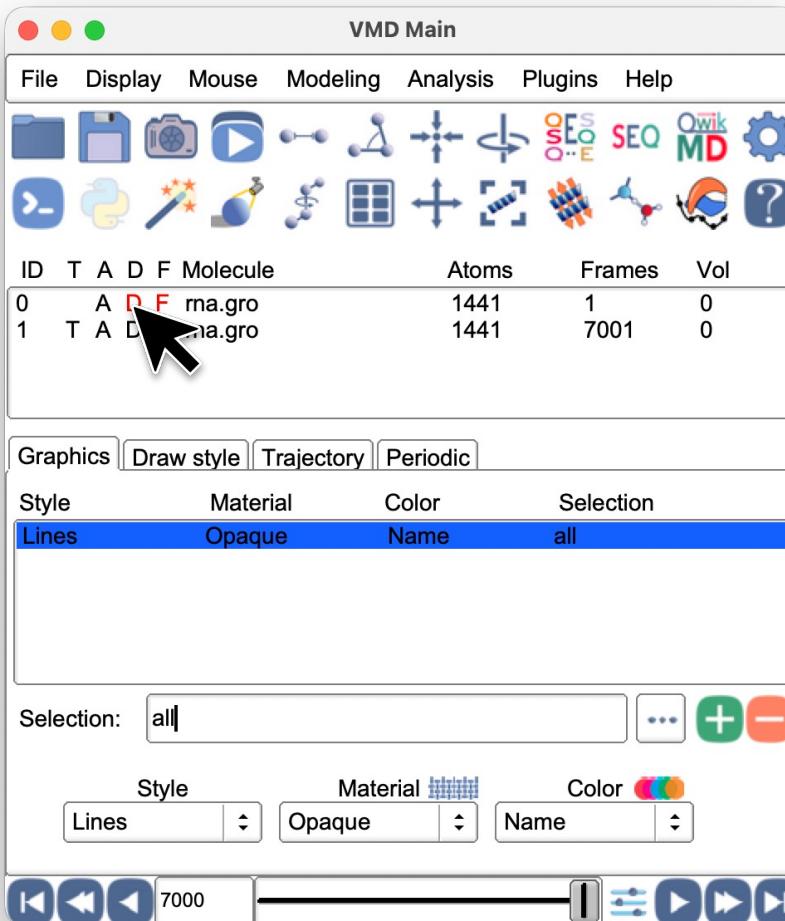
Load all at once loads faster,
especially for larger systems

The screenshot shows the VMD TkConsole window. It displays a command history in blue text:
Info) animate style Loop
Info) display resetview
Info) menu files off
Info) menu files on
Info) display resetview
(matheusmello) 49 % mol new{/Users/matheusmello/VMDscripting/rna.gro} type {gro}
} first 0 last -1 step 1 waitfor -1
Info) display resetview
Info) mol addrep 1
Info) display resetview
Info) mol new{/Users/matheusmello/VMDscripting/rna.gro} type {gro} first 0 last
-1 step 1 waitfor -1
Info) animate style Loop
1
Info) menu files off
Info) menu files on
Info) mol addfile{/Users/matheusmello/VMDscripting/rna.xtc} type {xtc} first 0
last -1 step 1 waitfor -1
Info) animate style Loop
(matheusmello) 50 % |
The console also shows the status 'Main slave' and the number '25.20'.

} Notice the new commands

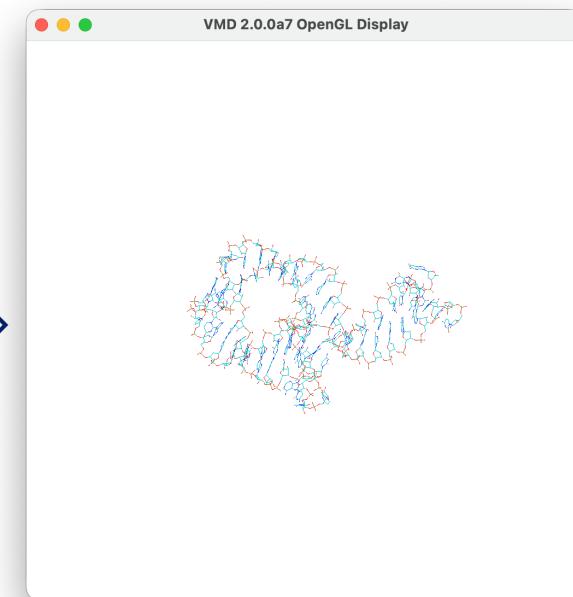
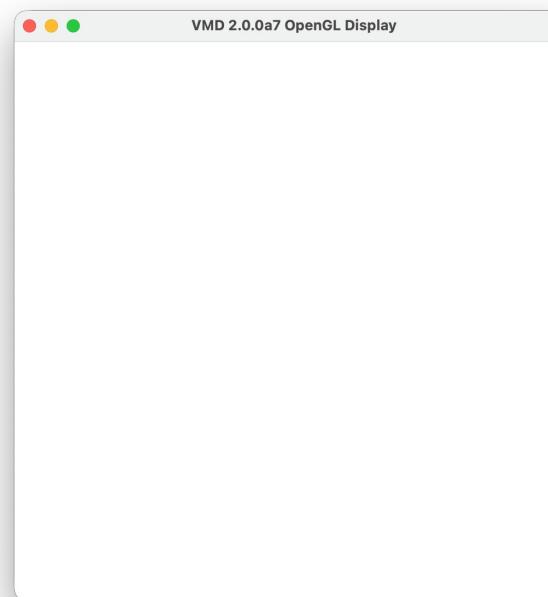
Managing trajectories

Let's hide the first molecule for now...



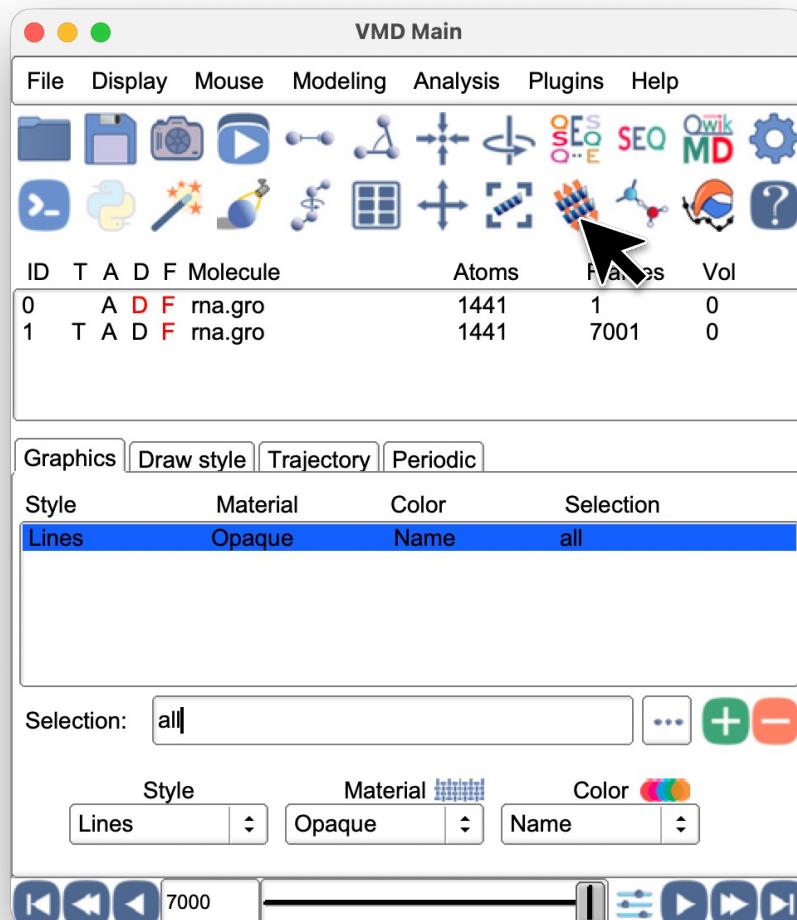
Solutions:

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

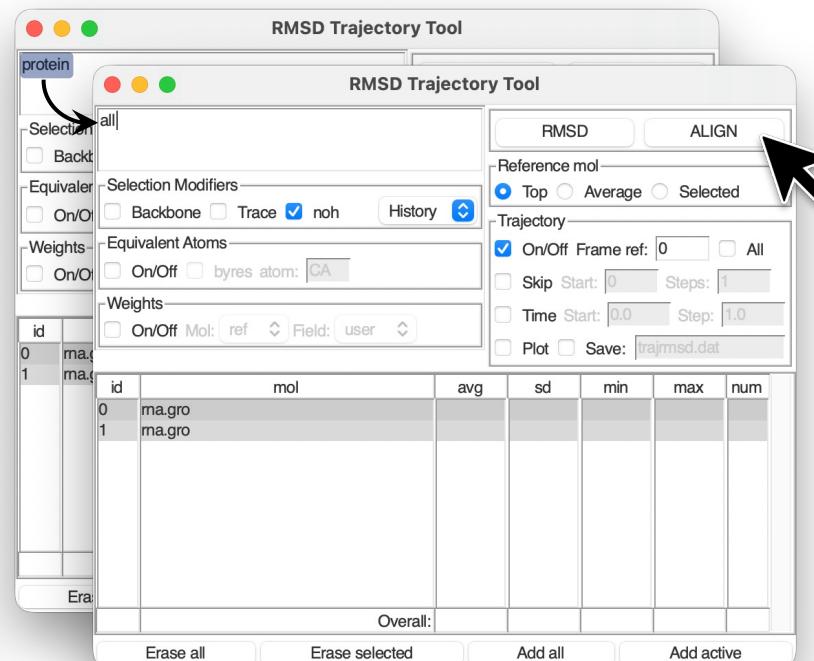


Aligning trajectories

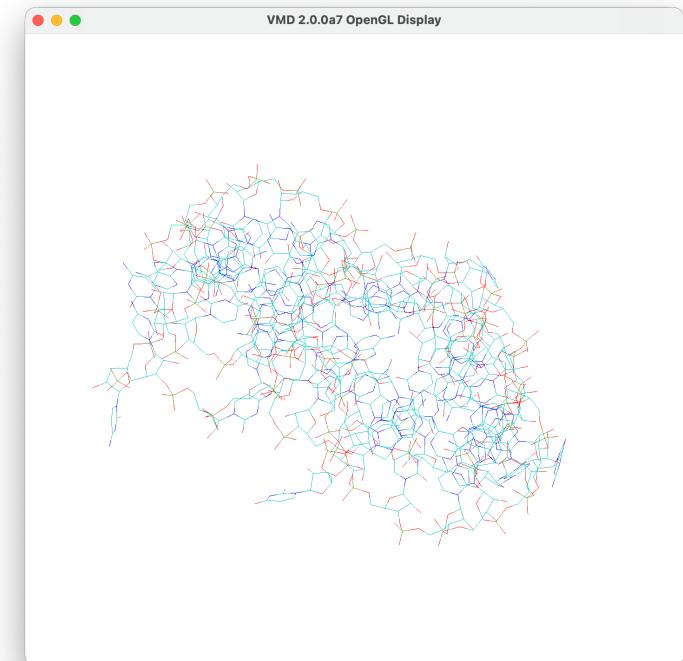
Open RMSD Trajectory tool



Change selection and align



Reset view (=)

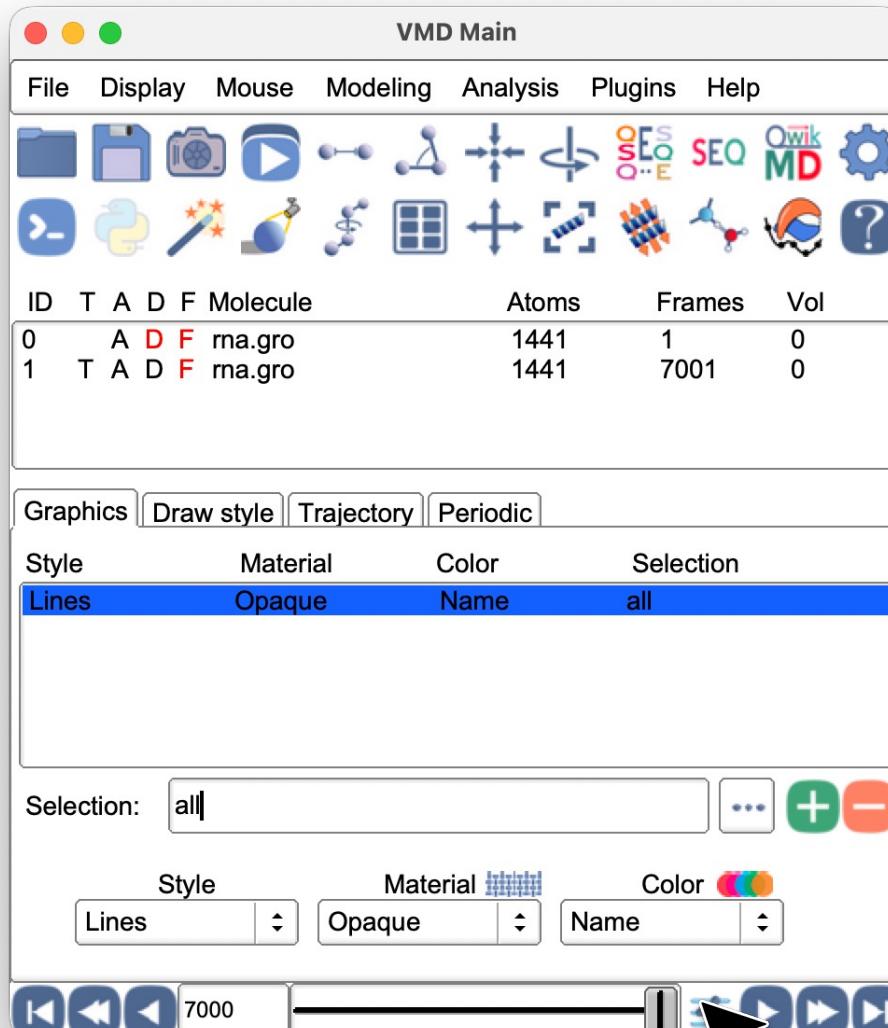


Try a custom selection and explore a representation that you like!

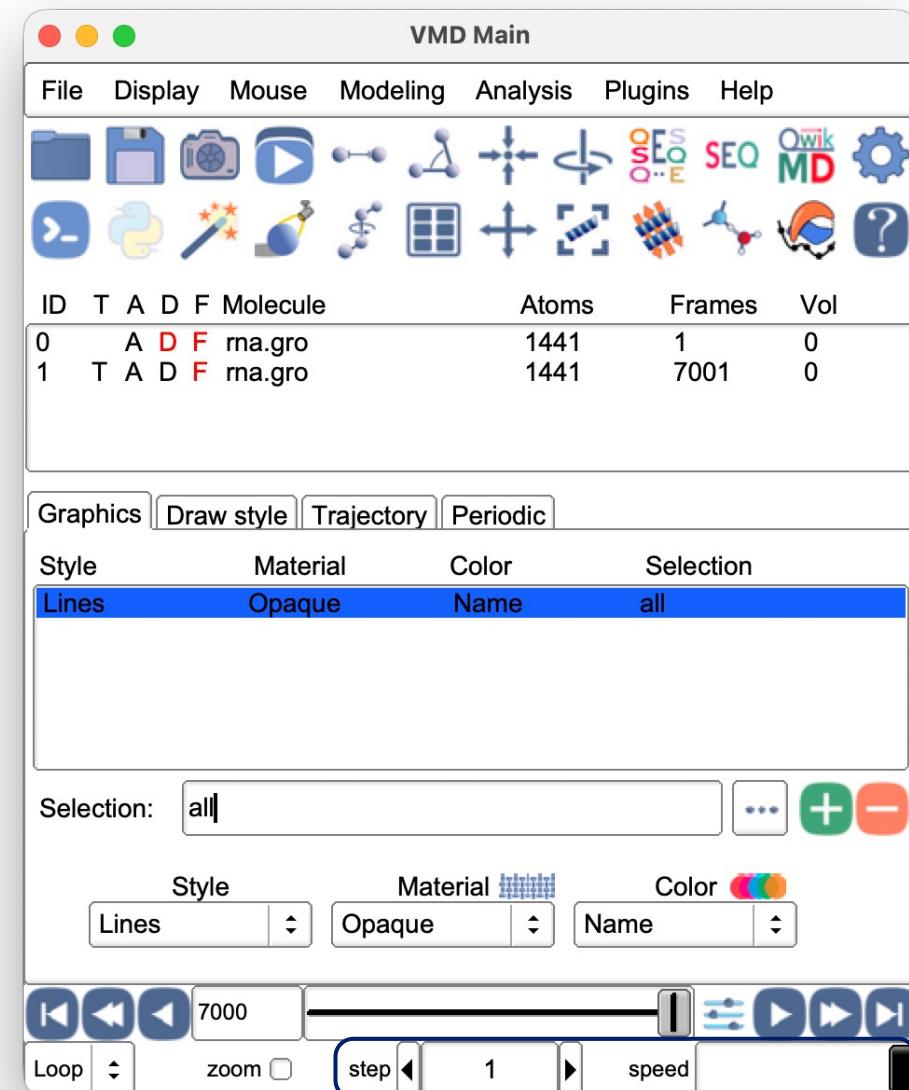
Alternatively : Extensions > Analysis > RMSD Trajectory Tool

