

# Advanced VMD: Trajectories, movies, scripting

## WestGrid webinar

Olivier Fisette [olivier.fisette@westgrid.ca](mailto:olivier.fisette@westgrid.ca)  
Advanced Research Computing, ICT  
University of Saskatchewan

2020-10-28  
CC BY 4.0

# Downloadable material

- All examples shown during the webinar use downloadable files.
  - Structures
  - Trajectories
  - Scripts
- Address
  - <https://nextcloud.computecanada.ca/index.php/s/kcHgbbEgwS3GkHM/download>
  - With wget, add -O *advanced\_vmd.tar*
- Extract the files from the archive.  
*tar -xf advanced\_vmd.tar*

# Contents

## In the previous webinar

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering

## New topics

- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- Combining trajectories, movies, and scripting

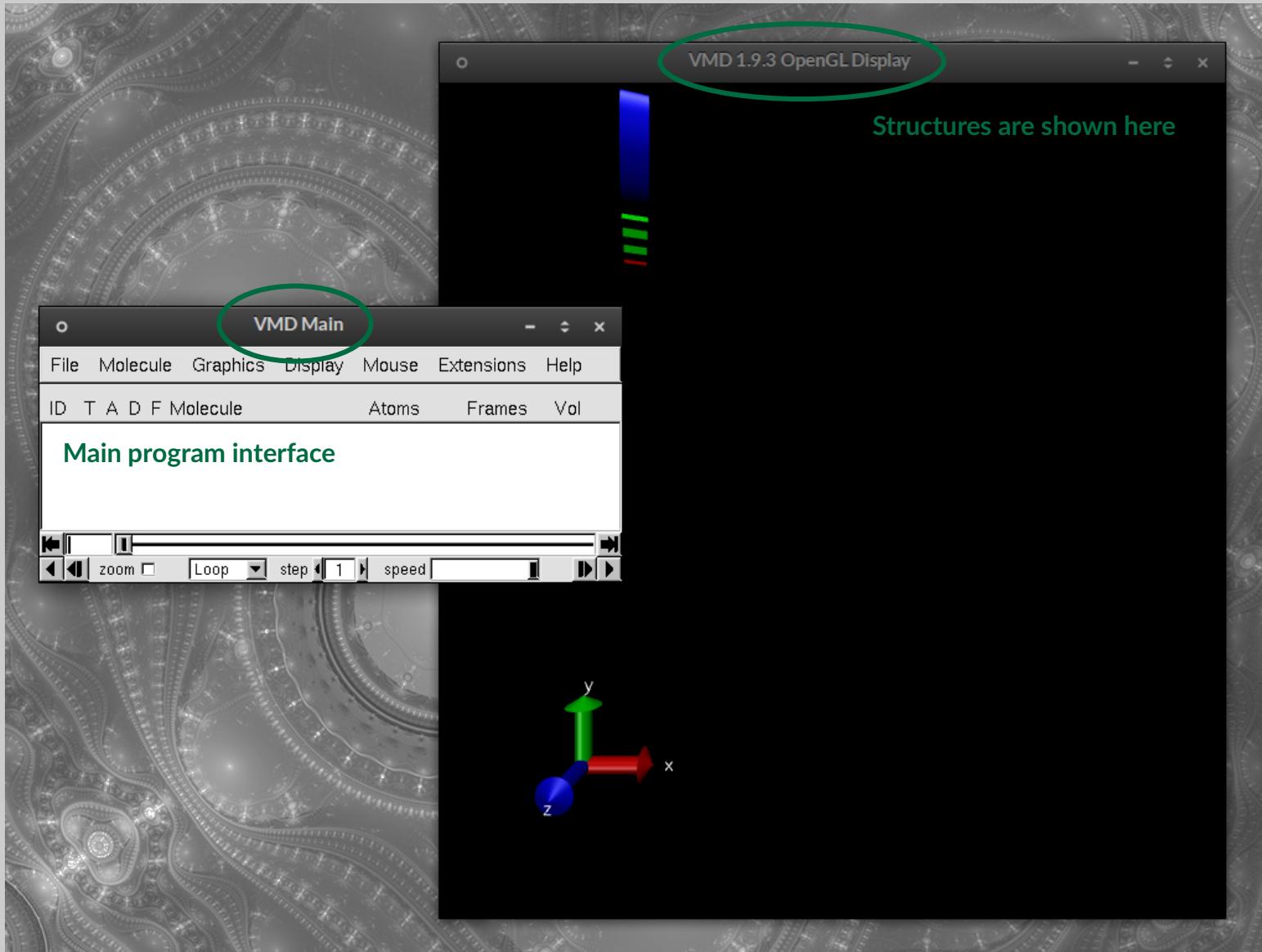
# Contents

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- Combining trajectories, movies, and scripting

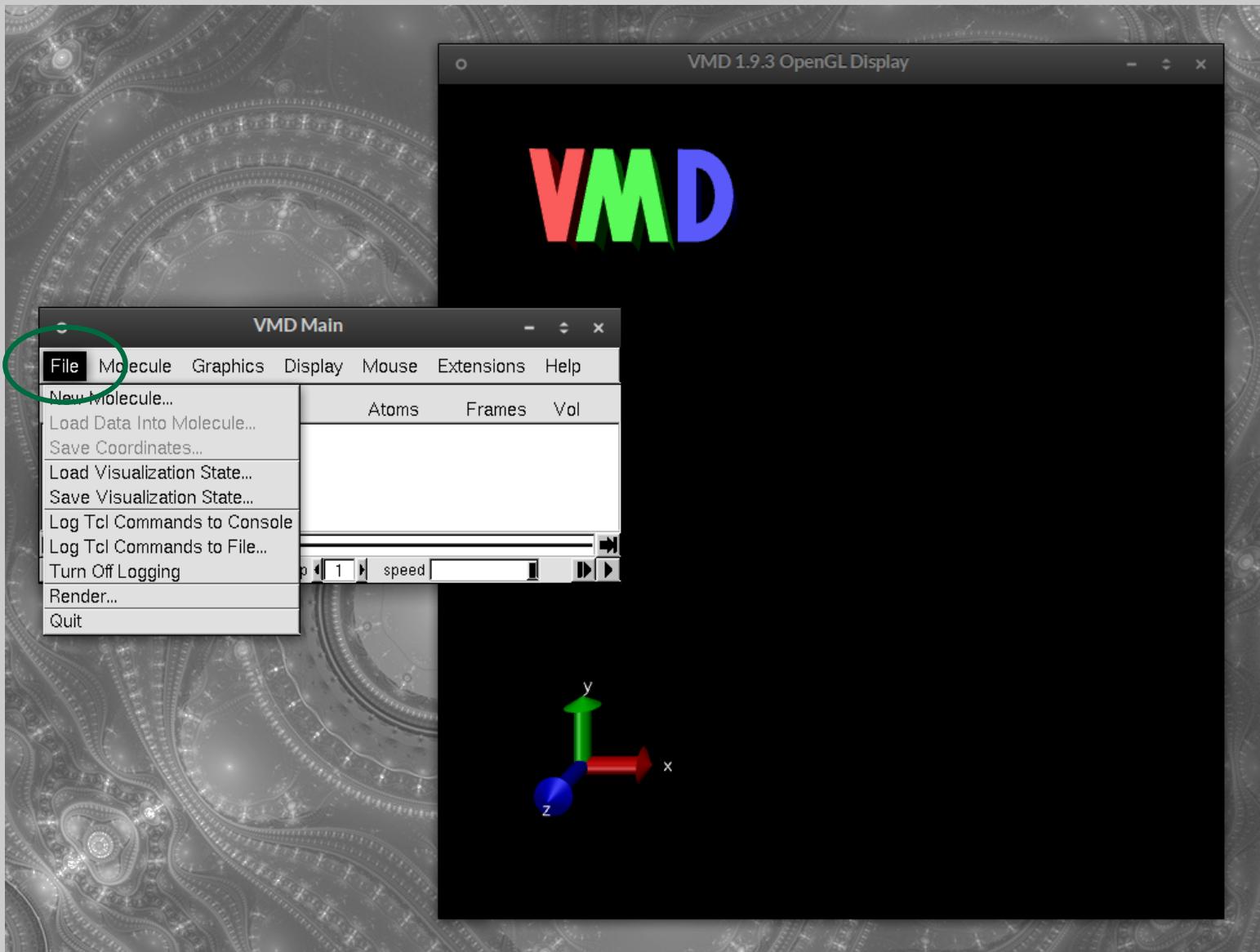
# Contents

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- Combining trajectories, movies, and scripting

