Molecular visualization with VMD

In this tutorial we'll cover some basics of VMD operation, taking as exemples a structure file and an MD trajectory that has been produced from it. Compared to PyMOL, VMD is better suited to larger multi-component systems, and trajectory handling. Here, we use it to create publication-quality images of a structure of a pore formed by several transmembrane helices, embedded in a POPC:POPG membrane.

Launching VMD

From the command line, in the tutorial's directory, you can start VMD and simultaneously load the **`peptide_membrane.gro`** structure file by running:

```
vmd peptide_membrane.gro
```

(the **`.gro`** is a structure format similar to the **`.pdb`**)

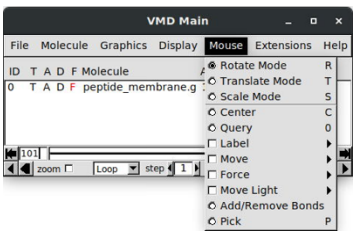
You should get a couple new windows: the display window and the menu window. Your shell will also have been taken over by the VMD console; you don't need to worry about it right now — it'll revert to your regular shell after you quit — but important information sometimes shows up there.

> **Note:** by default, VMD launches in a rendering mode that can be *extremely* slow when drawing spheres/curved surfaces. You'll improve your experience considerably if you change it as soon as you start. Go to **`Display->Rendermode`** and select **`GLSL`**.

In the display window you can now use the mouse to move the structure around. Use the left button and right button to rotate around different axes, and the mouse wheel to zoom in/out (laptop trackpads often achieve the same function via their scrolling gesture).

Mouse modes and actions

What the mouse does when dragged or clicked depends on the current *mouse mode*. The most used modes at this stage are the *Rotate mode* — the one you start in by default — and the *Translate mode*. You can access them under the **`Mouse`** menu, but it's much handier to use the keyboard shortcuts **`R`** and **`T`**, respectively (only usable when the display window has focus). In the translate mode you pan the image with left button, and zoom in/out with the right one.



Besides modes, there are mouse actions. These usually do something if you click on an atom/bond, but otherwise behave like the Rotate mode. A useful one is the *Center* action. It allows you to click on any atom, which will set it as the center of rotation.

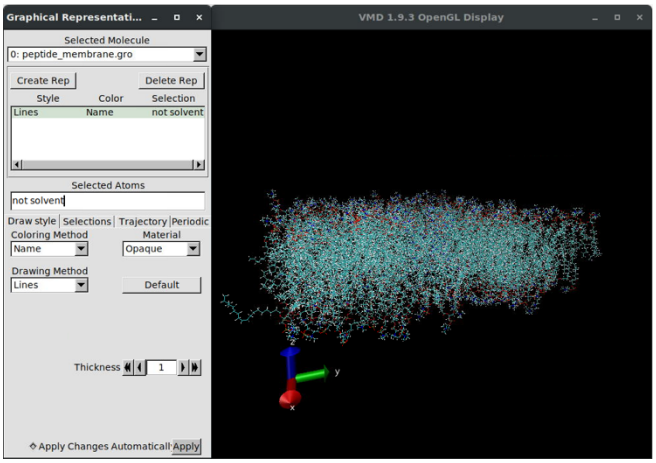
Try controlling the view to your liking using these tools. If you ever get lost from the scene, pressing the equals sign ('**`=`**') will re-center the structure.

Controlling looks – the Representations window

We'll now control how the system looks. By the end of this tutorial you'll have a publishable image (which the one you're looking at right now is not). From the menu, select **`Graphics->Representations`**. This new window allows you to control all aspects of what is

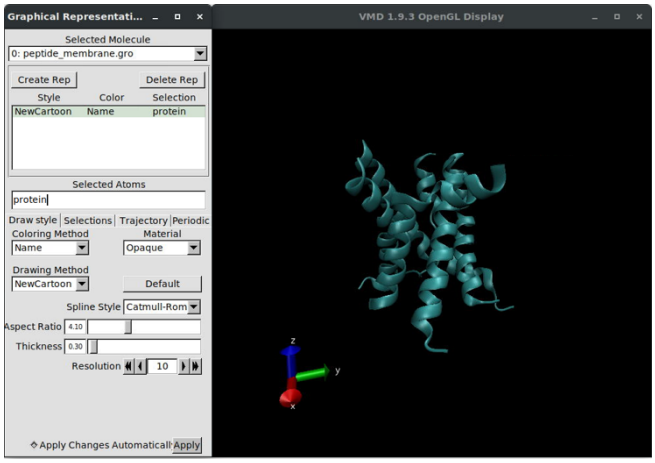
shown and how. Each representation consists in a selection of atoms, displayed in a given style/color. Multiple representations are possible, even representations covering overlapping sets of atoms. Right now you should have a single representation showing all atoms connected by lines.