version 2.0.0

Managing trajectories

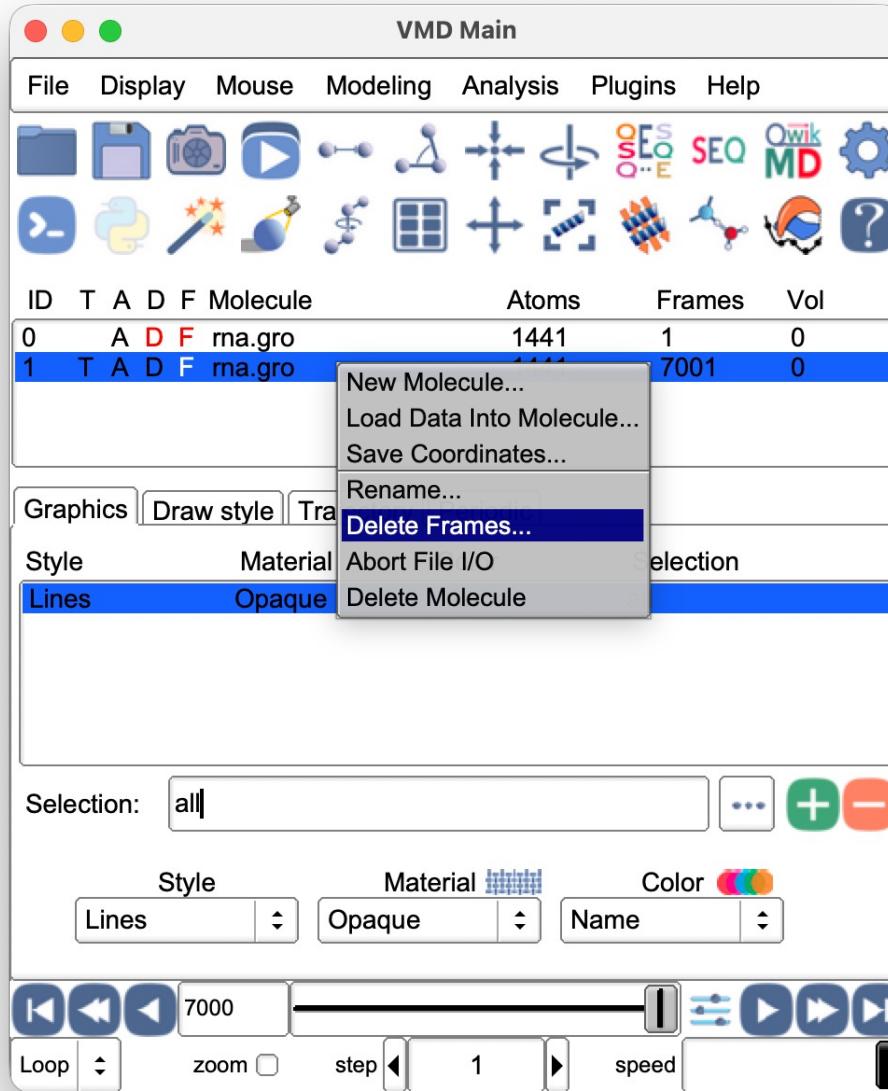


Skip to last frame
Next frame
Play forward

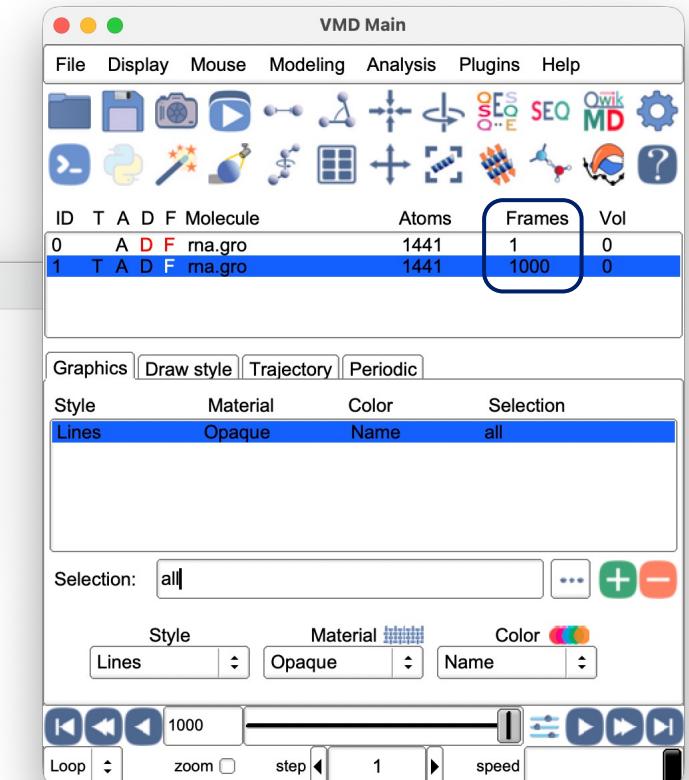
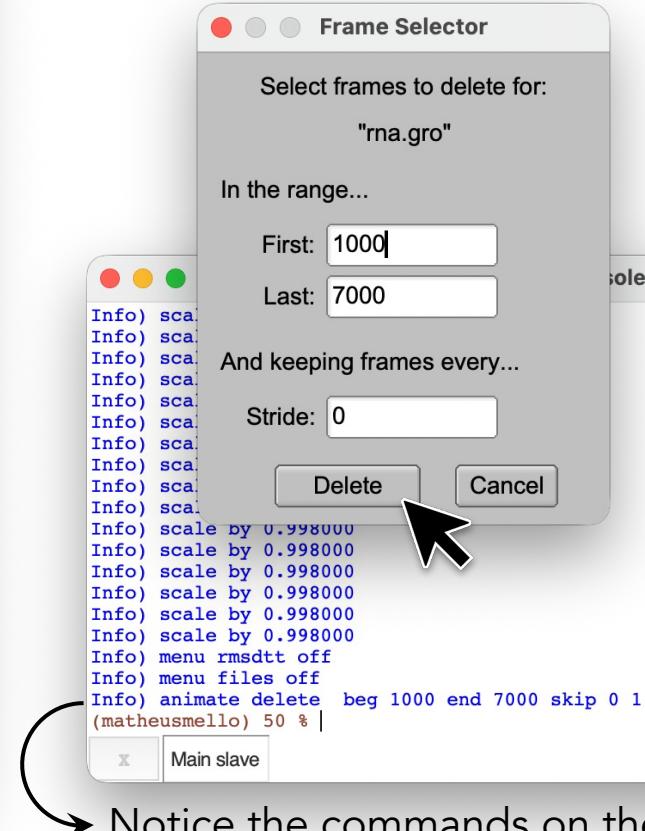


version 2.0.0

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)

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 % |
```

VMD Main

Main slave 7.20

File Display Mouse Modeling Analysis Plugins Help

ID T A D F Molecule Atoms Frames Vol

0	A	D	F	rna.gro	1441	1	0	
1	T	A	D	F	rna.gro	1441	1000	0

Aligning two different molecules

The atomselect object has some properties

The atomselect can also alter the selection

The figure consists of two side-by-side VMD OpenGL displays. Both displays show a complex arrangement of DNA molecules represented as ribbon-like structures. The left display has a pink label 'simulation' and a cyan label 'crystal'. The right display has a pink label 'simulation' and a cyan label 'crystal'. Above the left display is a VMD TkConsole window showing command-line input and output related to moving a crystal.

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

Aligning two different molecules

Now we compute the transformation matrix

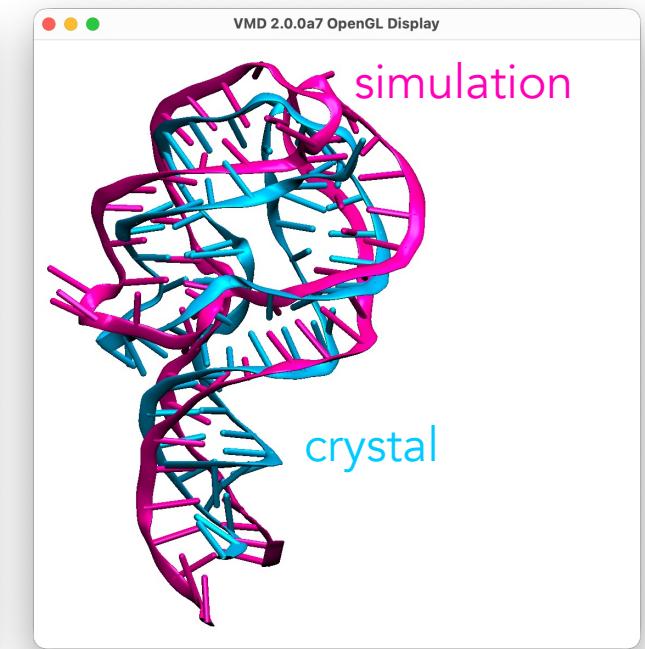
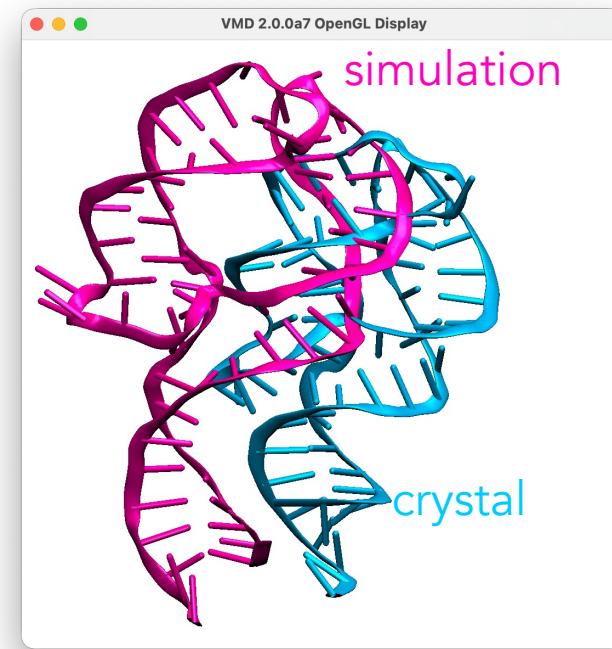
```
> measure fit atomsel1 atomsel2
```

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

VMD TkConsole