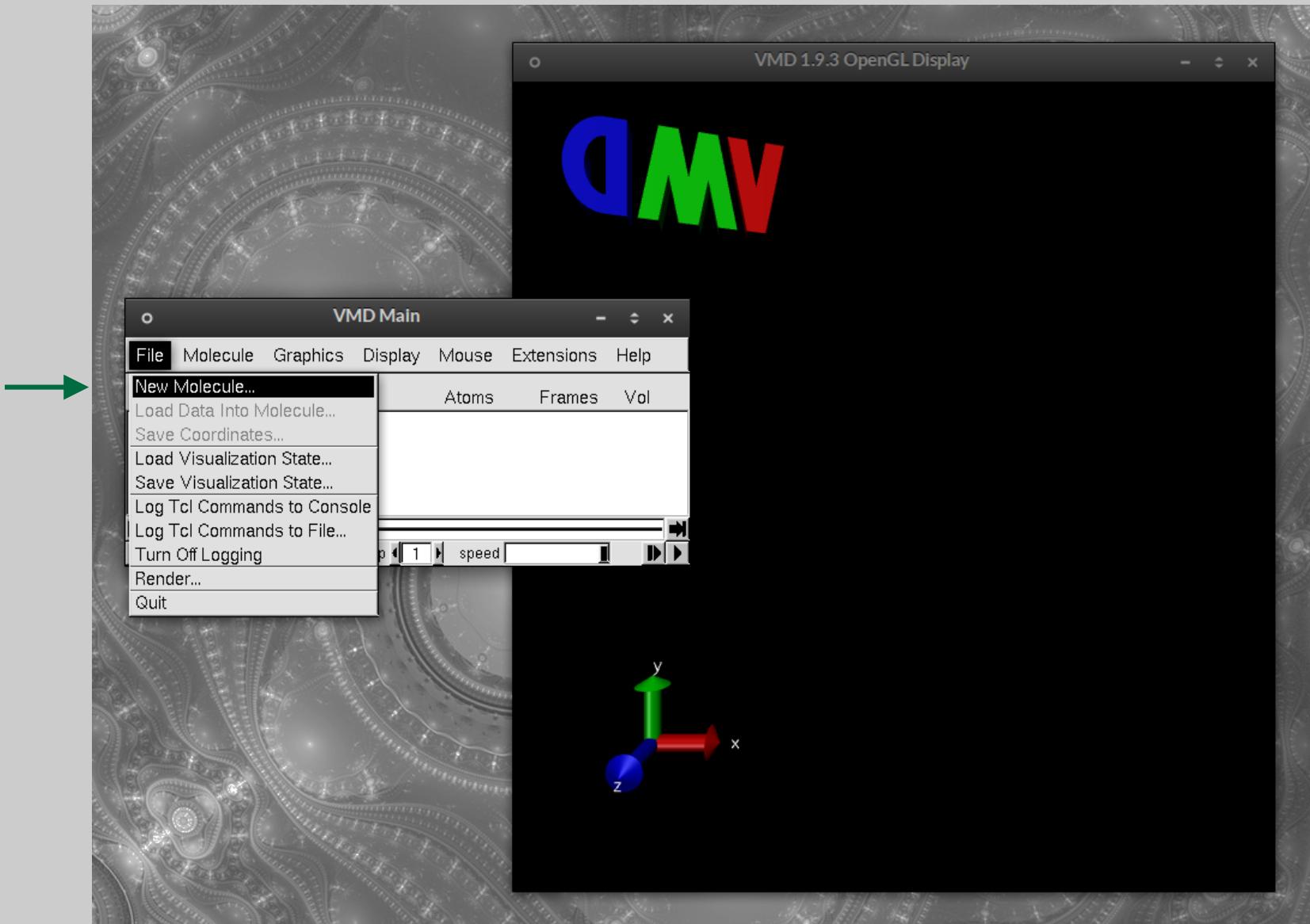
# VMD Windows



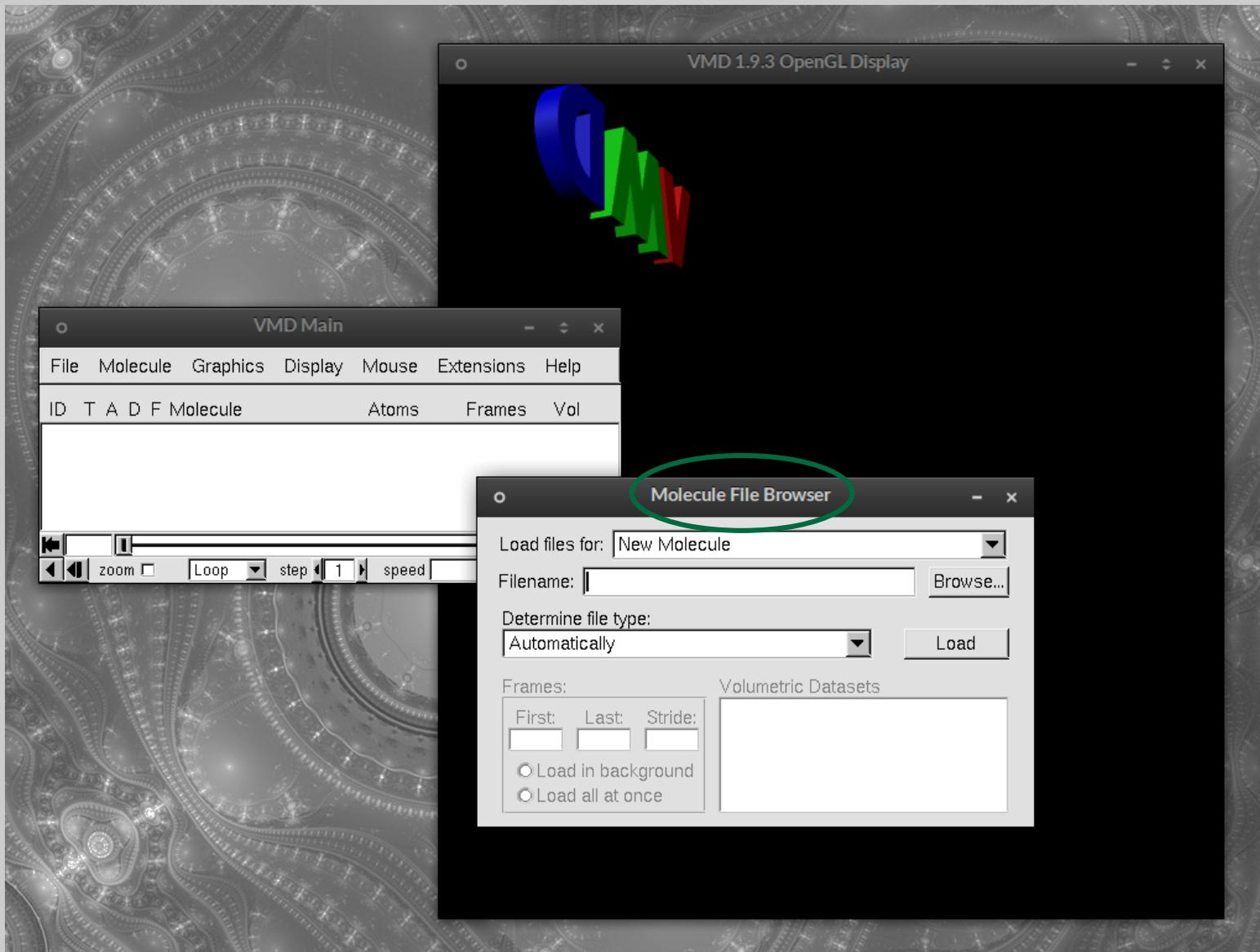
# Loading a structure



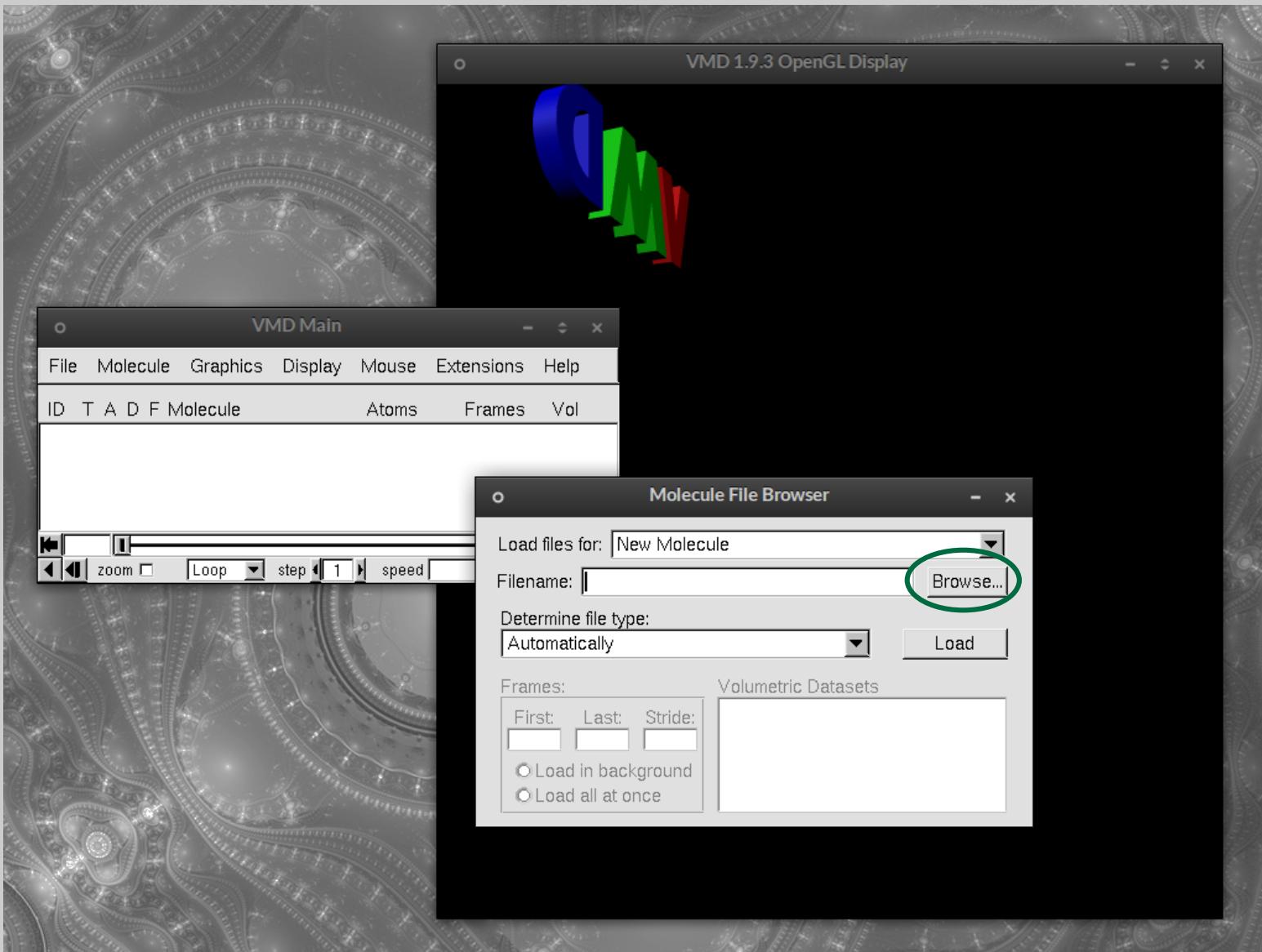
# Loading a structure



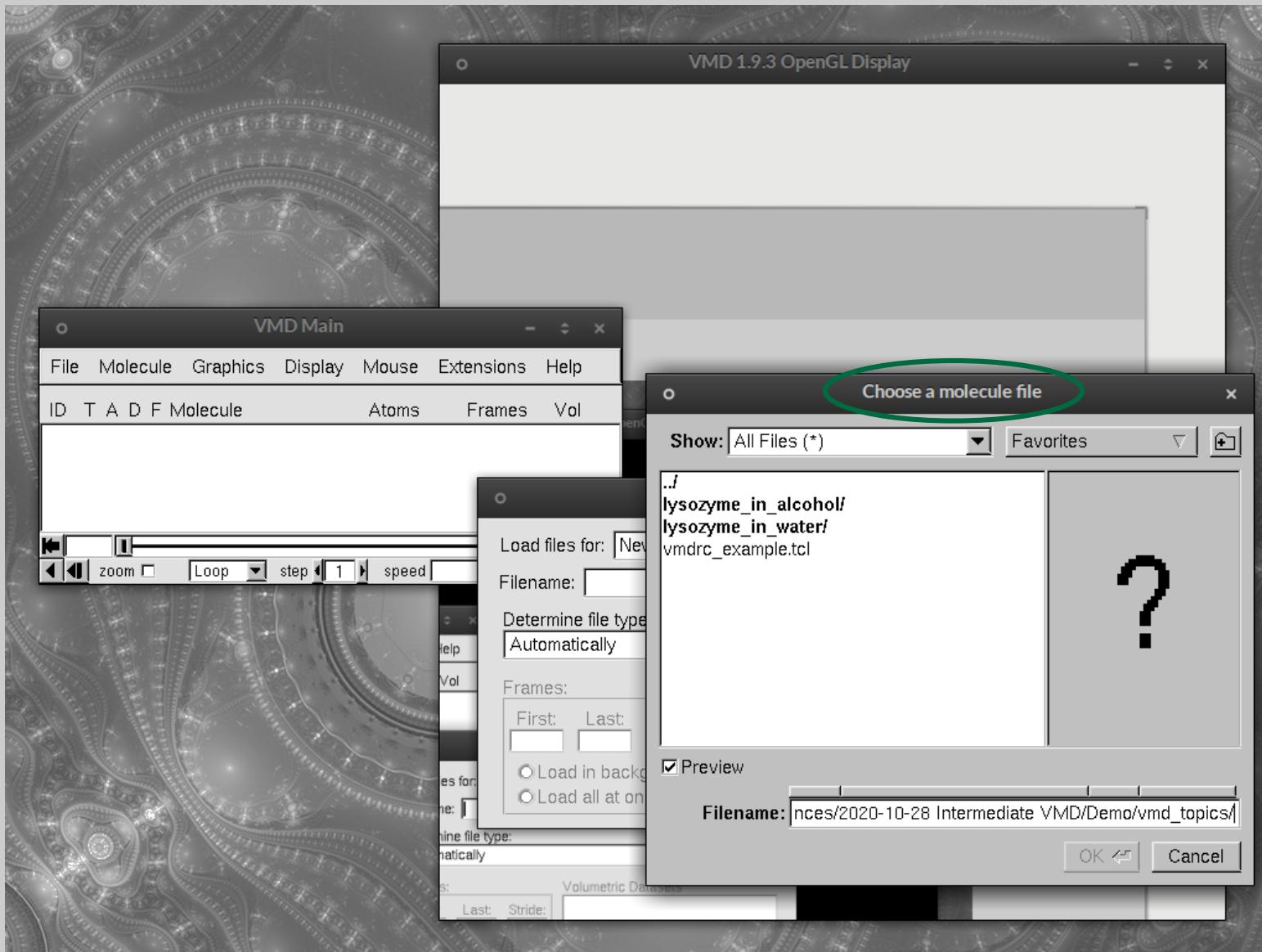
# Loading a structure



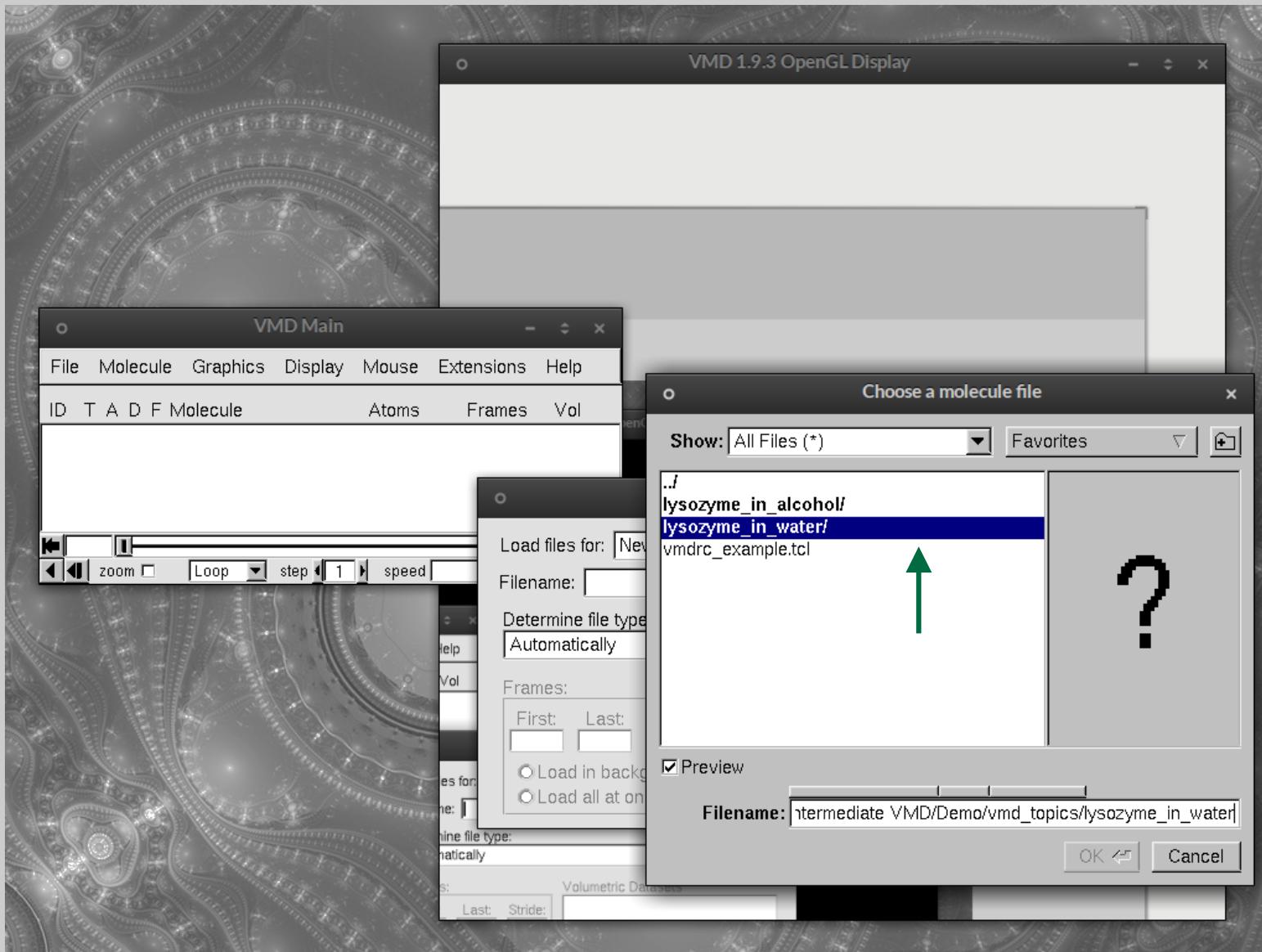
# Loading a structure



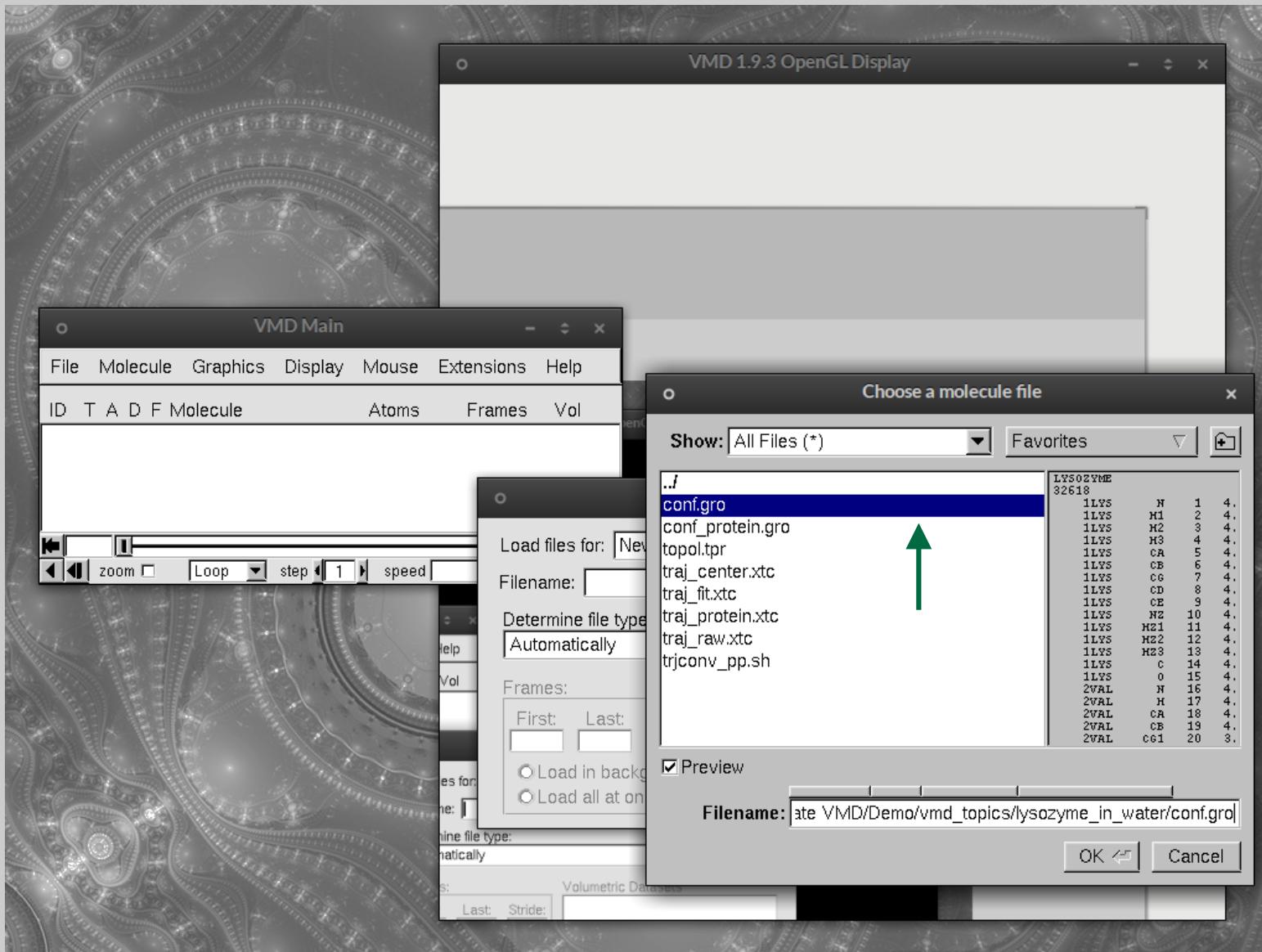
# Loading a structure



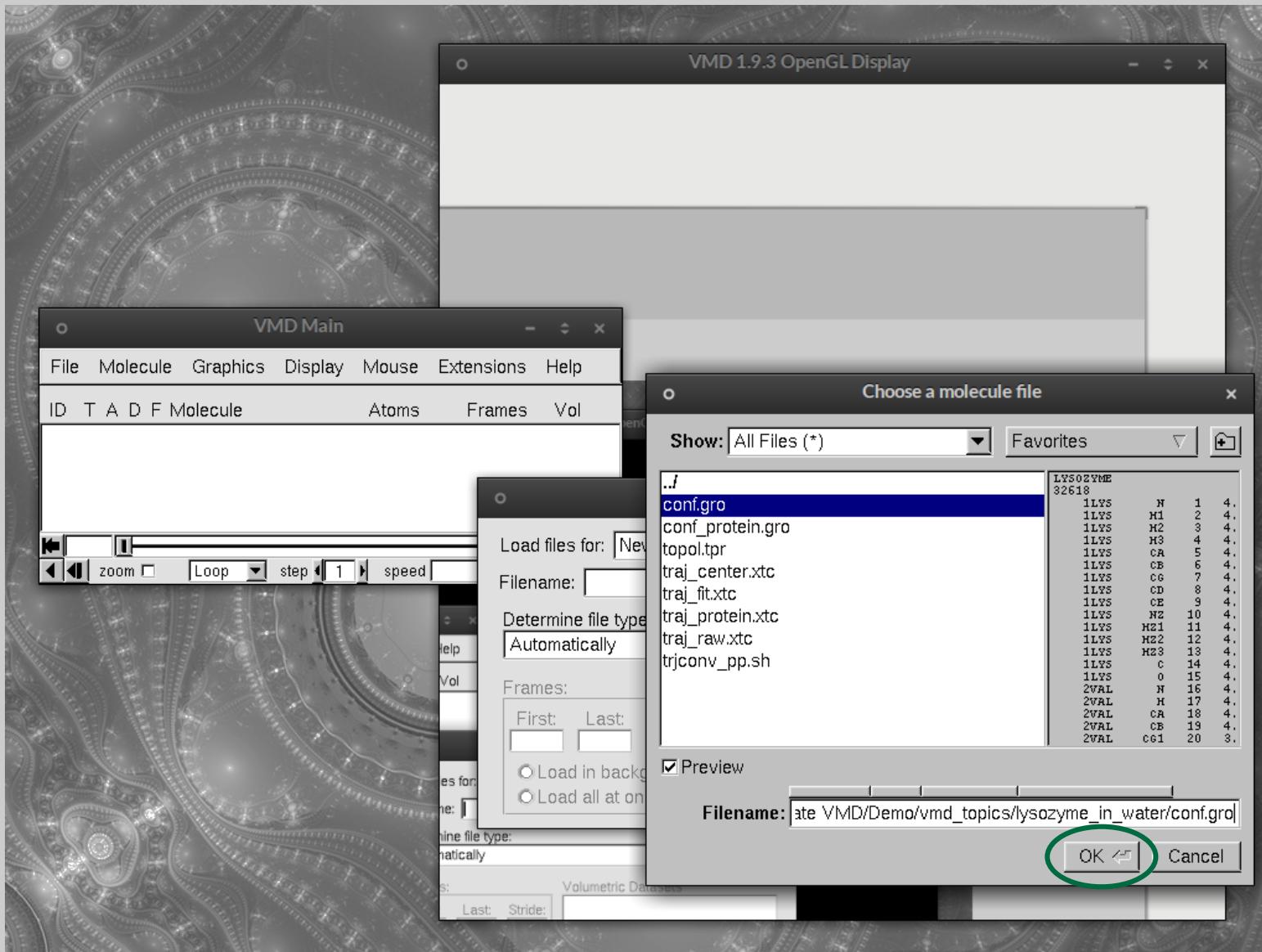
# Loading a structure



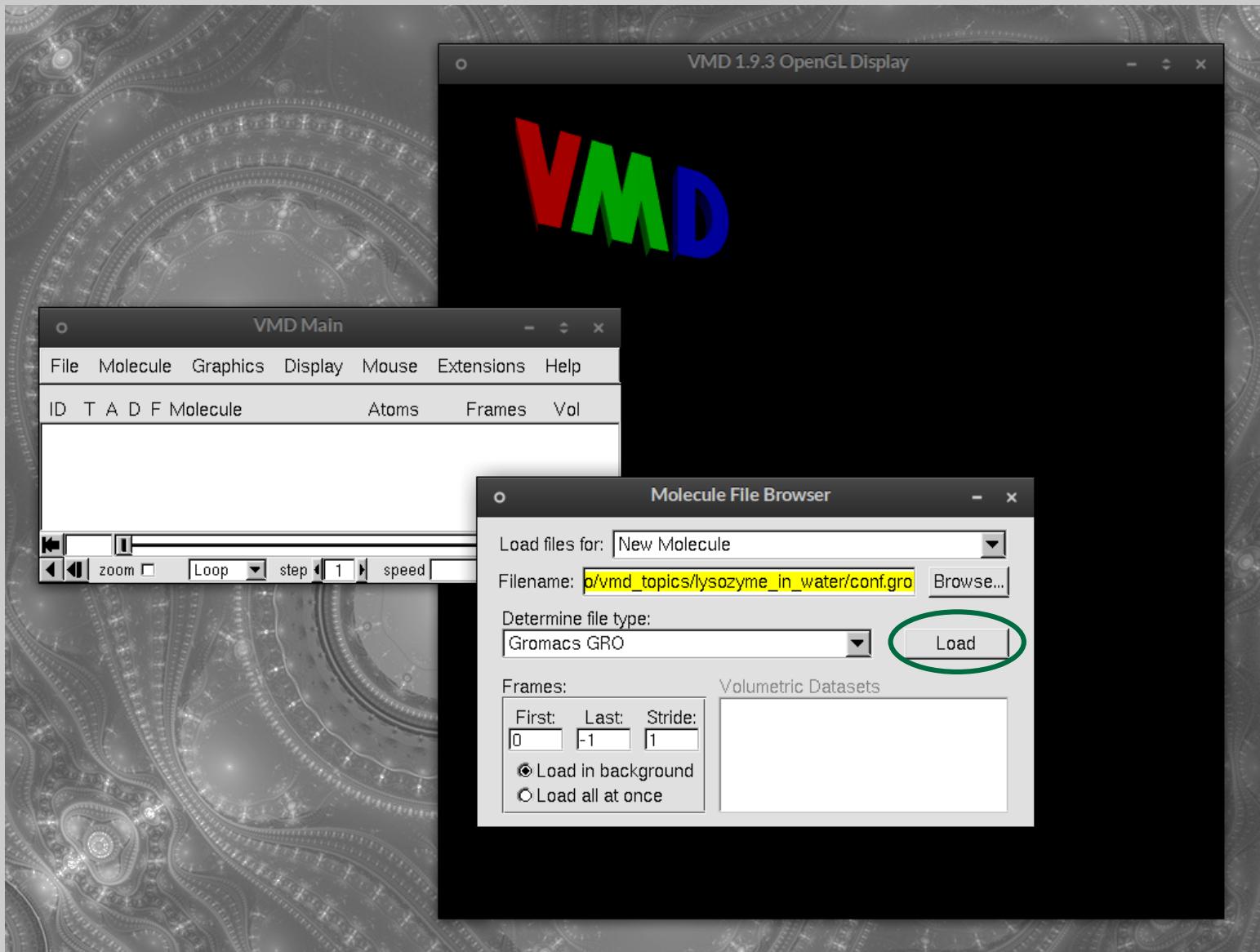
# Loading a structure



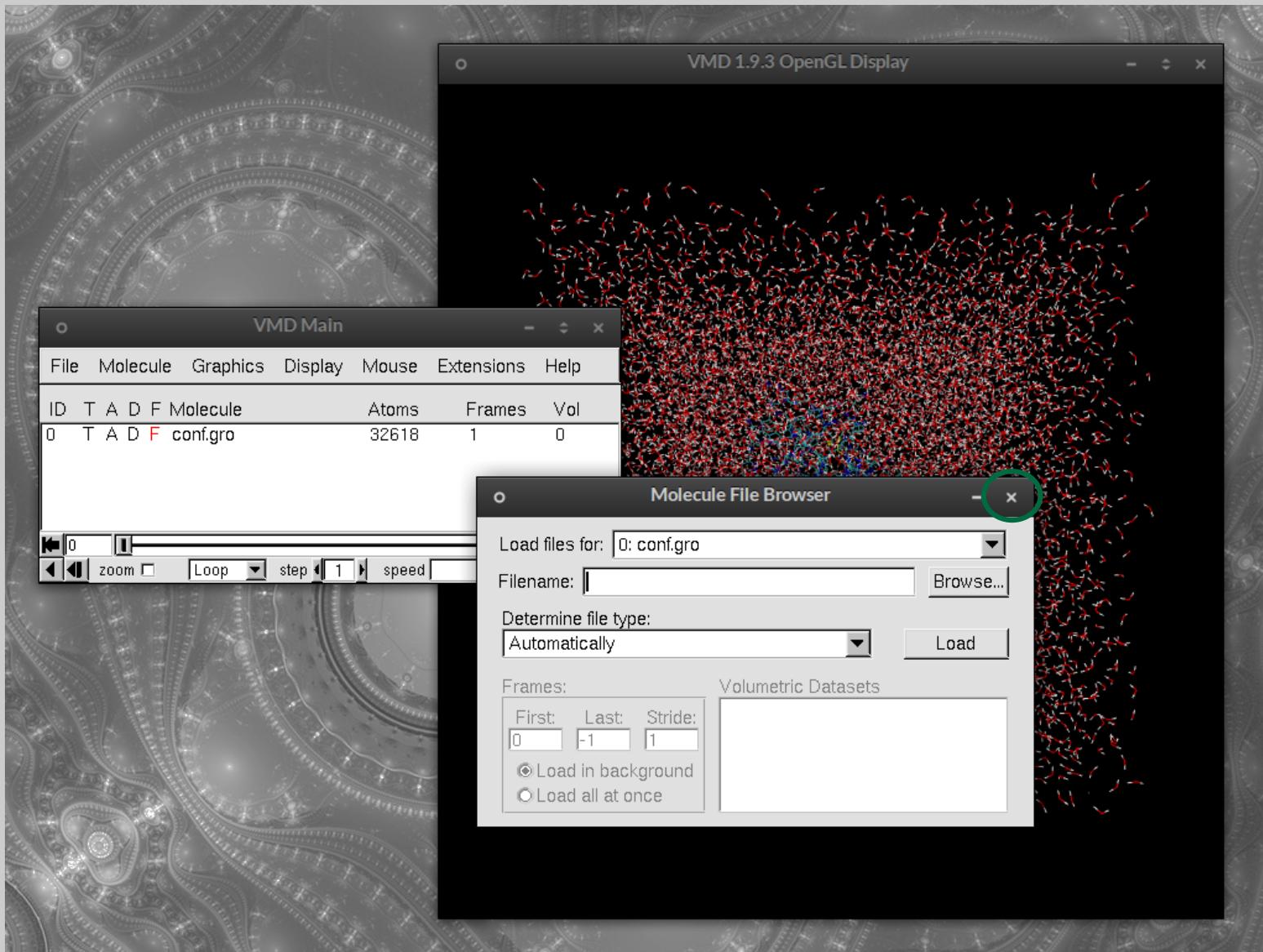
# Loading a structure



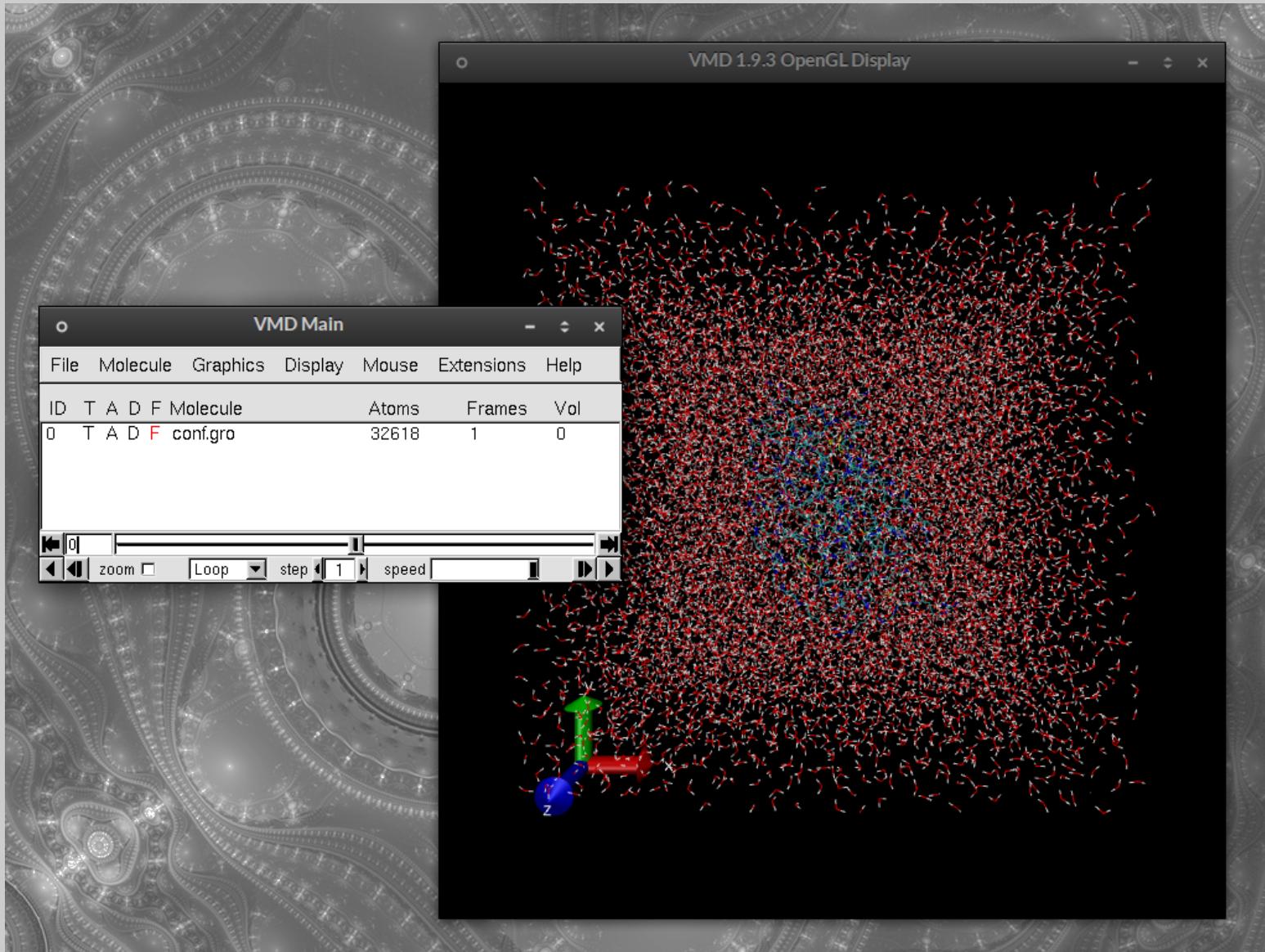
# Loading a structure



# Loading a structure



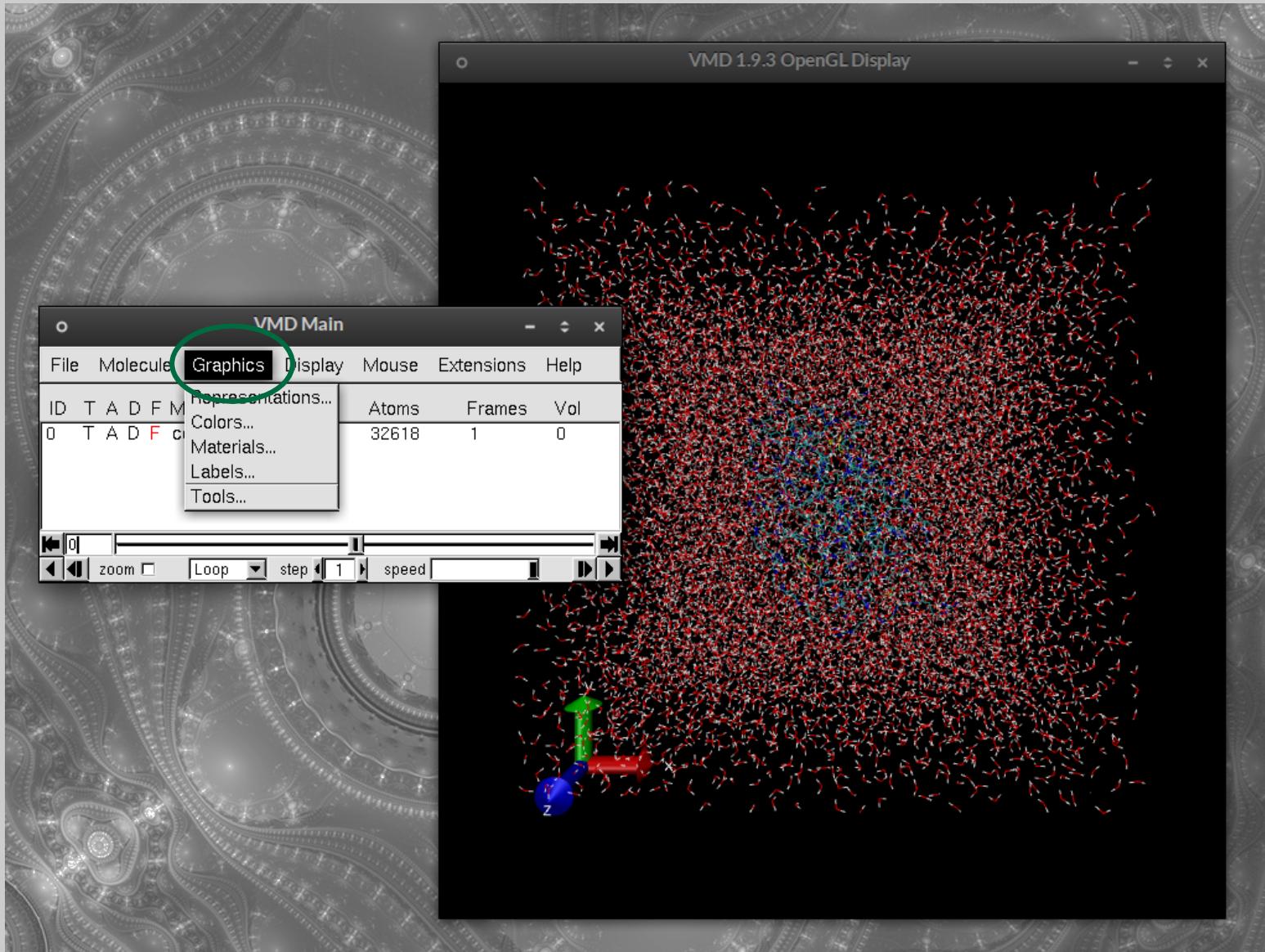
# Loading a structure



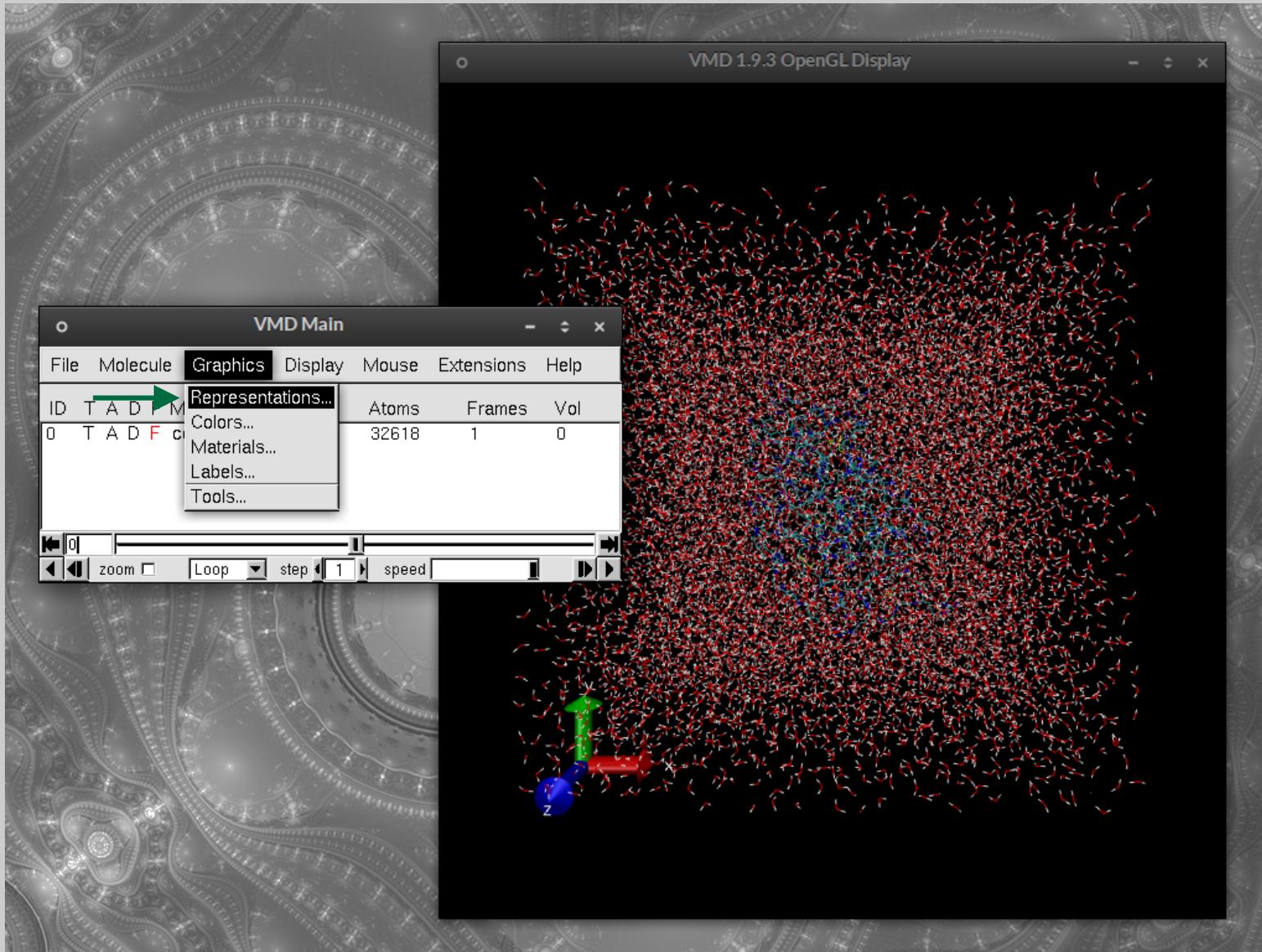
# Loading a structure

- The VMD interface uses many windows.
  - The “Main” window gives access to all controls and other windows.
  - A single “OpenGL Display” window shows all structures.
- File formats are automatically detected.
- A wide variety of formats (60+) are supported.