Selections and styles



The image as is is not clear at all. Much of the noise comes from having the water visible. Let's hide it. We do this by replacing **`all`** in the Selected Atoms string by a selection that excludes the water. Typing in **`not solvent`** (and pressing Enter) does the trick.

The image got a bit better, but still not very good. It's still practically impossible to tell the protein apart from the lipids (hint: it's in the dead center of the membrane). Let's hide everything except the protein. It is cumbersome to specify the protein as the negation of solvent and membrane; we can much simply replace the selection string by **`protein`**. Tadaa.

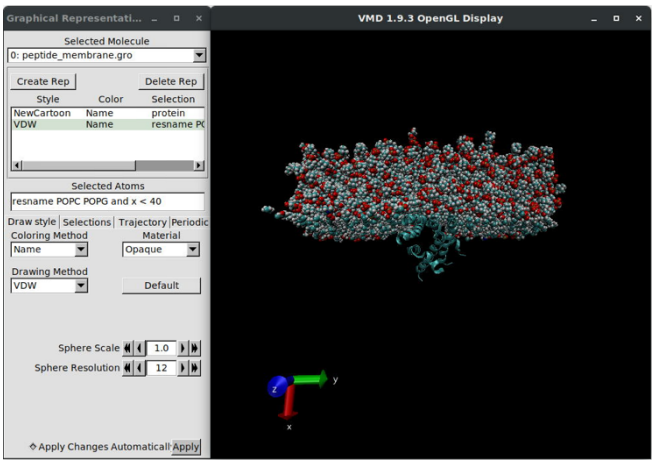


Having the peptide helices shown as lines is terrible. Let's show them as the usual cartoon style everyone's used to. Change the Drawing Method to **`New Cartoon`**. Much better! But now all we have is protein in empty space. How best to also represent the membrane in a clear way? The best now is to create a new representation, just for the membrane, that we will then adapt to our needs. Click **`Create Rep`** to duplicate the current representation.

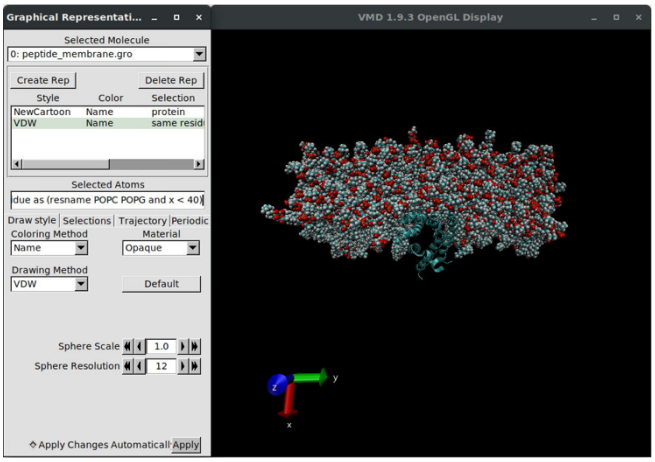
Not much happens because the two representations are the same. Let's change the selection to mean the membrane. Sadly, VMD doesn't have a shortcut such as "lipids", meaning we'll have to select them in a different way. Our membrane is composed of molecules with residue names POPC and POPG. We can then use the selection string **`resname POPC POPG`**.

If you tried that, you'll see that still, nothing changes : (The reason is that when you created the new representation, it retained the **`New Cartoon`** drawing method. This only applies to proteins, which is why nothing was shown. Choose something else, like the VDW sphere style.

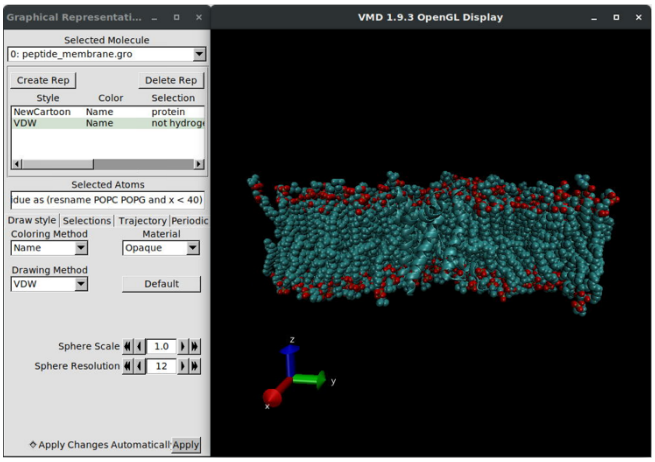
Ok we got somewhere, but the image is, again, too crowded to be able to see the protein. A typical trick in this situation is to only show lipids behind the protein. Let's hide all lipids in front of it. We do this by adding geometric constraints to the selection so that it reads **`resname POPC POPG and x < 40`**. This means that only atoms at x positions in the box before 40 Angstrom get displayed (the right value to choose depends on the system size, and may take some trial and error).



