# VMD Tutorial

Source: https://www.ks.uiuc.edu/Training/Tutorials/vmd-index.html

## Unit 1: Introduction

1. Download the pdb file of Ubiquitin (1UBQ.pdb)
2. To load,"New Molecule" → "Browse" → "Load"
3. Hotkeys

| | |
|---|---|
| c | Center: Change rotation point by clicking on an atom. Reset to default with Reset View. |
| 0 | Query: Click on an item to print its name in the console. |
| 1 | Label → Atom: Toggle atom label on/off by clicking on an atom. |
| 2 | Label → Bond: Toggle bond distance label by clicking on two atoms in sequence. |
| 3 | Label → Angle: Toggle angle label by clicking on three atoms in sequence. |
| 4 | Label → Dihedral: Toggle dihedral angle label by clicking on four atoms in sequence. |
| 5 | Move → Atom: Change atom position by clicking and dragging the atom. |
| 6 | Move → Residue: Move all atoms in a selected residue by dragging one atom. Shift key rotates about selected atom; middle button rotates about a line through the atom. |
| 7 | Move → Fragment: Move all atoms in a selected fragment similarly to residues. |
| 8 | Move → Molecule: Move all atoms in a selected molecule by dragging one atom. |
| 9 | Move → Rep: Move atoms in a selected representation by clicking on an atom within that rep. |

4. Representation style

| | |
|---|---|
| Lines | simple lines for bonds, points for atoms |
| Bonds | lighted cylinders for bonds |
| DynamicBonds | dynamically calculated distance-based bonds |
| HBonds | display hydrogen bonds |
| Points | just points for atoms, no bonds |
| VDW | atom as sphere |
| CPK | ball & stick model |
| Licorice | ball & stick model, radius cannot be changed |
| Polyhedra | polyhedra connecting atoms within a cutoff radius |
| Trace | connected cylindrical segments through C alpha atoms |
| Tube | smooth cylindrical tube through the C alpha atoms |
| Ribbons | flat ribbon through the C alpha atoms |
| NewRibbons | smooth ribbon through the C alpha atoms |
| Cartoon/ NewCartoon | secondary structure: helices=cylinders, beta-sheets=solid ribbons, other=tube |
| PaperChain | display ring structures as polygons, colored by ring pucker |

| | |
|---|---|
| Twister | flat ribbon tracing glycosidic bonds, with twists oriented by sugar residues |
| QuickSurf | molecular surface (Gaussian density surface) |
| MSMS | molecular surface as determined by the program MSMS |
| Surf | molecular surface as determined by SURF |
| VolumeSlice | display a texture mapped slice from a volumetric data set |
| Isosurface | display an isovalue surface from a volumetric data set |
| FieldLines | field lines generated by integrating particles by volume gradient vectors |
| Orbital | molecular orbital selected by wavefunction type, spin, excitation, and orbital ID |
| Beads | per-residue approximate bounding spheres |
| Dotted | dotted van der Waals spheres for atoms, no bonds |
| Solvent | dotted representation of the solvent accessible surface |

5. Color categories

| | |
|---|---|
| Display | Color of background, gradient, depth cueing, text |
| Axes | The components of the axes |
| Name | The available atom names (color by Name) |
| Type | The available atom types (color by Type) |
| Element | Atomic elements (color by Element), with "X" for unknown |
| Resname | The residue names (color by ResName) |
| Restype | The residue types<br>- non-polar=white, basic=blue, acidic=red, polar=green |
| Chain | The one-character chain identifier. |
| Segname | The segment names (color by SegName) |
| Conformation | The available conformation codes (color by Conformation) |
| Molecule | The names assigned to each molecule (color by Molecule) |
| Highlight | The protein, nucleic, and non-backbone colors |
| Structure | The secondary structure type (helix, sheet, coil) (color by Structure) |
| Surface | The surface types |
| Labels | The different labels (atoms, bonds, etc.) |
| Stage | The colors for the checkboard stage |