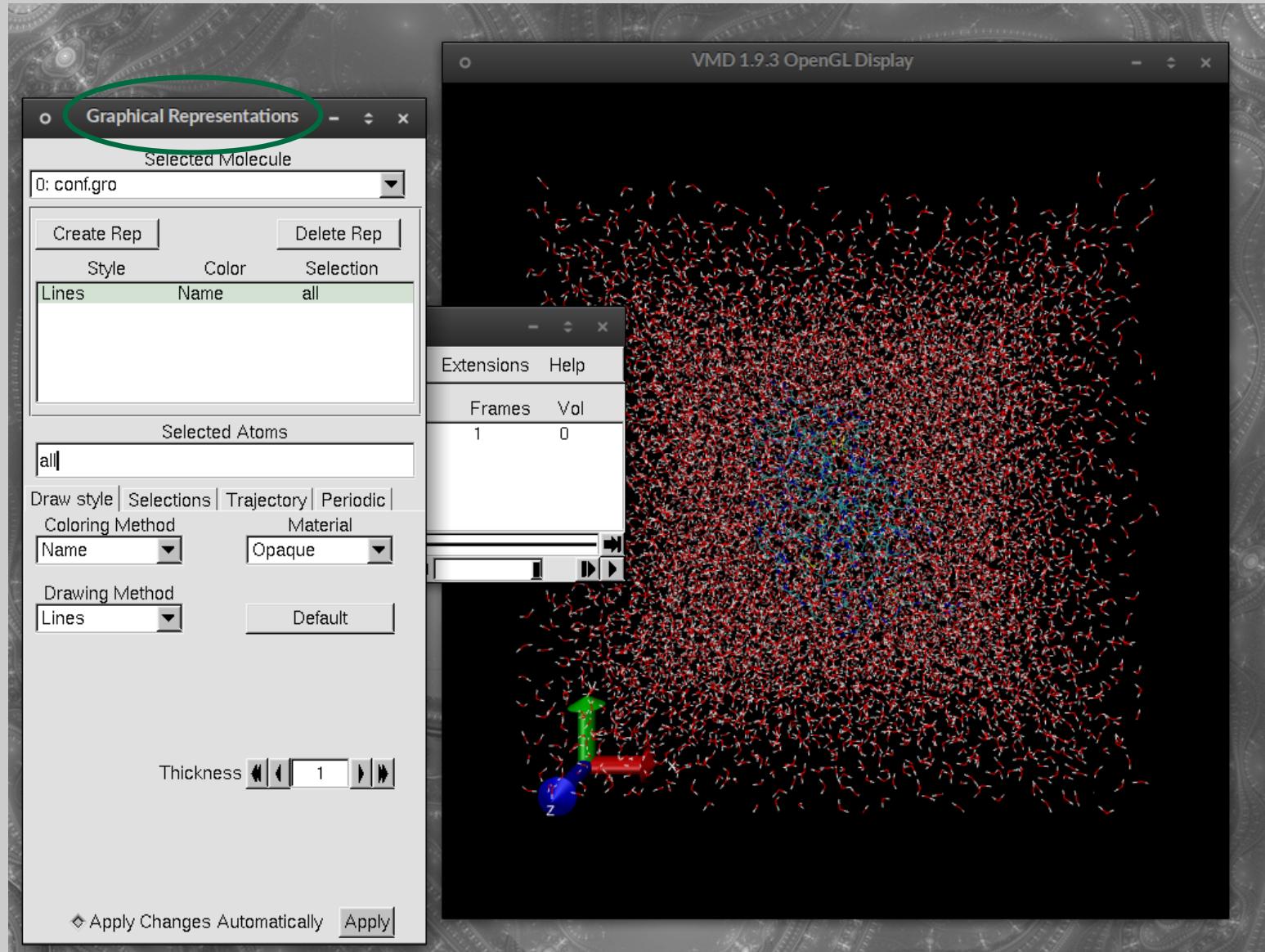
# Graphical representations



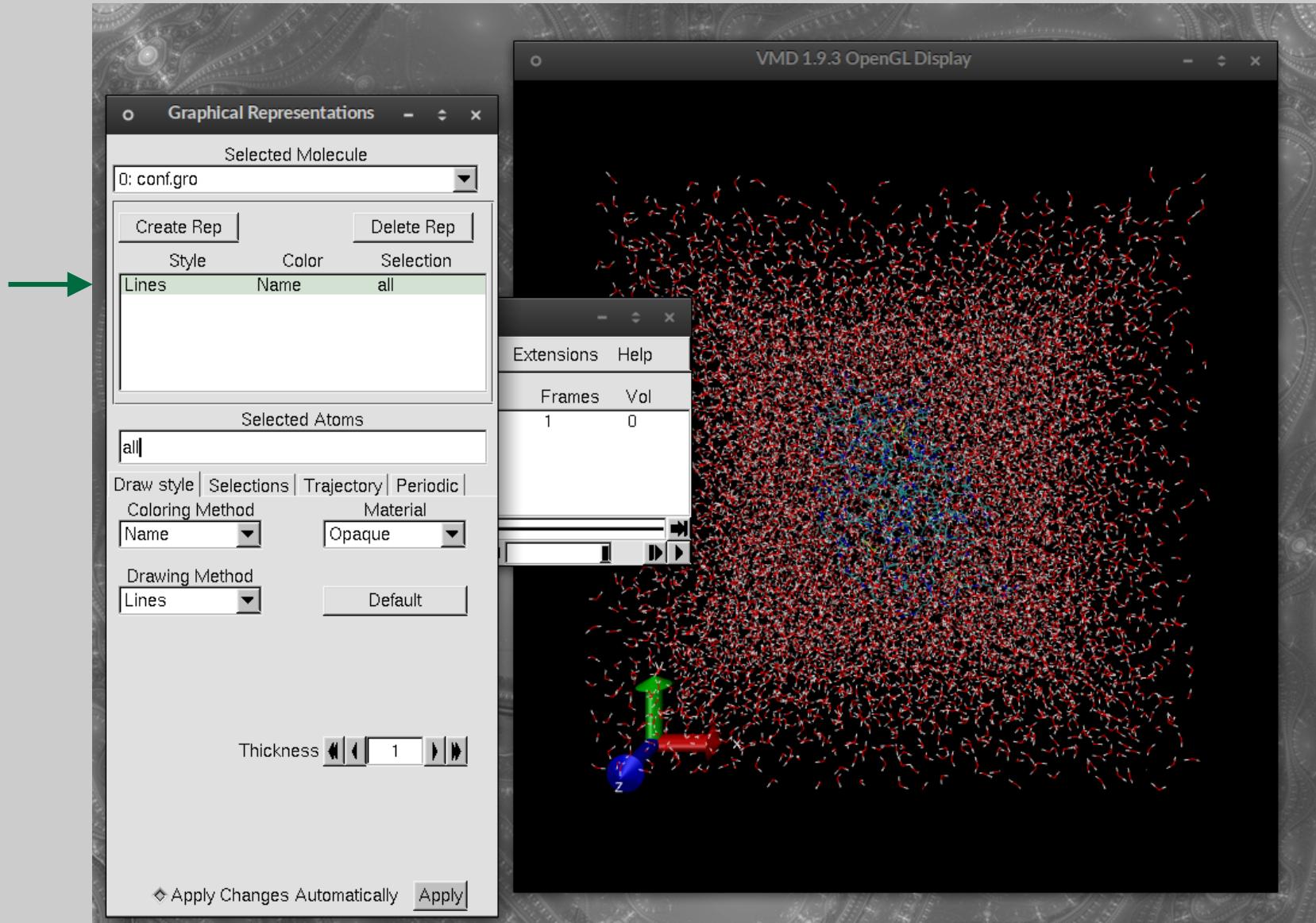
# Graphical representations



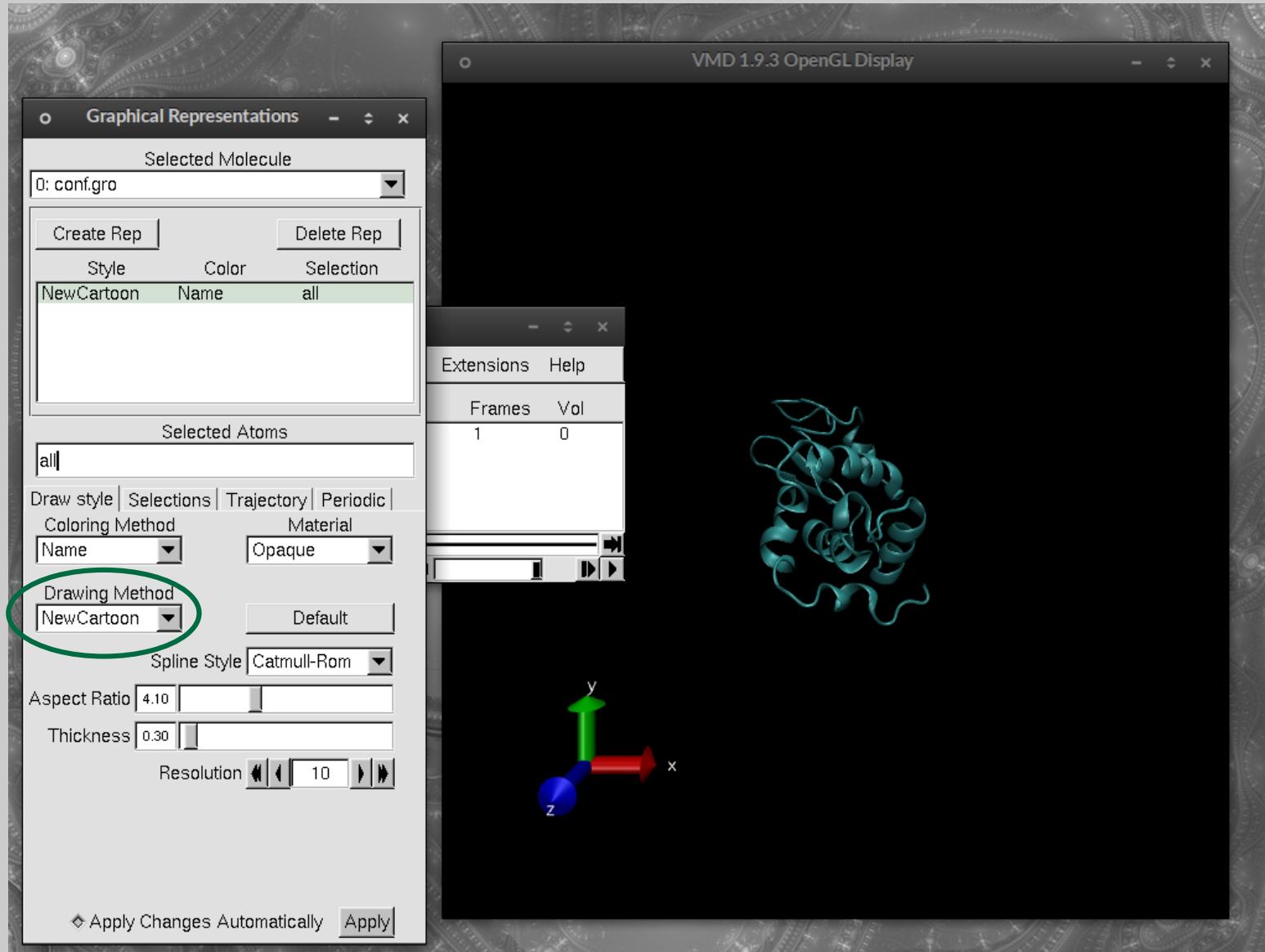
# Graphical representations



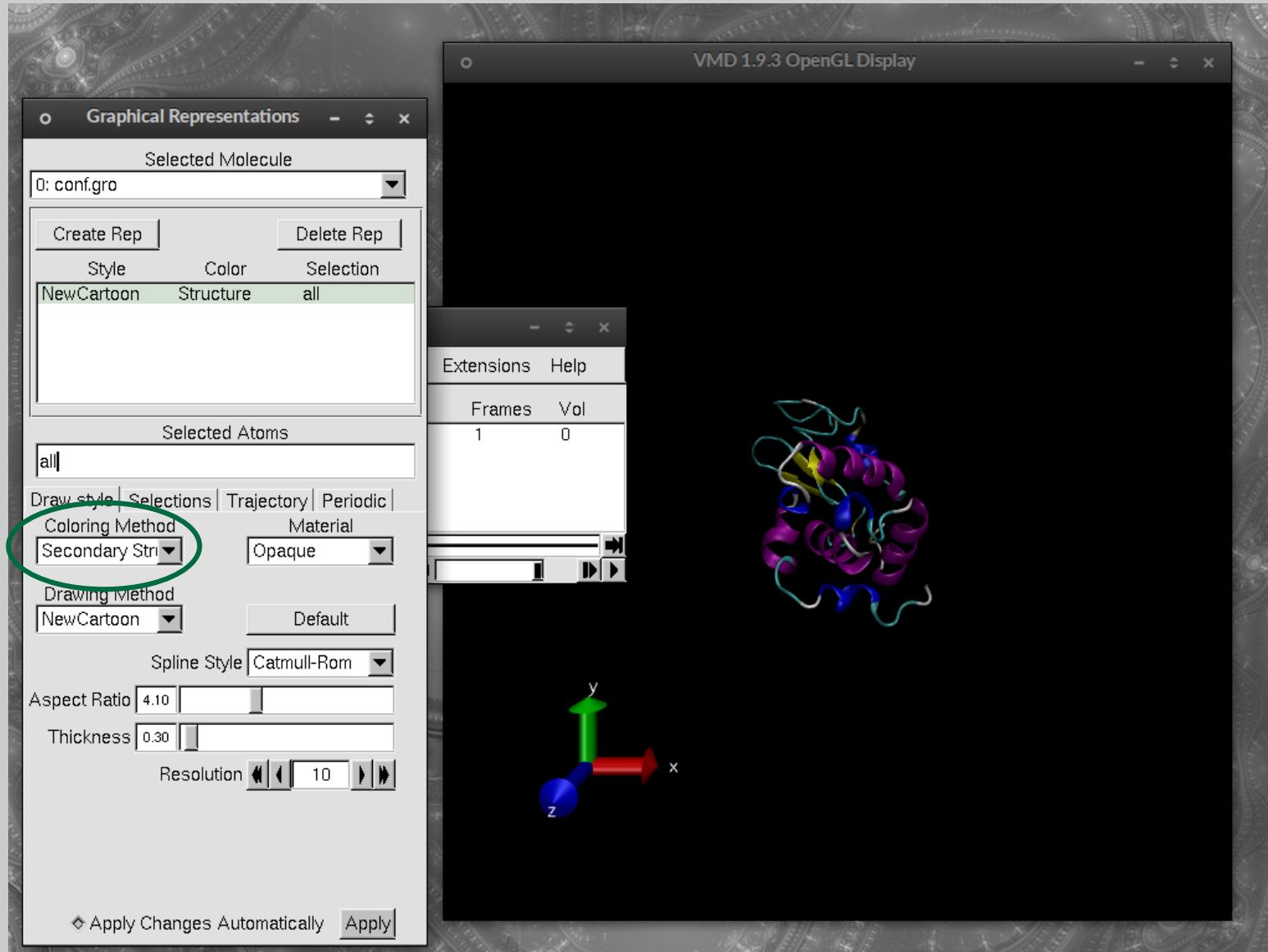
# Graphical representations



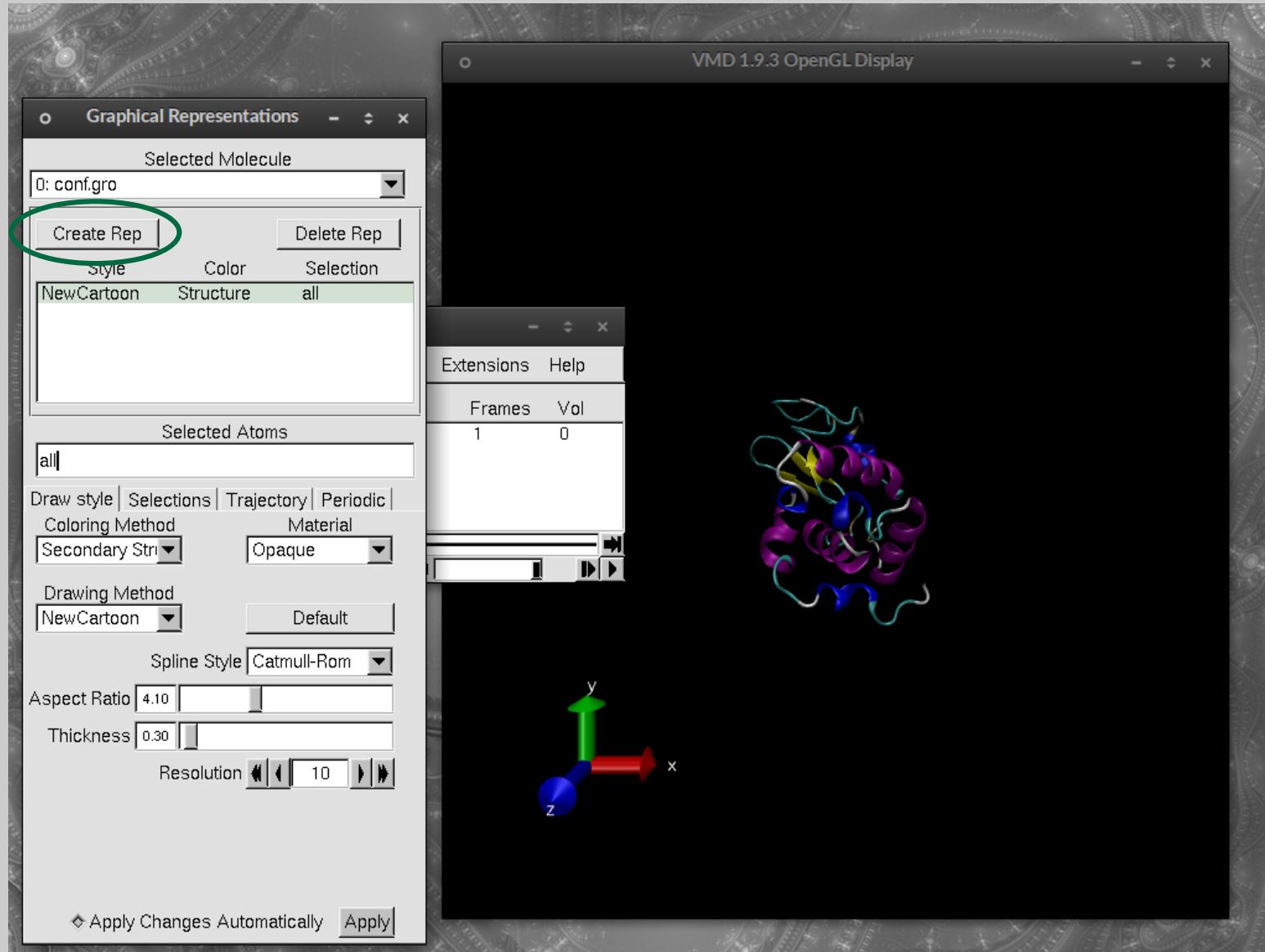
# Graphical representations



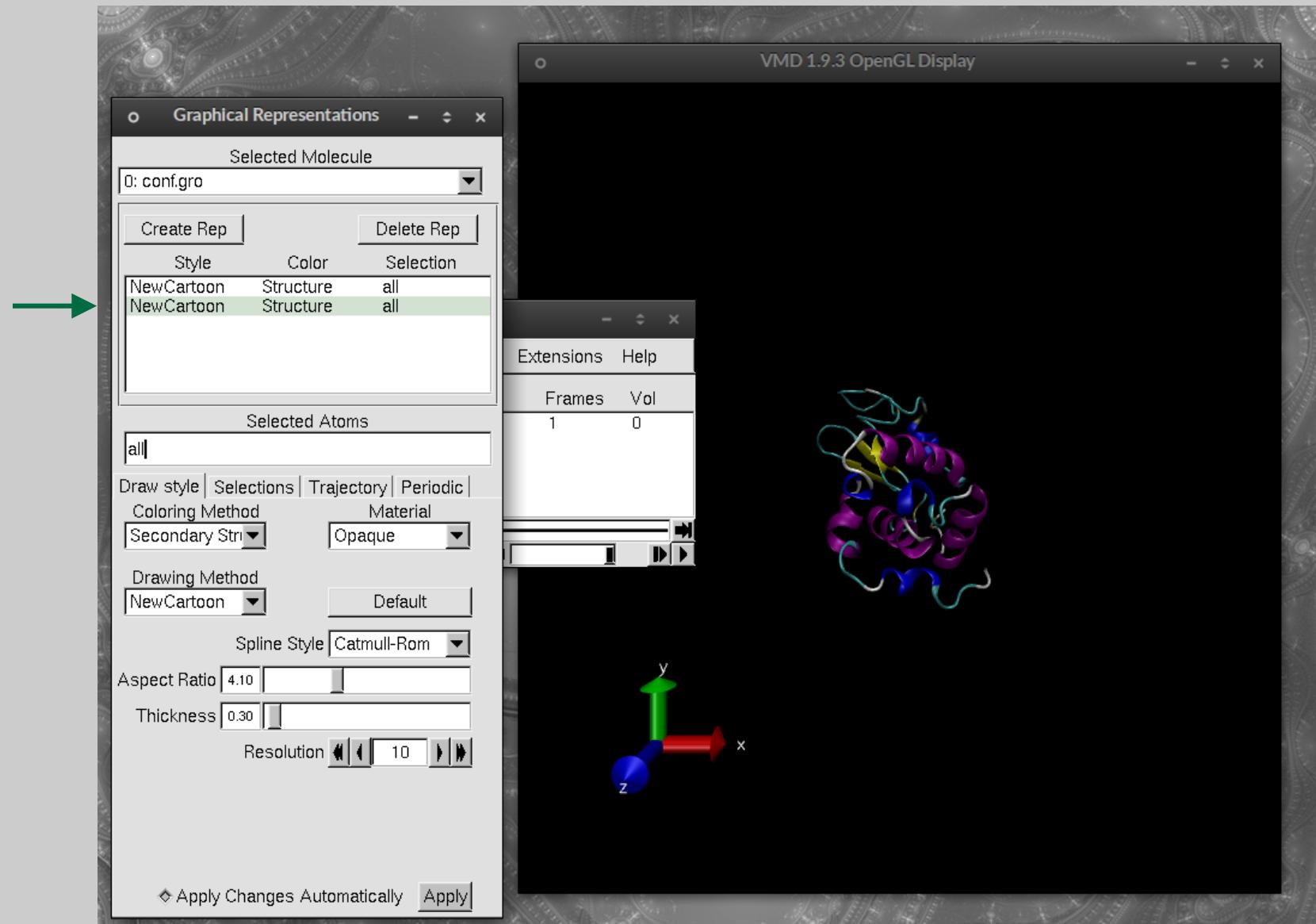
# Graphical representations



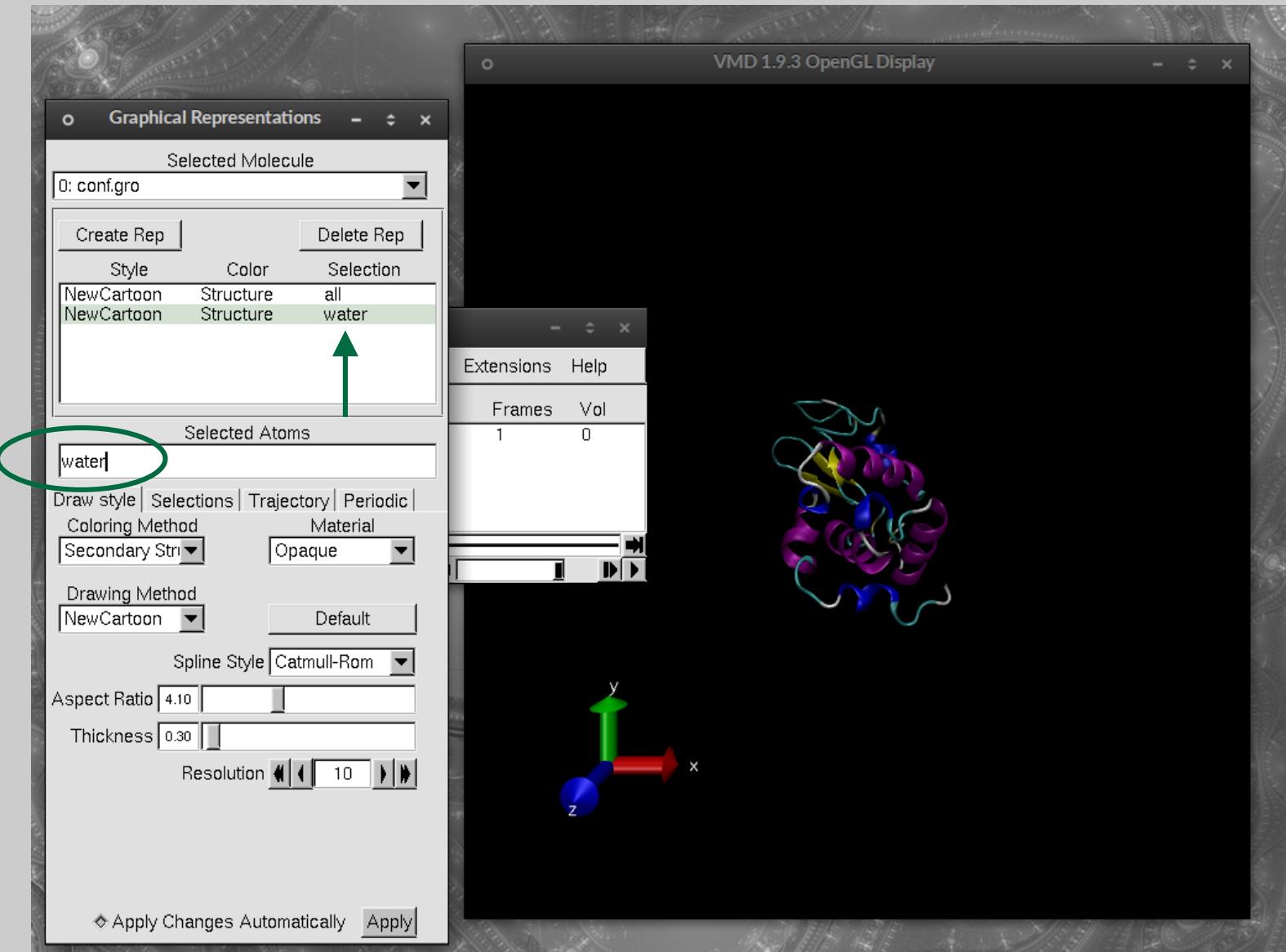
# Graphical representations



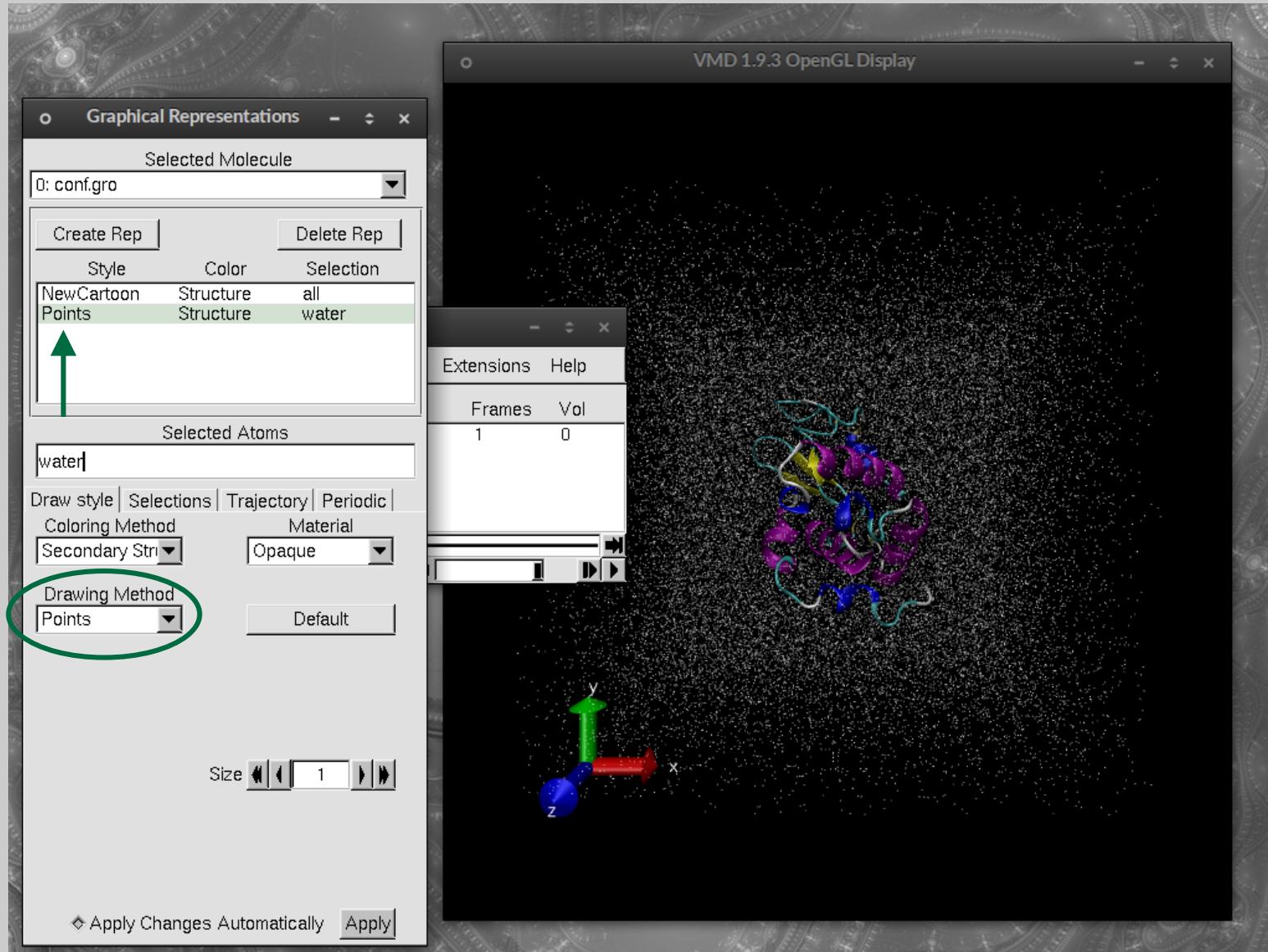
# Graphical representations



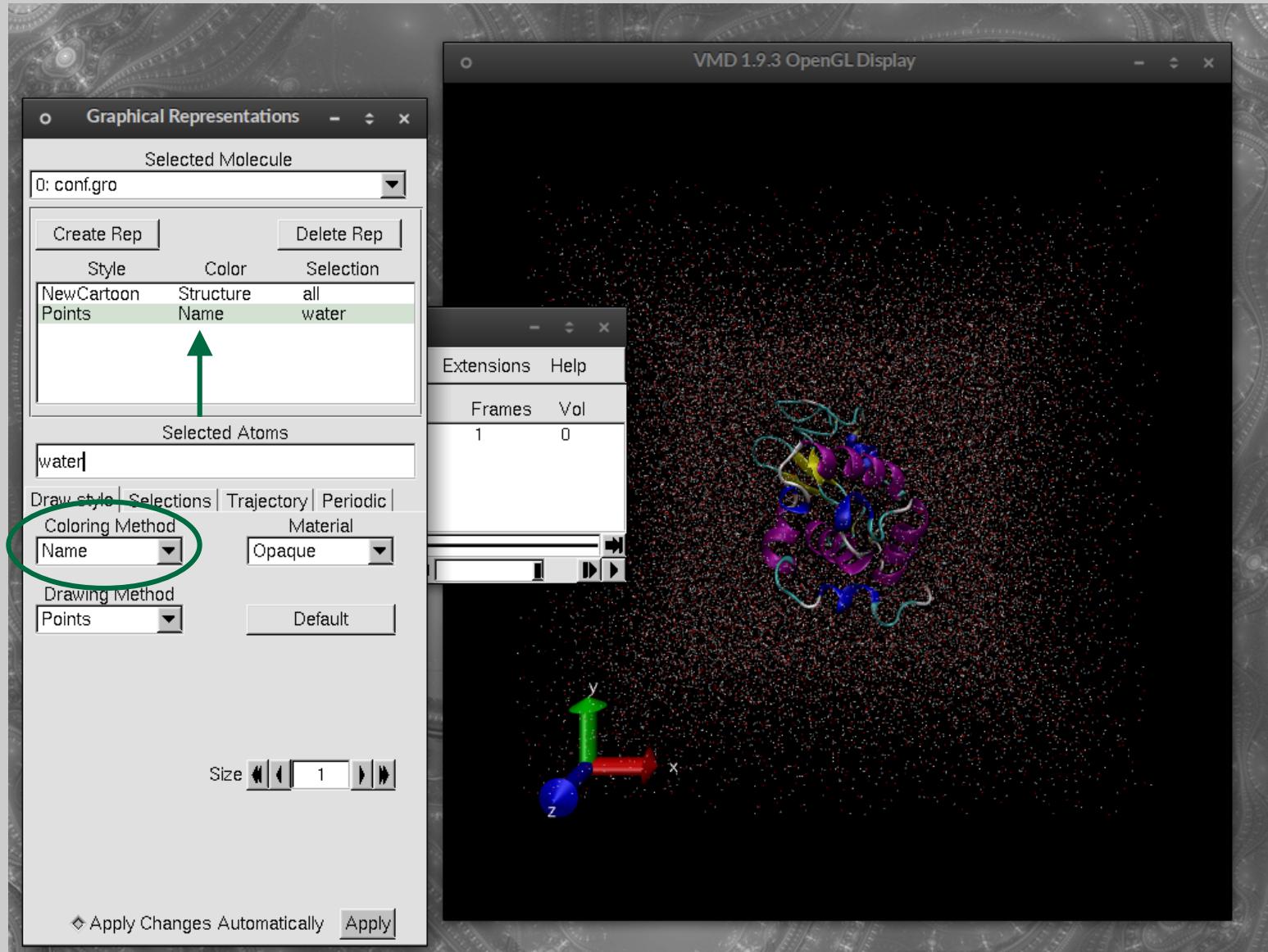
# Graphical representations



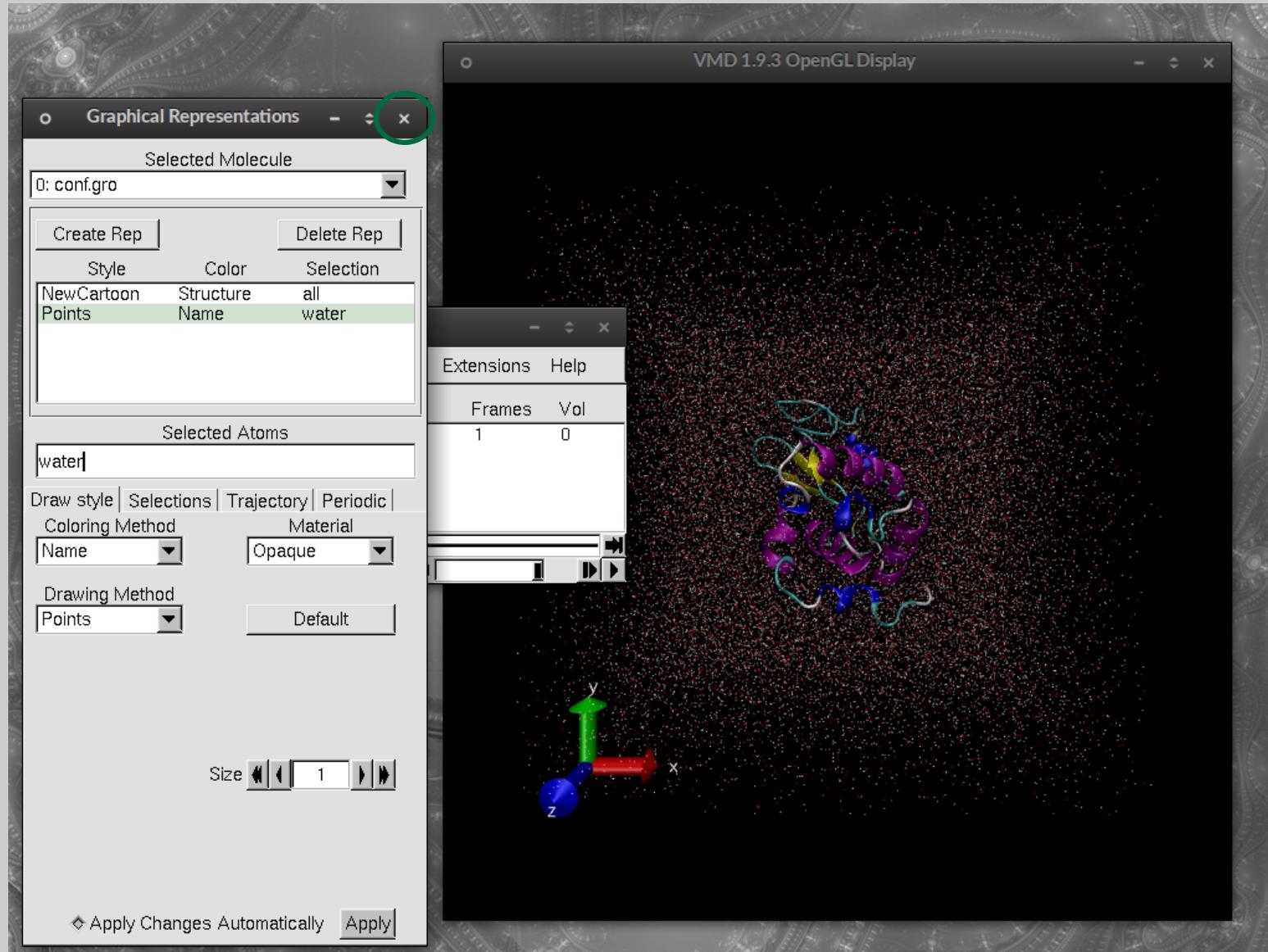
# Graphical representations



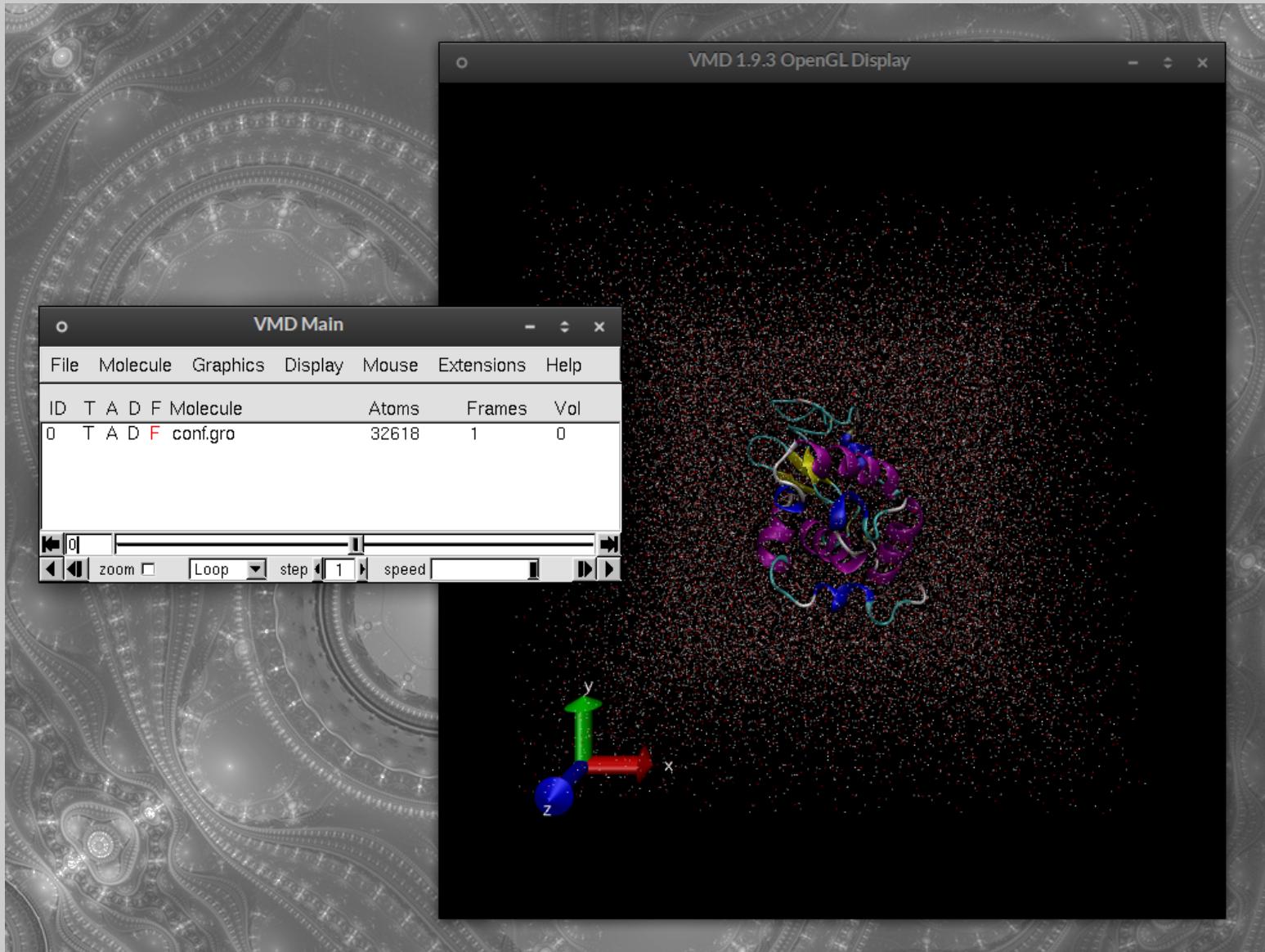
# Graphical representations



# Graphical representations



# Graphical representations



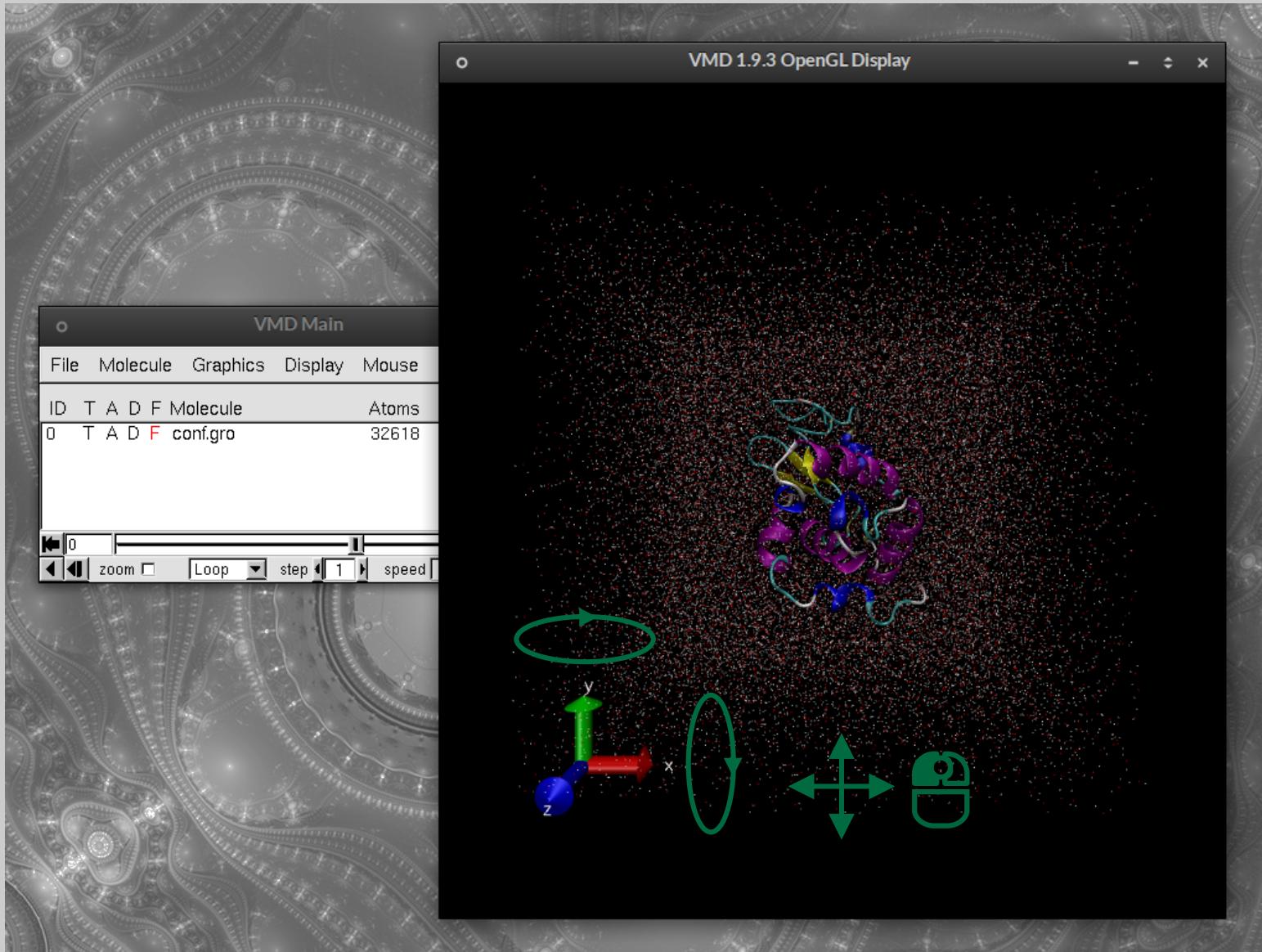
# Graphical representations

- Each structure (VMD calls it a “Molecule”) can have one or more graphical representations.
- There is a wide array of drawing styles.
  - Lines, licorice, VdW spheres, surface, balls and sticks, cartoons, etc.
- Representations can be coloured.
  - By atom name, chain, residue type (hydrophobic, polar, acid, basic), rainbow chain (ResID), secondary structure, B-factor (Beta)
- Atom selections can be used to display only part of a structure.
  - A selection applies to a single representation.
  - Powerful “Atom Selection Language”

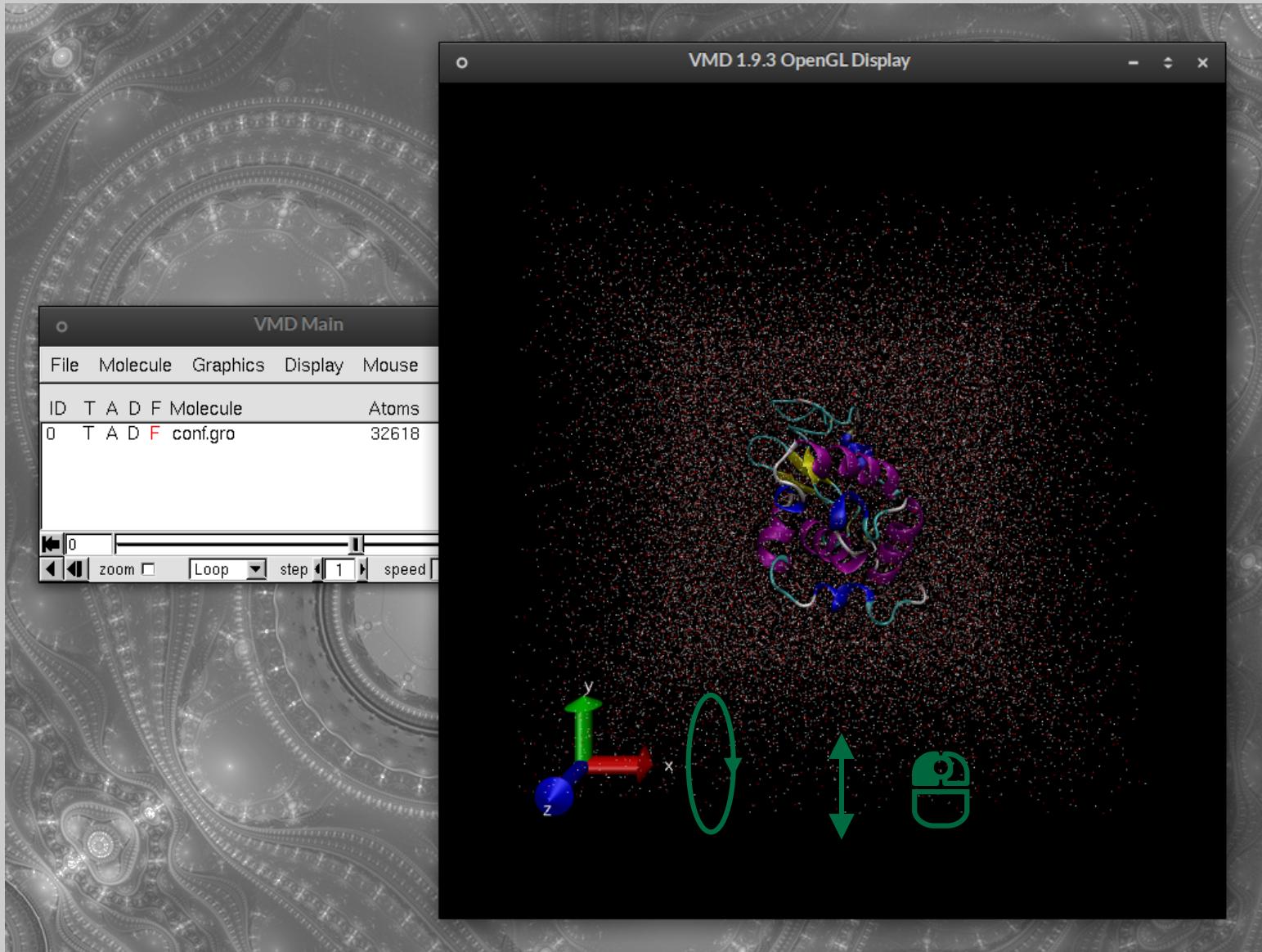
# Atom selection examples

name CA	Alpha carbons
resid 35	Residue 35
resid 35 to 40	Residues 35–40
name CA and resname ALA	Alanine alpha carbons
backbone	Protein backbone atoms
not protein	Any atom not in a protein
name "C.*"	Carbon atoms
mass < 5	Atoms heavier than 5 a.u.
numbonds = 2	Atoms bonded to 2 other atoms
abs(charge) > 1	Atoms with a large net charge
z < 6 and z > 3	Atoms between 3 and 6 Å in z
within 5 of name FE	Atoms within 5 Å of iron atoms
protein within 5 of nucleic	Protein atoms close to nucleic acids
water within 3 of protein	Water close to proteins