```
(matheusmello) 63 % set trans_matrix [measure fit $crystal $simulation]
{1.0 -2.873695947869237e-8 4.456467905811223e-8 -20.0} {2.873695770233553e-8 1.0
2.2902424845483438e-8 -1.52587890625e-5} {-4.456467905811223e-8 -2.290242484548
3438e-8 1.0 3.30178918375168e-5} {0.0 0.0 0.0 1.0}
(matheusmello) 64 % $crystal move $trans_matrix
(matheusmello) 65 % |
```

Main slave 4.20



Aligning two different molecules

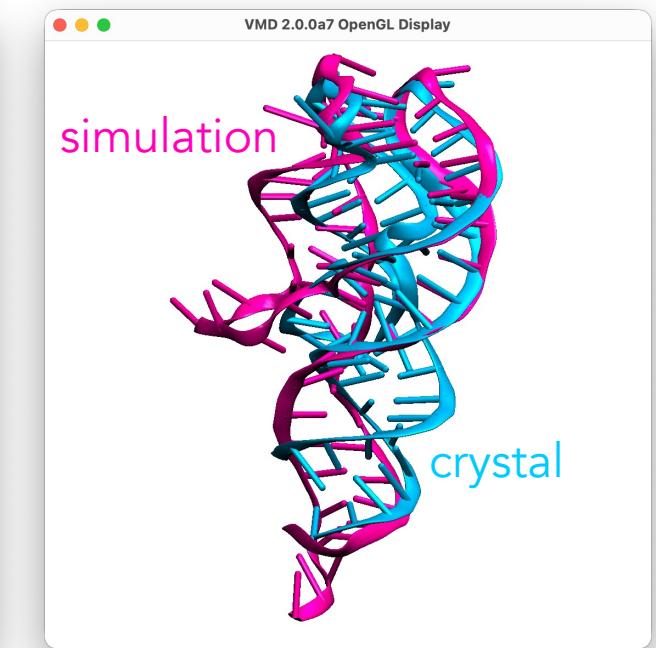
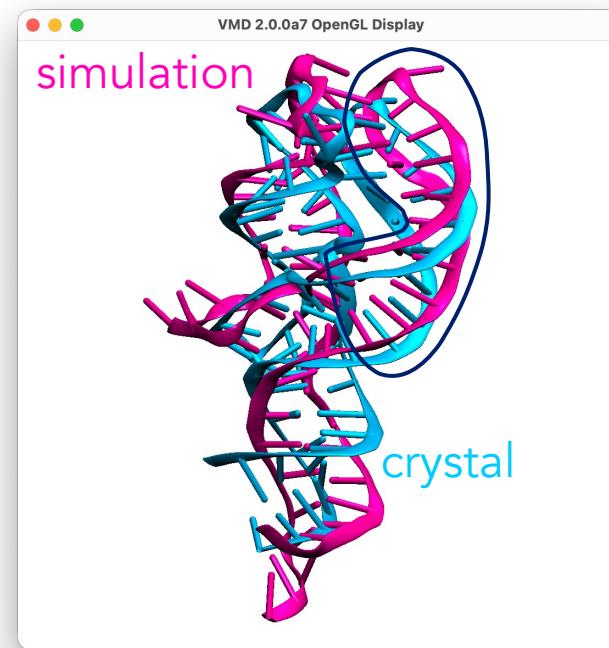
This procedure works for any two selections that contain the same number of atoms

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

VMD TkConsole

```
(matheusmello) 77 % set loop_crystal [atomselect 0 "resid 40 to 60"]
atomselect7011
(matheusmello) 78 % set loop_sim [atomselect 1 "resid 40 to 60"]
atomselect7012
(matheusmello) 79 % set M [measure fit $loop_crystal $loop_sim]
{1.0 -9.201023232208172e-8 -5.784955803278535e-8 3.0517578125e-5} {9.20102252166
5436e-8 1.0 -8.840941490007026e-8 0.0} {5.784956513821271e-8 8.84094077946429e-8
1.0 -3.665703843580559e-5} {0.0 0.0 0.0 1.0}
(matheusmello) 80 % $crystal move $M
(matheusmello) 81 %
```

Main slave 8.20



This approach aligns only the current frame shown on display!

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.

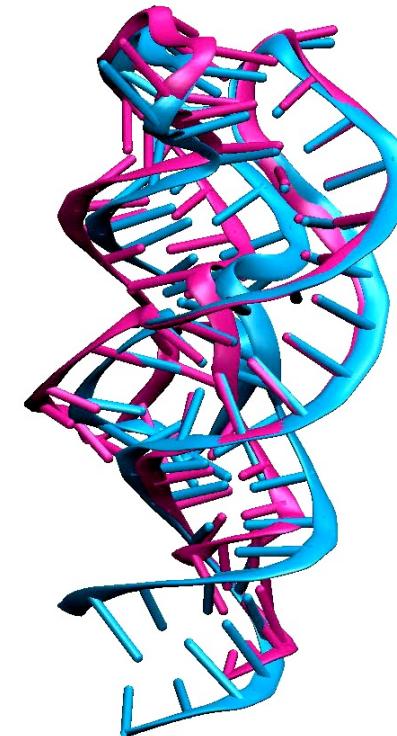


Hint: what gets logged on the Tk console when you change frames in the UI?

