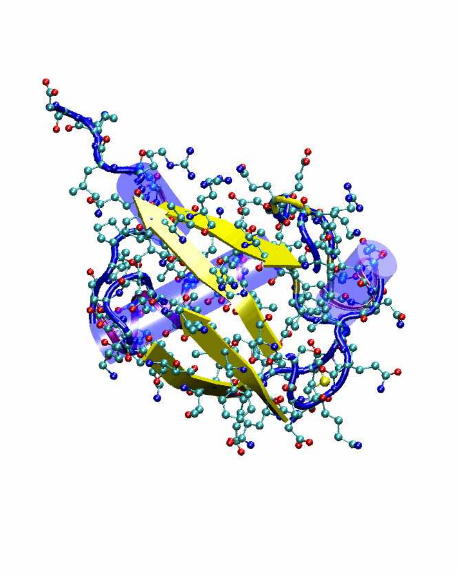
University of Illinois at Urbana-Champaign
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group

VMD Molecular Graphics



VMD Developer: John Stone

Jordi Cohen

Marcos Sotomayor

Elizabeth Villa

May 2003.

Contents

1 Basics of VMD **4**
1.1 Loading a Molecule 4
1.2 Displaying the Protein 5
1.3 Exploring Different Drawing Styles 6
1.4 Exploring Different Coloring Methods 8
1.5 Exploring Different Selections 8
1.6 Multiple Representations 10
1.7 Sequence Extension 11
1.8 Saving your Work 12

2 Multiple Molecules and Scripting **14**
2.1 Loading Multiple Molecules 14
2.2 Using the Main Window 15
2.3 Tcl Scripting Basics and TkCon 16
2.4 The `atomselect` Command 17
2.5 Aligning Two Molecules 20
2.6 Using Color to Display the Deviation 22

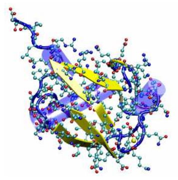
3 Trajectories, Macros and Labels **24**
3.1 Loading trajectories 24
3.2 Macros 25
3.3 Main Menu Animation Tools 28
3.4 Labels 31
3.5 An Example Tcl Script: Calculating the RMSD of a trajectory . 35

Introduction

This tutorial introduces new users to VMD and its capabilities. It can also be used as a refresher course for the occasional VMD user wishing to employ this program more productively. *This tutorial has been designed specifically for VMD 1.8.1 and should take about 3 hours to complete in its entirety.*

The tutorial is subdivided into three separate units of increasing complexity. The first unit covers the basics of molecular graphics representations and will introduce everything you need to know to generate nice graphics. The other two units are targeted towards the scientifically-oriented user and focus on scripting in VMD. While scripting may be skipped by the non-technical users, we encourage everyone to give it a try as it provides some very powerful (and easy to use) tools that cannot be offered by a simple graphical user interface.

The examples in the tutorial will focus on the study of ubiquitin – a small protein with interesting properties. Throughout the text, some material will be presented in separate "boxes". These boxes include complementary information to the tutorial, such as information about the biological role of ubiquitin, and tips or shortcuts for using VMD.



Ubiquitin. This tutorial will focus on the visualization of *ubiquitin* with VMD. Ubiquitin is a small protein of 76 amino acids, that is believed to be present in all eukaryotic cells. It is one of the most conserved of all eukaryotic proteins (the first 74 amino acids form a structure that is identical in insects, trout, bovines and human) and it has been identified in the nucleus, cytoplasm and on the cell-surface. It's primary role is in protein degradation, where it acts as a tag for intracellular proteolysis.

First time

Begin by typing in the tbss terminal:

```
tbss> vmd-tutorial
```

This will run a script that will create all the necessary files to use this tutorial. It will also put you in the correct directory. You are ready to start.

Continue using the tutorial

If you logged out, and want to continue the tutorial, type in the tbss terminal:

```
tbss> cd vmd-tutorial-files
```

This will place you in the directory containing all the necessary files.

1 Basics of VMD

In this unit you will build a nice image of ubiquitin while becoming accustomed to basic VMD commands. In addition, you will learn how to look for interesting structural properties of proteins using VMD.