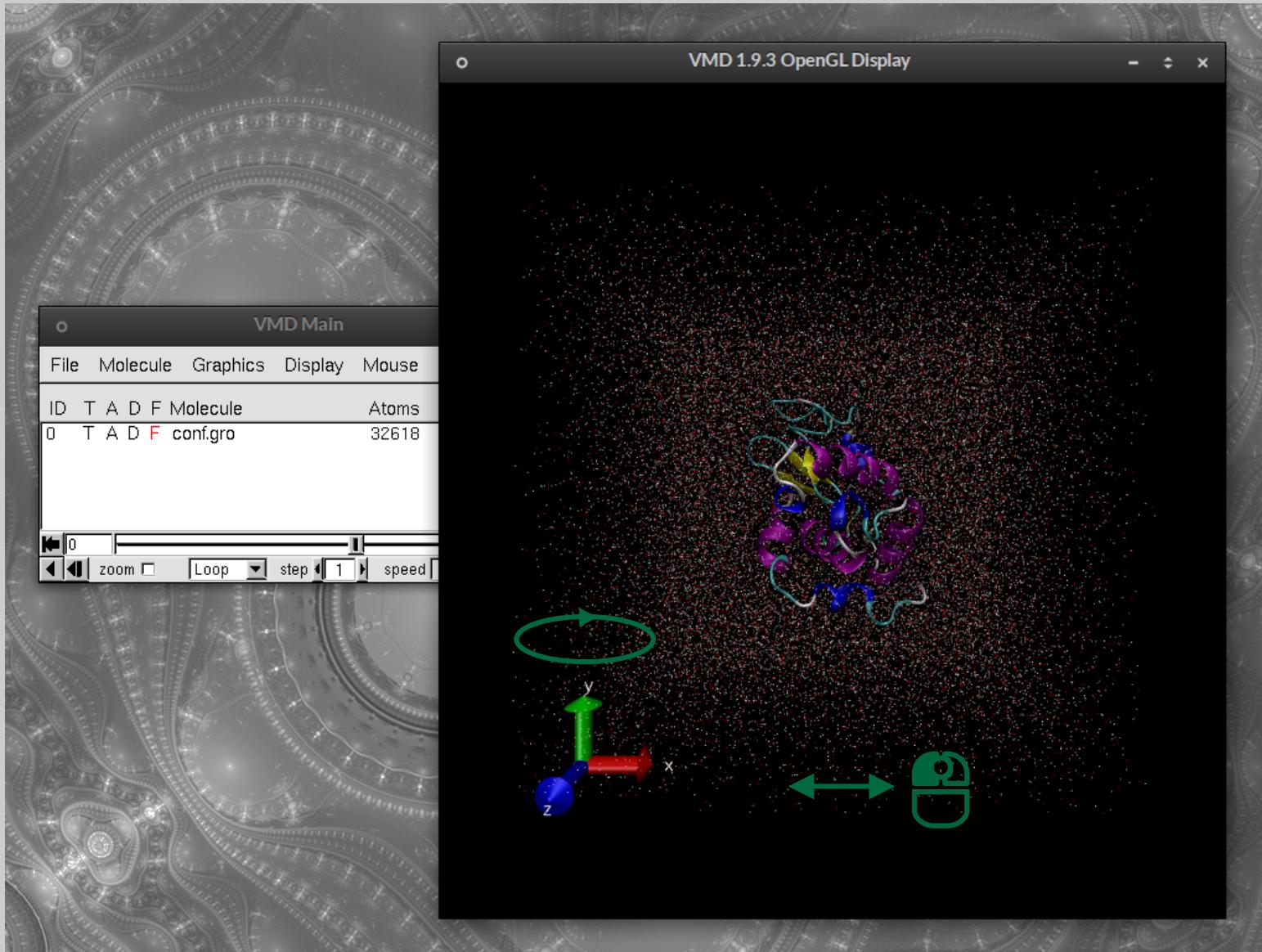
# 3D navigation



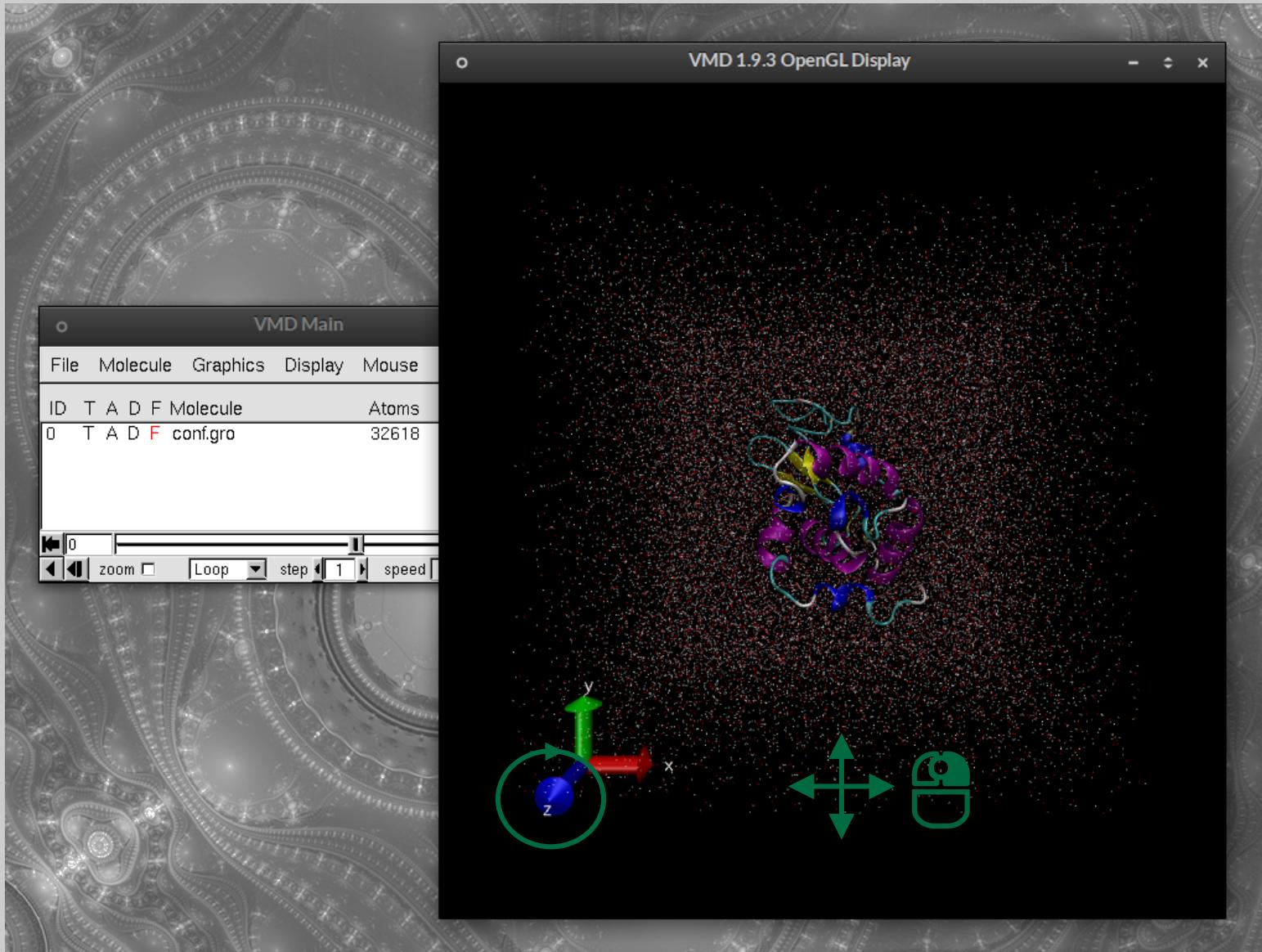
# 3D navigation



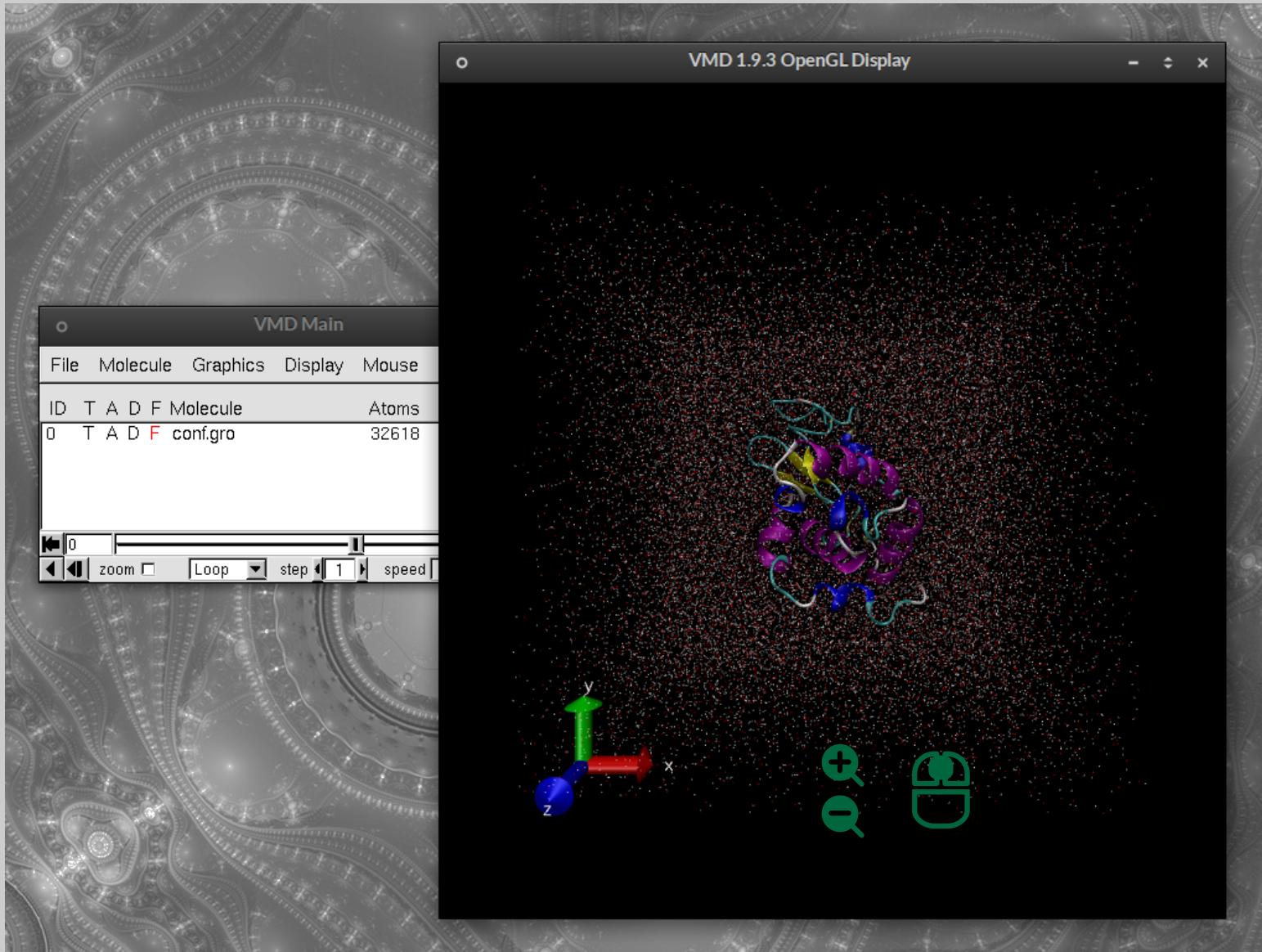
# 3D navigation



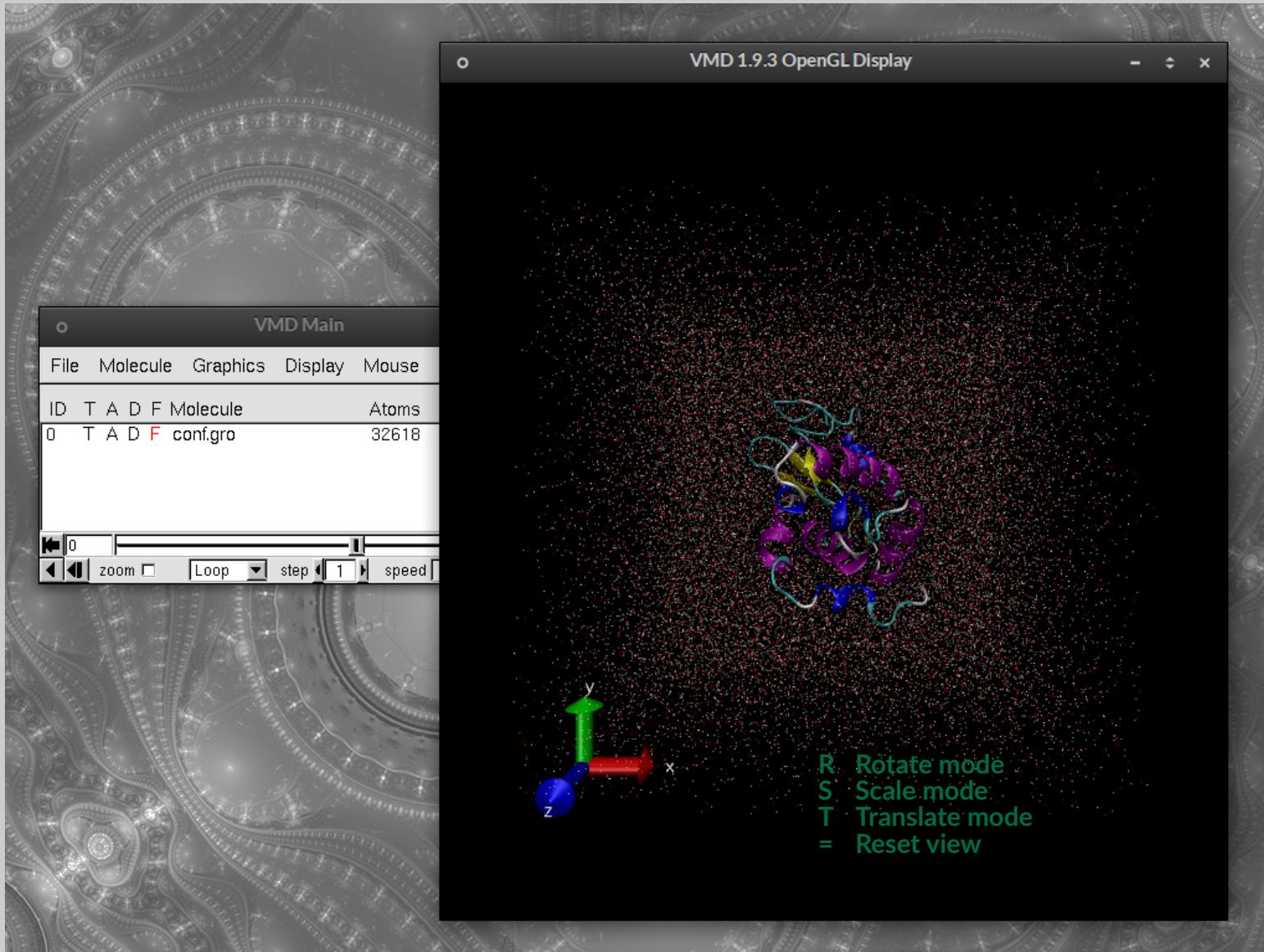
# 3D navigation



# 3D navigation



# 3D navigation

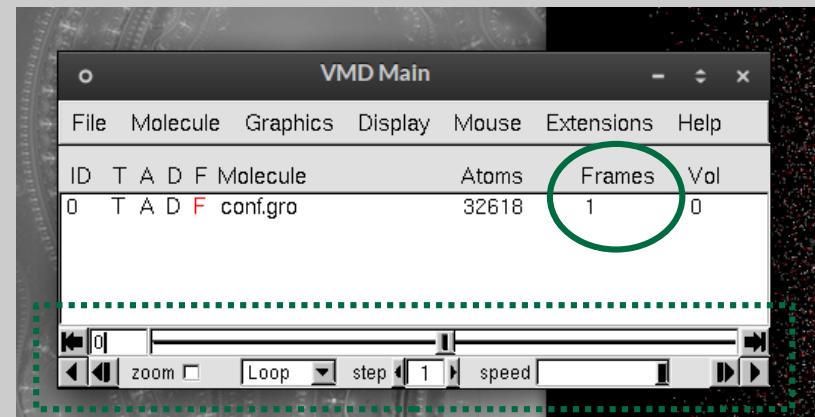


# Contents

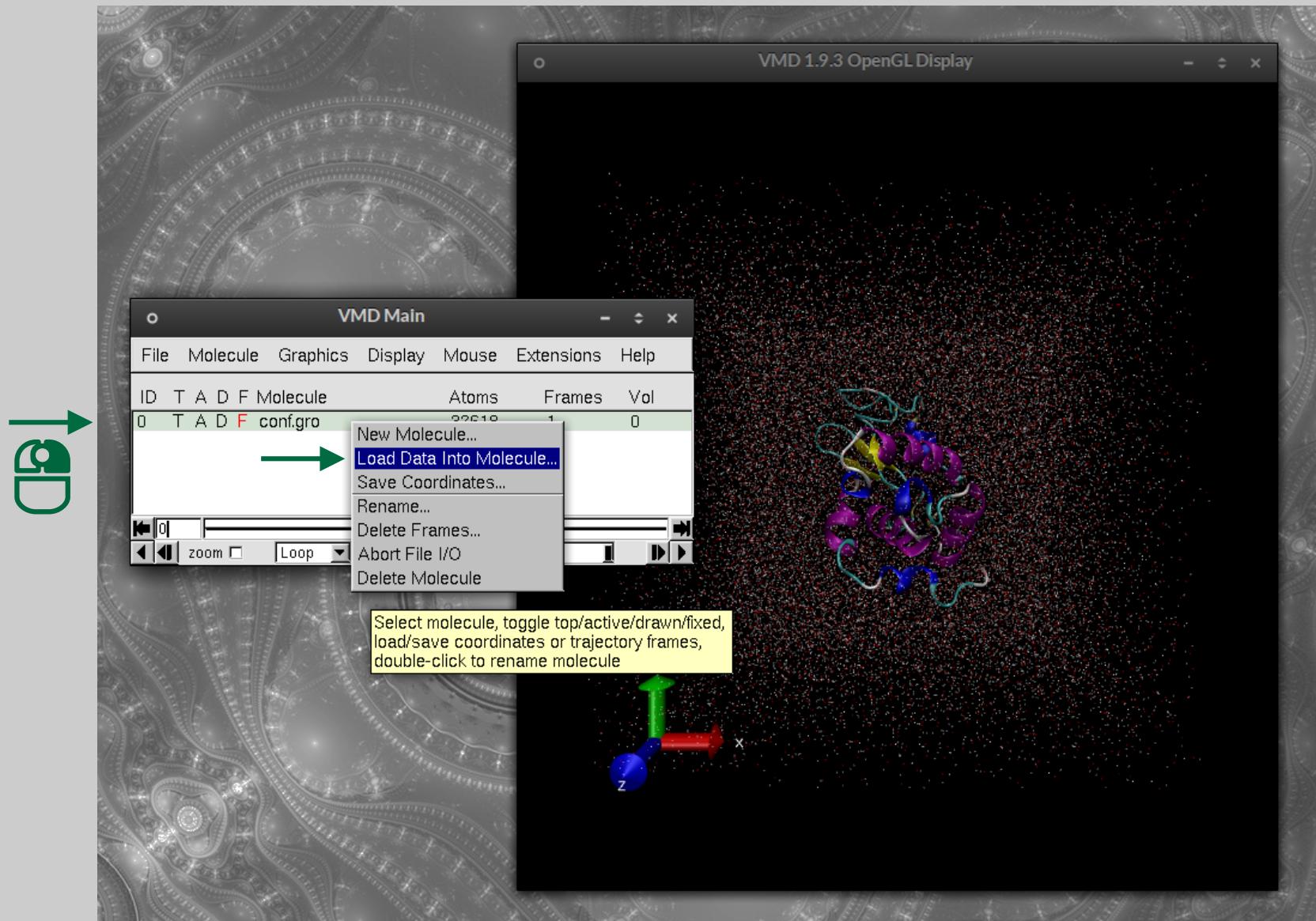
- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- Combining trajectories, movies, and scripting

# Trajectory basics

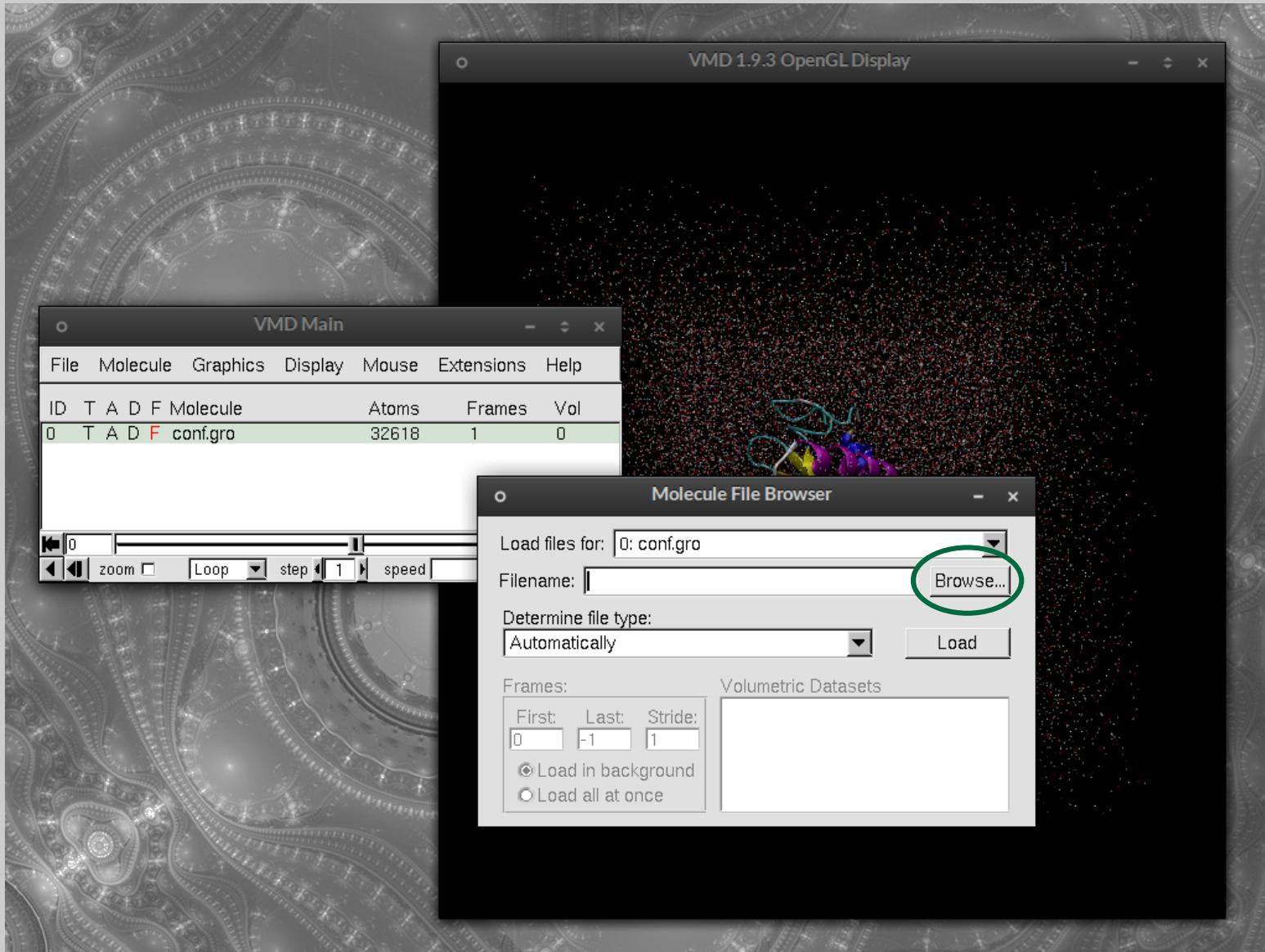
- VMD can read molecular trajectories in a variety of formats:
  - DCD (NAMD, CHARMM)
  - XTC, TRR (Gromacs)
  - CRD (Amber)
- For interactive visualisation, analysis, and making movies
- Most trajectory controls are available from the “Main” window.
- Loading trajectories is done through the “Molecule File Browser” (as for structures).



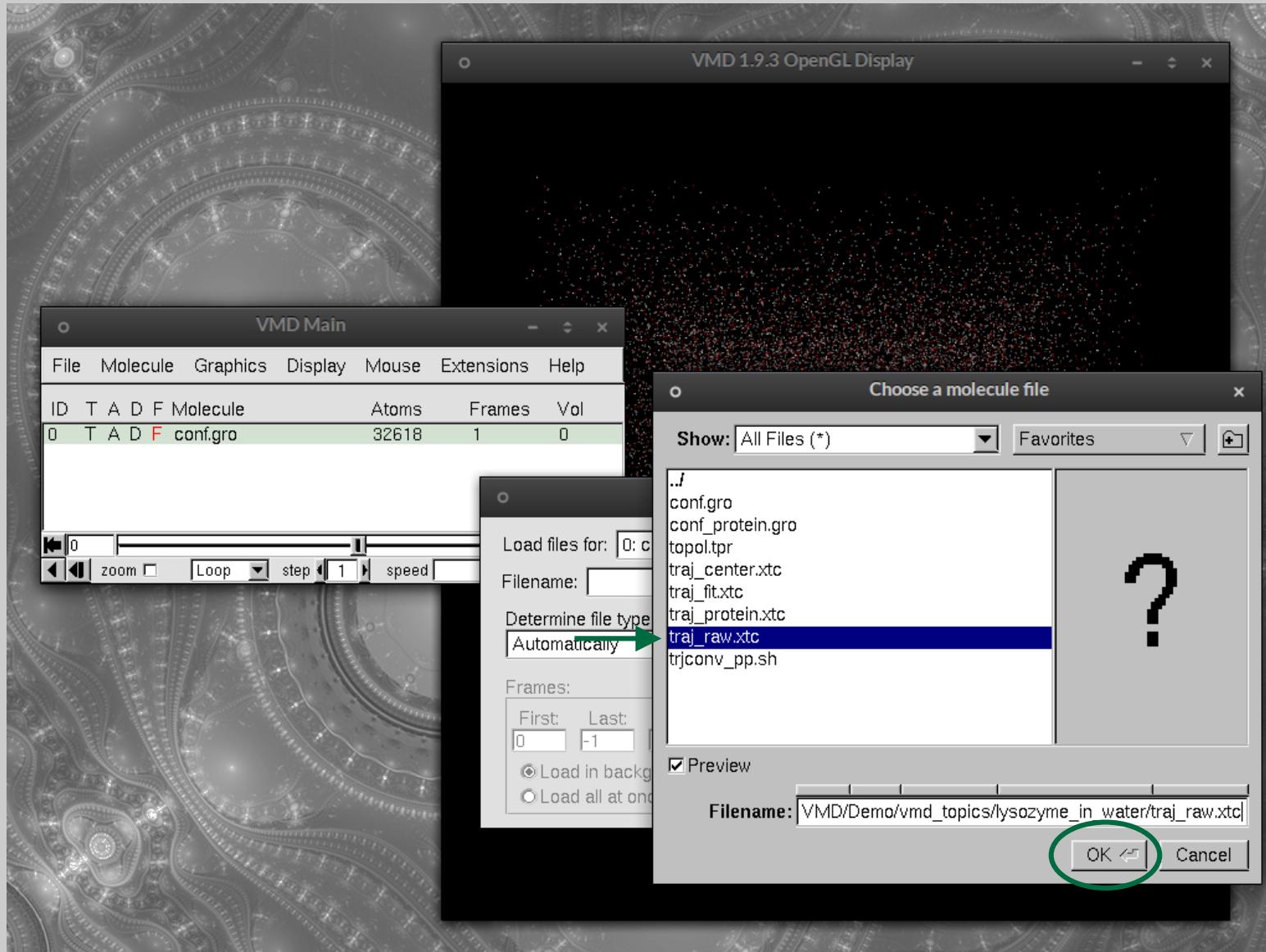
# Trajectory basics



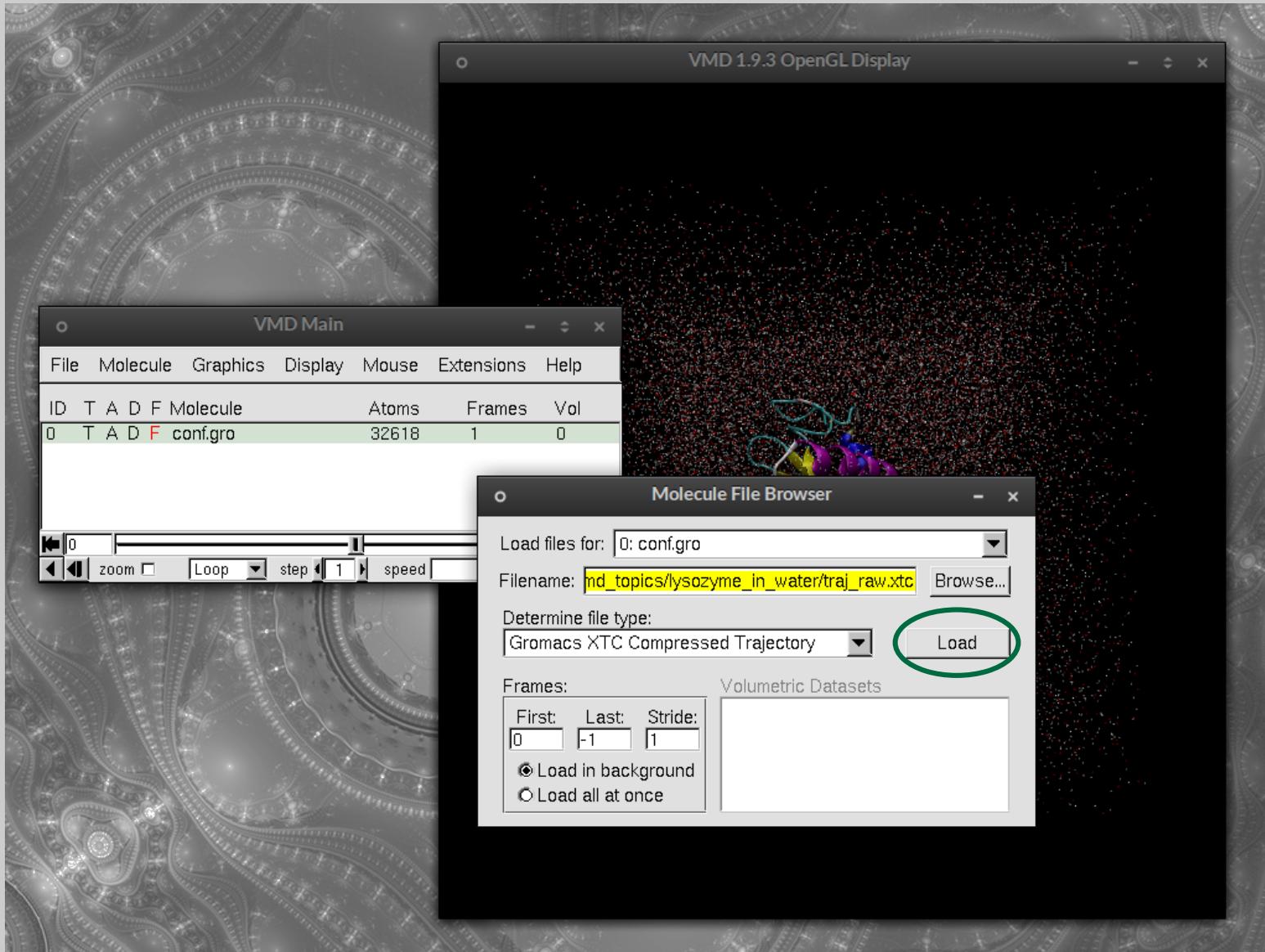
# Trajectory basics



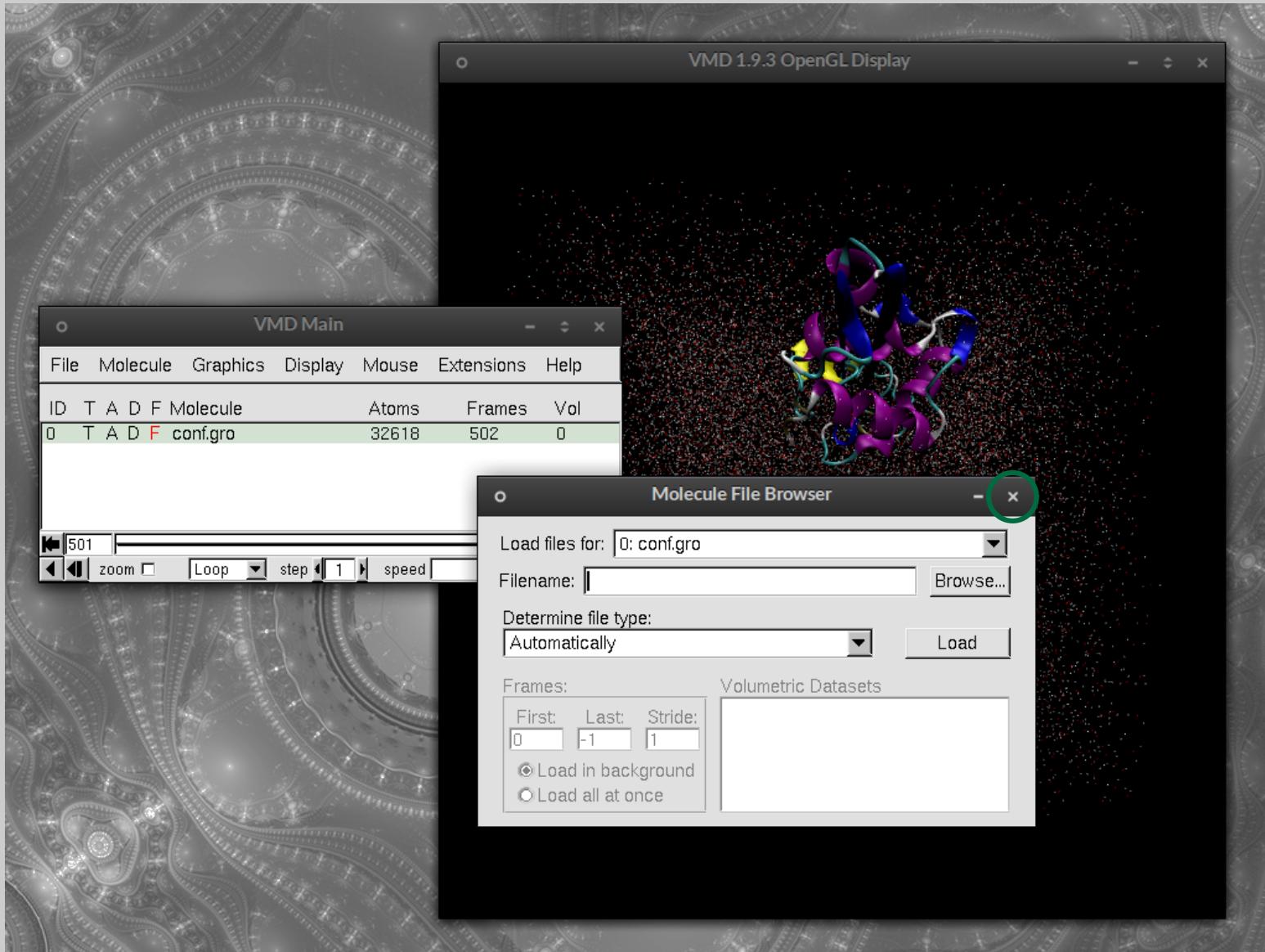
# Trajectory basics



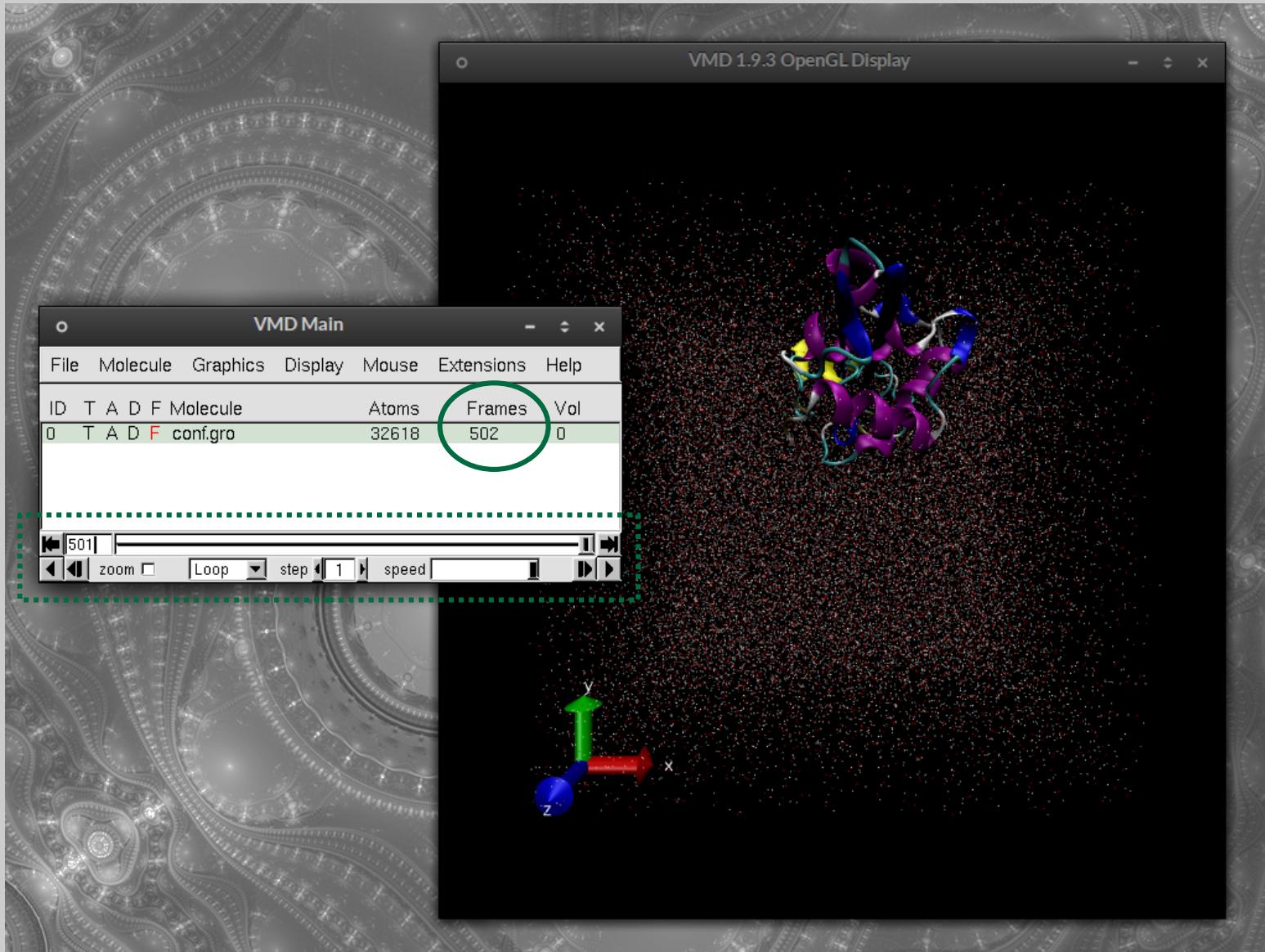
# Trajectory basics



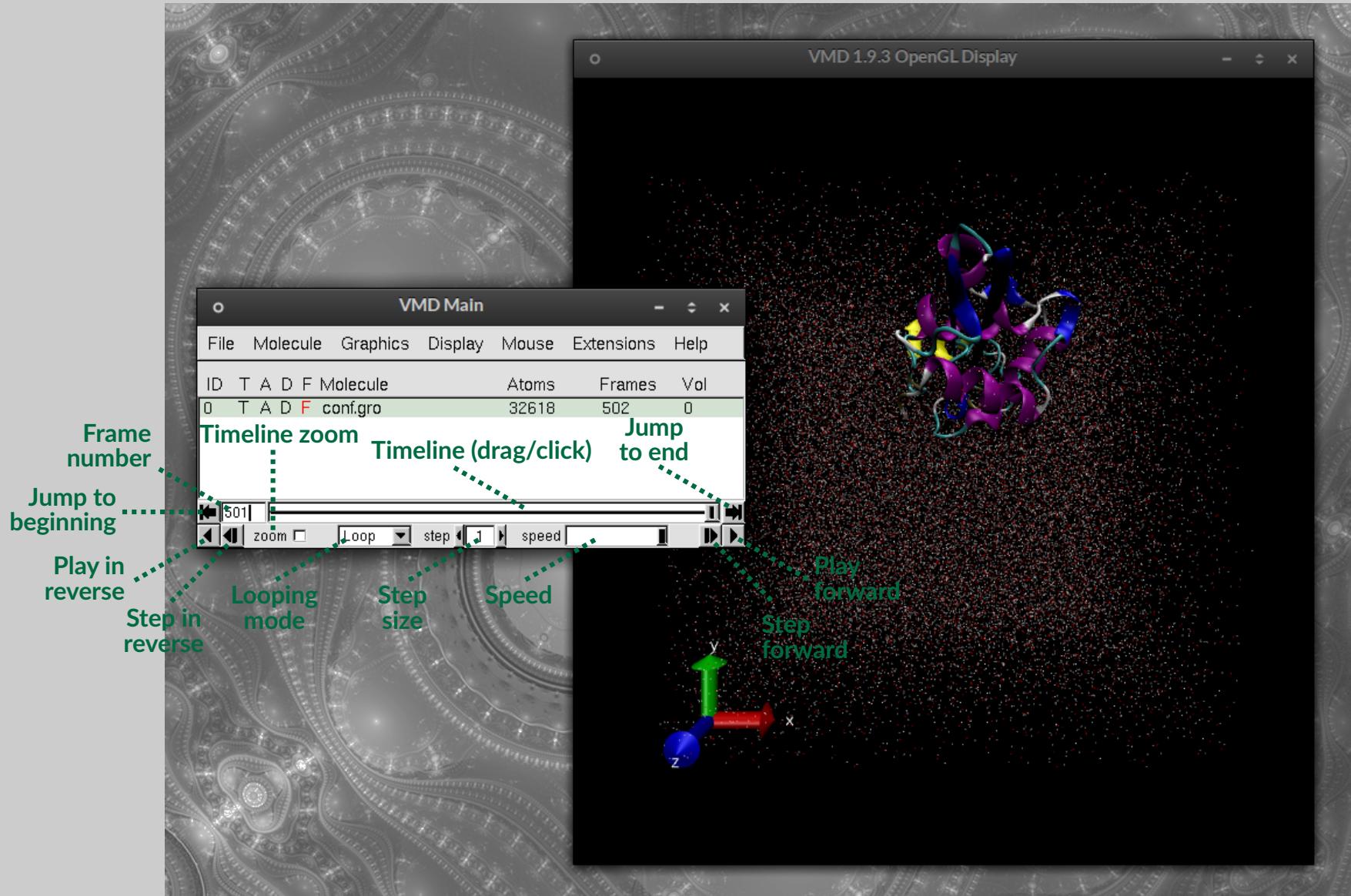
# Trajectory basics



# Trajectory basics

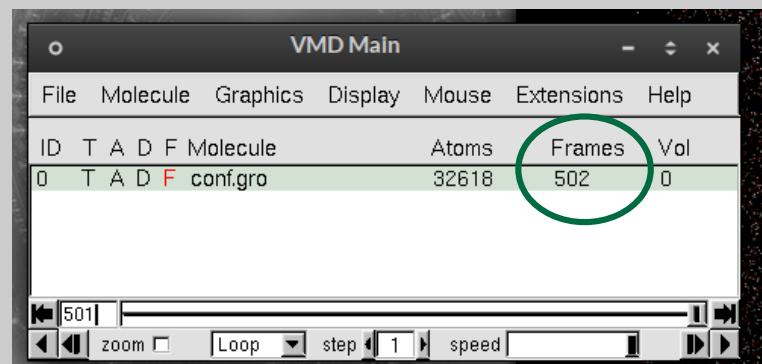


# Trajectory basics



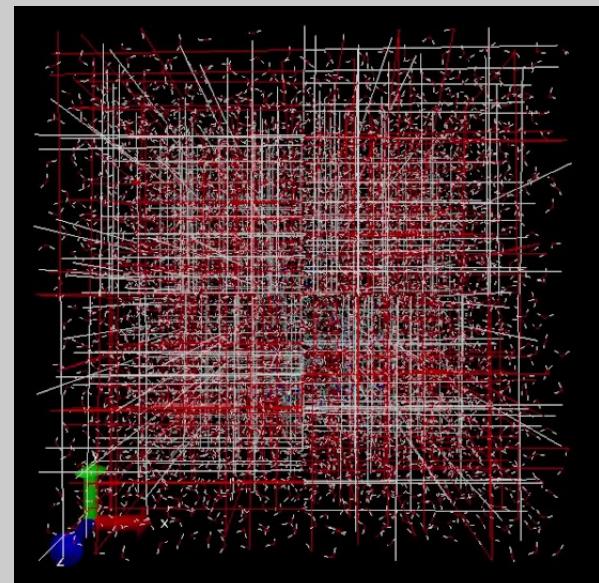
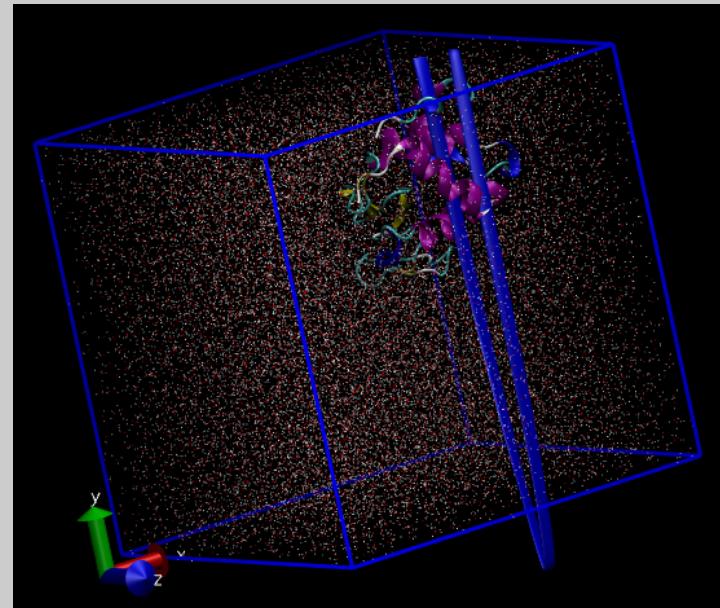
# Trajectory troubleshooting

- Unexpected number of frames
  - VMD has no separate “topology” vs “trajectory” concepts.
  - The first frame is the original “topology”.
  - Trajectories typically store initial coordinates as a frame.
  - 2 more frames than expected



# Trajectory troubleshooting

- Lines/bonds all over the place
  - Atoms are bonded across periodic boundary conditions (PBC).
  - Trajectory post-processing is required to make molecules whole.