6. Select atoms by typing “Singlewords” in “Selected Atoms”
    a. And, or, not
    b. “Keyword”
7. View sequence in “Sequence Viewer”, color code obtained from STRIDE

a. B-value = temperature factor
b. Struct = secondary structure

| | |
|---|---|
| T | Turn |
| E | Extended conformation ($\beta$ sheets) |
| B | Isolated bridge |
| H | Alpha helix |
| G | 3-10 helix |
| I | Pi helix |
| C | Coil |

8. Save visualisation state:"File" → "Save State" (.vmd)
9. Change background: "Graphics" → "Colors" → "Display" → "Background"
10. Reder image: "File" → "Render" → "Render using" → "TachyonInternal" (.tga)

# Unit 2: Multiple Molecules and Scripting

Description:

- Experiment: Equilibration simulation of ubiquitin
- Medium: Water
- Duration: 1 ns
- Task: Compare conformation of ubiquitin at the end of this simulation with the initial crystal structure

1. Download the pdb file of Ubiquitin {1UBQ.pdb} → Rename to "crystal"
2. "New Molecule" → "Browse" → "Load"
3. Add the structure file containing Ubiquitin at the end of simulation {ubiquitin.psf} to "New Molecule"
4. Add the coordinate file at the end of simulation {ubiquitin-equilibrated.coor} to "ubiquitin.psf" → Rename to "simulation"
5. Unfix both molecules (F in red) and "Reset View"
6. Only display the "crystal" molecule (D in black)
7. Set "crystal" molecule as top (T)
8. "Extensions" → "Tk Console"

| **Commands** | **set crystal [atomselect 0 "all"]** | **Set all atoms in molecule 0 as "crystal"** |
|---|---|---|
| | $crystal num | Obtain the number of crystal |
| | $crystal set beta 0 | Set "crystal" as beta 0 |
| | $crystal get resname | |
| | set M [measure fit $alpha1 $alpha2] | Transformation matrix M that map first selection onto second |
| | $crystal move $M | |
| | atomselect macro bstrand1 {protein and resid 2 to 6} | Create macro for bstrand1 including residues 2 to 6, found in Selections > singlewords |

| | | |
|---|---|---|
| | source filename.tcl | Import source code |
| Selection methods with “” | all | All atoms |
| | hydrophobic | |
| | alpha | alpha carbon |
| Display methods | beta 0 | |
| | radius 1.5 | |
| Obtain information | resname | Residual name |
| | resid | Residual index |
| | index | Atom index |
| | name | Atom name |
| | {x y z} | Coordinates |
| file | psf | Protein structure |
| | vmd | Status |
| | dcd | Trajectory |
| | tcl | Text script |