```
VMD TkConsole
(matheusmello) 83 % set loop_crystal [atomselect 0 "resid 40 to 60"]
atomselect7013
(matheusmello) 84 % set loop_sim [atomselect 1 "resid 40 to 60"]
atomselect7014
(matheusmello) 85 % for {set frame 0} {$frame < 1000} {incr frame} {
    animate goto $frame
    set M [measure fit $loop_sim $loop_crystal]
    $simulation move $M
}
```

Main slave 9.1

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

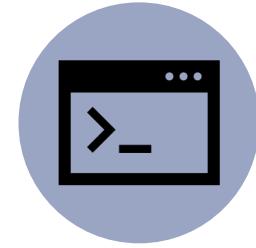


Movie making in VMD



GUI

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

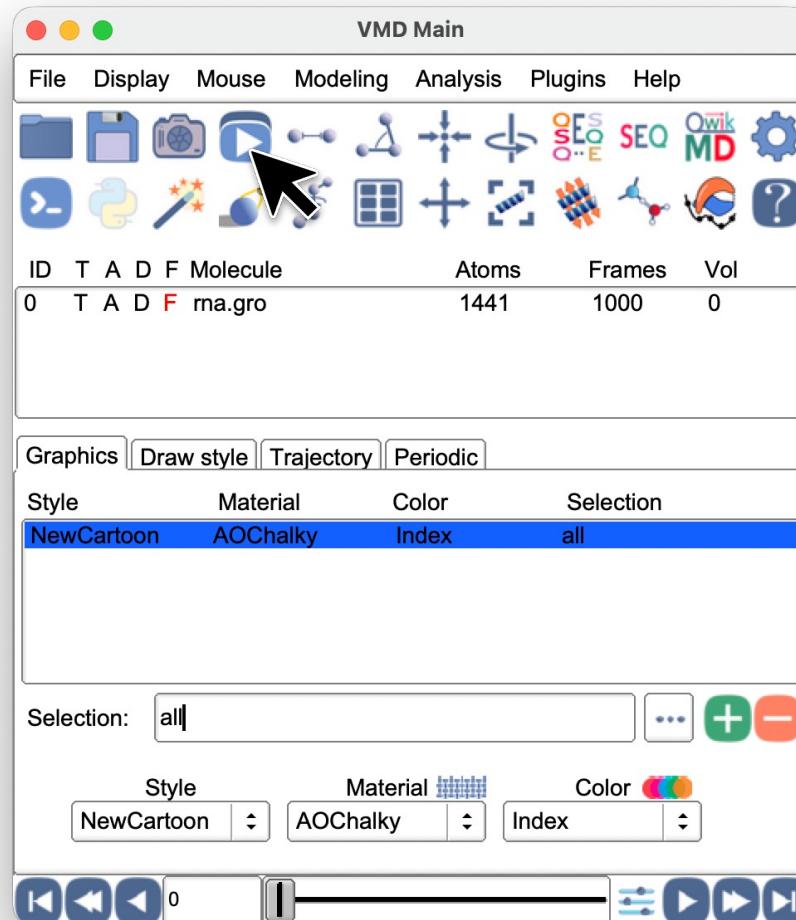


Scripting

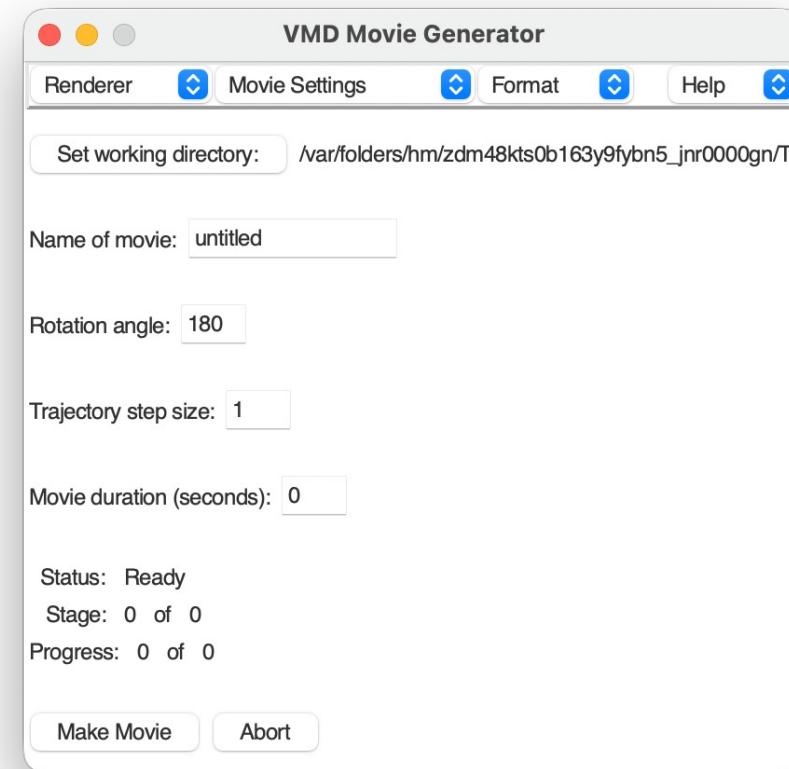
May take more time to set up but can achieve more polished results.

Movie making with the GUI

Let's restart and open the RNA model again



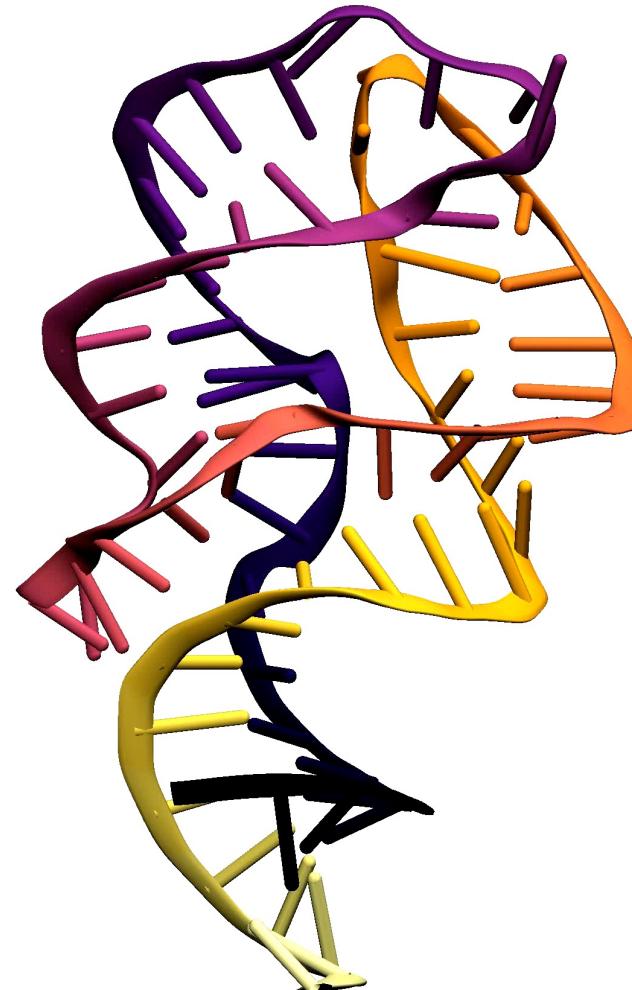
Alternatively: Extensions > Visualization > Movie Maker
(version 1.9.4)



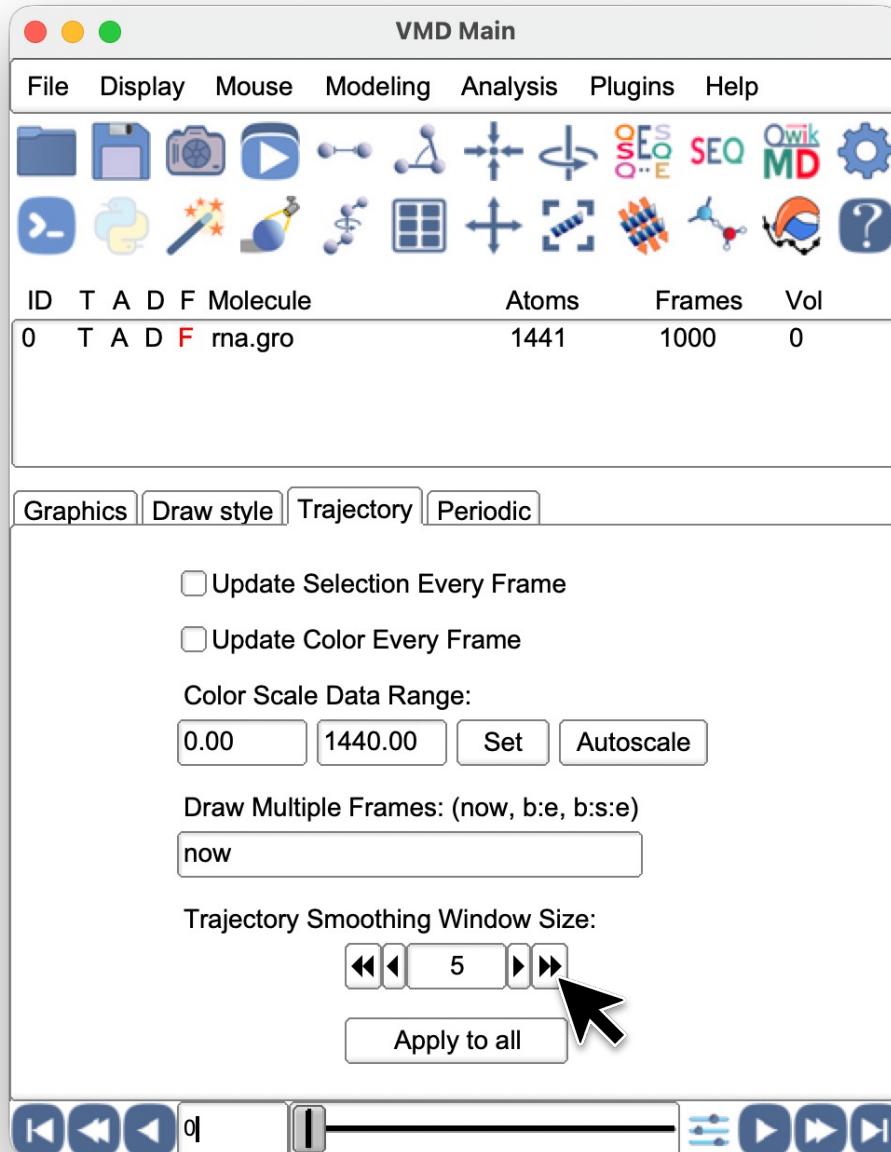
version 2.0.0

Movie making with the GUI

Our results so far...

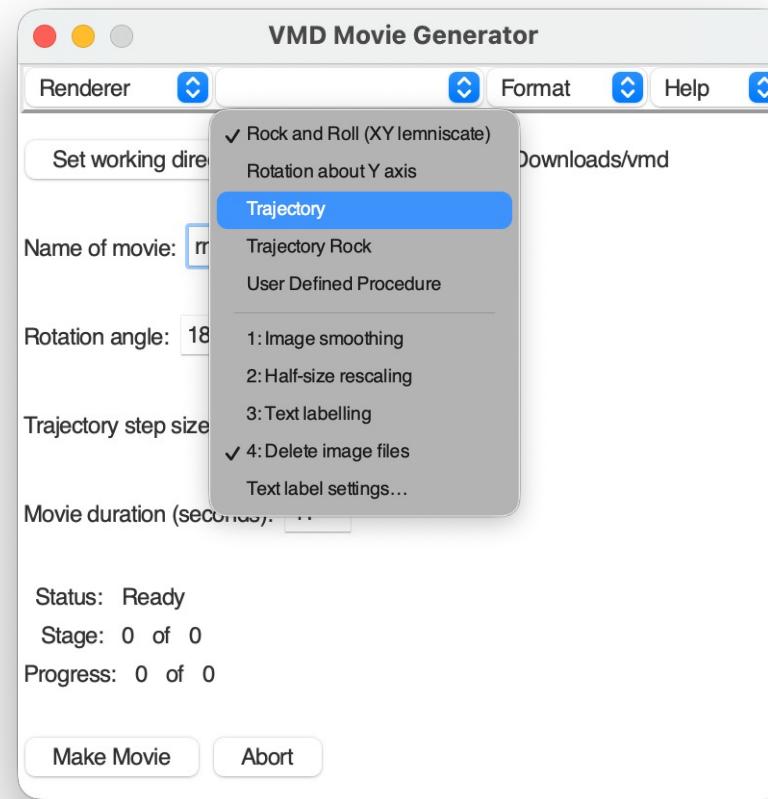


Movie making with the GUI



Change Movie Settings to Trajectory

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

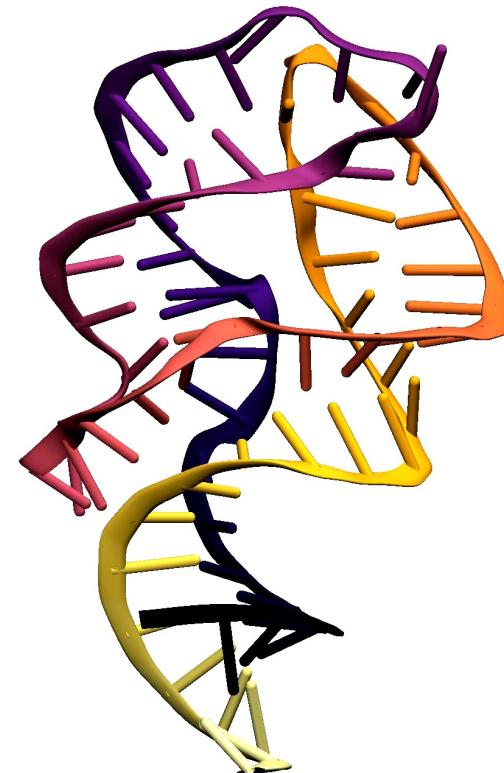


Movie making with the GUI

Smoothing = 5



Smoothing = 0

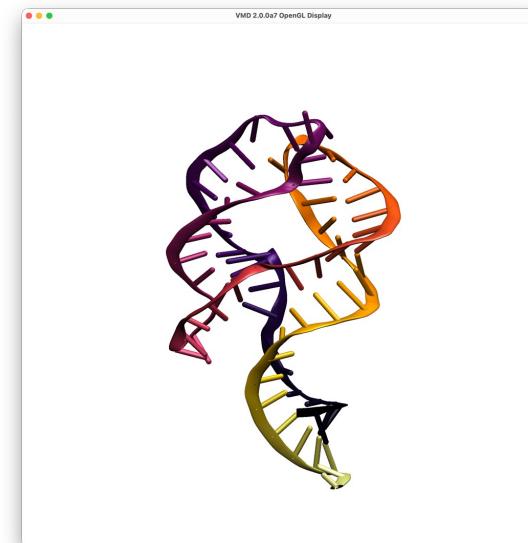


Movie making with the Tk console

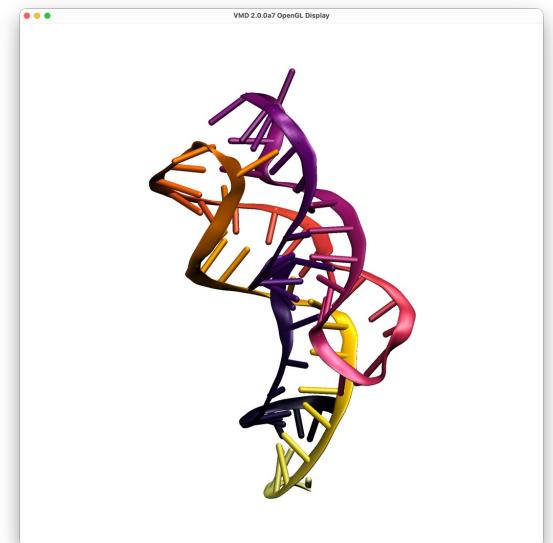
Download the file `visualization.tcl` from the Git Repo.