# Trajectory troubleshooting

- Number of atom mismatch between topology and trajectory
  - The files must match exactly.
  - If atoms are removed from the trajectory by post-processing, a corresponding topology must also be created.
- Out of memory errors or crash
  - VMD loads all trajectory frames in memory.
  - Compressed trajectories require more memory than file size.
    - The compressed XTC in this demo is 60M but requires 250M of memory, or about 4 times the file size.
  - Post-processing can be used to reduce file size.
    - Removing atoms outside of a selection
    - Removing frames outside of a time range
    - Skipping frames

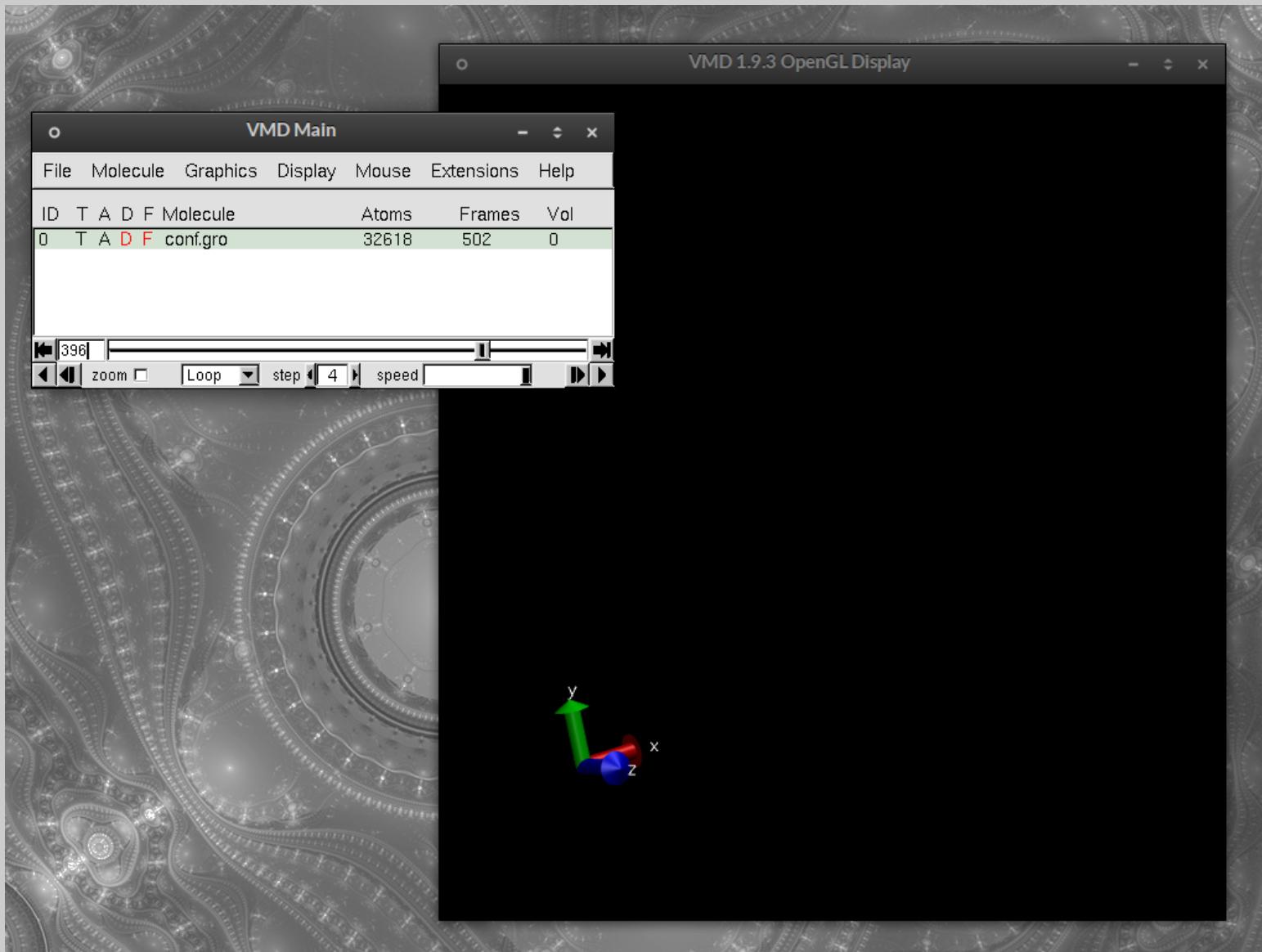
# Trajectory post-processing

- Transforming raw MD trajectories through operations
  - Making molecules whole (PBC treatment)
  - Centering a selection
  - Fitting (RMSD minimisation) a selection
  - Removing atoms outside of a selection
  - Removing frames outside of a time range
  - Skipping frames
- Makes certain analyses easier
- Makes visualisation easier

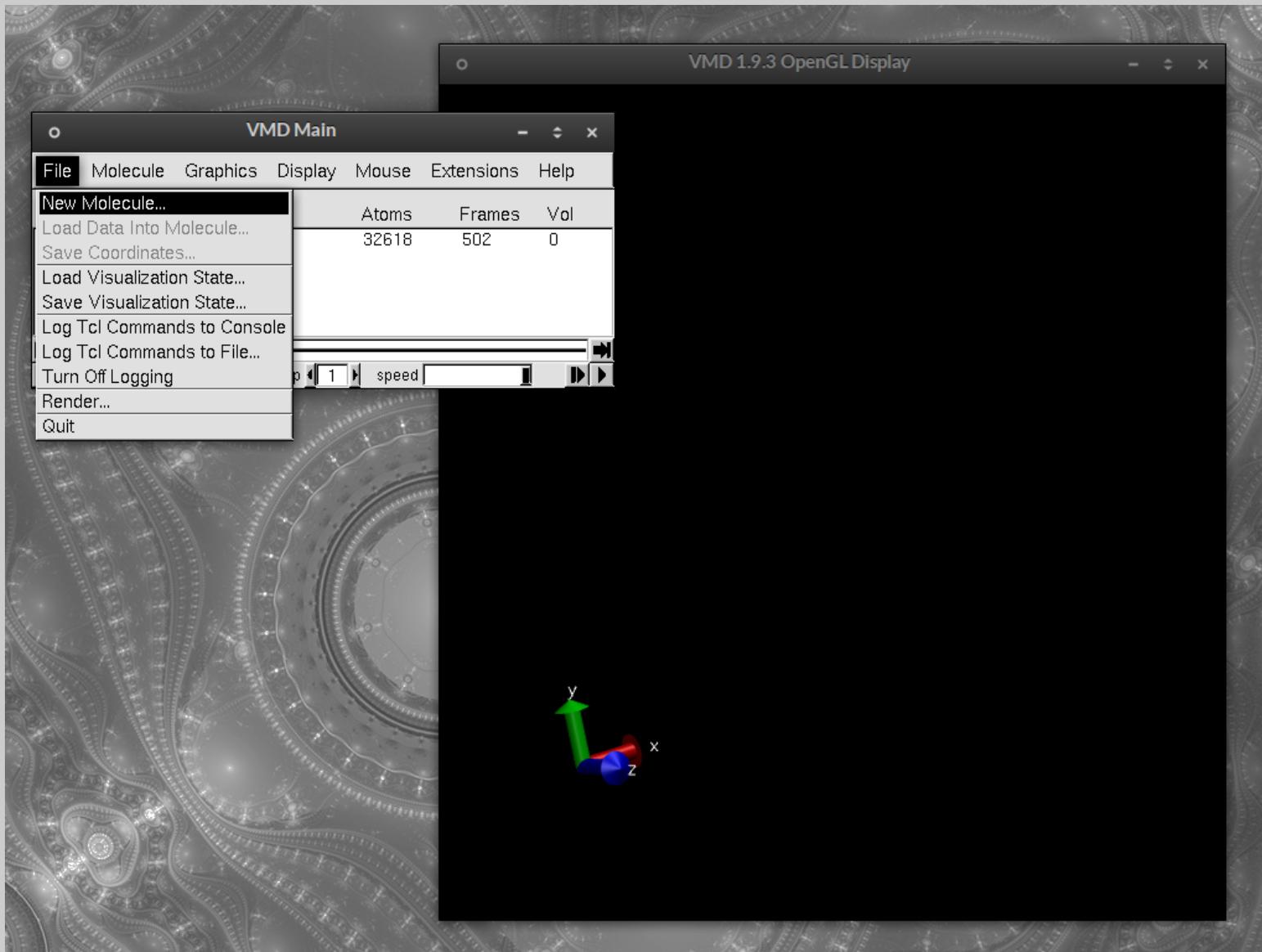
# Trajectory post-processing

```
1 #!/usr/bin/env bash
2
3 # trjconv_pp.sh - Post-process raw GROMACS trajectory
4
5 # Center protein, and make molecules whole
6 gmx trjconv -s "topol.tpr" -f "traj_raw.xtc" -center -pbc mol \
---    -o "traj_center.xtc" <<-eof
|     Protein
|     System
10 eof
11
12 # Apply RMSD-minimising fit to protein
13 gmx trjconv -s "topol.tpr" -f "traj_center.xtc" -fit progressive \
---    -o "traj_fit.xtc" <<-eof
|     Protein
|     System
17 eof
18
19 # Remove non-protein atoms
20 gmx trjconv -s "topol.tpr" -f "traj_fit.xtc" \
---    -o "traj_protein.xtc" <<-eof
|     Protein
23 eof
24
25 # Create topology for protein-only trajectory
26 gmx trjconv -s "topol.tpr" -f "conf.gro" \
---    -o "conf_protein.gro" <<-eof
|     Protein
29 eof
```

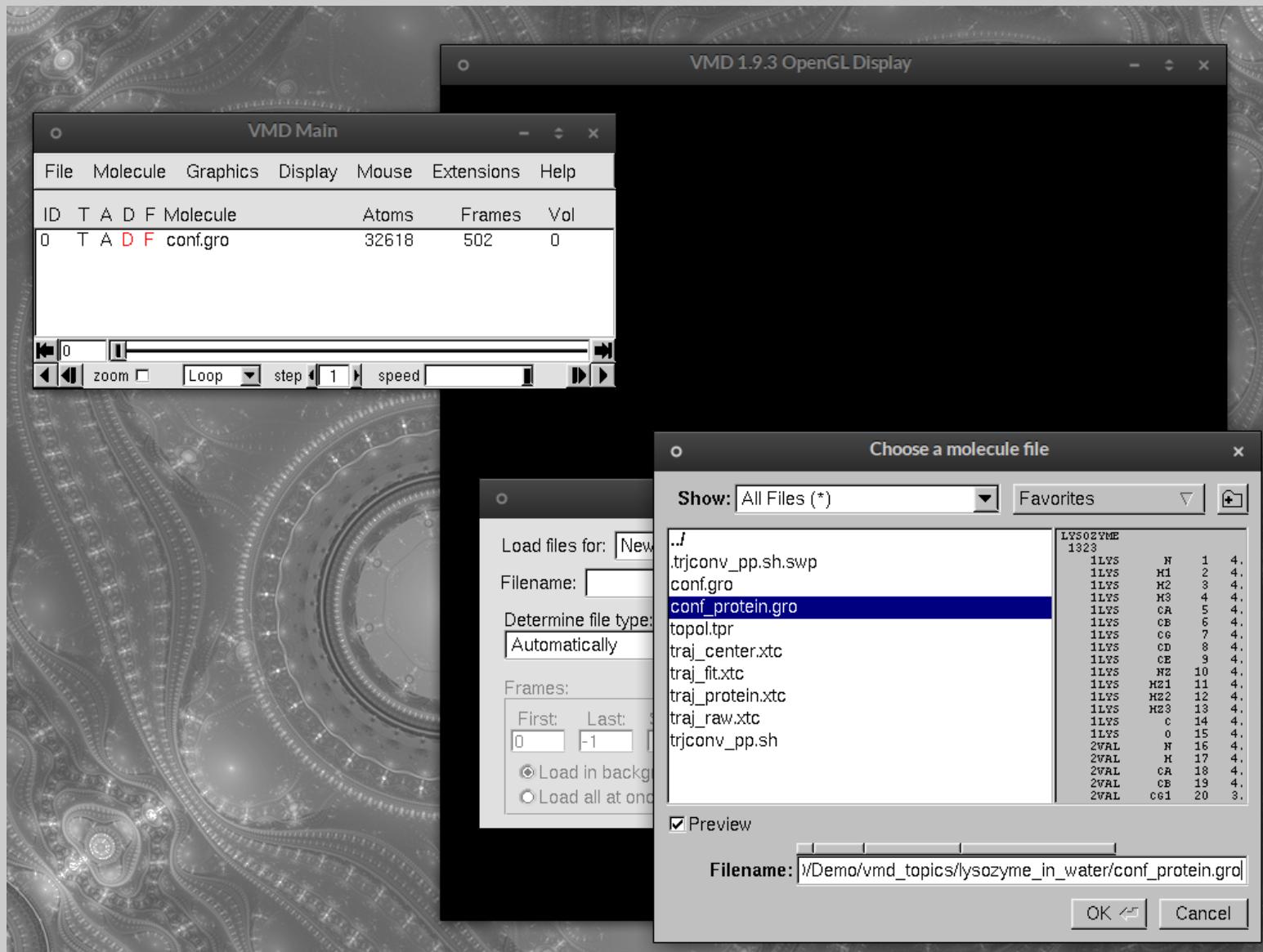
# Trajectory post-processing



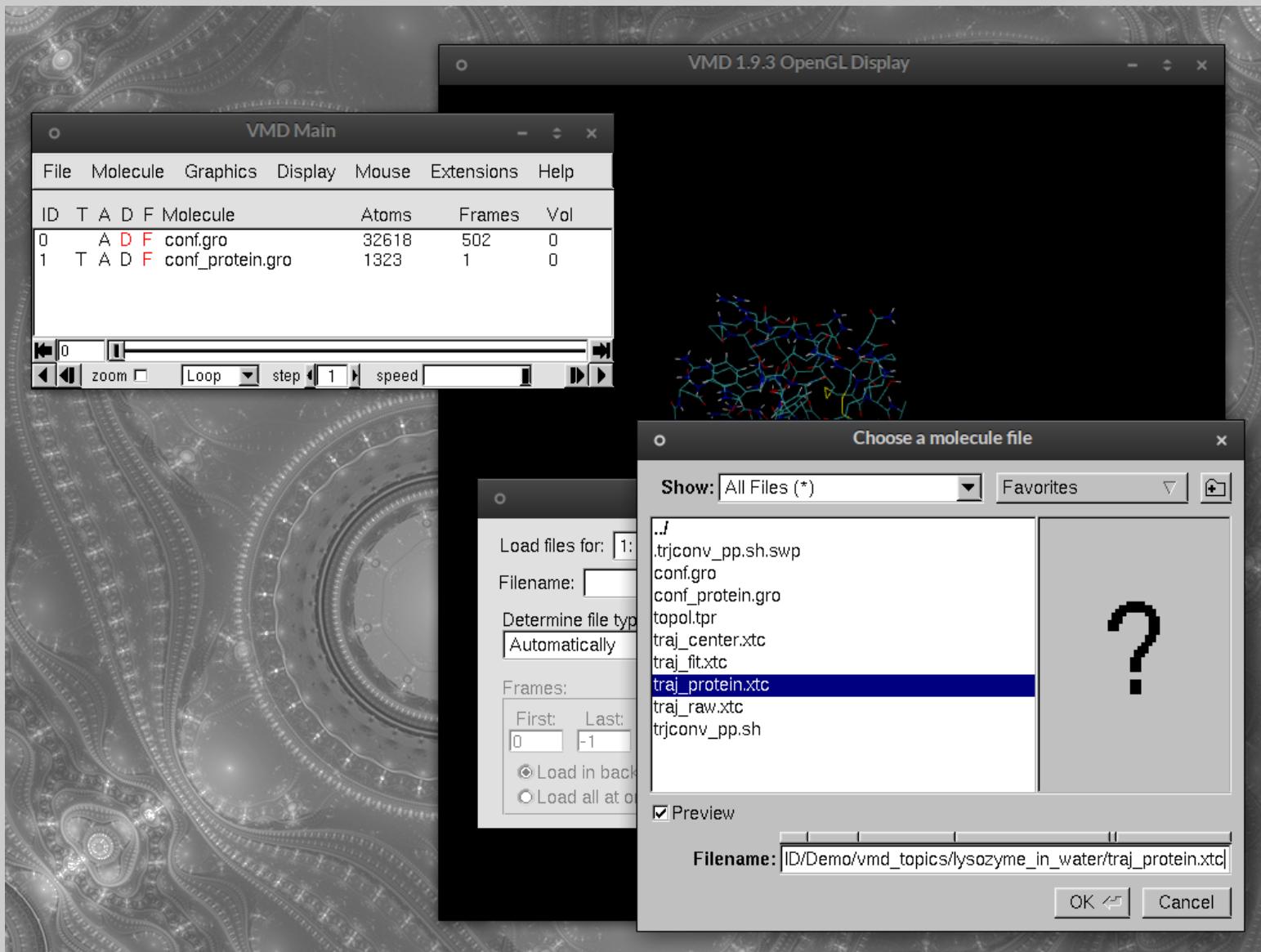
# Trajectory post-processing



# Trajectory post-processing



# Trajectory post-processing



# Contents

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- Combining trajectories, movies, and scripting

# Movie-making strategies

- Three-step process
  - Animating a trajectory or structure
  - Rendering movie frames
  - Encoding the movie from the separate frames
- Animation and rendering
  - Movie Maker
  - Scripting
- Encoding
  - Movie Maker
  - External tools

# Animating

- Movie Maker
  - Molecular trajectory
  - Single structure with changing viewpoint (XY 8-loop or Y rotation)
  - Easy
  - Limited
- Movie Maker and scripting
  - User-defined procedure to step through a trajectory, change the viewpoint, etc.
  - More flexible
- Pure scripting
  - Total control

# Rendering

- Movie Maker
  - No need to deal explicitly with rendering
    - Frames are created in a working directory, then deleted
    - For internal encoding
  - Individual frames in JPEG, TARGA
    - For encoding with external tools
    - Additional step, but better quality
- Scripting
  - Create frames using commands
  - Encoding through external tools

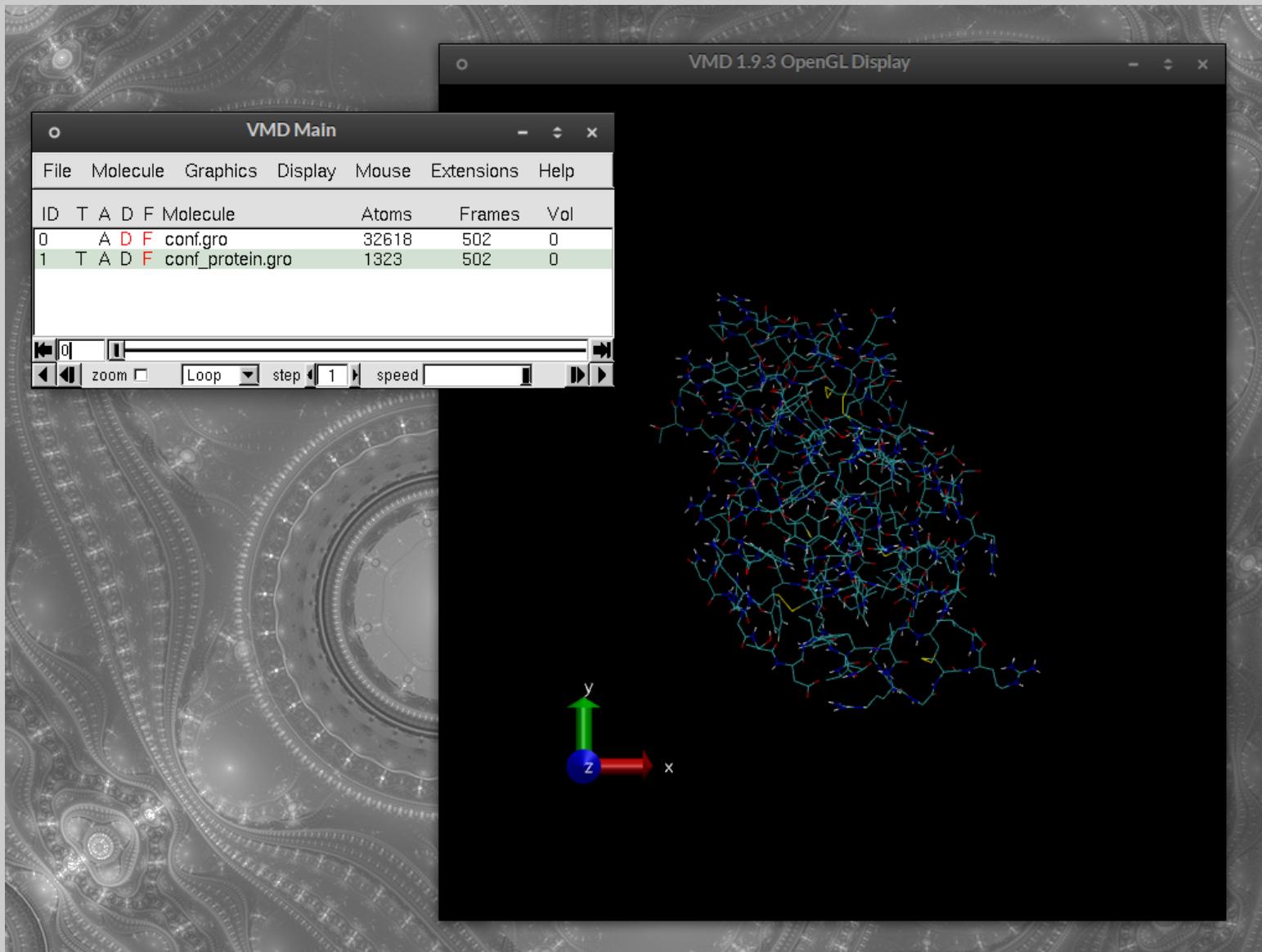
# Encoding

- Movie Maker
  - Movies in MPEG-1, MPEG-2, animated GIF
  - Using external *ppmtompeg*, *mencoder*, *ffmpeg*
  - Somewhat low-quality movies
  - Old codecs...
- External tools
  - Encoding (*ffmpeg* to produce MP4 with H.265 codec)
  - Post-processing frames (ImageMagick to add text, graphs...)
  - Add audio (Audacity)

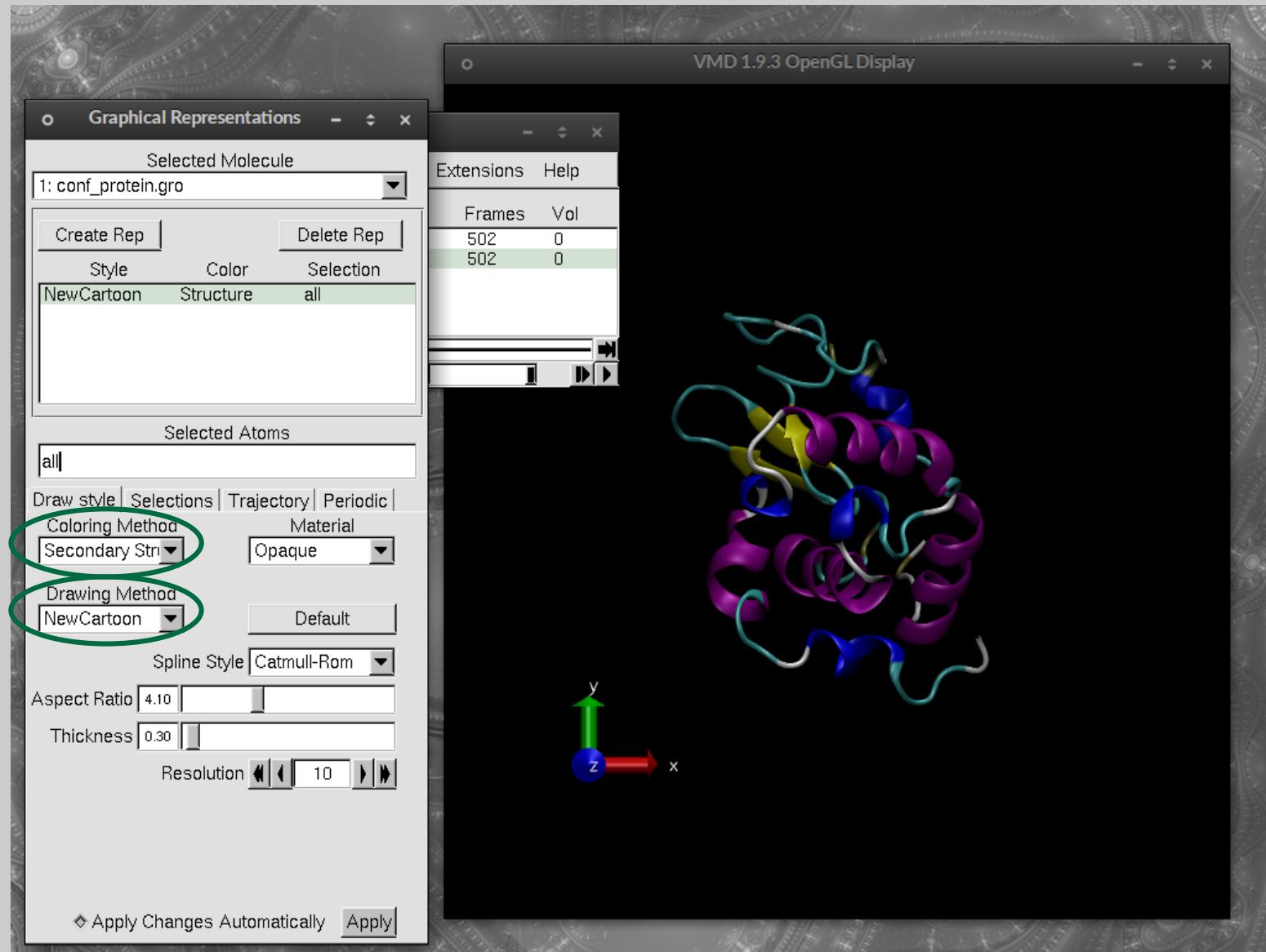
# Movie Maker

- Load your trajectory/structure
- Set graphical representations
  - Trajectory smoothing
- Movie Maker
  - Set animation options
  - Choose rendering and encoding options
  - Make tea (it can take a while...)