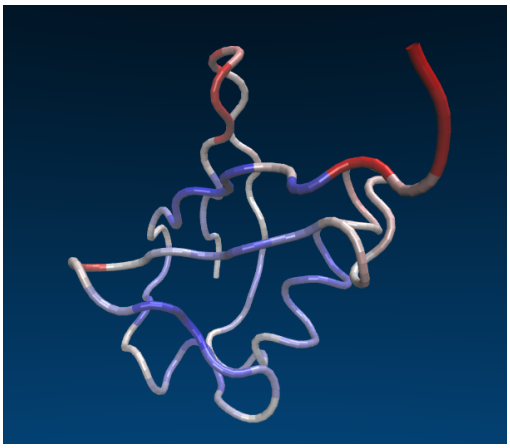
9. Display both molecules (D in black)
10. For both crystal and simulation, set coloring method to “ColorID” and “0 blue” and “1 red” respectively, drawing method to “Tube”
11. In Tk Counsole window, select the protein backbone of both proteins
    a. set alpha1 [atomselect 0 “alpha”]
    b. set alpha2 [atomselect 1 “alpha”]
12. Calculate and apply transformation matrix M to best map alpha2 to alpha1
    a. set M [measure fit $alpha1 $alpha2]
    b. set crystal [atomselect top “all”]
    c. $crystal move $M
13. Run pre-written script to compare atoms’ positions
    a. source coloring.tcl

```
(input) 55 % source coloring.tcl
-1.4344735145568848 1.5170034170150757 -11.387672424316406, -1.8203625679016113
1.046593189239502 -11.700506210327148
-1.4344735145568848 - -1.8203625679016113
-1.2171720266342163 4.893920421600342 -9.64897632598877, -1.5955531597137451 4.4
61385726928711 -10.035935401916504
-1.2171720266342163 - -1.5955531597137451
-2.2546496391296387 5.21519660949707 -6.0121073722839355, -2.407890558242798 4.8
41493606567383 -6.30332612991333
-2.2546496391296387 - -2.407890558242798
-2.1305925846099854 8.245089530944824 -3.6896939277648926, -2.268209218978882 8.
021492004394531 -4.130740642547607
-2.1305925846099854 - -2.268209218978882
-0.6730284690856934 8.250974655151367 -0.16197004914283752, -0.6497521996498108
8.03714656829834 -0.564492404460907
-0.6730284690856934 - -0.6497521996498108
-1.9813416004180908 11.187929153442383 1.9380509853363037, -1.056281328201294 11
.034486770629883 1.8269755840301514
-1.9813416004180908 - -1.056281328201294
```

```
 # Define the coloring procedure:
proc tutorialcoloring {} {

 # Get the molIDs of the first 2 molecules.
 set mol1 [lindex [molinfo list] 0]
 set mol2 [lindex [molinfo list] 1]

 # Create our two sels
 set sel0 [atomselect $mol1 "alpha and protein"]  ;# crystal
 set sel1 [atomselect $mol2 "alpha and protein"]  ;# simulation

 # Create an empty list (for all the displacements)
 set mylist {}
```

```
 # loop over the position vectors v1 and v2 for each atom:
 foreach v0 [$sel0 get {x y z}] v1 [$sel1 get {x y z}] {
  puts "$v0, $v1"
  puts "[lindex $v0 0] - [lindex $v1 0]"
  set dx [expr [lindex $v0 0] - [lindex $v1 0]]
  set dy [expr [lindex $v0 1] - [lindex $v1 1]]
  set dz [expr [lindex $v0 2] - [lindex $v1 2]]

  # Calculate displacement for a given atom
  set disp [expr ($dx*$dx + $dy*$dy + $dz*$dz)]
  lappend mylist $disp
 }

 # Assign the displacements to beta values of the crystal molecule
 $sel0 set beta $mylist

}

# Run the procedure here
tutorialcoloring
```

14. For “crystal” molecule, change Coloring Style to “Beta”
15. Hind “simulation” molecule
16. “Graphical Representations” → “Trajectory” → “Color Scale Range” →”crystal” → “0””5” (A) → “Set”
17. “Graphics” → “Colors” → “Color Scale” → “Method” → “BWR” low displacement=blue; high displacement=red, in between=white
18. Shift “Midpoint” to 0.1
19. Output as {1ubq-equilibration-in-water.vmd}



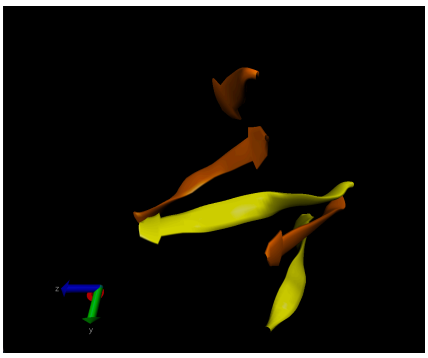
# Unit 3: Trajectories, Macros and Labels

Description:

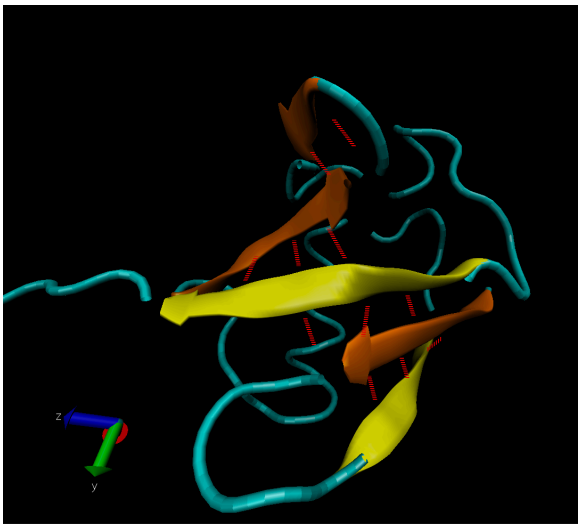
- FMA experiment

1. Add frames {pulling.dcd} to equilibrated structure file {ubiquitin.psf} → “Load”
2. Go to the first frame

3. Drawing Method = “Tube”; Selected Atoms = “protein”
4. “Create Rep” → Drawing Method = “Lines”; Selected Atoms = “water” →Turn off representation
5. To create macros of each strand in the mixed beta sheet, in Tk Console
    a. atomselect macro bstrand1 {protein and resid 2 to 6 }
    b. Repeat for the rest of the strands
    c. Sequence of each beta strand can be found in “Sequence Viewer”, coloured in yellow
6. To create a representation with the 3rd and 5th beta strands, “Create Rep” → “Selected Atoms” → “Selections” → “Sinlewords”: “bstrand3 or bstrand5” → Draw Style = “Cartoon”, yellow
7. Create the representation with bstrand1,2,4. As bstrand4 only has two residues, it is not shown

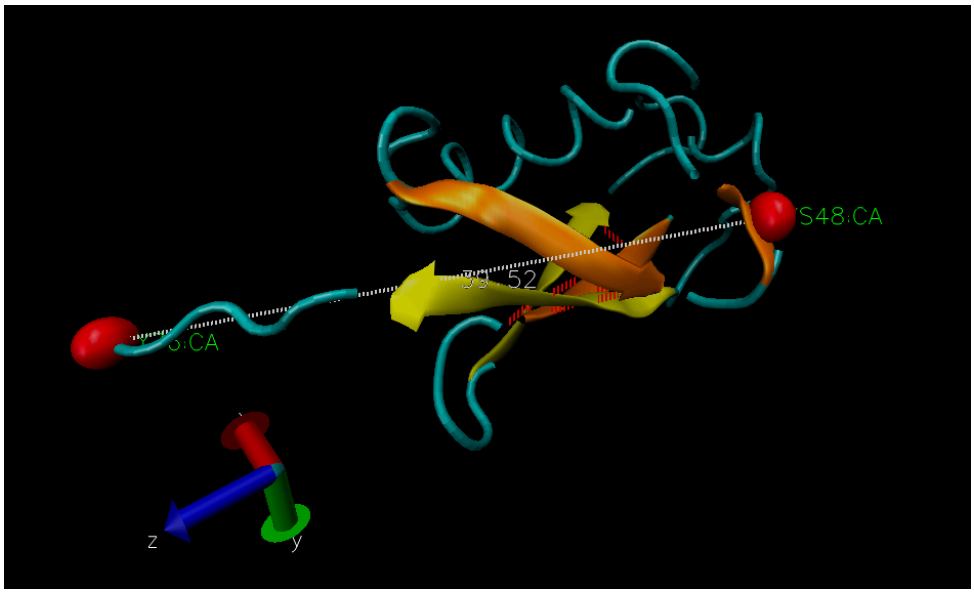


8. Change “protein” to “protein and not betasheet”
9. To see the features of ubiquitin unfolding, “Create Rep” → “Selected Atoms”= ”betasheet and backbone” → Draw Style = “Hbonds”; Drawing Method = “Color ID”, red; Distance Cutoff = “3.2”; Angle Cutoff = “30”; Line Thickness “5”