```
VMD TkConsole  
  
(VMDscripting) 67 % ls  
/Users/matheusmello/VMDscripting:  
alignment.mpg      la.tcl          orient.tcl      powers_of_2.dat  
rna.gro           rna.xtc        visualization.tcl vmd-tutorial.vmd  
  
(VMDscripting) 68 % source visualization.tcl  
(VMDscripting) 69 % save_vp 1  
(VMDscripting) 70 % save_vp 2  
(VMDscripting) 71 % move_vp 1 2  
(VMDscripting) 72 % move_vp 2 1  
(VMDscripting) 73 % mkdir video  
(VMDscripting) 74 % move_vp_render 1 2 0 . "animate"  
Rendering frame ./animate.0000.dat  
Rendering frame ./animate.0001.dat  
Rendering frame ./animate.0002.dat  
Rendering frame ./animate.0003.dat  
Rendering frame ./animate.0004.dat  
Rendering frame ./animate.0005.dat  
Rendering frame ./animate.0006.dat  
Rendering frame ./animate.0007.dat  
Rendering frame ./animate.0008.dat  
  
X Main slave 62.20
```

```
> 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 1



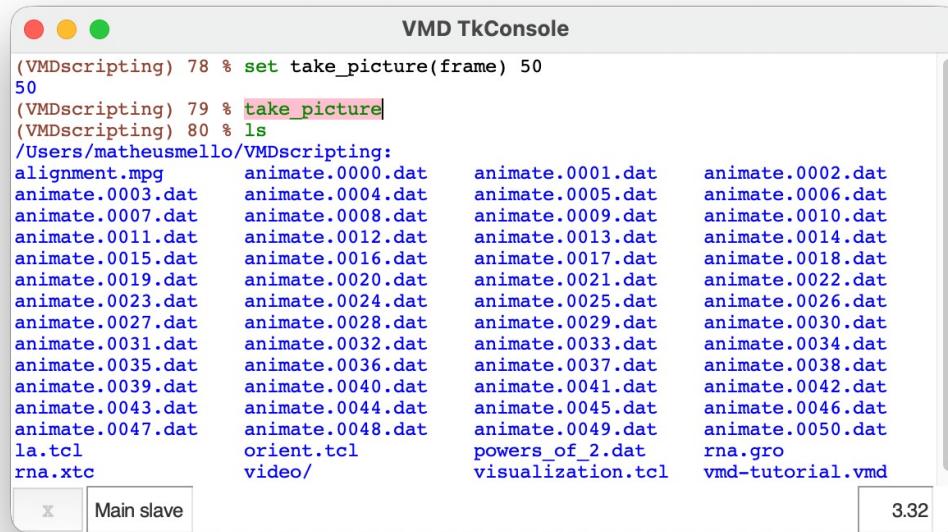
Viewpoint 2



Viewpoint refers to the camera position, orientation and zoom.

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.



The screenshot shows a terminal window titled "VMD TkConsole". The session starts with some environment variables and then runs a script:

```
(VMDscripting) 78 % set take_picture(frame) 50
50
(VMDscripting) 79 % take_picture
(VMDscripting) 80 % ls
```

The directory listing shows files named `animate.0000.dat` through `animate.0050.dat`, `alignment.mpg`, `la.tcl`, `orient.tcl`, `rna.gro`, and `vmd-tutorial.vmd`.

At the bottom left is a small window titled "Main slave" with a "X" button. At the bottom right is the number "3.32".

Our movie script:

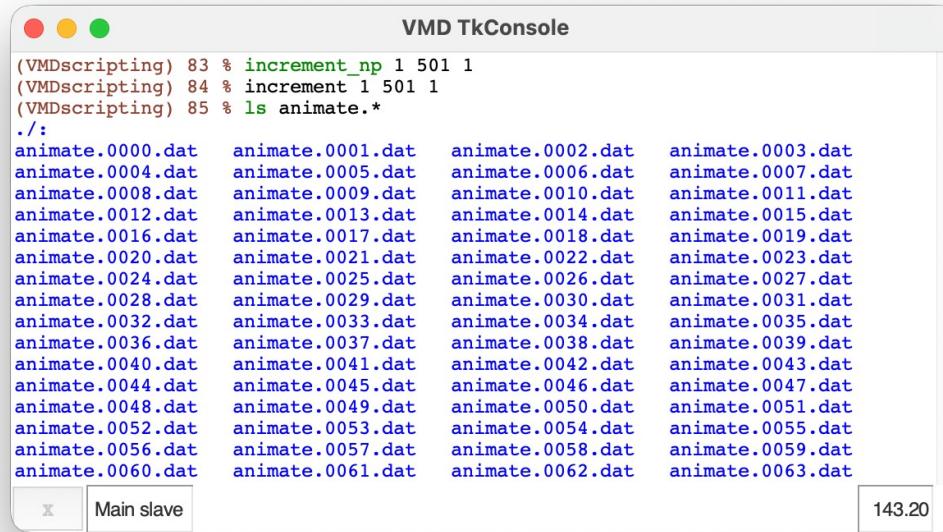
- Frames 0-49: moving between viewpoints 1 and 2
- Frame 50: snapshot of viewpoint 2 in frame 0



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.