1.1 Loading a Molecule

Our first step is to load our molecule. A pdb file, `1UBQ.pdb`, that contains the atom coordinates of ubiquitin is provided with the tutorial.

1 Choose the File → New Molecule... menu item Fig 1(a) in the VMD Main window. Another window, the Molecule File Browser (b), will appear in your screen.

2 Use the Browse... (c) button to find the file `1UBQ.pdb` in the appropriate directory. Note that when you select the file, you will be back in the Molecule File Browser window. In order to actually load the file you have to press Load (d). Do not forget to do this!

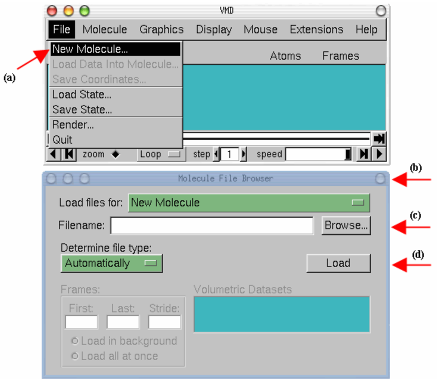
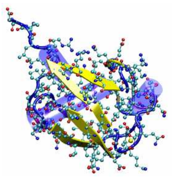


Figure 1: Loading a Molecule.

Now, ubiquitin is shown in your screen in the OpenGL Display window. You may close the Molecule File Browser window at any time.

Webpdb. VMD can download a pdb file from the Protein Data Bank if a network connection is available. Just type the four letter code of the protein in the File Name text entry of the Molecule File Browser window and press the Load button. VMD will download it automatically.



Coordinates file. The file `1UBQ.pdb` corresponds to the X-ray structure of ubiquitin refined at 1.8 Å resolution provided by Senadhi Vijay-Kumar, Charles E. Bugg and William J. Cook, J. Mol. Biol. (1987) **194**, 531. Note that the protein is sorrounded by 58 water molecules, and that hydrogen atoms are not included.

1.2 Displaying the Protein

In order to see the 3D structure of our protein we will use the mouse and its multiple modes.

1 While holding the left button pressed over the protein in the OpenGL Display, move the mouse and explore what happens. This is the rotation mode of the mouse and allows you to rotate the molecule around an axis parallel to the screen Fig. 2(a).

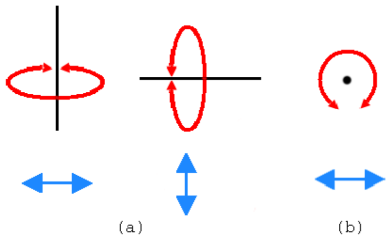


Figure 2: Rotation modes.

2 If you press the second button and repeat the previous step, the rotation will be done around an axis perpendicular to your screen (b) (For Mac users, the second button is equivalent to press the command key while holding the mouse button pressed).

3 In the VMD Main window, look at the Mouse menu (Fig 3). Here, you will be able to switch the mouse mode from Rotation to Translation or Scale modes.

4 The Translation mode will allow you to move the molecule around the screen while holding the first button pressed.

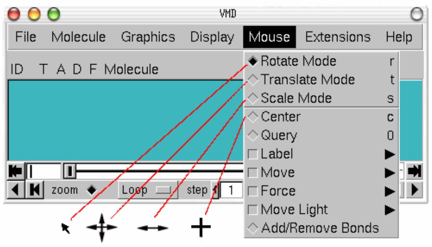


Figure 3: Mouse modes.

5 The Scale mode will allow you to zoom in or out by moving the mouse horizontally while holding the first button pressed.

It should be noted that the previous actions performed with the mouse do not change the actual coordinates of the molecule atoms.

Mouse modes. Note that each mouse mode has its own characteristic cursor and its own shortcut key (`r`: Rotate, `t`: Translate, `s`: Scale) that could be used instead of the Mouse menu. (Be sure to have the OpengGL Display window active when using the shortcuts.) Additional information can be found in the VMD user's guide.

Another useful option is the Mouse → Center menu item. It allows you to specify the point around which rotations are done.