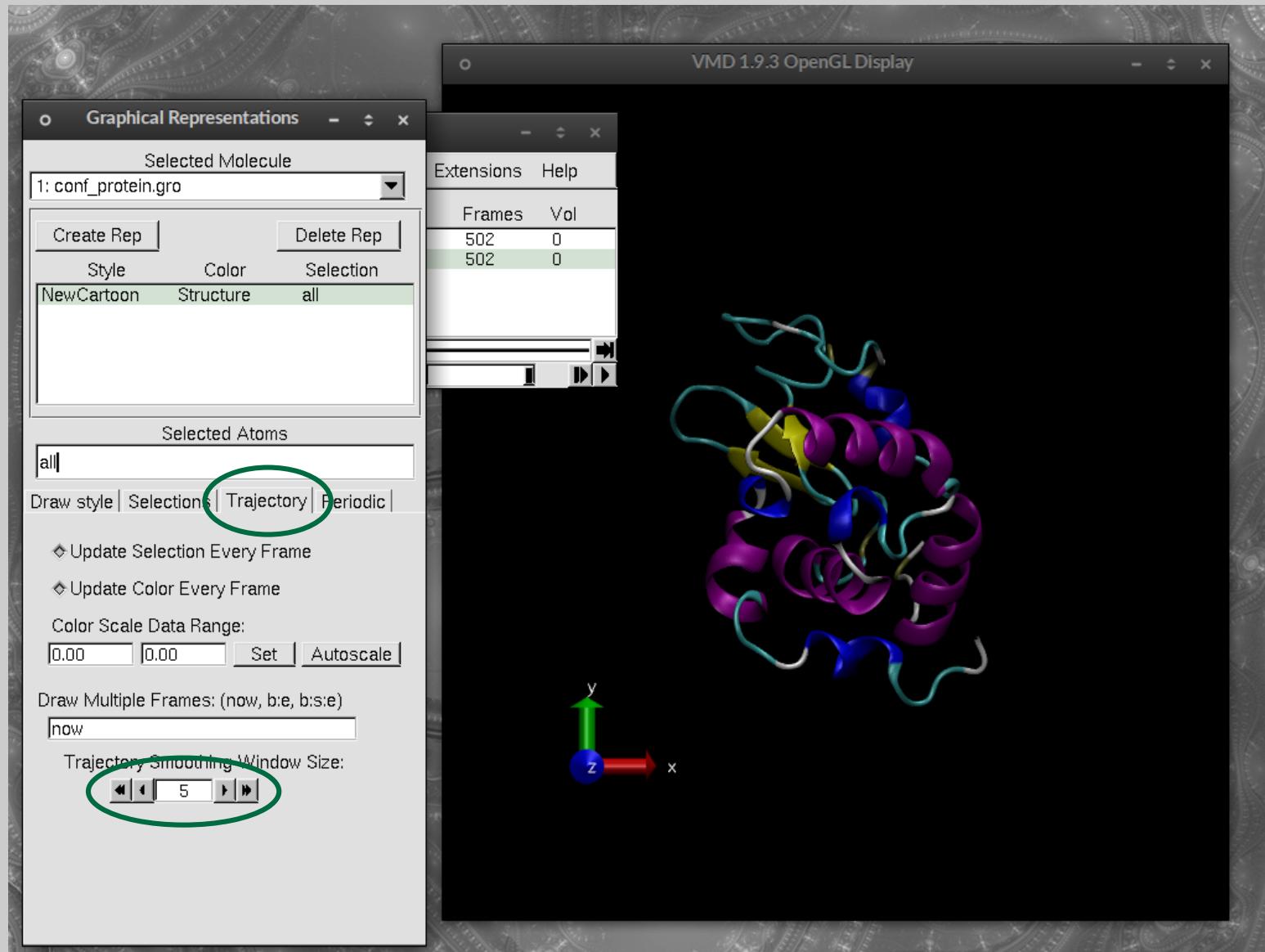
# Movie Maker



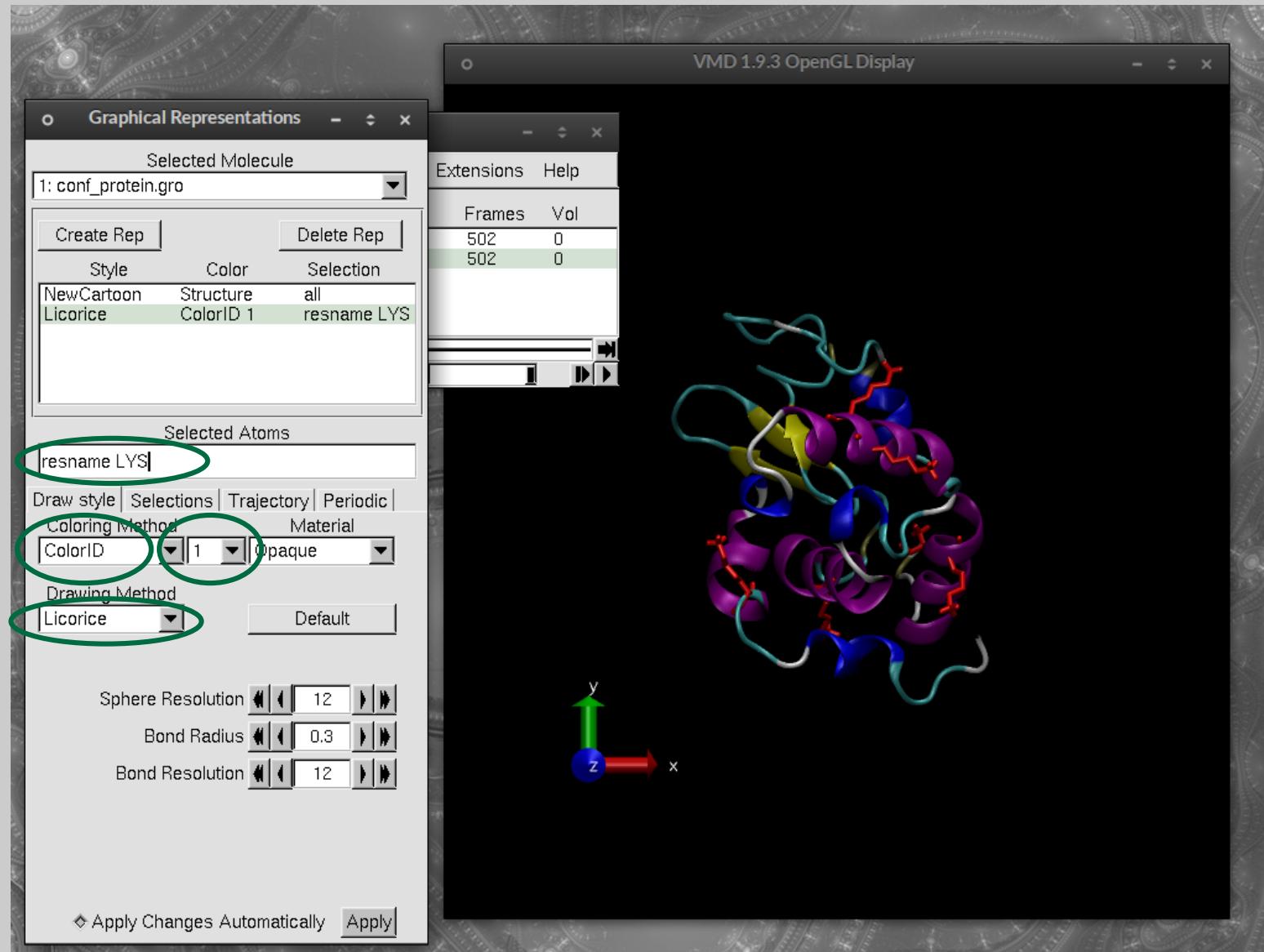
# Movie Maker



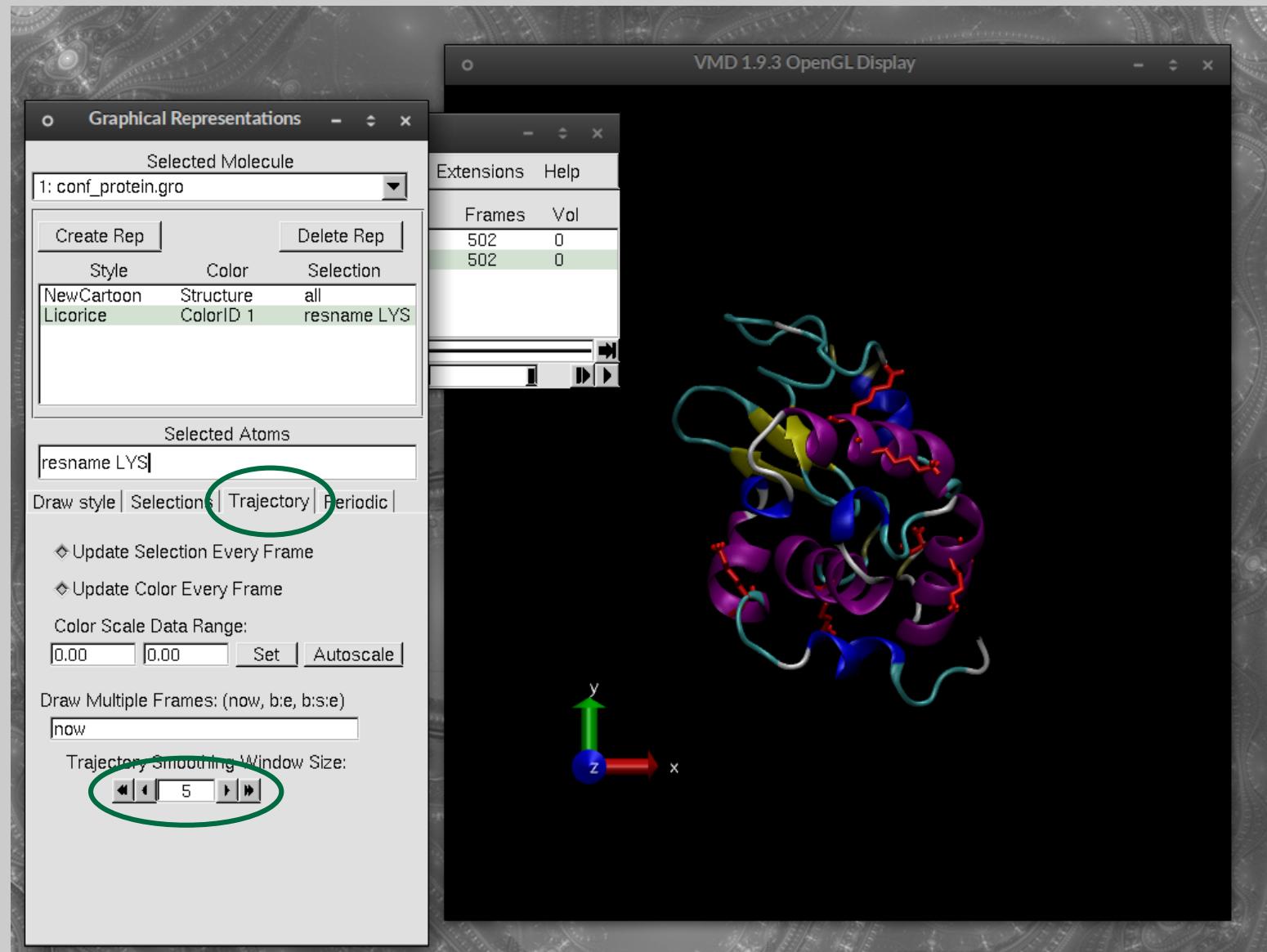
# Movie Maker



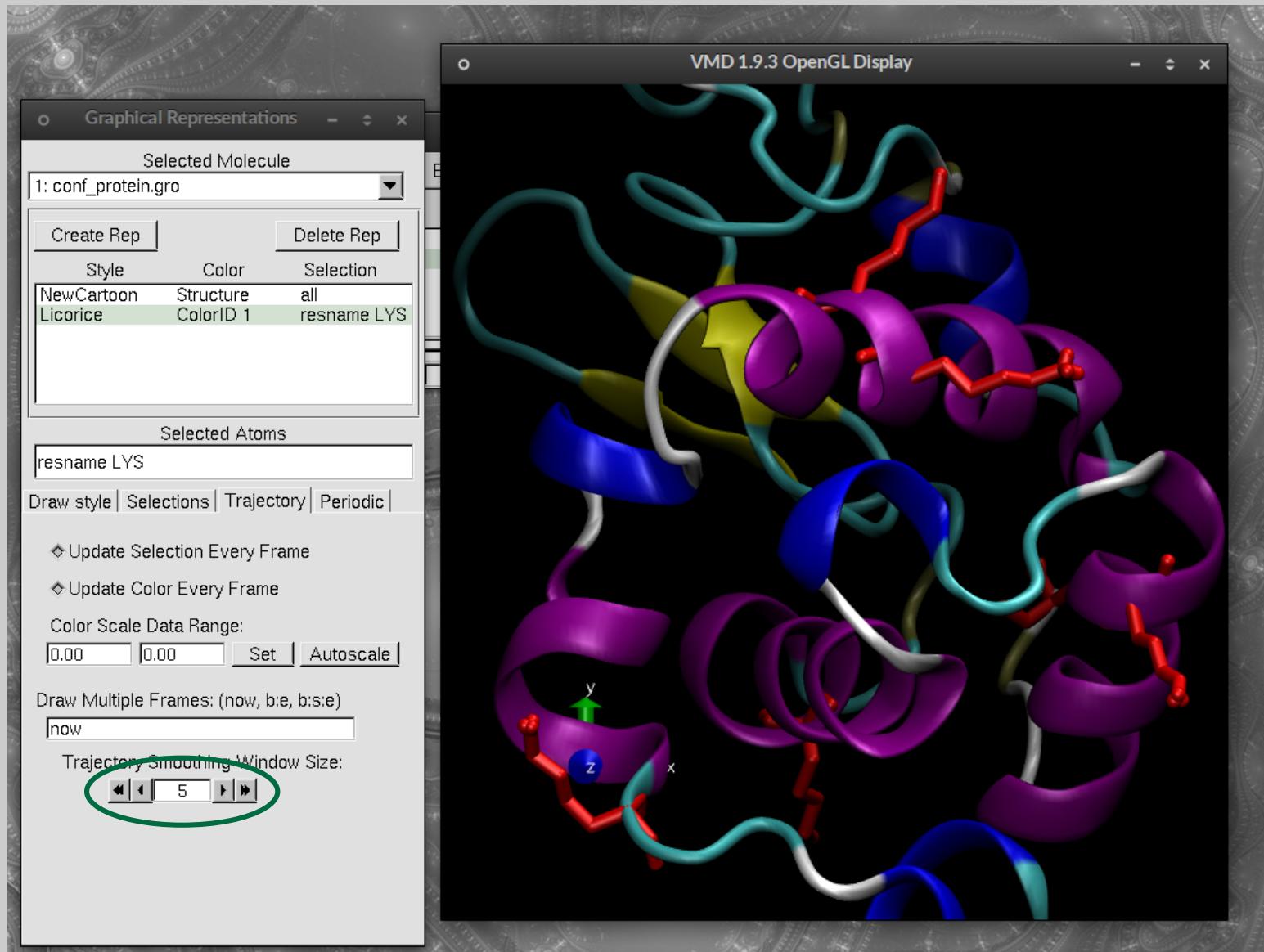
# Movie Maker



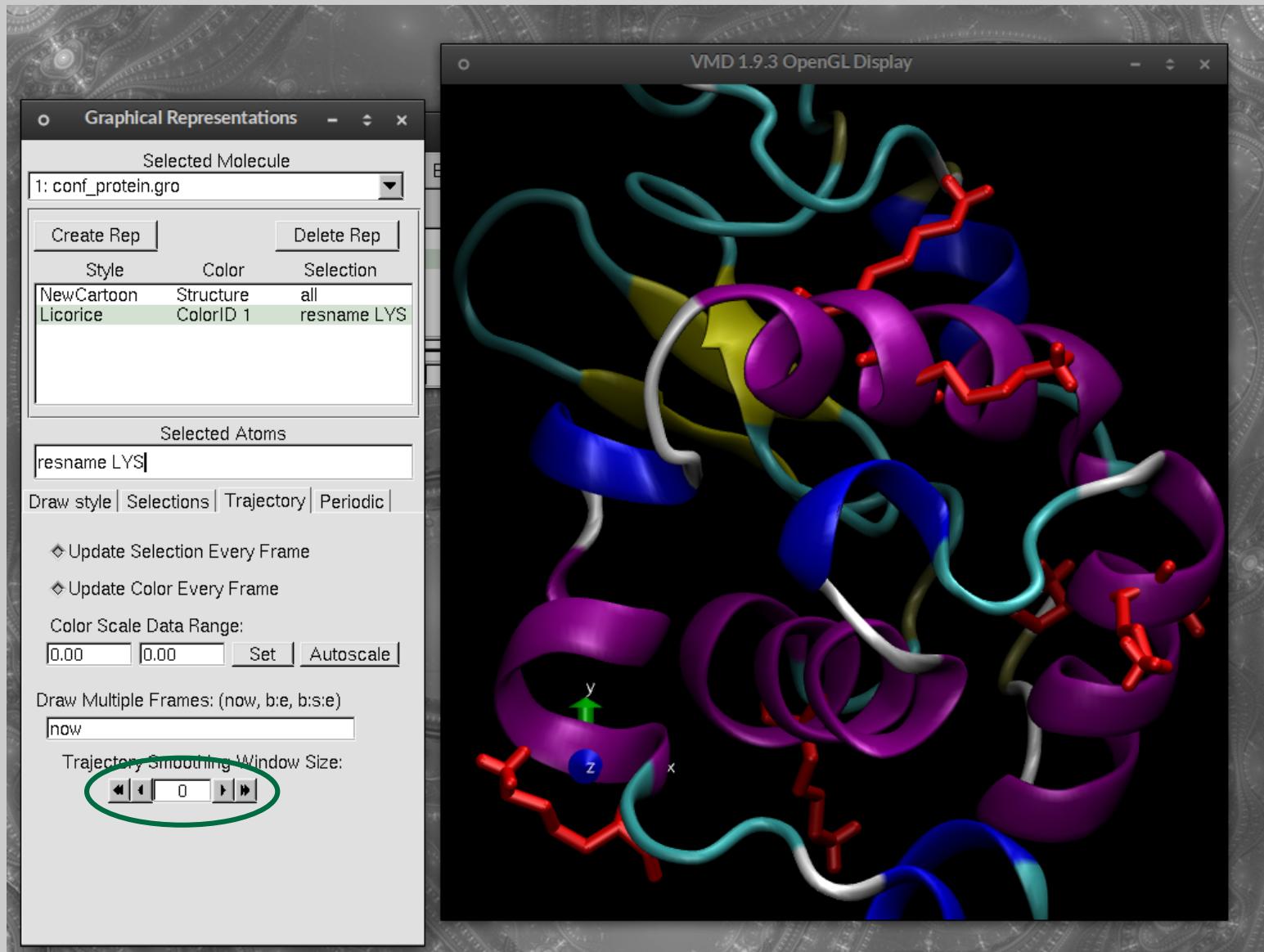
# Movie Maker



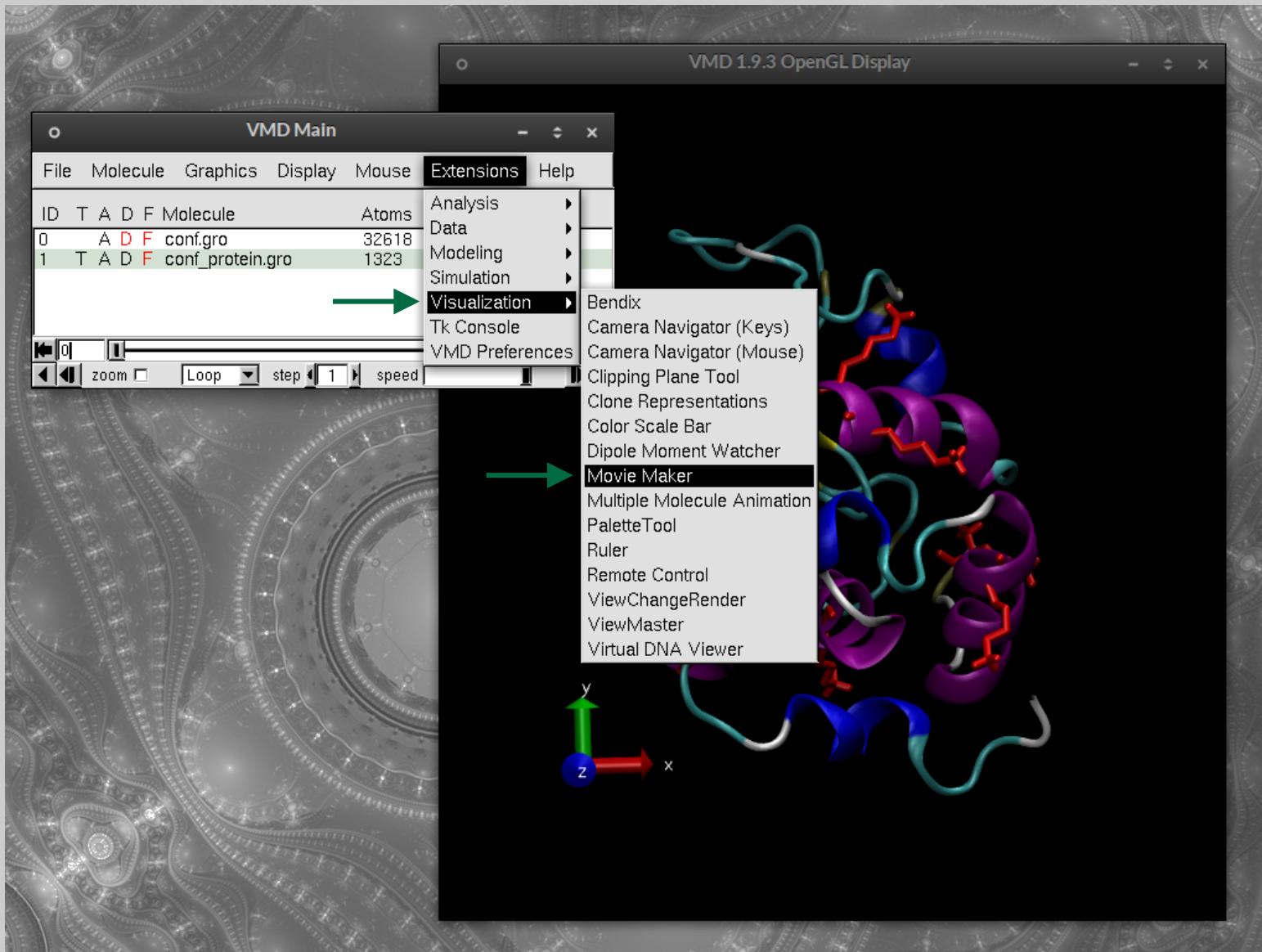
# Movie Maker



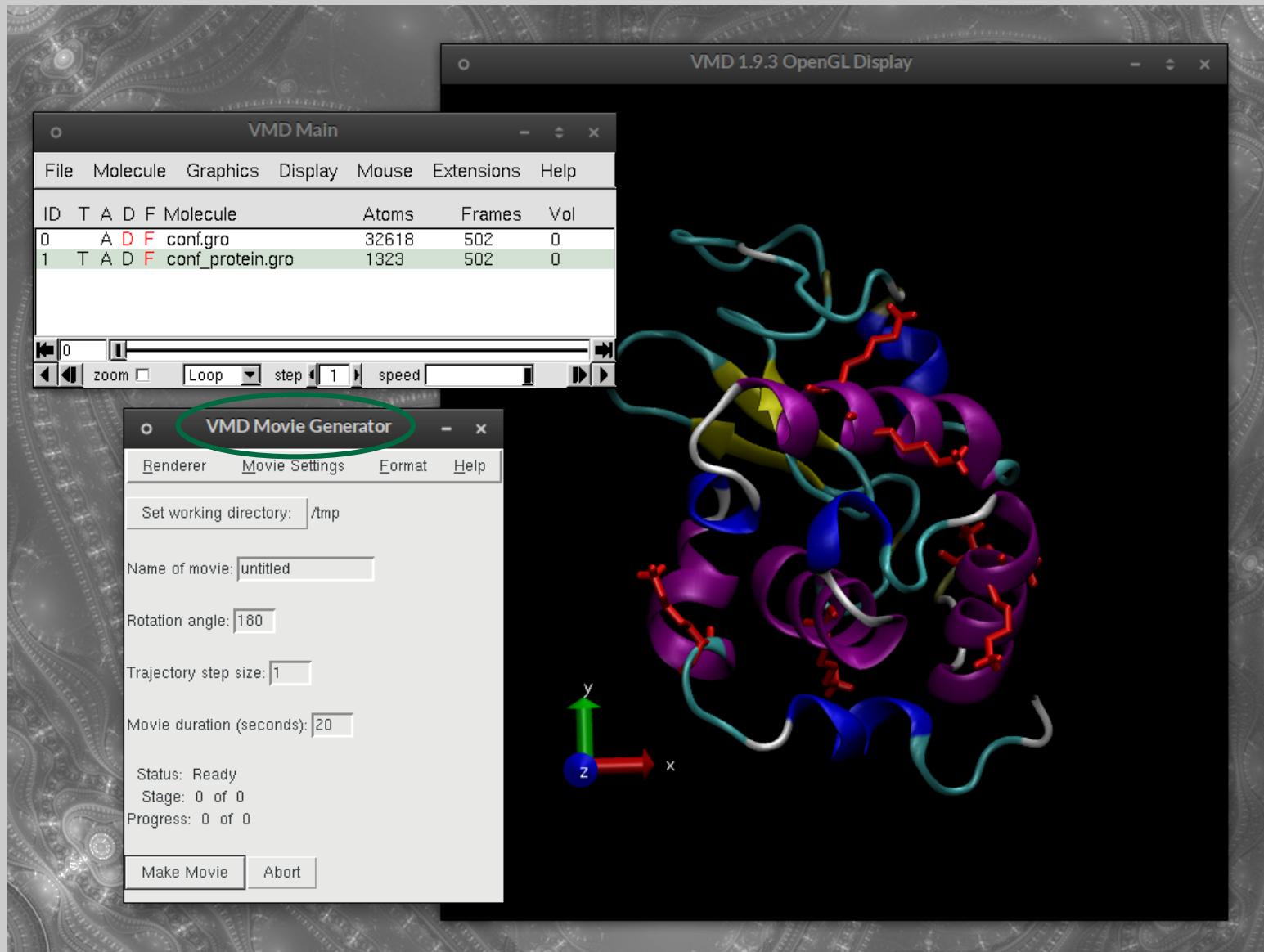
# Movie Maker



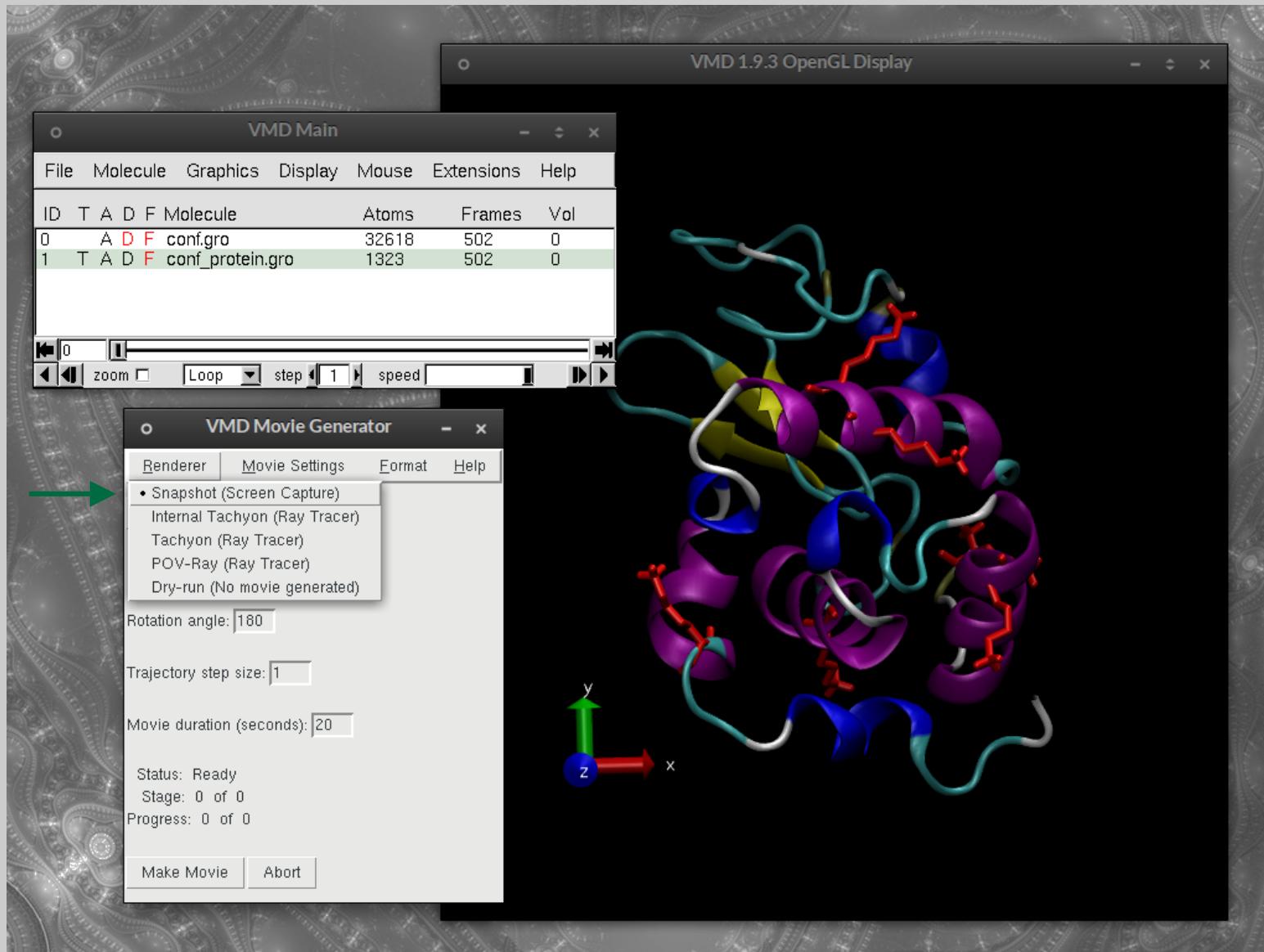
# Movie Maker



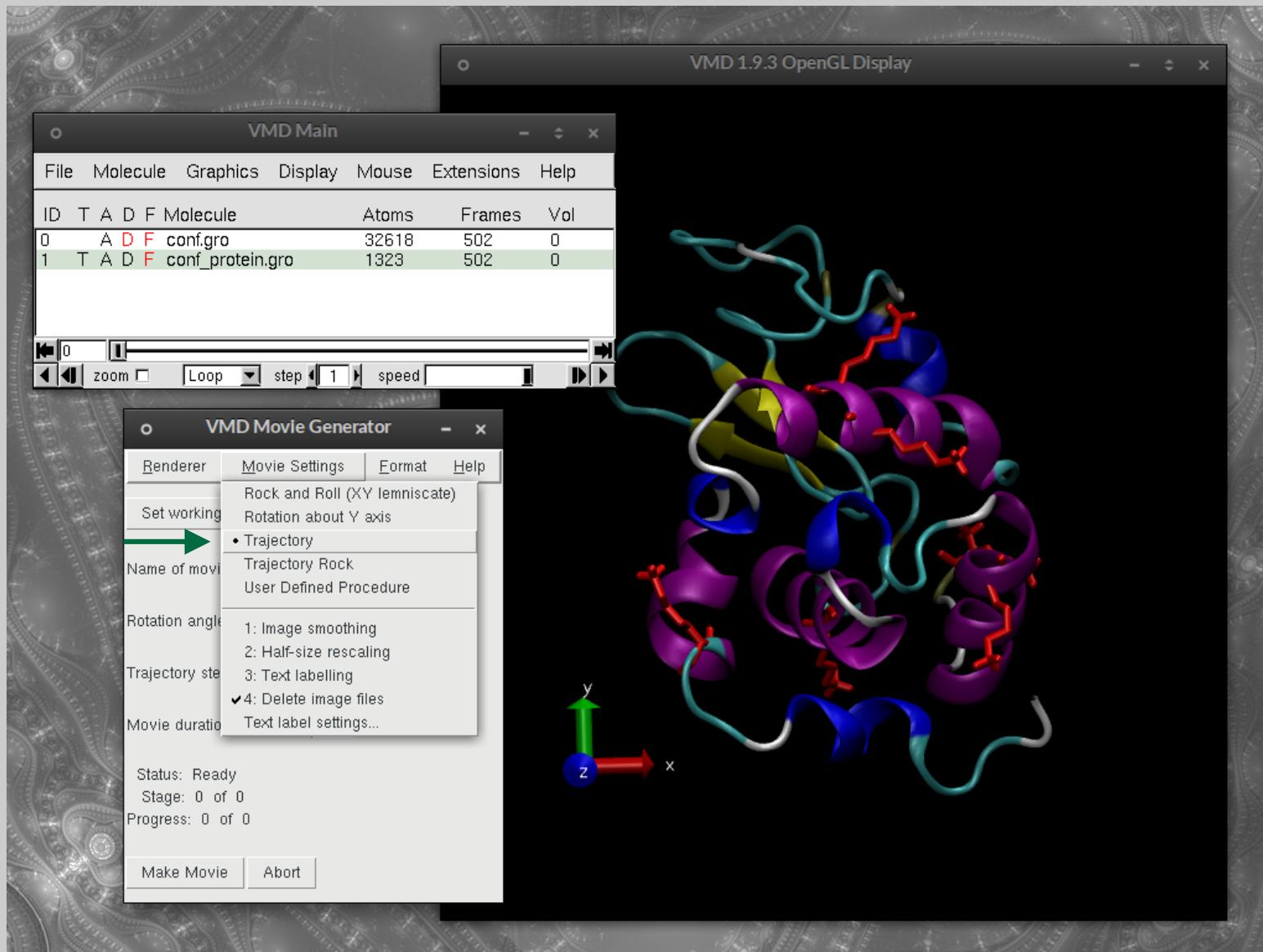
# Movie Maker



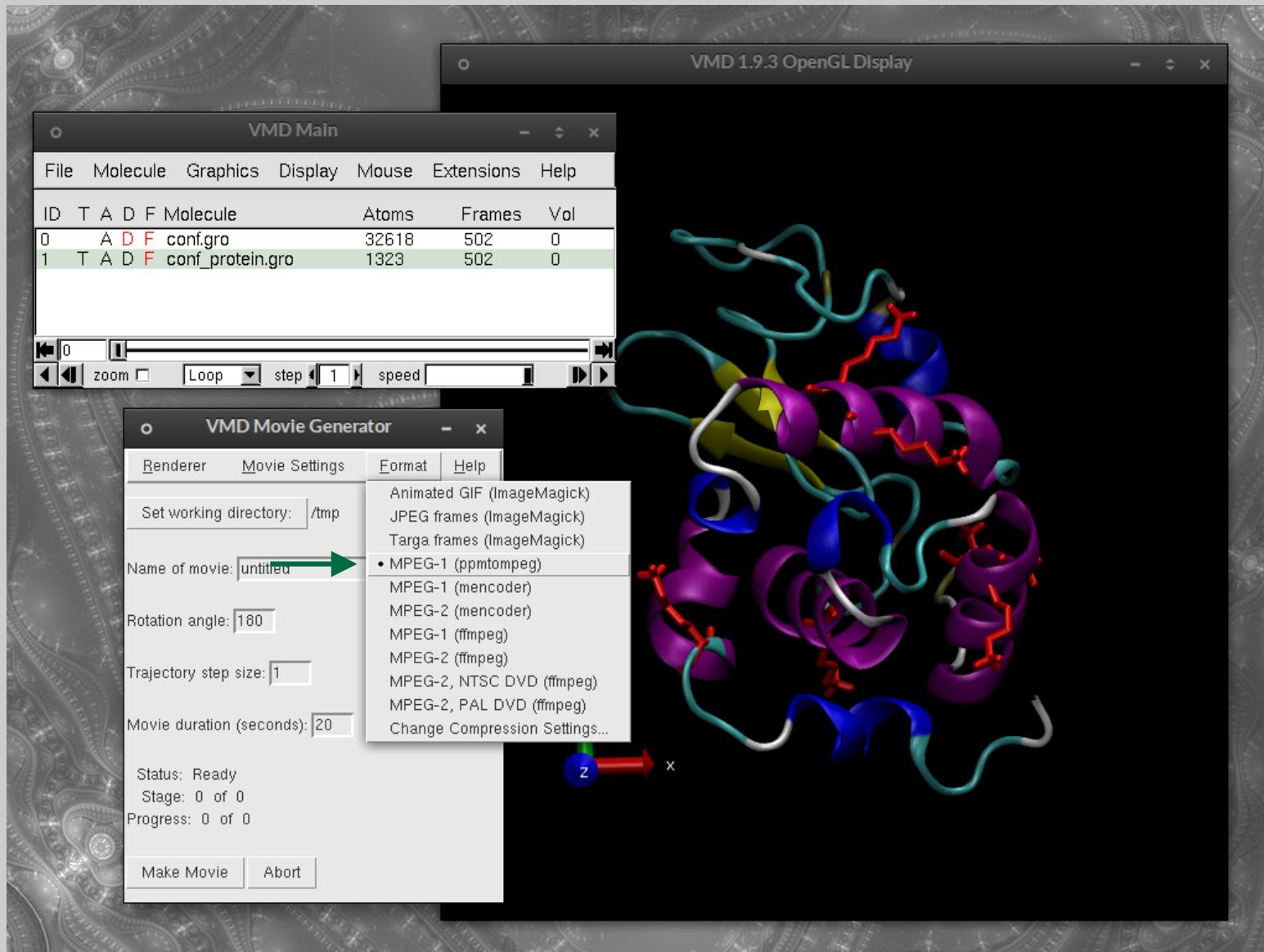
# Movie Maker



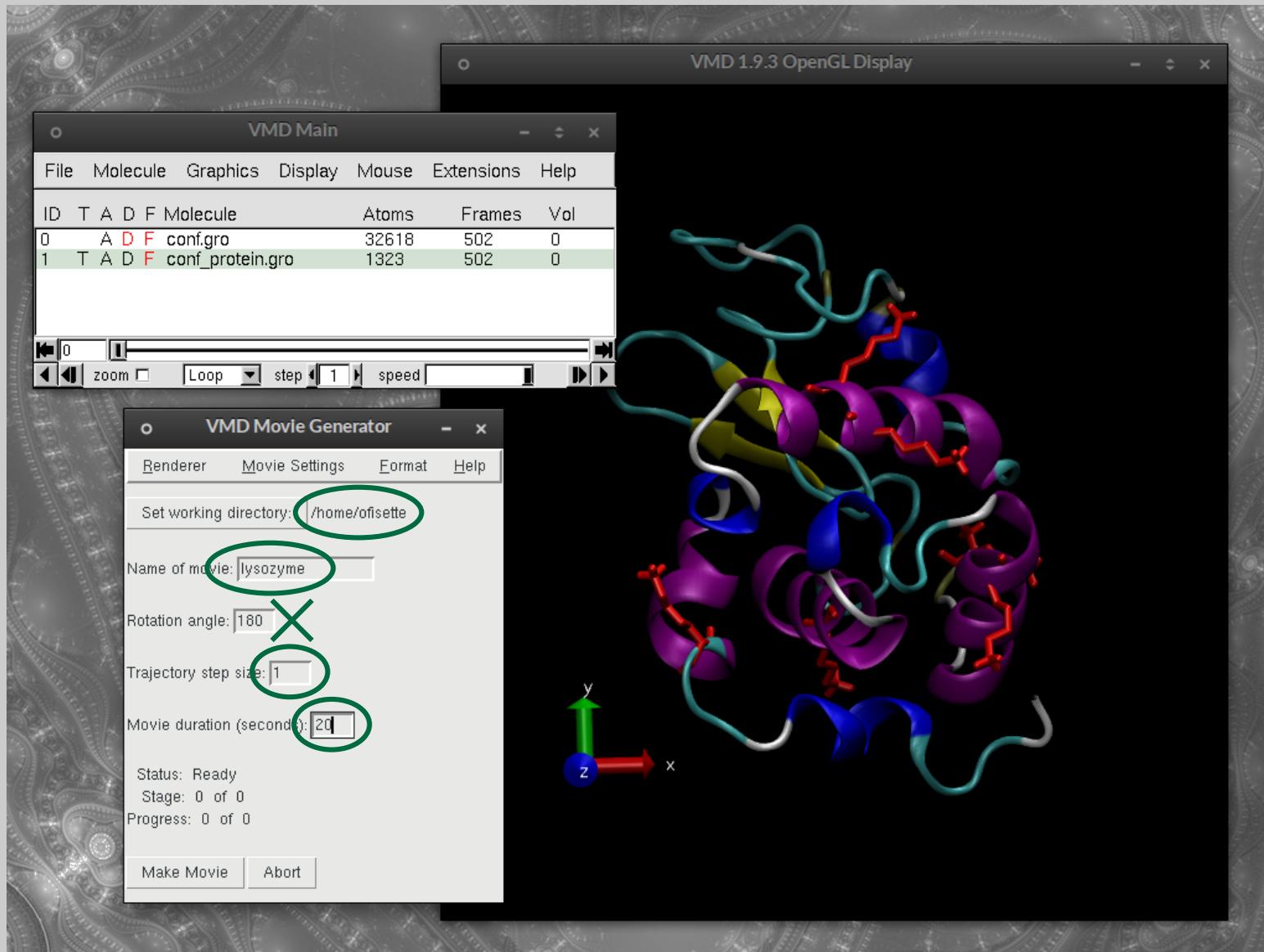
# Movie Maker



# Movie Maker



# Movie Maker



# Movie Maker



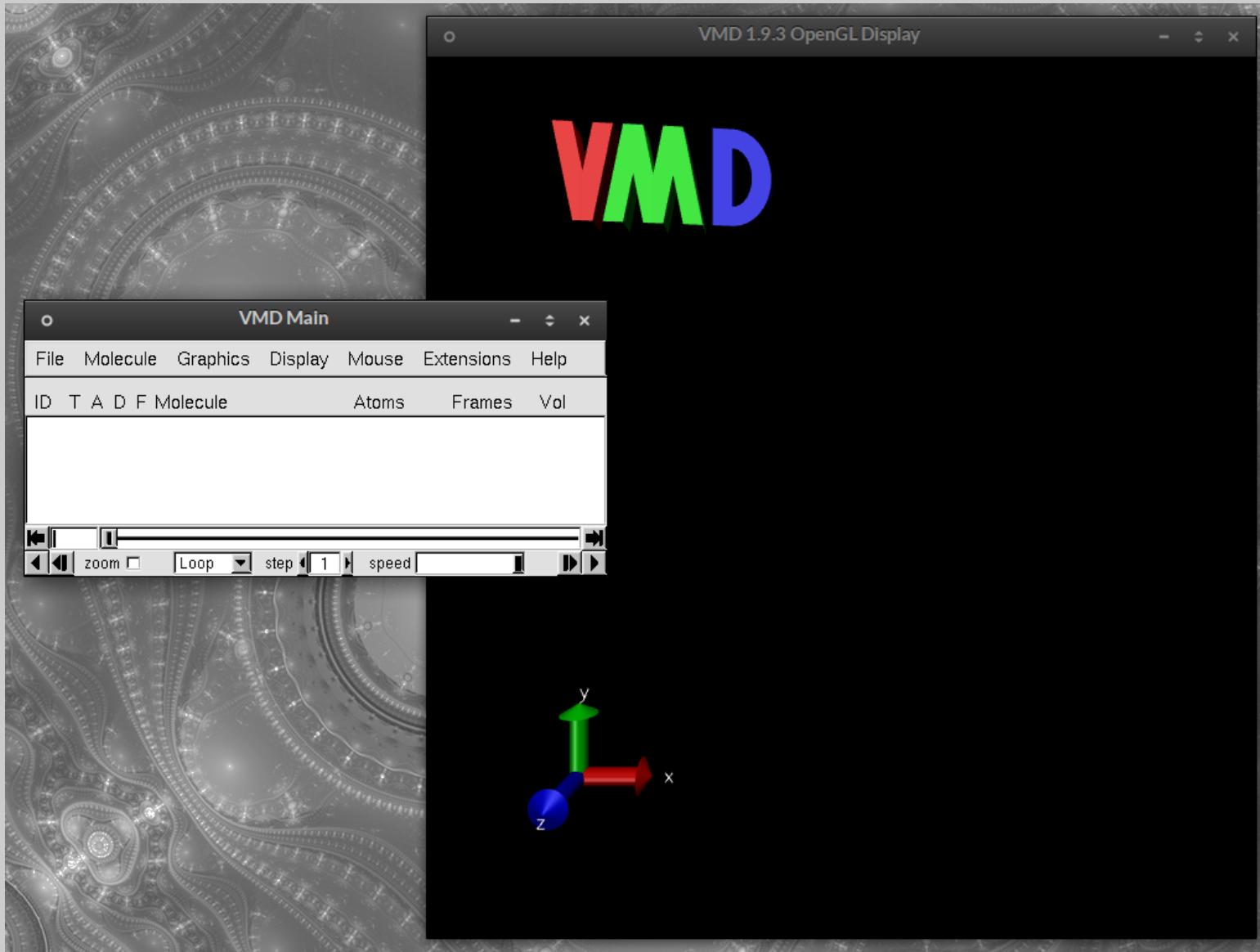
# Contents

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - **Tk Console**
  - **Tcl scripts**
  - ***vmdrc***
- Combining trajectories, movies, and scripting

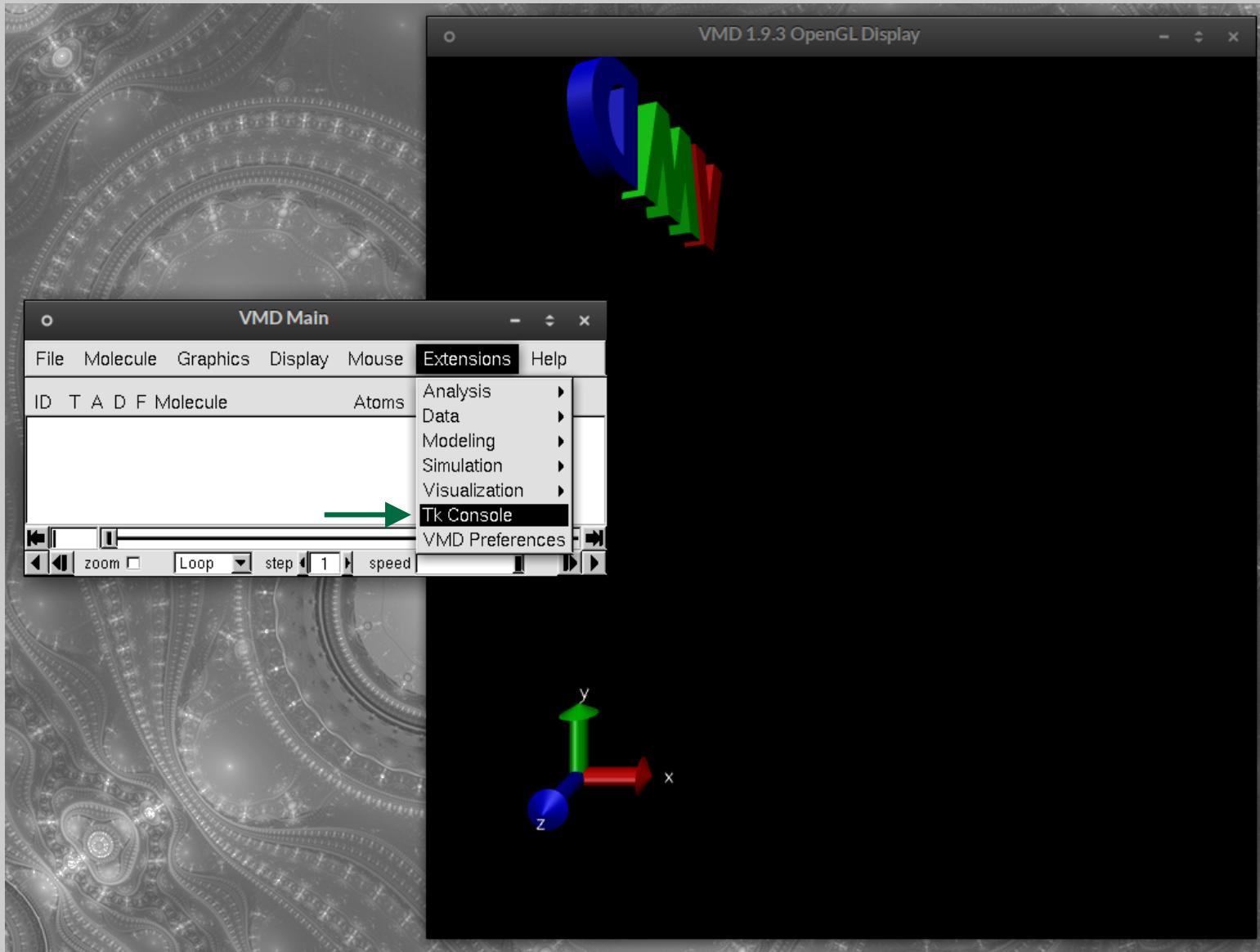
# Scripting

- VMD can be controlled through text commands.
- This has advantages over the graphical interface.
  - Easier (avoid convoluted menus and windows)
  - Faster (automated command repetition)
  - Reproducible (commands stored in a script)
- There are many ways to control VMD through text commands.
  - Typing in the “Tk Console”
  - Loading scripts (*play* command) from the “Tk Console”
  - Creating a *vmdrc* file