The screenshot shows a terminal window titled "VMD TkConsole". The session starts with three command-line entries:

```
(VMDscripting) 83 % increment_np 1 501 1
(VMDscripting) 84 % increment 1 501 1
(VMDscripting) 85 % ls animate.*
```

Following these, there is a large list of files:

```
./:
animate.0000.dat  animate.0001.dat  animate.0002.dat  animate.0003.dat
animate.0004.dat  animate.0005.dat  animate.0006.dat  animate.0007.dat
animate.0008.dat  animate.0009.dat  animate.0010.dat  animate.0011.dat
animate.0012.dat  animate.0013.dat  animate.0014.dat  animate.0015.dat
animate.0016.dat  animate.0017.dat  animate.0018.dat  animate.0019.dat
animate.0020.dat  animate.0021.dat  animate.0022.dat  animate.0023.dat
animate.0024.dat  animate.0025.dat  animate.0026.dat  animate.0027.dat
animate.0028.dat  animate.0029.dat  animate.0030.dat  animate.0031.dat
animate.0032.dat  animate.0033.dat  animate.0034.dat  animate.0035.dat
animate.0036.dat  animate.0037.dat  animate.0038.dat  animate.0039.dat
animate.0040.dat  animate.0041.dat  animate.0042.dat  animate.0043.dat
animate.0044.dat  animate.0045.dat  animate.0046.dat  animate.0047.dat
animate.0048.dat  animate.0049.dat  animate.0050.dat  animate.0051.dat
animate.0052.dat  animate.0053.dat  animate.0054.dat  animate.0055.dat
animate.0056.dat  animate.0057.dat  animate.0058.dat  animate.0059.dat
animate.0060.dat  animate.0061.dat  animate.0062.dat  animate.0063.dat
```

The bottom left corner of the window says "Main slave" and the bottom right corner shows the number "143.20".

Our movie script:

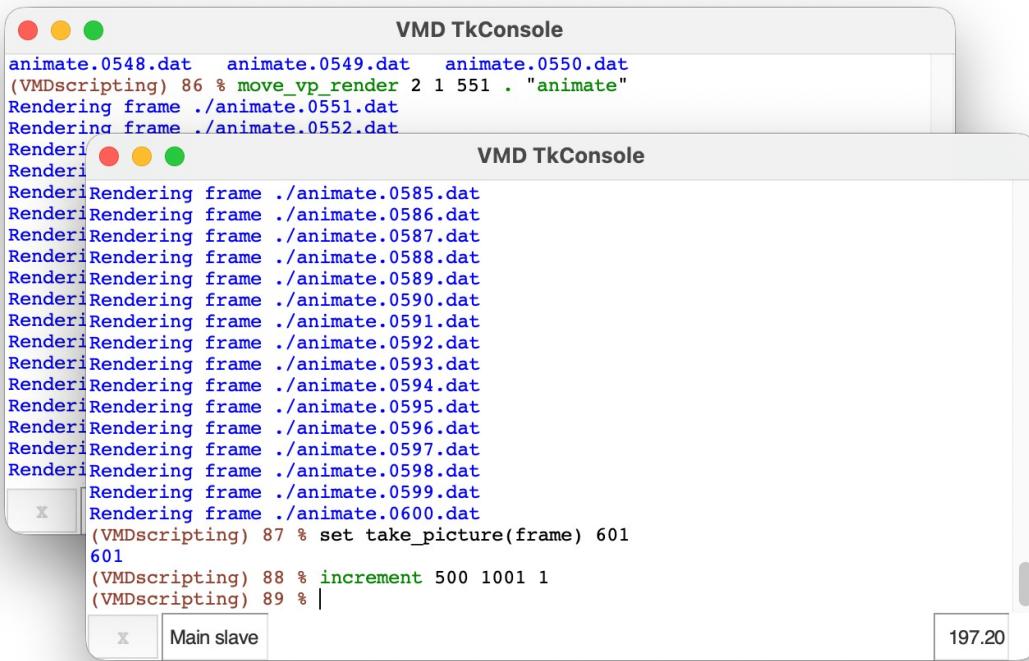
- Frames 0-49: moving between viewpoints 1 and 2
- Frame 50: snapshot of viewpoint 2 in frame 0
- Frame 51-550: dynamics of the RNA in viewpoint 2 until frame 500

```
> 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.



The image shows two overlapping windows of the VMD TkConsole. The top window displays the command `move_vp_render 2 1 551 . "animate"`. The bottom window shows a series of rendering commands from frame 0548 to 0600, followed by a command to set the take picture frame to 601. The bottom window also shows the increment command being issued.

```
VMD TkConsole
animate.0548.dat  animate.0549.dat  animate.0550.dat
(VMDscripting) 86 % move_vp_render 2 1 551 . "animate"
Rendering frame ./animate.0551.dat
Rendering frame ./animate.0552.dat
Renderi
Renderi
Rendering frame ./animate.0585.dat
Rendering frame ./animate.0586.dat
Rendering frame ./animate.0587.dat
Rendering frame ./animate.0588.dat
Rendering frame ./animate.0589.dat
Rendering frame ./animate.0590.dat
Rendering frame ./animate.0591.dat
Rendering frame ./animate.0592.dat
Rendering frame ./animate.0593.dat
Rendering frame ./animate.0594.dat
Rendering frame ./animate.0595.dat
Rendering frame ./animate.0596.dat
Rendering frame ./animate.0597.dat
Rendering frame ./animate.0598.dat
Rendering frame ./animate.0599.dat
Rendering frame ./animate.0600.dat
(VMDscripting) 87 % set take_picture(frame) 601
601
(VMDscripting) 88 % increment 500 1001 1
(VMDscripting) 89 %

X Main slave
197.20
```

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

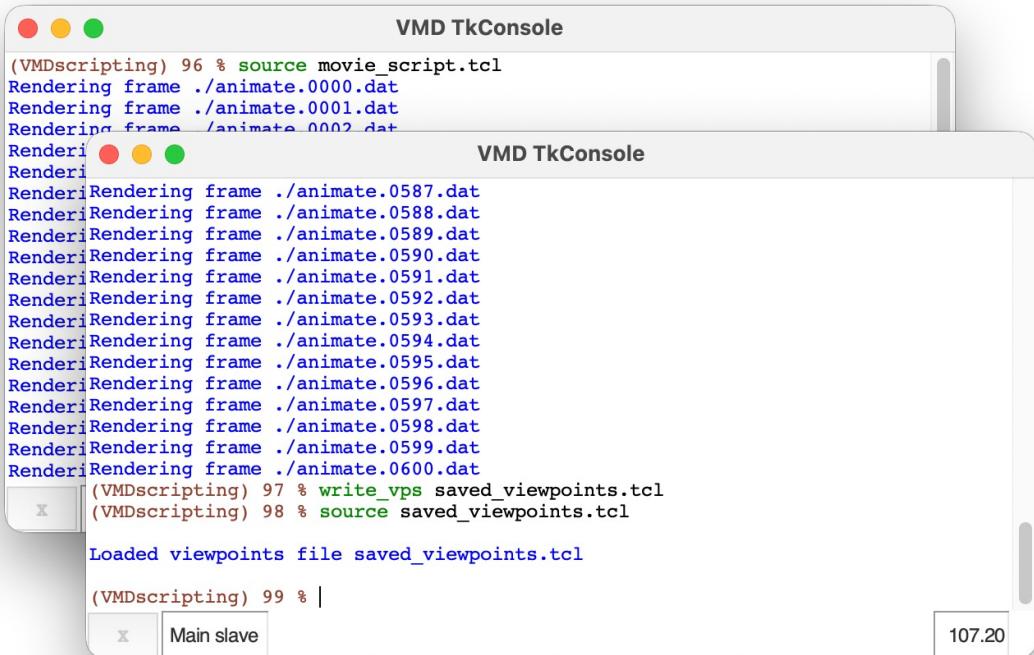
→ end frame is not included!

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

Movie making with the Tk console

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



The image shows two overlapping windows of the VMD TkConsole. The top window has the title 'VMD TkConsole' and contains the following command history:

```
(VMDscripting) 96 % source movie_script.tcl
Rendering frame ./animate.0000.dat
Rendering frame ./animate.0001.dat
Rendering frame ./animate.0002.dat
Renderi
Renderi
Renderi Rendering frame ./animate.0587.dat
Renderi Rendering frame ./animate.0588.dat
Renderi Rendering frame ./animate.0589.dat
Renderi Rendering frame ./animate.0590.dat
Renderi Rendering frame ./animate.0591.dat
Renderi Rendering frame ./animate.0592.dat
Renderi Rendering frame ./animate.0593.dat
Renderi Rendering frame ./animate.0594.dat
Renderi Rendering frame ./animate.0595.dat
Renderi Rendering frame ./animate.0596.dat
Renderi Rendering frame ./animate.0597.dat
Renderi Rendering frame ./animate.0598.dat
Renderi Rendering frame ./animate.0599.dat
Renderi Rendering frame ./animate.0600.dat
(VMDscripting) 97 % write_vps saved_viewpoints.tcl
(VMDscripting) 98 % source saved_viewpoints.tcl