Getting better — at least we can now see the protein and membrane simultaneously along the x axis. However, the lipids were brutally cut along that x = 40 plane and look unnatural with chopped tails. It's better if we instead select "whole lipids with at least one atom before x = 40". We can do that by writing **`same residue as (resname POPC POPG and x < 40)`**. Nice.



Still a big source of noise is the hydrogen atoms. Let's get rid of them. VMD does have the **`hydrogen`** selection shortcut, so we can just write **`not hydrogen and same residue as (resname POPC POPG and x < 40)`**.

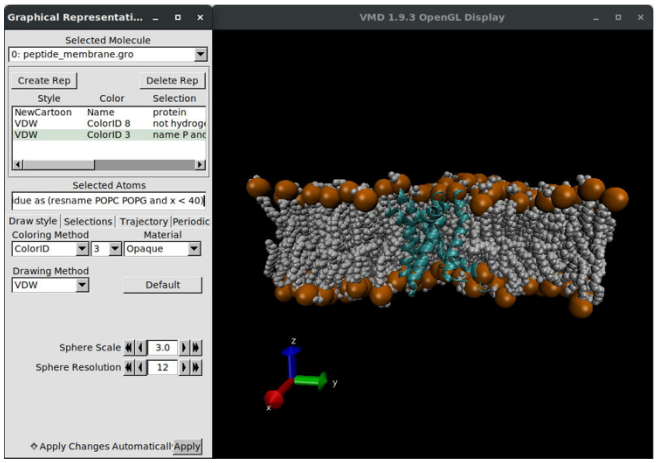


Ok, now the color is the problem. Since membranes are often depicted in lighter colors let's change it. The Coloring Method menu has many options, but the most used ones are by atom **`Name`** (what you have currently) and by **`ColorID`**. Choose **`ColorID`** and note that a new

menu shows up; it allows you to assign arbitrary colors to the selection. Choose something like **8 white**.

Uniform coloring discards useful membrane highlights (such as where the heads are vs the tails). How can we recover and enhance that perception? A way is to create yet another representation and set it to the phosphorus atoms of those headgroups. You can select them all simply by typing **name P**. This selects also the phosphorus of lipids that aren't being shown; to limit the selection to the lipids behind the peptides you can extend it with the earlier concept: **name P and same residue as (resname POPC POPG and x < 40)**.

Well, that didn't do much. That's because the Ps are still being represented as small spheres, and completely overlap with the representation already in place. Make the spheres bigger using the **Sphere Scale** arrows and color them, say, orange.



We're almost there, selection-wise! You'll notice that there are some atoms poking out of the headgroup spheres. Those are the choline and glycerol groups of POPC and POPG respectively, that lie beyond the phosphate. Let's hide them from the lipids' representation so as to have a cleaner image. You'll want to hide the lipid atoms named **C11 C12 C13 C14 C15 N OC2** and **OC3** (no copy-paste example this time; time to think!).

Note: knowing which atoms to hide involves understanding the structure of the molecule at hand and then finding out the relevant names. One way to do that is to use the **Mouse** action **Query** (keyboard shortcut **0**): click any atom, and information about it will show up in VMD's console.

Note: remembering the selection syntax, knowing available shortcuts, and finding out the names and resnames in your system can be difficult. The **Selections** tab helps by providing lists of keywords, logical operators, and available values. You can even fully construct your selection just via the buttons there instead of typing! Just don't forget to return to the **Draw style** tab.

Cleaning up and rendering

Fixing defaults

VMD starts with a number of, frankly, poorly chosen defaults that get in the way of making a good image. Let's fix them one-by one:

Perspective: it is switched on by default but often feels exaggerated (especially the spheres) and can be misleading. Change it to **Ortographic** under **Display**.

Depth cueing: it's the dark fog that darkens far away stuff. Switch it off also under **Display**.

Axes: can be useful, initially, to get your bearings but are almost never needed in a rendered image. They're also the sign of a VMD n00b. Get rid of them under **Display->Axes**.

Background: the black backdrop is jarring and seldom fits well in publications. Change it under **Graphics->Colors->Category 'Display'->Name 'Background'**. Set color **8 white**.