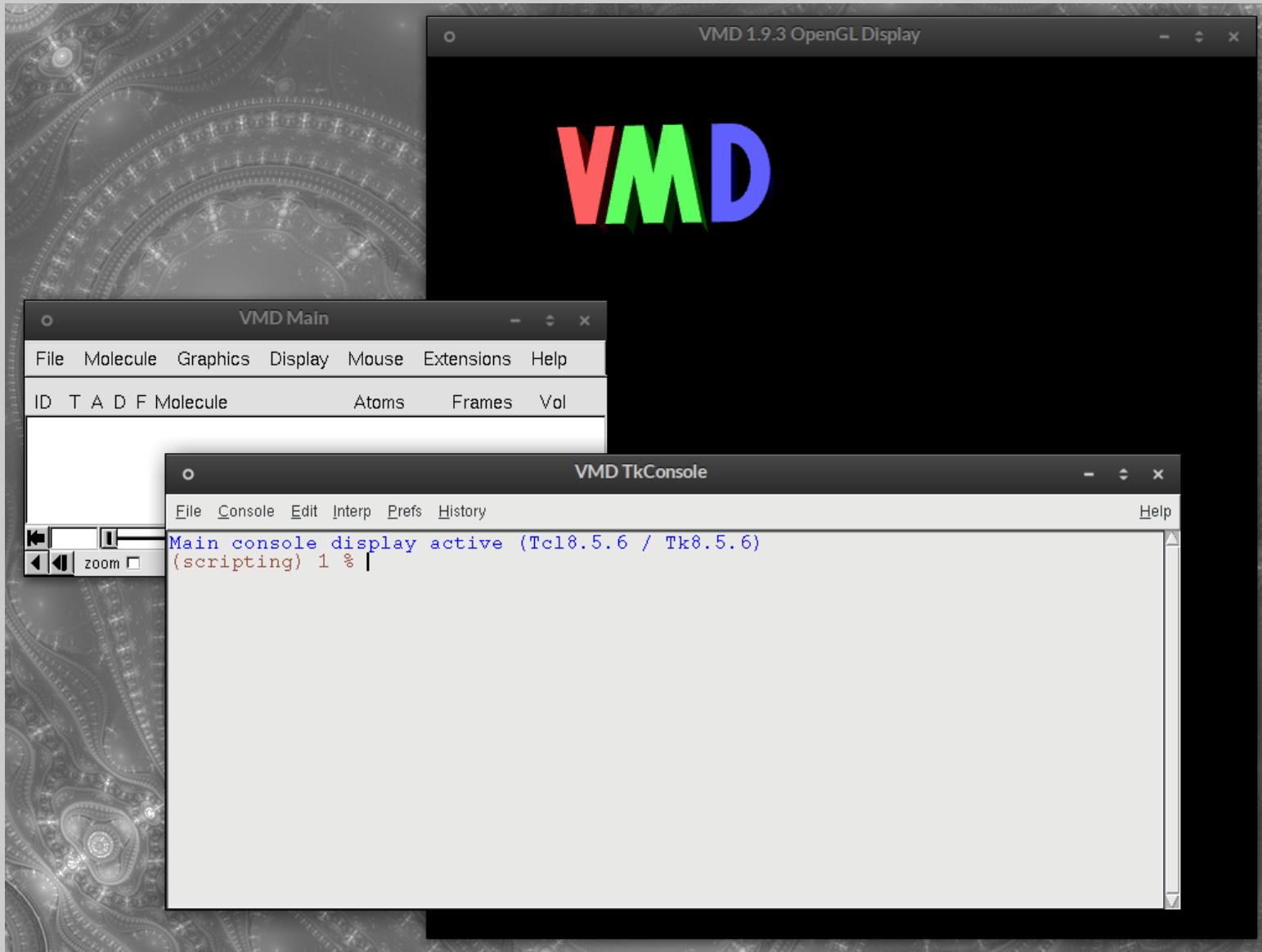
# Tk Console



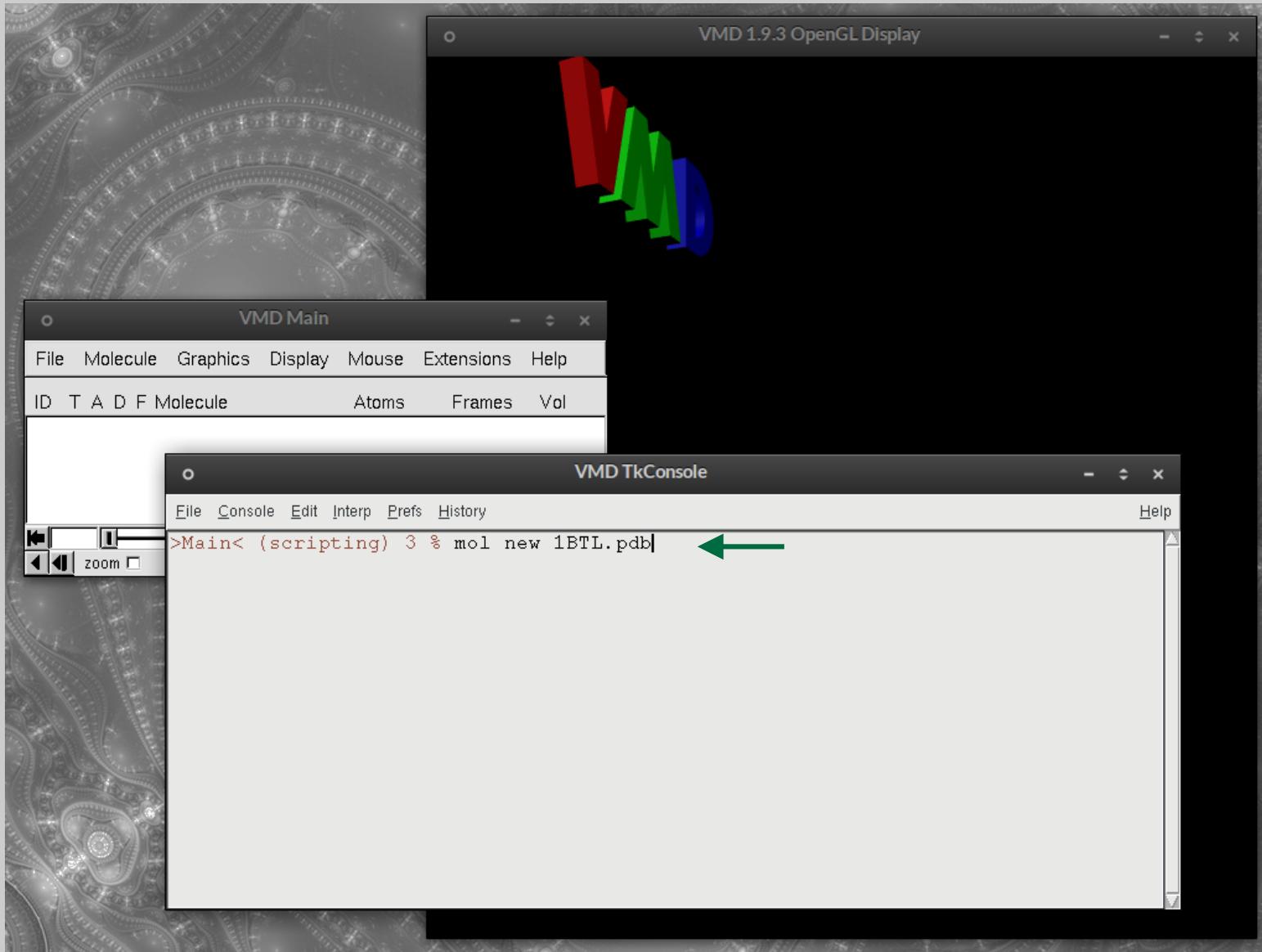
# Tk Console



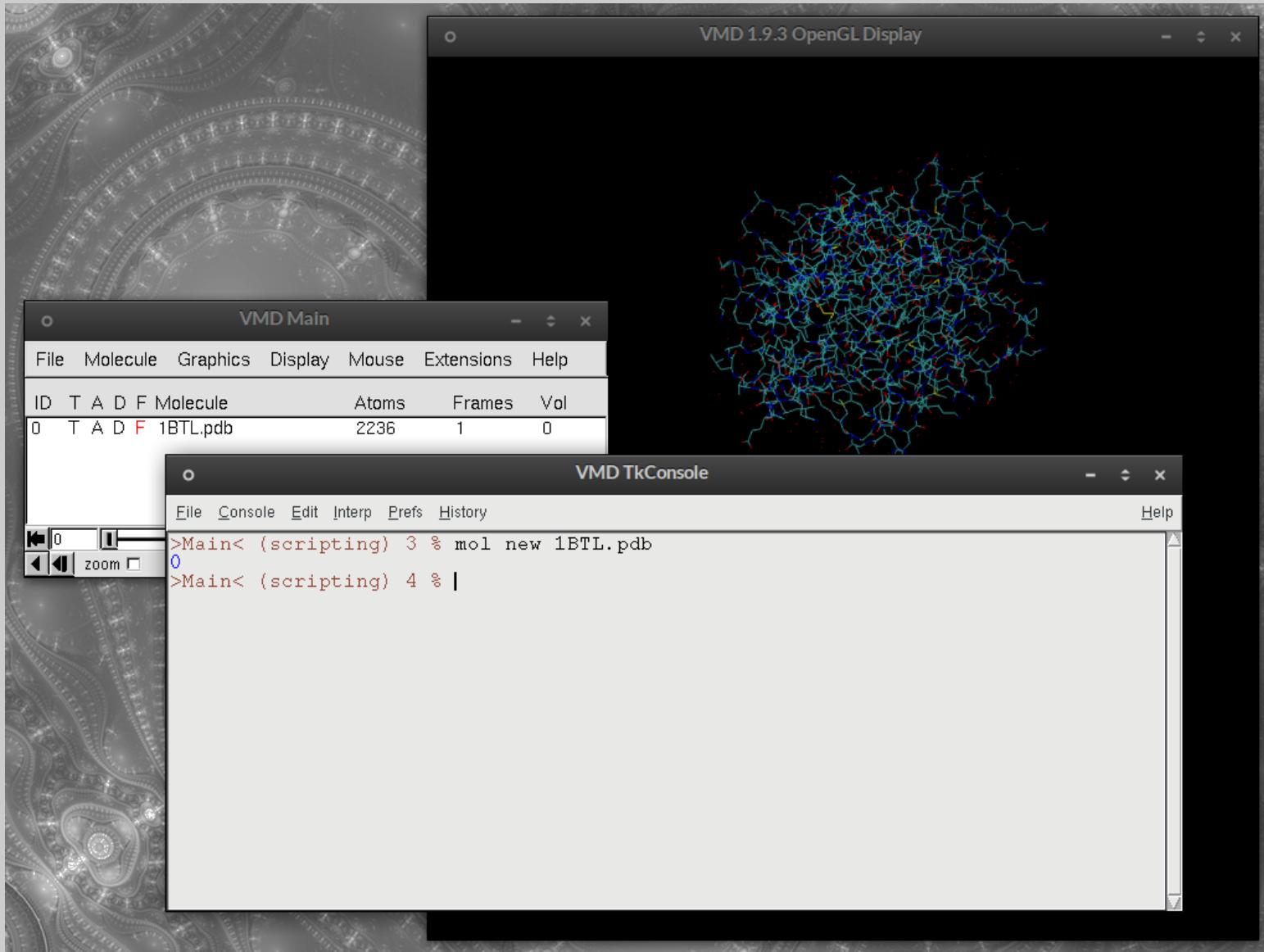
# Tk Console



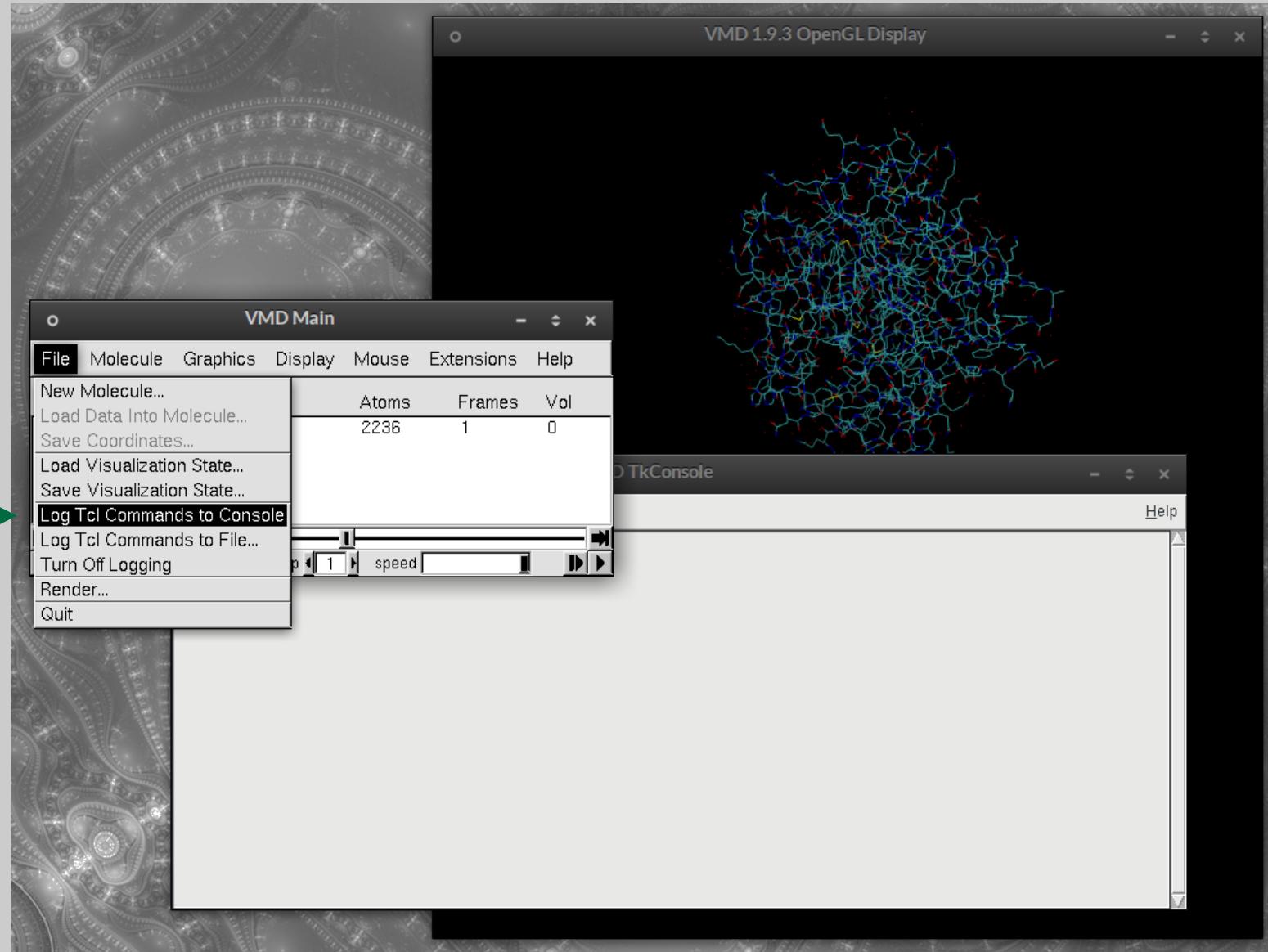
# Tk Console



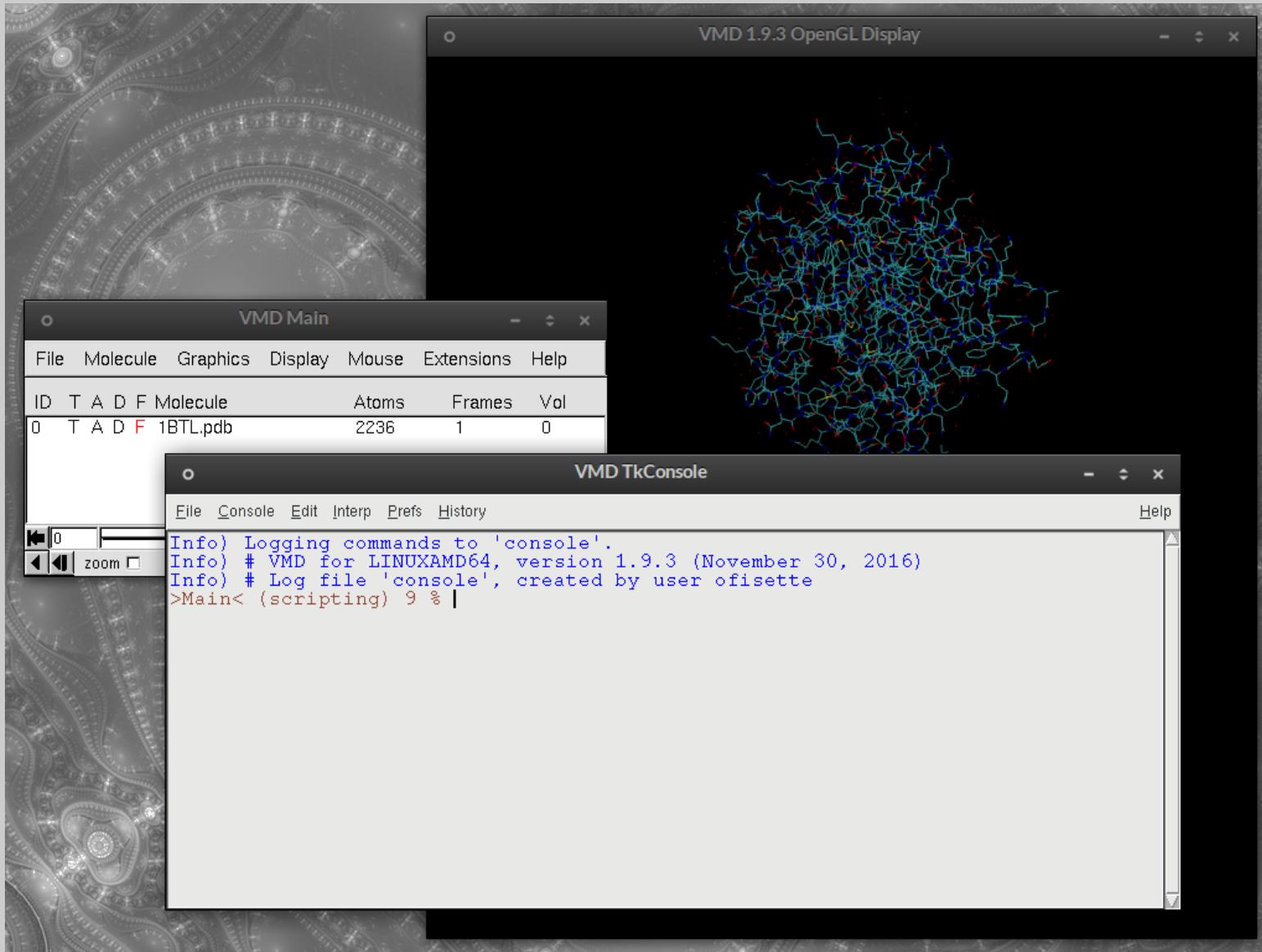
# Tk Console



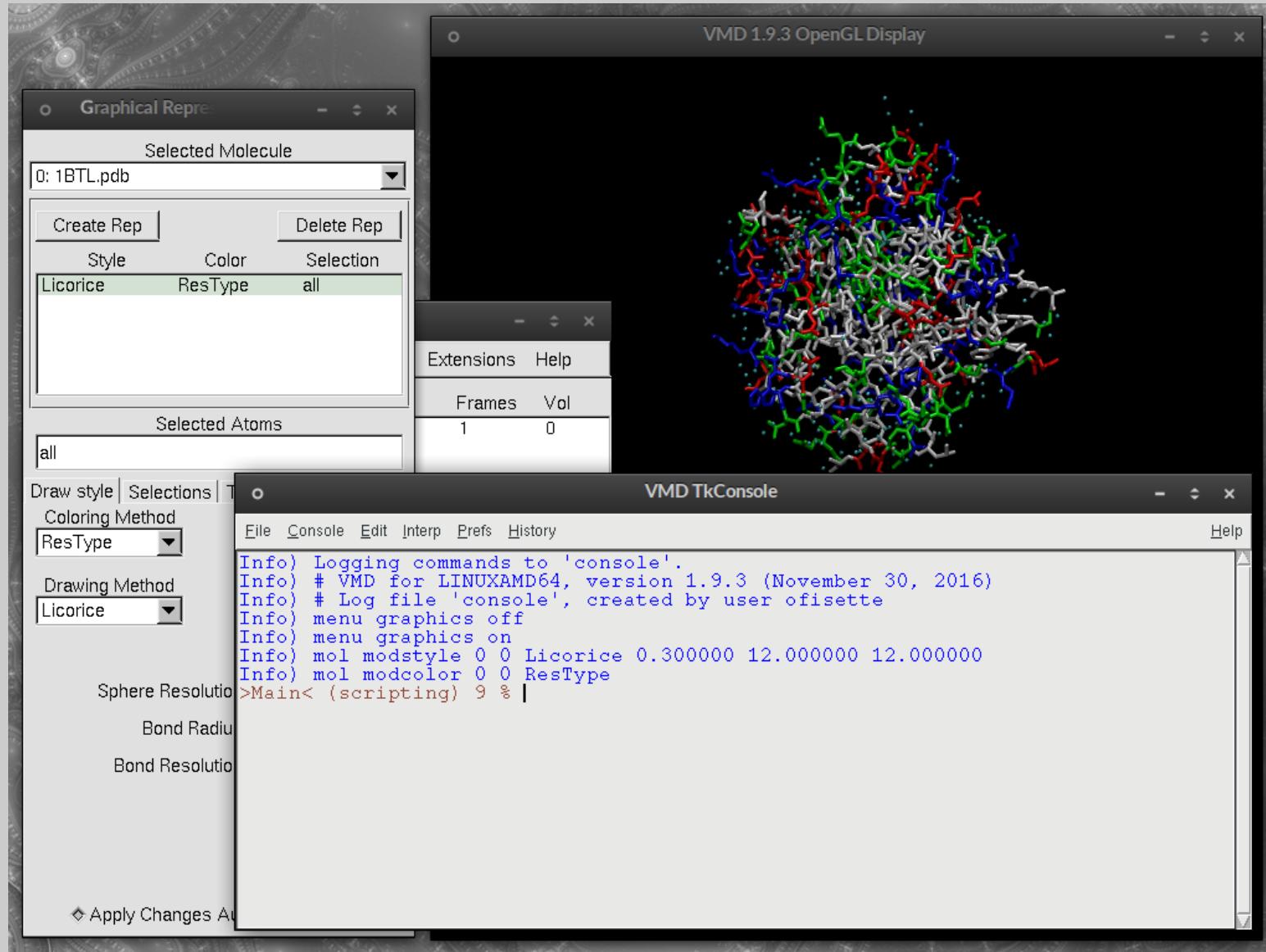
# Tk Console



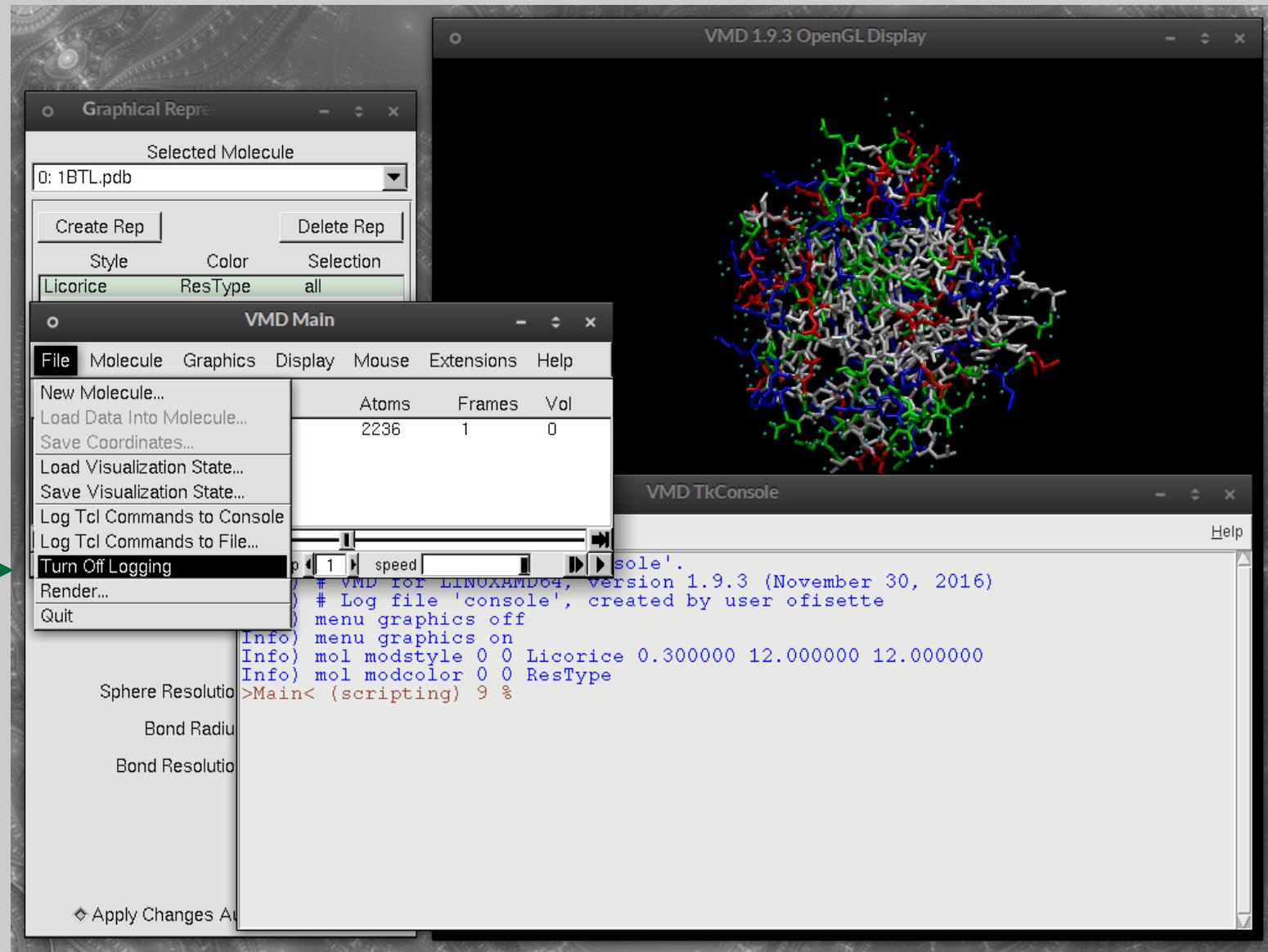
# Tk Console



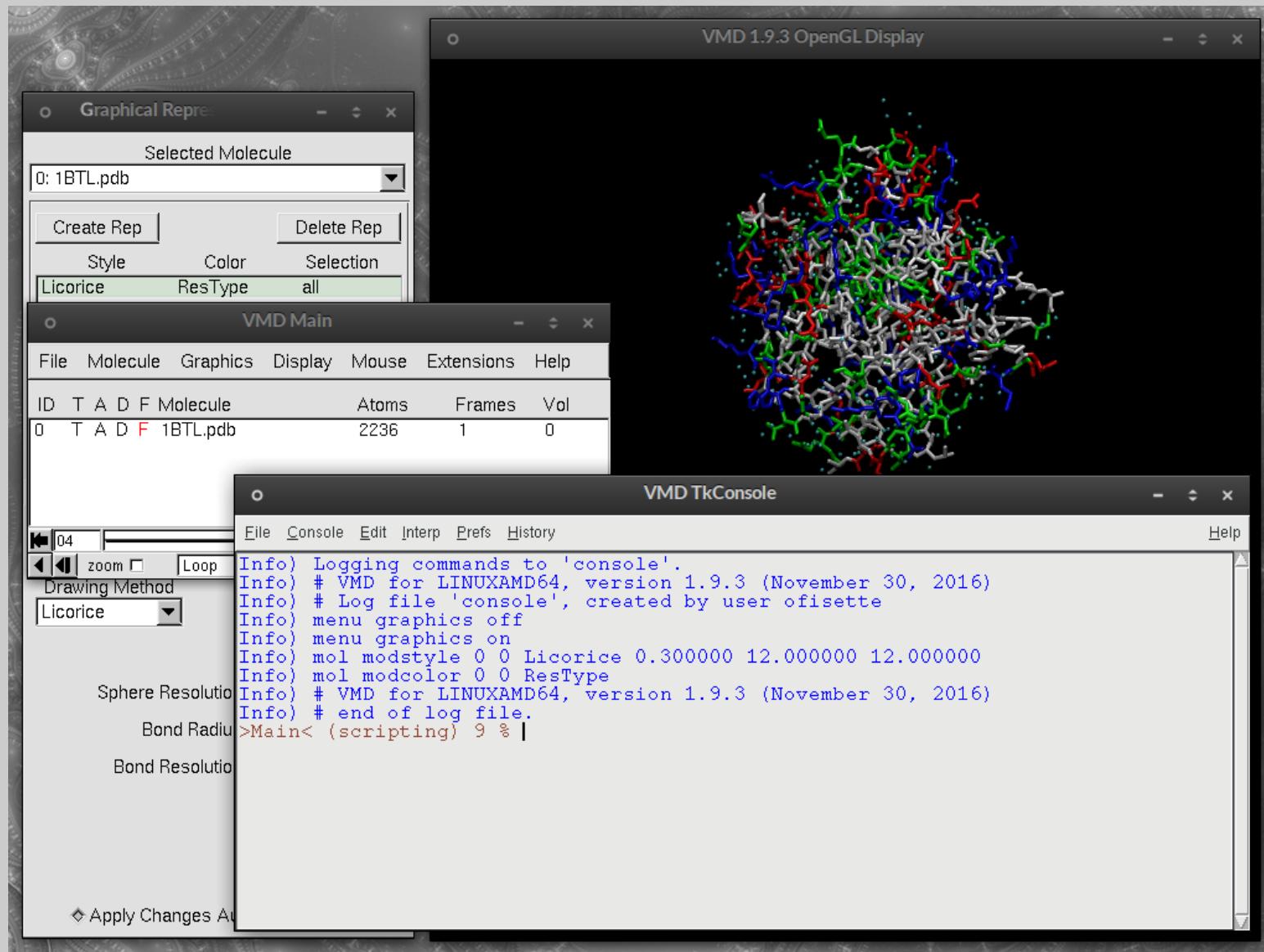
# Tk Console



# Tk Console



# Tk Console



# VMD User's Guide: Tcl Text Interface

<https://www.ks.uiuc.edu/Research/vmd/current/ug/node117.html>

The screenshot shows a web browser window with the title "Tcl Text Interface". The address bar displays the URL: <https://www.ks.uiuc.edu/Research/vmd/vmd-1.8.3/ug/node117.html>. Below the address bar is a navigation menu with links: "Next", "Up", "Previous", "Contents", and "Index". Underneath this menu, there is a breadcrumb trail: "Next: Using text commands Up: VMD User's Guide Previous: Making a Movie Contents Index". The main content area is titled "Tcl Text Interface". A text block states: "The Tcl text interface provides complete access to all the VMD commands. Anything that can be done from the menus can be done with VMD text commands." Below this text is a section titled "Subsections" with a list of links:

- [Using text commands](#)
- [Tcl/Tk](#)
- [Tcl Text Commands](#)
  - [animate](#)
  - [atomselect](#)
  - [axes](#)
  - [color](#)
  - [colorinfo](#)
  - [display](#)
  - [draw](#)
  - [exit](#)
  - [graphics](#)
  - [help](#)
  - [imd](#)
  - [label](#)
  - [light](#)
  - [logfile](#)
  - [material](#)
  - [measure](#)
  - [menu](#)
  - [mol](#)
  - [molecule](#)
  - [molinfo](#)
  - [mouse](#)
  - [play](#)
  - [quit](#)
  - [render](#)
  - [rock](#)
  - [rotate](#)
  - [scale](#)
  - [stage](#)
  - [tool](#)
  - [translate](#)
  - [user](#)
  - [vmdinfo](#)
  - [wait](#)
  - [sleep](#)

# Tcl scripts

```
1 # Global parameters
2 display projection orthographic
3 color Display Background white
4 axes location Off
5
6 # Delete default representation
7 mol delrep 0 top
8
9 # Show protein as cartoon, color by secondary structure
10 # (thickness, resolution, aspect ratio, spline type)
11 mol representation NewCartoon 0.3 10.0 4.1 0
12 mol color Structure
13 mol selection {protein}
14 mol addrep top
```

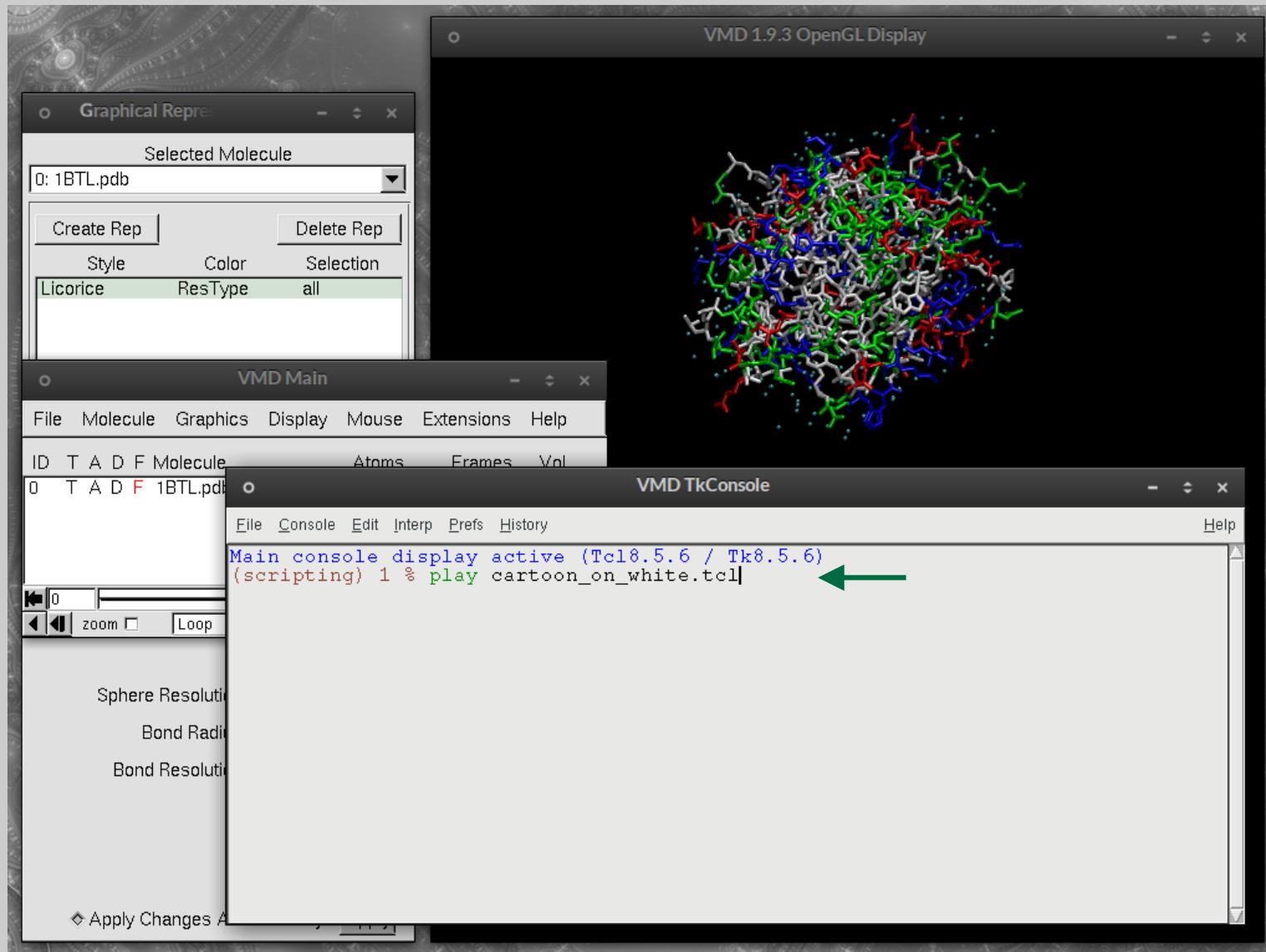
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~

"cartoon\_on\_white.tcl" 14L, 365C écrit(s)

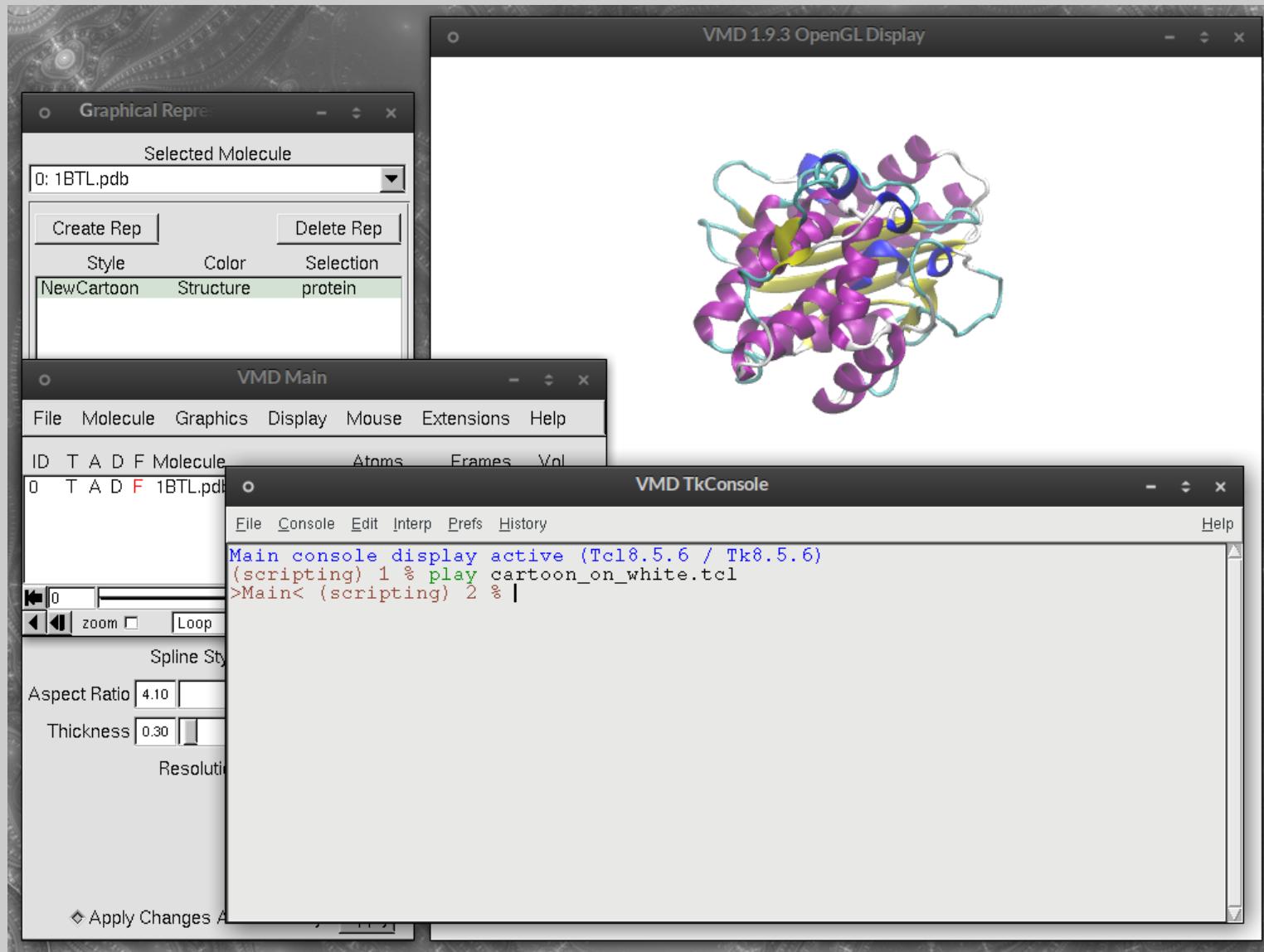
14,14

Tout

# Tcl scripts



# Tcl scripts



# *vmdrc* Initialisation file

- You can create a Tcl script that will be “played” when VMD starts.
  - UNIX: `$HOME/.vmdrc`
  - Windows: `$HOME/vmd.rc`
- This enables customisation
  - Display options
  - Keyboard shortcuts
  - Colour definitions

# vmrdc Initialisation file

```
1 # Global parameters
2 display projection orthographic
3 color Display Background white
4 axes location Off
5
6 # Keyboard shortcuts
7 user add key a "axes location off"
8 user add key A "axes location lowerleft"
9
10 # Universal palette
11 # blue (blue)
12 color change rgb 0 0.00 0.47 0.72
13 # red (orange red)
14 color change rgb 1 0.96 0.25 0.00
15 # 2 gray
16 # orange (dark yellow)
17 color change rgb 3 0.99 0.63 0.00
18 # yellow (bright yellow)
19 color change rgb 4 0.94 0.89 0.26
20 # 5 tan (dark blue)
21 color change rgb 5 0.00 0.30 0.45
22 # 6 silver
23 # green (green)
24 color change rgb 7 0.00 0.62 0.45
25 # 8 white
26 # pink (pink)
27 color change rgb 9 0.89 0.49 0.68
28 # cyan (sky blue)
29 color change rgb 10 0.25 0.75 1.00
30
31 # Show main window
32 menu main on
```