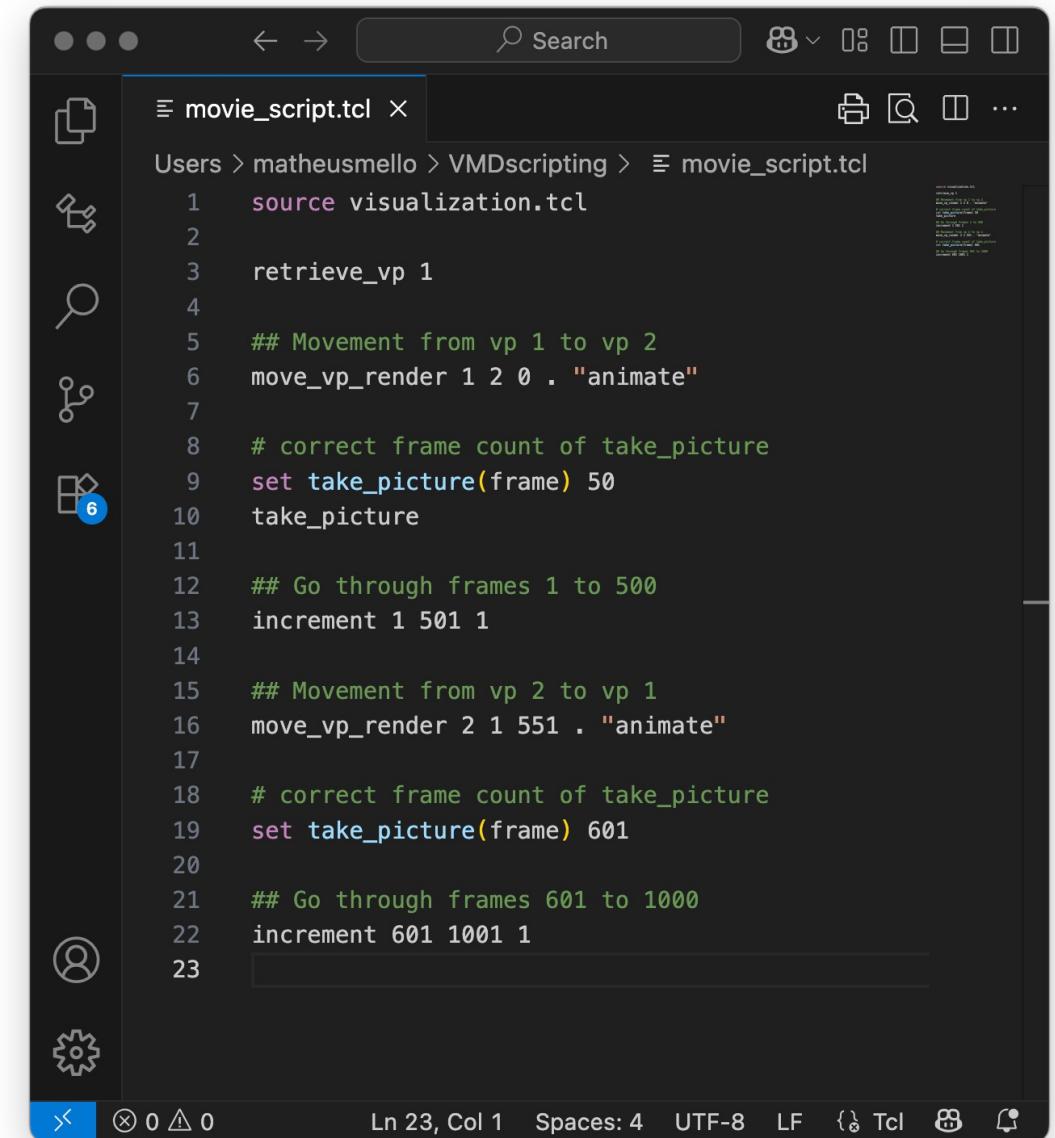
Loaded viewpoints file saved_viewpoints.tcl

(VMDscripting) 99 % |
```

The bottom window also has the title 'VMD TkConsole' and shows the same command history. Both windows have a status bar at the bottom indicating '107.20'.

To save and load viewpoints

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



The image shows a code editor window titled 'movie_script.tcl'. The script content is as follows:

```
source visualization.tcl
retrieve_vp 1
## Movement from vp 1 to vp 2
move_vp_render 1 2 0 . "animate"
# correct frame count of take_picture
set take_picture(frame) 50
take_picture
## Go through frames 1 to 500
increment 1 501 1
## Movement from vp 2 to vp 1
move_vp_render 2 1 551 . "animate"
# correct frame count of take_picture
set take_picture(frame) 601
## Go through frames 601 to 1000
increment 601 1001 1
```

The code editor interface includes a sidebar with icons for file operations, a search bar, and status indicators at the bottom.

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.

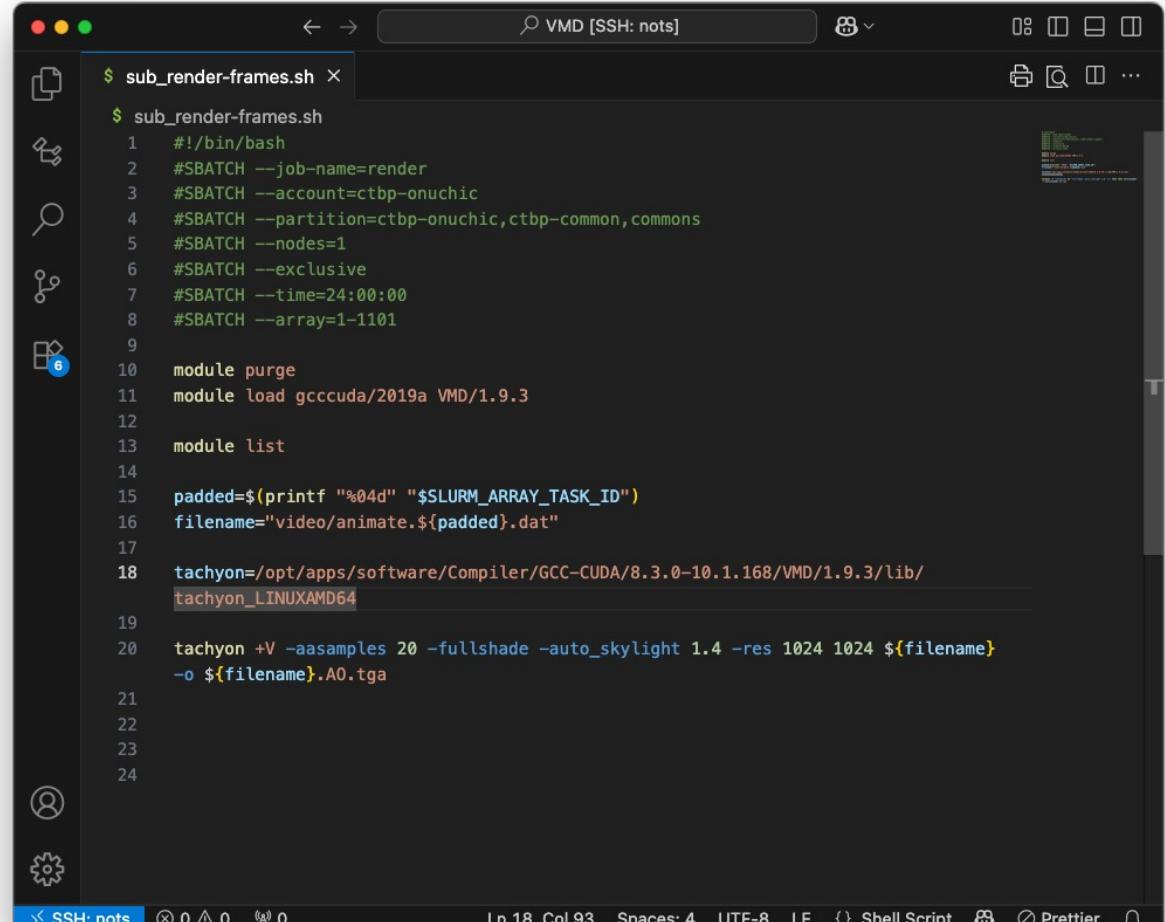
```
> mkdir video  
> mv *.dat video  
> rsync -avP video remote_loc
```

On remote:

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

To assemble the video:

```
> ffmpeg -framerate 60 -i video/animate.%04d.dat.A0.tga -c:v libx264  
-crf 10 -pix_fmt yuv420p rna_dynamics.mp4
```



```
$ sub_render-frames.sh ×  
$ sub_render-frames.sh  
1  #!/bin/bash  
2  #SBATCH --job-name=render  
3  #SBATCH --account=ctbp-onuchic  
4  #SBATCH --partition=ctbp-onuchic,ctbp-common,commons  
5  #SBATCH --nodes=1  
6  #SBATCH --exclusive  
7  #SBATCH --time=24:00:00  
8  #SBATCH --array=1-1101  
9  
10 module purge  
11 module load gcccuda/2019a VMD/1.9.3  
12  
13 module list  
14  
15 padded=$(printf "%04d" "$SLURM_ARRAY_TASK_ID")  
16 filename="video/animate.${padded}.dat"  
17  
18 tachyon=/opt/apps/software/Compiler/GCC-CUDA/8.3.0-10.1.168/VMD/1.9.3/lib/  
tachyon_LINUXAMD64  
19  
20 tachyon +V -aasamples 20 -fullshade -auto_skylight 1.4 -res 1024 1024 ${filename}  
-o ${filename}.AO.tga  
21  
22  
23  
24
```

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>