6 Select the Center menu item and pick one atom at one of the ends of the protein. (The cursor should display a cross.)

7 Now, press `r`, rotate the molecule with the mouse and see how your molecule moves around the point you have selected.

1.3 Exploring Different Drawing Styles

VMD can display your molecule using a wide variety of drawing styles. Here, we will explore those that can help you to identify different structures in the protein.

1 Choose the Graphics → Representations... menu item. A window called Graphical Representations will appear and you will see in yellow Fig 4(a) the current graphical representation used to display your molecule.

2 In the Draw Style tab (b) we can change the style (d) and color (c) of the representation. In this section we will focus in the drawing style (the default is Lines).

3 Each drawing style has its own parameters. For instance, change the Thickness of the lines by using the controls on the right bottom part (e) of the Graphical Representation window.

4 Now, choose from Drawing Method the VDW (van der Waals) menu item. Each atom is now represented by a sphere. In this way you can see more easily the volumetric distribution of the protein.

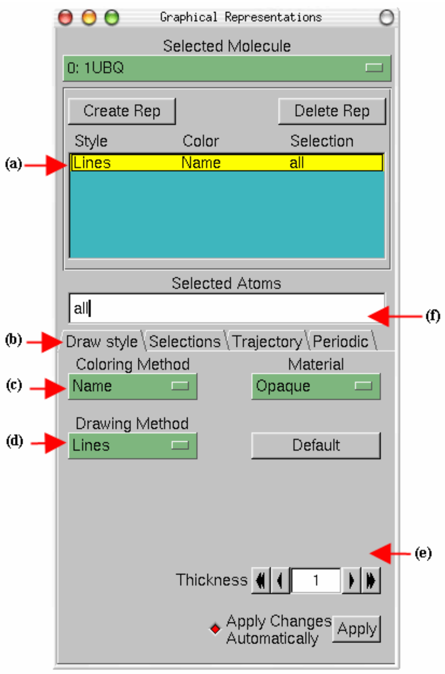


Figure 4: Graphical Representations window.

5 In order to see the arrangements of atoms in the interior of the protein, use the new controls on the right bottom part of the window (e) to change the

Sphere Radius to 0.5 and the Sphere Resolution to 13. Be aware that the higher the resolution you choose, the slower the display of your molecule will be.

6 Note that in the Name coloring method, each atom has its own color, i.e: O is red, N is blue, C is cyan and S is yellow.

7 Press the Default button. This allows you to return to the default properties of the drawing method.

More representations. Other interesting representations are CPK and Licorice. In the first one, like in old chemistry ball & stick kits, each atom is represented by a sphere and each bond is represented by a cylinder. (Radius and resolution of both the sphere and the cylinder can be modified independently.) The Licorice drawing method (widely used) also represents each atom as a sphere and each bond as a cylinder, but the sphere radius cannot be modified independently.

The previous representations allows you to see the micromolecular details of your protein. However, more general structural properties can be seen by using more abstract drawing methods.

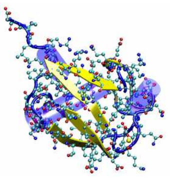
8 Choose the Tube style under Drawing Method and observe the backbone of your protein. Set the Radius at 0.8.

9 By looking at your protein in the tube mode, can you distinguish how many helices, β sheets and coils are present in the protein?

The last drawing method we will explore here is called Cartoon. It gives a simplified representation of a protein based in its secondary structure. Helices are drawn as cylinders, β sheets as solid ribbons and all other structures as a tube. This is probably the most popular drawing method to view the overall architecture of a protein.

10 Choose the Cartoon style and set the Beta Sheet Thickness as 3, the Helix/Coil Radius as 1.5.

11 Identify now how many helices, betasheets and coils are present in the protein.



Structure of ubiquitin. Ubiquitin has three and one half turns of α-helix (residues 23 to 34, three of them hydrophobic), one short piece of 3_{10}-helix (residues 56 to 59) and a mixed β sheet with five strands (residues 1 to 7, 10 to 17, 40 to 45, 48 to 50, and 64 to 72) and seven reverse turns. VMD calculates the secondary structure using STRIDE, which uses an heuristic algorithm that in this case shows only four of the five β strands.