10. “File” → “Save State” (.vmd)
11. Use the slider to obtain the shape of protein and surrounding water molecules at different times
12. To label bonds or atoms, “Mouse” → “Label” → “Atoms”
    a. To remove labels, “Graphics” → “Labels” → “Delete”
13. To make VDW Representation for alpha carbon of Lysine 48 and of the C terminus
    a. (Tk console) Set sel [atomselect top “resid 48 76 and name CA”]

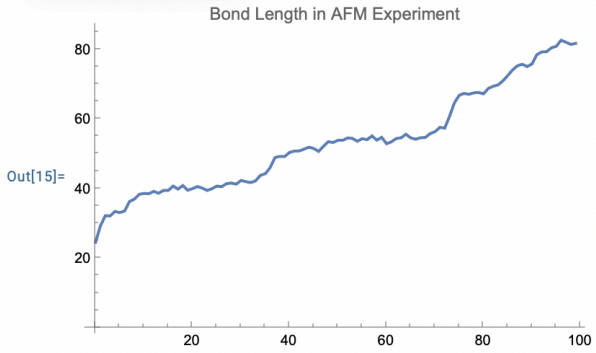
b. (Tk console) $sel get index >> “770 1242”
c. Create VDW Representation with selection index 770 1242
d. Label the 2 atoms




e. “Graphics” → “Labels” → “Bonds” → “Graph” (.dat)

```
In[18]:= data = Import["/Users/heiley/Desktop/SURP/vmd-tutorial/output/2-1ubq-AFM-length.dat"];
In[19]:= plot = ListPlot[data, Joined → True, PlotLabel → "Bond Length in AFM Experiment", LabelStyle → "Black,Bold"]
```

Choose new cell type



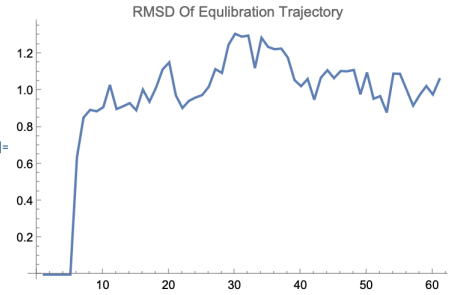

```
In[16]:= Export["/Users/heiley/Desktop/SURP/vmd-tutorial/output/2-1ubq-AFM-length.png", plot]
Out[16]= /Users/heiley/Desktop/SURP/vmd-tutorial/output/2-1ubq-AFM-length.png
```

14. To calculate RMSD of equilibration trajectory
    a. “Molecule” → “Delete Frames”
    b. “File” → “Load Data to Molecule” {equilibration.dcd}
    c. Turn on water representation
    d. (Tk console) source rmsd.tcl

e. Generate graph using {rmds.dat}

```
In[19]:= data = Import["/Users/heiley/Desktop/SURP/vmd-tutorial/output/3-1ubq-equilibration-evolution-rmsd.dat"];
In[20]:= plot = ListPlot[Flatten[data], Joined → True, PlotLabel → "RMSD Of Equlibration Trajectory", LabelStyle → "Black,Bold"]
```

Out[20]=




```
In[21]:= Export["/Users/heiley/Desktop/SURP/vmd-tutorial/output/3-1ubq-equilibration-evolution-rmsd.png", plot]
Out[21]= /Users/heiley/Desktop/SURP/vmd-tutorial/output/3-1ubq-equilibration-evolution-rmsd.png
```

f. {rmsd-fullthrottle.tcl} calculates the average RMSD for each residue in a selection over all frames in a trajectory

**More RMSD.** You can try sourcing another script called `rmsd-fullthrottle.tcl` Take a look at this script to learn simple procedures in tcl, as well as calculating the RMSD of each residue over time and coloring residues according to their RMSD similarly that in Unit 2.