13,18

Tout

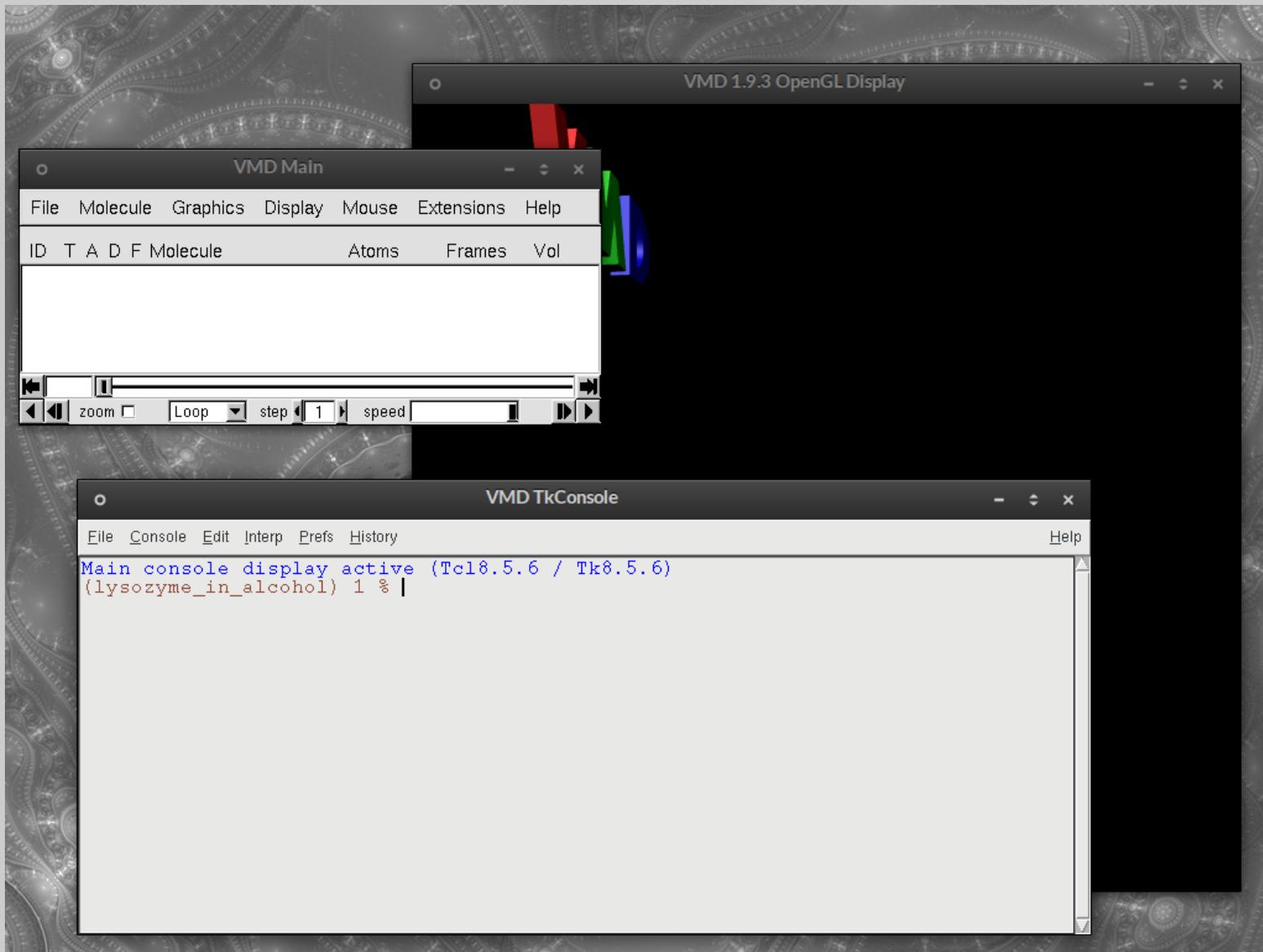
# Contents

- Basics
  - Loading structures
  - Representations
  - Atom selections
  - 3D navigation
- Drawing methods
- Materials and colouring
- Image rendering
- Trajectories
  - Troubleshooting
  - Post-processing
- Movies
  - Using the “Movie Maker”
- Scripting
  - Tk Console
  - Tcl scripts
  - *vmdrc*
- **Combining trajectories, movies, and scripting**

# Combining trajectories, movies, and scripting

- Objective
  - Animate a trajectory of lysozyme unfolding in ethanol solvent
  - Rotate the viewpoint while the trajectory advances
  - Recalculate secondary structure at each trajectory frame
  - Produce an MP4 movie with the H.265 codec
- Strategy
  - Animate in Movie Maker with a user-defined procedure (Tcl script)
  - Render TGA frames from the Movie Maker
  - Encode the movie outside VMD with *ffmpeg* (Bash script)

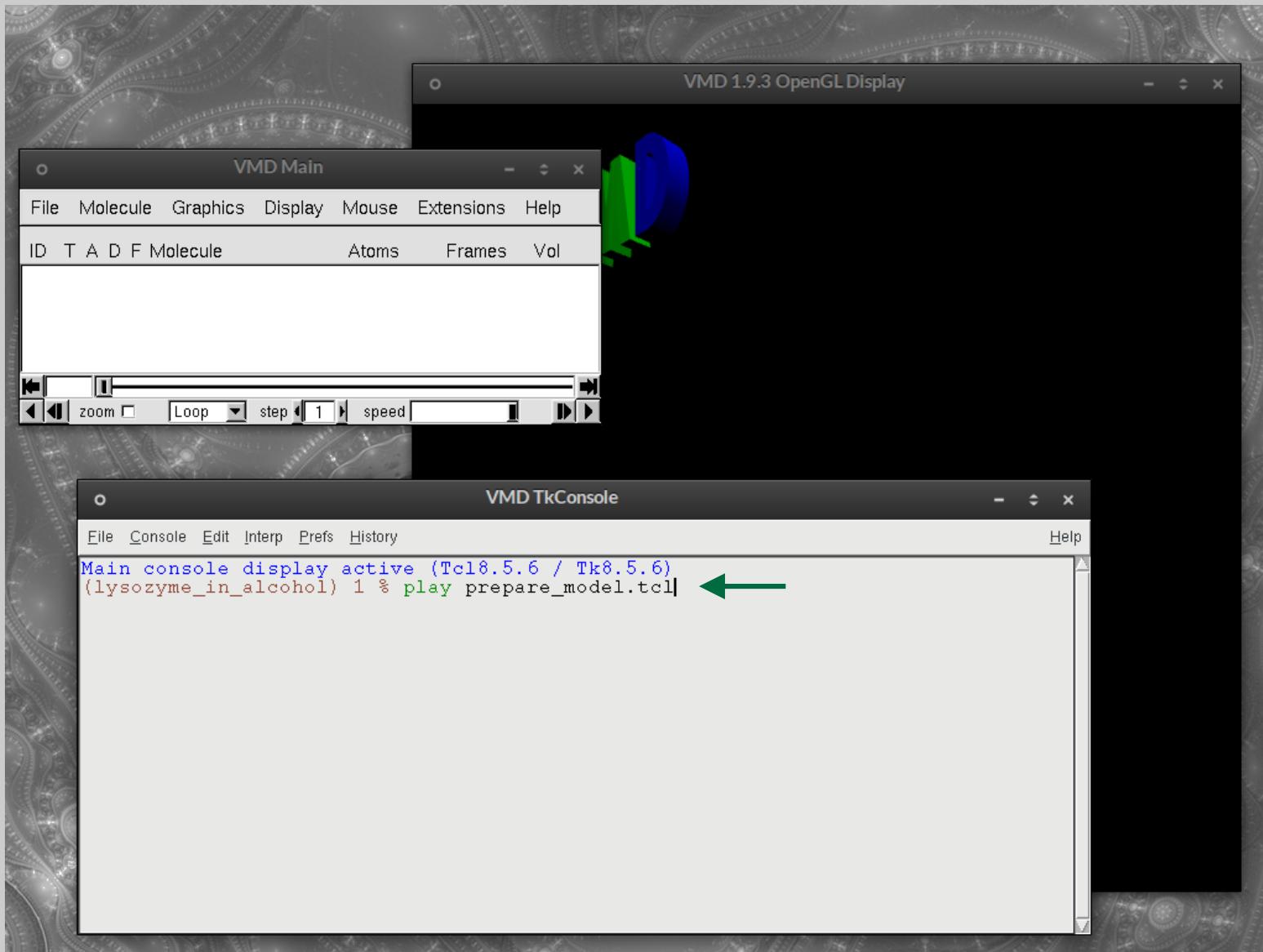
# Combining trajectories, movies, and scripting



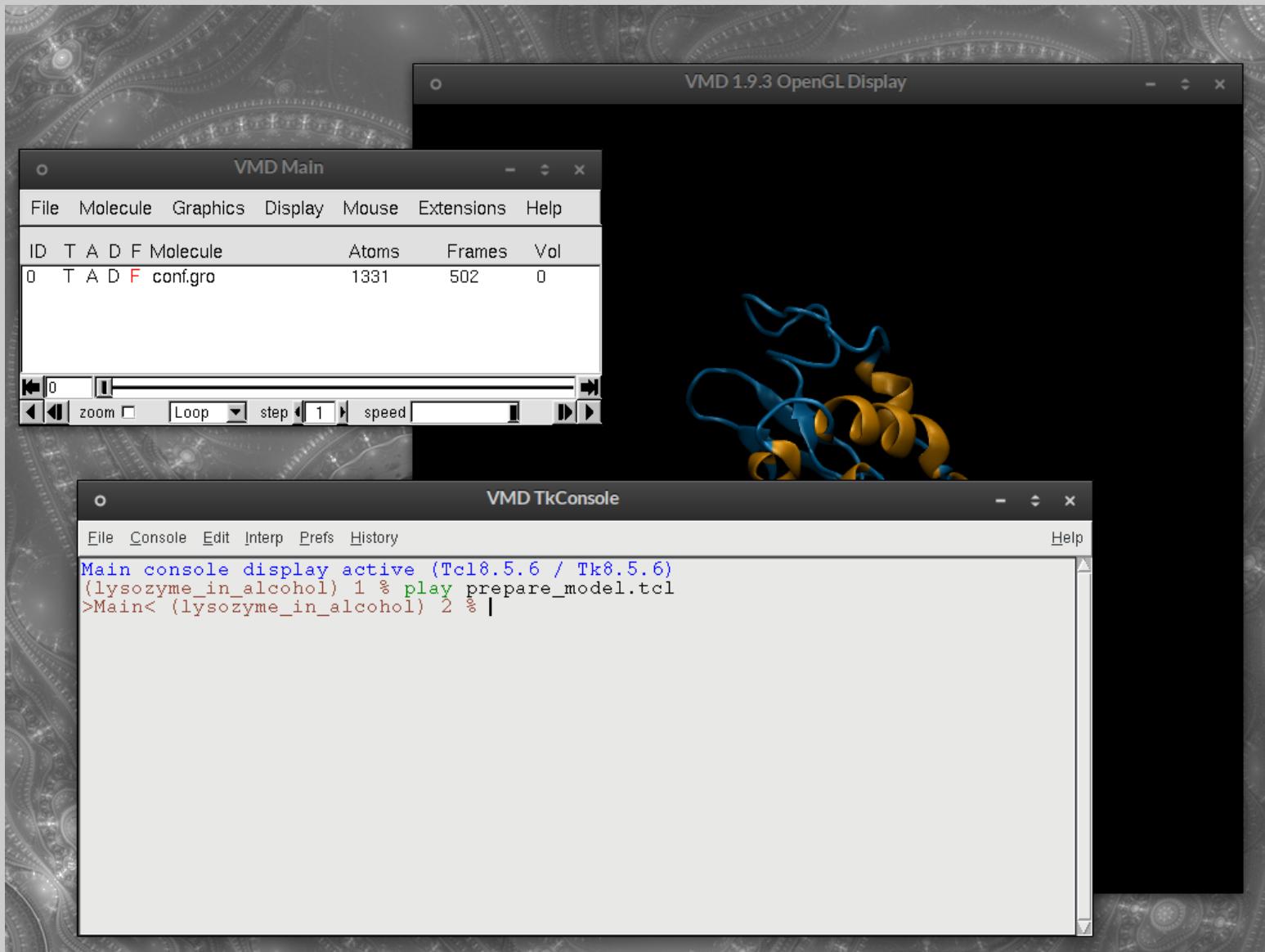
# Combining trajectories, movies, and scripting

```
1 # Load topology and trajectory
2 mol new conf.gro
3 mol addfile traj.xtc waitfor all
4
5 # Universal palette
6 color change rgb  0 0.00 0.47 0.72
7 color change rgb  1 0.96 0.25 0.00
8 color change rgb  3 0.99 0.63 0.00
9 color change rgb  4 0.94 0.89 0.26
10 color change rgb 5 0.00 0.30 0.45
11 color change rgb 7 0.00 0.62 0.45
12 color change rgb 9 0.89 0.49 0.68
13 color change rgb 10 0.25 0.75 1.00
14
15 # Global parameters
16 display projection orthographic
17 color display background white
18 axes location off
19
20 # Delete default representation
21 mol delrep 0 top
22
23 # Show protein as cartoon, color by secondary structure
24 mol representation NewCartoon 0.3 10.0 4.1 0
25 mol color Structure
26 mol selection {protein}
27 mol addrep top
28
29 # Trajectory smoothing
30 mol smoothrep 0 0 5
31
32 # Secondary structure colours
33 color Structure "Alpha Helix" orange
34 color Structure 3_10_Helix orange
35 color Structure Pi_Helix orange
36 color Structure Extended_Beta blue
37 color Structure Bridge_Beta blue
38 color Structure Turn blue
39 color Structure Coil blue
40
41 # Go to the first frame (topology)
42 animate goto 0
43
44 # (Re)calculate secondary structure
45 vmd_calculate_structure top
```

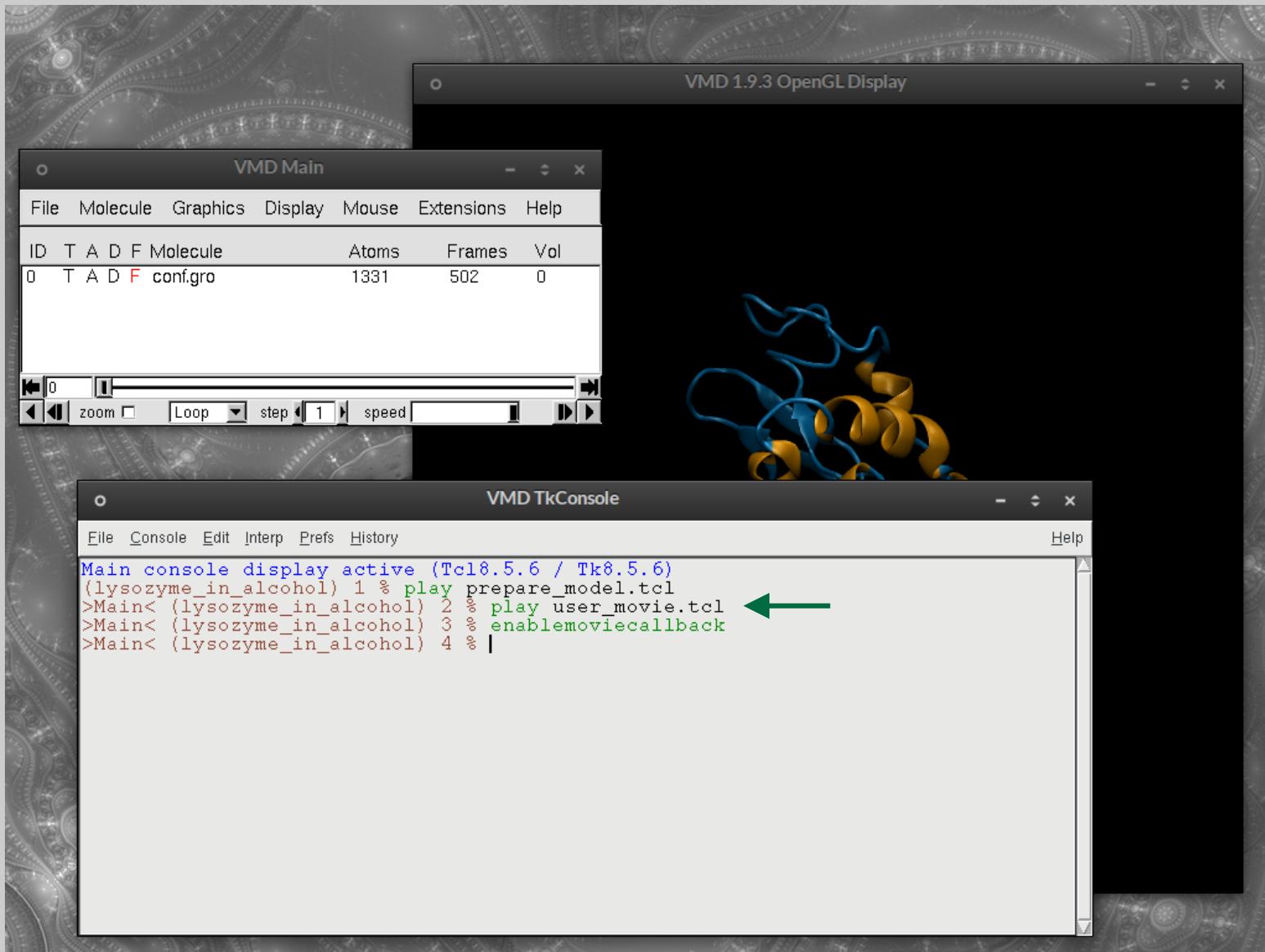
# Combining trajectories, movies, and scripting



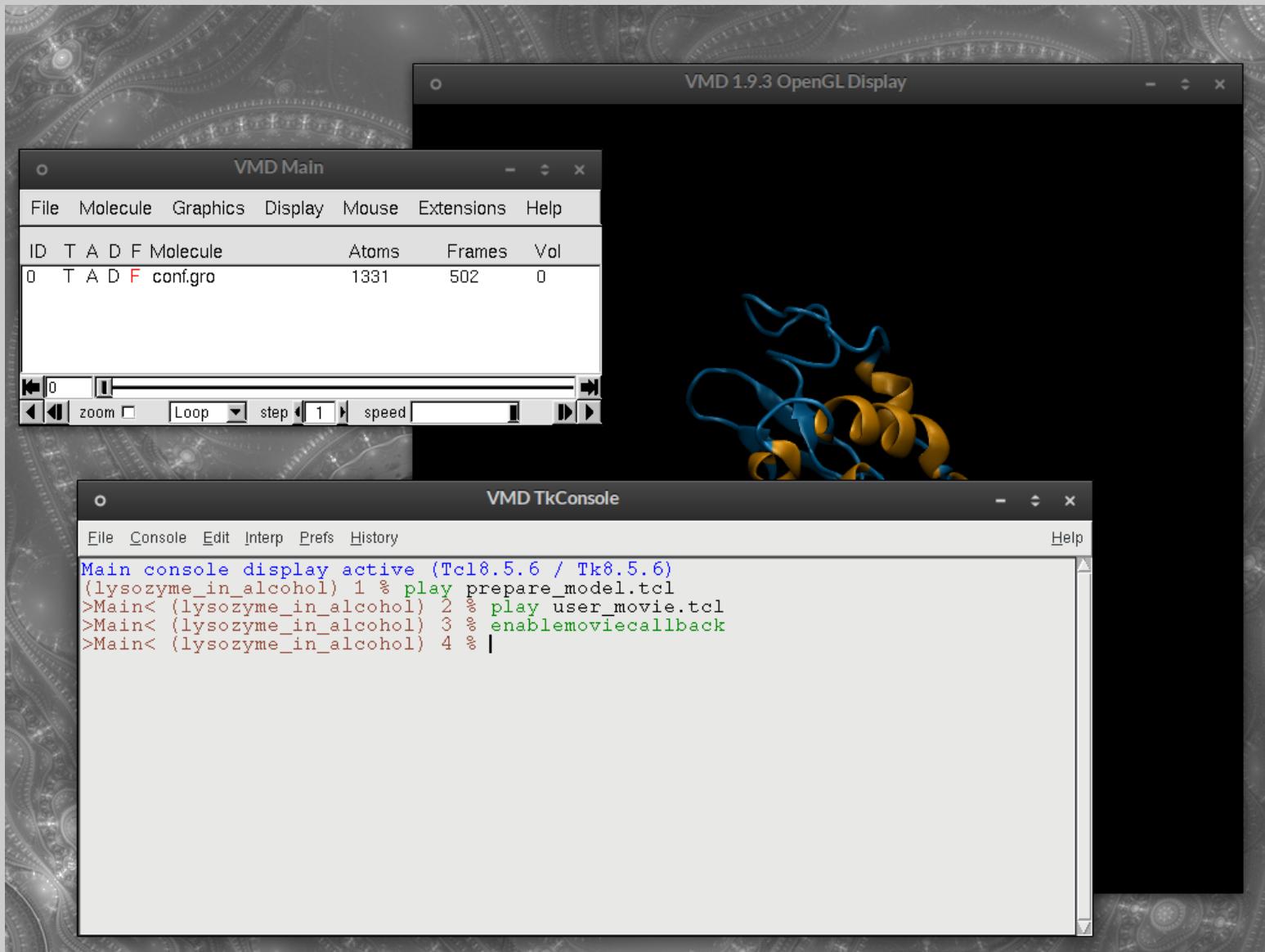
# Combining trajectories, movies, and scripting



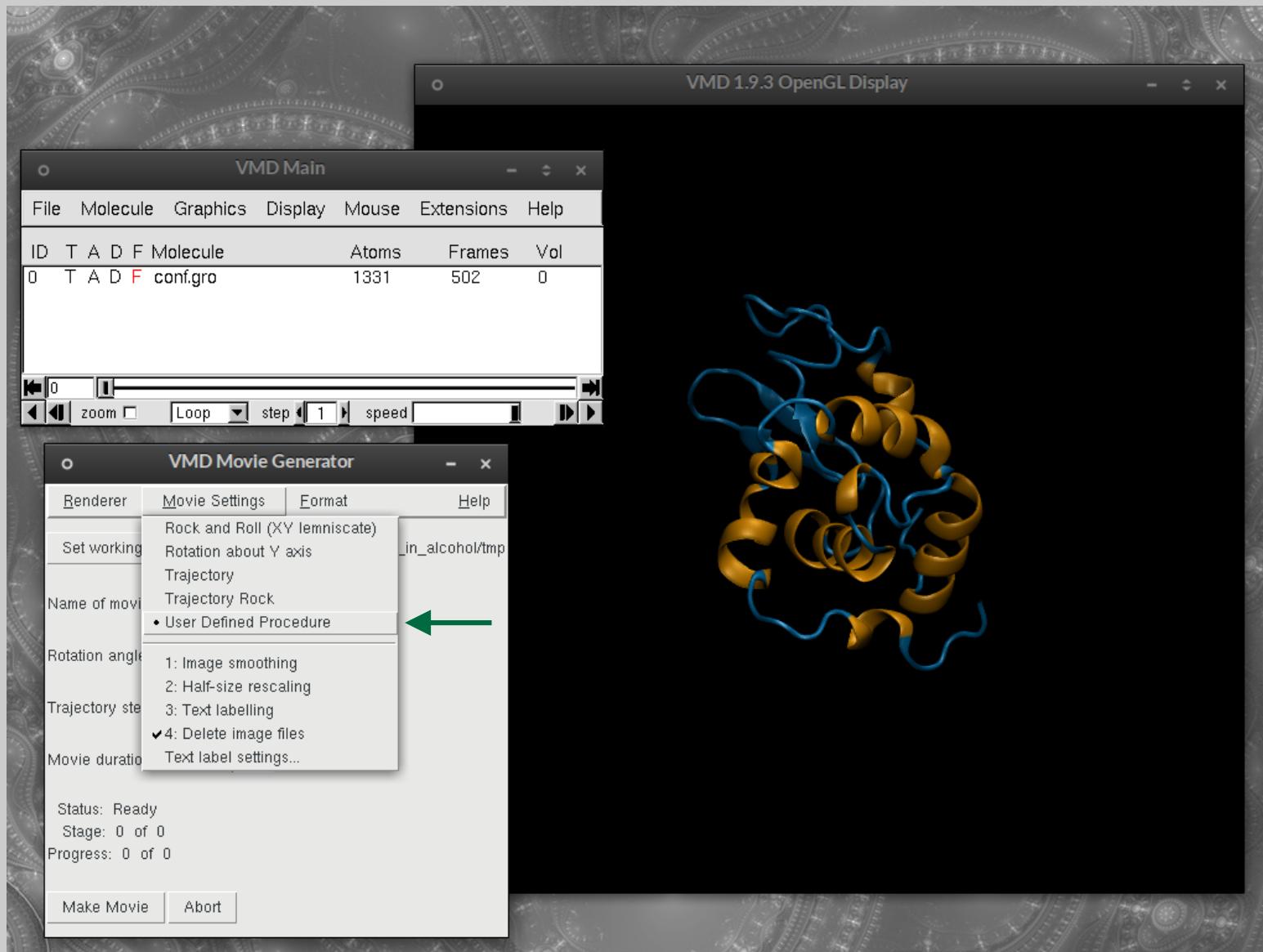
# Combining trajectories, movies, and scripting



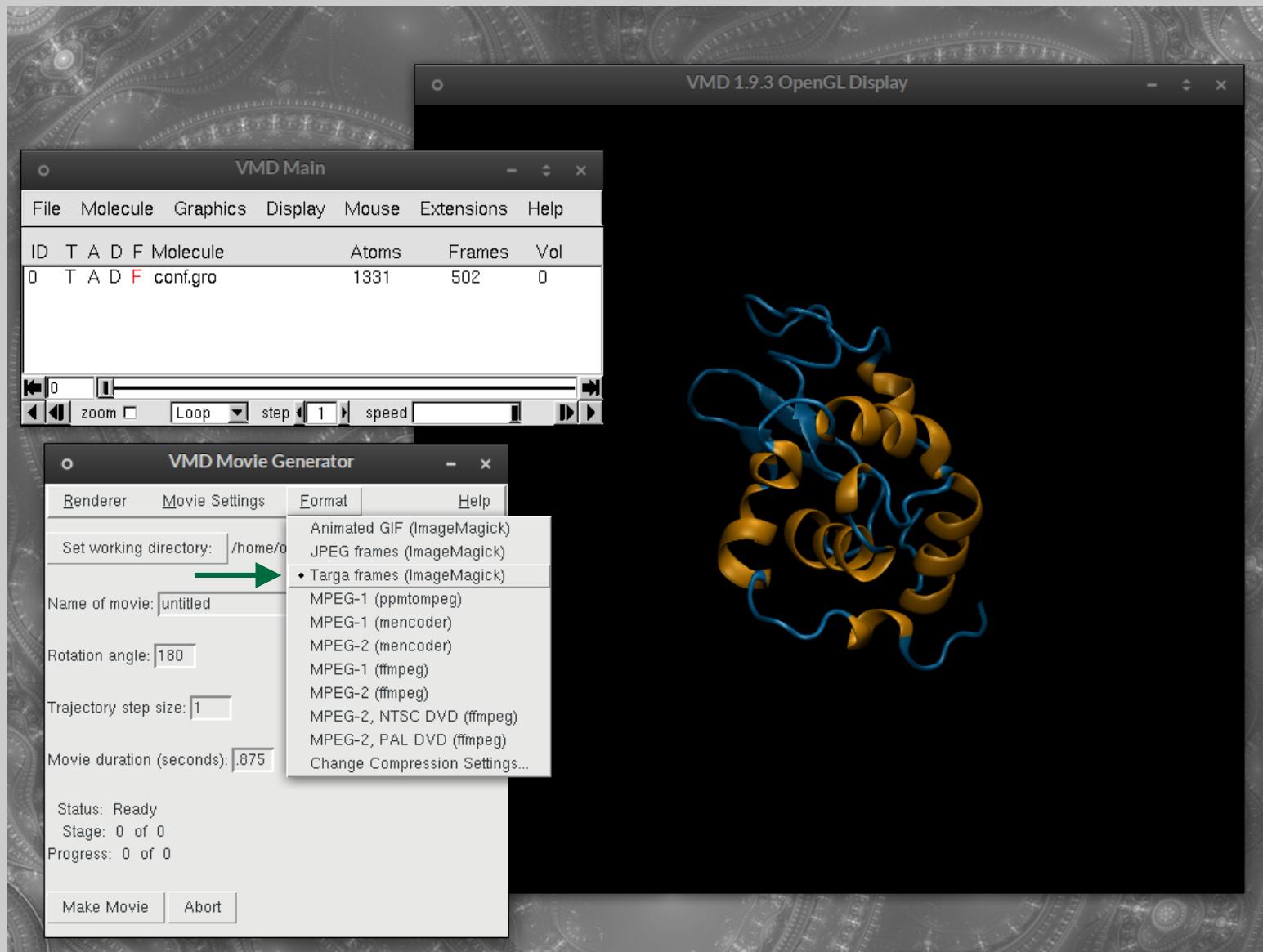
# Combining trajectories, movies, and scripting



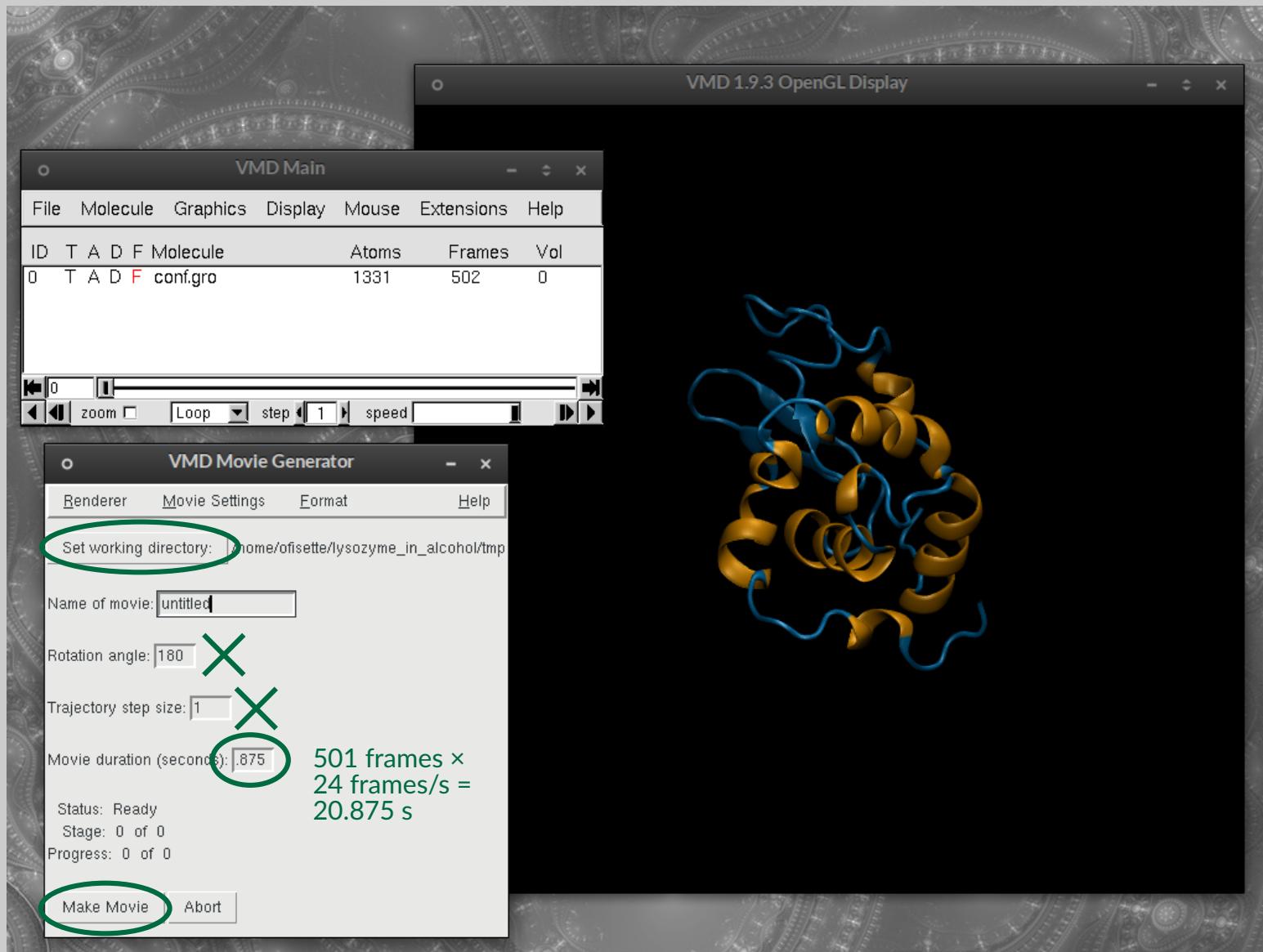
# Combining trajectories, movies, and scripting



# Combining trajectories, movies, and scripting



# Combining trajectories, movies, and scripting



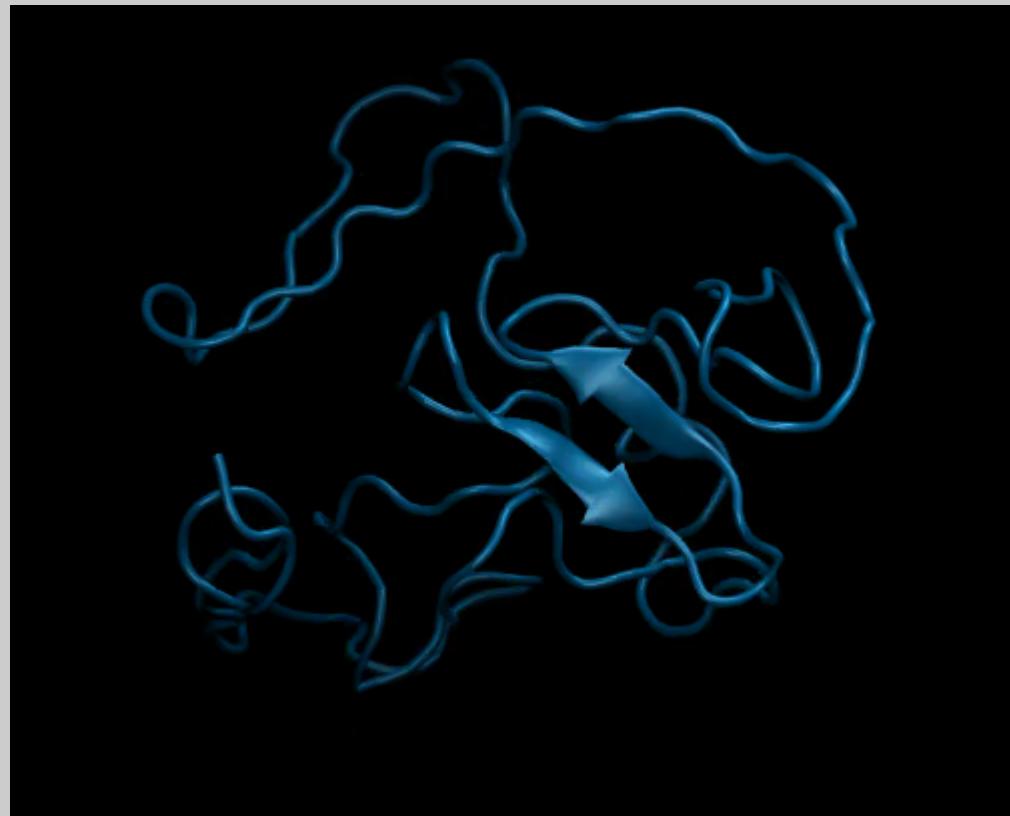
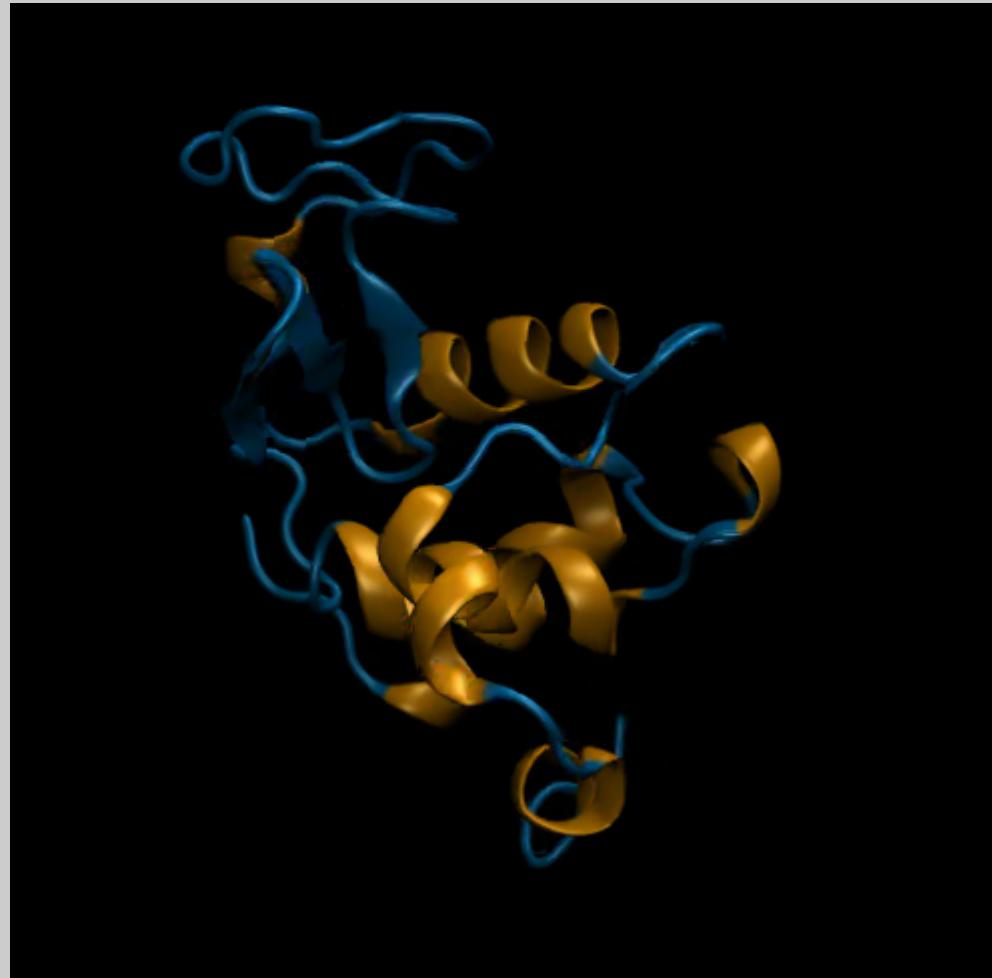
# Combining trajectories, movies, and scripting

```
1 #!/usr/bin/env bash
2 set -e; shopt -s expand_aliases
3
4 ffmpeg -framerate 24 \
5         -i "tmp/final.untitled.%05d.tga" \
6         -vf vflip \
7         -c:v libx265 "lysozyme_ethanol.mp4"
```

7,42

Tout

# Combining trajectories, movies, and scripting



# Resources

- VMD website
  - <https://www.ks.uiuc.edu/Research/vmd/>
- Using VMD (tutorial)
  - <https://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/>
- VMD User's Guide
  - <https://www.ks.uiuc.edu/Research/vmd/vmd-1.8.3/ug/>
- Compute Canada Documentation (CC Doc): VMD
  - <https://docs.computecanada.ca/wiki/VMD>
- WestGrid webinar: Molecular visualization with VMD
  - [https://www.westgrid.ca/events/molecular\\_visualization\\_vmd](https://www.westgrid.ca/events/molecular_visualization_vmd)

# Questions

?