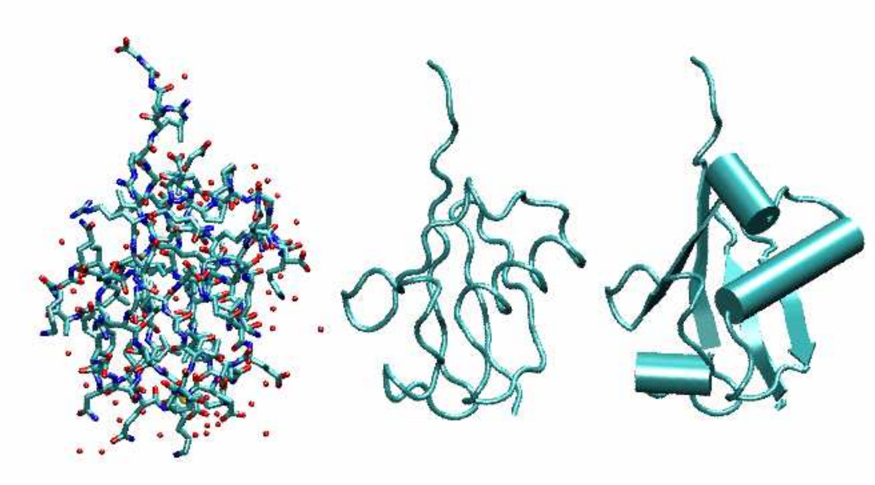


Figure 5: Licorice, Tube and Cartoon representations of Ubiquitin

1.4 Exploring Different Coloring Methods

1 Now, let's modify the colors of our representation. Choose the ResType coloring method Fig. 4(c). This allows you to distinguish non-polar residues (white), basic residues (blue), acidic residues (red) and polar residues (green).

2 Select now the Structure coloring method (c) and confirm that the cartoon representation displays colors consistent with secondary structure.

1.5 Exploring Different Selections

Let's look at different independent (and interesting) parts of our molecule.

1 In the Selected Atoms text entry Fig. 4(f) of the Graphical Representations window delete the word `all`, type `helix` and press the Apply button or hit the Enter key. (Do this every time you type something.) VMD will show just the helices present in our molecule.

2 In the Graphical Representations window choose the Selections tab Fig. 6(a). In section Singlewords (b) you will find a list of possible selections you can type. For instance, try to display β sheets instead of helices by typing the appropriate word in the Selected Atoms text entry.

Combinations of boolean operators can also be used when writing a selection.

3 In order to see all that is not helix and not β sheet, type the following `(not helix)and(not betasheet)`

4 In the section Keyword (c) of the Selections tab (a) you can see properties that can be used to select parts of a protein with their possible values. Look at possible values of the Keyword resname (d). Display all the Lysines and Glycines presents in the protein by typing `(resname LYS)or(resname GLY)`. Lysines play a fundamental role in the configuration of polyubiquitin chains.

5 Now, change the current representation to CPK style and the coloring method to ResID by using the previous described buttons in the Draw Style tab. In the screen you will be able to see the different Lysines and Glysines. How many of each one can you see?

6 In the Selected Atoms text entry type `water`. Choose the coloring method Name. You should see the 58 water molecules (in fact only the oxygens) present in our system.

7 In order to see wich water molecules are closer to the protein you can use the command `within`. Type `water and within 3 of protein`. This selects all the water molecules that are within a distance of 3 angstroms of the protein.

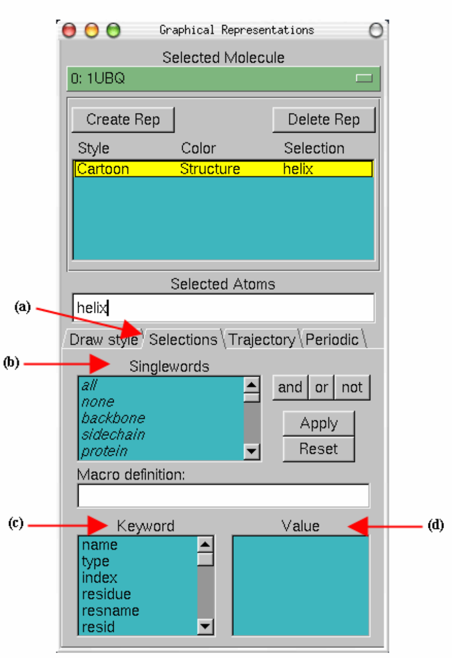


Figure 6: Graphical Representations window and the Selections tab.