Note: these defaults can be permanently changed so that you don't need to re-fix them every time you use VMD. If it exists, VMD reads a file **.vmdrc** from your home every time it starts and executes the commands in there. To implement these and some other useful fixes use:

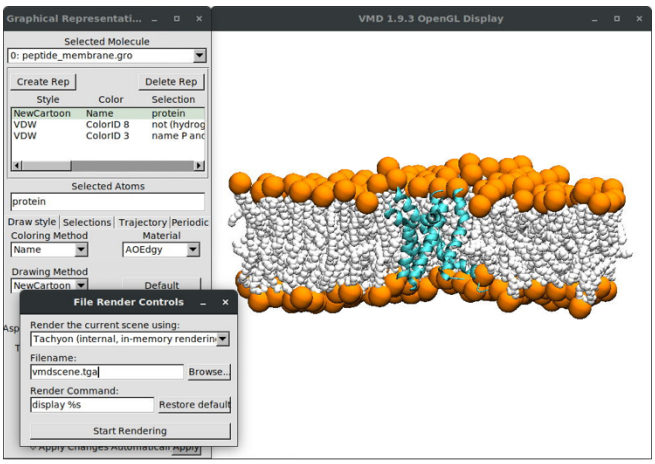
```
display rendermode GLSL
display projection orthographic
display depthcue off
axes location off
color Display Background white
animate style once
display culling off
display nearclip set 0.01
display shadows on
display ambientocclusion on
display aoambient 1.0
display aodirect 0.1
```

Rendering

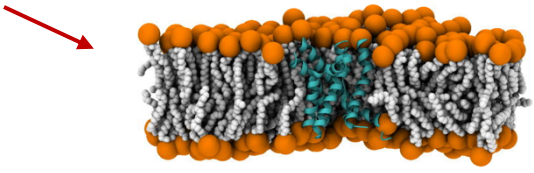
When rendering, VMD uses algorithms to try and mimic realistic-ish lighting and material responses. These can be slow, and therefore aren't used in the interactive display window.

Some tweaks will help with the final look, though it's an area where different applications certainly have different needs. Under **Display->Display Settings** switch on **Shadows** and ambient occlusion (**Amb. Occl.**), and set **AO Ambient** to 1.0 and **AO Direct** to 0.1. These changes don't affect how the image looks in the interactive window, but will when it gets rendered.

Now go to each representation and change its **Material** to **AOChalky** (or **AOEdgy** for the protein). This changes how light is processed in interaction with the objects, but note that the interactive display does a poor job at previewing; the end result should be much better.



Finally, under **File->Render...** select the **Tachyon internal** renderer. Press **Start Rendering**. The console will show the progress. When it's done, check out your nice rendered image in the same folder where you started VMD :)



Handling trajectory data

When dealing with molecular dynamics data you'll often need to visualize trajectories — progressions of atom positions over time. VMD can use trajectory data to animate images.

Depending on their type, trajectory data can be standalone or dependent on an existing structure. The `.xtc` format of the `trajectory.xtc` you have in the tutorial is the dependent type, in that it only stores atom positions over time. No information is present about atom types, atom names, or residue names. In order for VMD to be able to use the trajectory information you must load it onto an already read system (VMD calls it 'molecule'), such as the one we have been working on.

To load the trajectory data first select the molecule name in the main menu window. Then right-click it and choose `Load Data Into Molecule...`; a new window appears. Using `Browse...` find and choose the `trajectory.xtc` file. Back in the previous window tick `Load all at once` (this will cause VMD to read the entire file before trying to display the frames, else it may be rather slow). Finally, click `Load`.

After some moments you'll see the information in the menu changing to indicate that several frames are available. You can now use the arow buttons there to play the trajectory as if it were a movie (you'll notice the movie loops when it reaches the end; you may want to change that option from `Loop` to `Once`).

Time-dependent tweaks

You may notice that lipids start passing the 40 Angstrom barrier we set earlier. This happens because, by default, the selections are only calculated when they are defined. If you want a selection to be applied every frame go to the `Graphics->Representations` window and, for each representation, go to the `Trajectory` tab and tick `Update Selection Every Frame`.

Finally, molecular motion often looks jittery due to the several timescales of movement occurring simultaneously. We usually are interested in the longer scale movements, rather than the thermal noise. A neat way to filter out such noise is to represent each frame's positions as an average of past *n* frames. You can activate this also in the `Trajectory` tab, by setting the number of frames to average over as the `Trajectory Smoothing Window Size`. Try 4 and see how it looks (you'll need to set that for each representation independently).

Making movies

Being able to make movies is one of the perks of working with molecular dynamics. With VMD, the concept is to simply render each frame separately and then to combine them into a video. There's an extension that aids in that, under `Extensions->Visualization->Movie Maker`. It can do different types of movies, either on movements/rotations of a static structure, or of frames along a trajectory. Under `Movie Settings` select `Trajectory`. As `Renderer`, choose `Internal Tachyon` and as `Format` choose `JPEG frames`. Set an appropriate working directory (such as the tutorial one) and press `Make Movie`. Frames will be progressively rendered into that directory. When it's done (it may take quite a while) you can use those images to make a movie (VMD fails at that step because it can't find in the virtual machine encoding software).

The end!