8 Finally, try the following selections:

Selection	Action
`protein`	Shows the Protein
`resid 1`	The first residues
`(resid 1 76)and(not water)`	The first and last residues
`(resid 23 to 34)and(protein)`	The α helix

All the previous options provide you with a powerful tool to explore different parts of your protein or molecule.

1.6 Multiple Representations

The button Create Rep Fig 7(a) in the Graphical Representations window allows you to create multiple representations and therefore have a mixture of different selections with different styles and colors, all displayed at the same time.

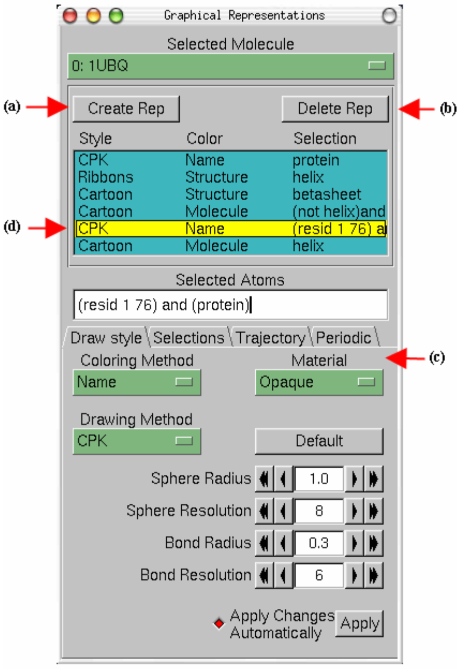


Figure 7: Multiple Representations of Ubiquitin.

1 Be sure that the current representation is in CPK style and coloring method Name

2 Set the current selection as `protein`.

3 Press the Create Rep button (a). Now, using the menu items of the Draw Style tab and the Selected Atoms text entry, modify the new representation in order to get Ribbons as the drawing method, Structure as the coloring method, and `helix` as the current selection.

4 Repeating the previous procedure, create the following three new representations:

Drawing Style	Coloring Method	Selection
Cartoon	Structure	`betasheet`
Cartoon	Molecule	`(not helix)and(not betasheet)`
CPK	Name	`(resid 1 76) and (protein)`

5 Create a final representation by pressing again the Create Rep button. Select the Cartoon drawing method, the Molecule coloring method and type `helix` in the Selected Atoms entry. For this last representation choose in the Material section (c) the Transparent menu item.

6 Note that with the mouse you can select the different representations you have created and modify each one independently. Also, you can switch each one on/off by double clicking on it or delete each one by using the Delete Rep button (b). At the end of this section, the Graphical Representations window should look like Fig. 7.

1.7 Sequence Extension

When dealing with a protein for the first time, it is very useful to find and display different amino acids quickly. The sequence extension allows you to pick and display one or more residues easily.

1 Choose the Extensions → Sequence menu item. A window Fig. 8(a) with a list of the amino acids (e) and their properties (b)&(c) will appear in your screen.

2 With the mouse, click over different residues (e) in the list and see how they are highlighted. In addition, the highlighted residue will appear in your OpenGL Display window in yellow and bond drawing style, so you can visualize it easily.

3 Using the Zoom controls (f) you can display the entire list of residues in the window. This is especially useful for larger proteins

4 Using the shift key while pressing the mouse button allows you to pick multiple residues at the same time. Look at residues 48, 63, 11 and 29 (e).

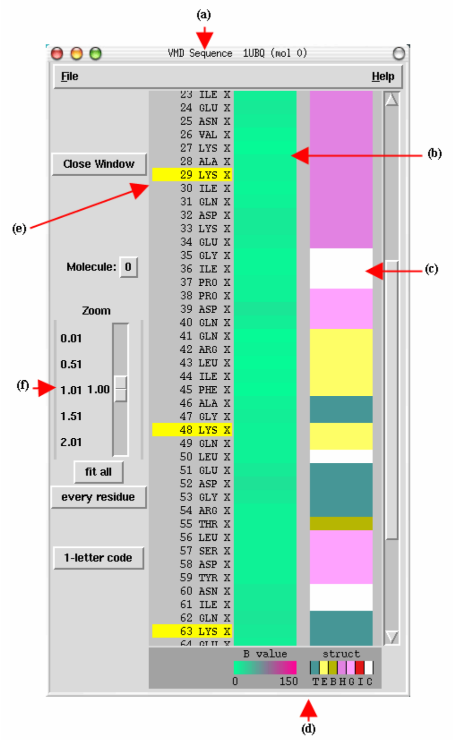
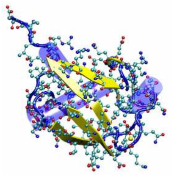


Figure 8: Sequence window.

5 Look at the Graphical Representations window, you should find a new representation with the residues you have selected using the Sequence

Extension. As you already did before, you can modify, hide or delete this representation.



Relevance of Lysines. Polyubiquitin chains can be linked by a peptide bond between C- and N-termini or through linkages involving lysines 48, 63, 11 or 29 (the ones that you selected using the sequence window). Different linked chains have different properties that are related to the functionality of the chain.

Information about residues is color-coded (d) in columns and obtained from STRIDE. The B-value column (b) shows the B-value field (temperature factor). The struct column shows secondary structure, where each letter means (d):

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

Table 1: Secondary Structure.

1.8 Saving your Work

The image that you have created using VMD can be saved, along with all representations you have created, as a VMD state. This VMD state contains all the information needed to start a new VMD session from it, without losing what you have done.

1 Choose the File → Save State menu item. Write an appropriate name (`myfirststate.vmd`) and save it.

The File → Load State menu item will allow you to load a previous saved VMD state, like the file you just saved. Although the VMD state allows you to work with the image and explore the properties of our protein using VMD, you usually need pictures that can be used in articles or other kind of documents. VMD can render the image you created and generate an image file that can be used in other applications, as it is shown in the following steps.

2 Using all that you have learned until now, find an appropriate view of the protein by scaling, rotating and translating the molecule. Turn different representations on and off and improve the resolution and different properties of the selections you have made. If you want an image of good quality, put special attention to the resolution of each representation.

3 Be aware of the new representations you created with the Sequence Extension and hide or delete them if it is neccesary.

4 Before rendering the image, change the background color by choosing the Graphics → Colors menu item. There, Choose the Display category, the Background name and the 8 white color. The background should be white now.

5 Choose the File → Render... menu item. A window called File Render Controls will appear in your screen.

6 You can render the image using different packages. Pick Tachyon in the Render using menu.

7 Write the name of the file where the image will be saved in the Filename text entry (i.e `picture`).

8 Press the Start Rendering button and the file with your image will be created. Note that this could take some time. You should end up with an image file named `picture.tga` (MacOS X or Unix) or `picture.bmp` (on Windows).

Now you are done with unit 1. We hope you have learned the basic commands of VMD. Also, you have generated two files. The first one is a VMD state that allows you to restart a VMD session and use/modify all that you did in this unit. The second file is an image file of your protein that can be used in some other application.

2 Multiple Molecules and Scripting

In this unit, you will learn to deal with multiple molecules simultaneously. You will also cover the basics of Tcl scripting and use it to edit atomic data, align two molecules together and color a molecule according to calculated properties.

1 Start with a new VMD session. If you have just completed unit 1, you should quit VMD and then launch it again.

2.1 Loading Multiple Molecules

First, you will load the molecules that you will need.

1 Open the Molecule File Browser from the File → New Molecule... menu item.

2 Load the X-ray crystal structure file `1UBQ.pdb` (just as you did in Unit 1).

An equilibration simulation for ubiquitin solvated in a water box has been performed for a duration of 1 ns. You will now compare the conformation of ubiquitin at the end of this simulation with the initial crystal structure.

Coordinate files and structure files. To save space, simulation output files usually contain only the atom coordinates and do not store unchanging information such as the atom types, atom charge, segment names and bonds. The latter information is stored in a separate "structure" file (*e.g.*, a PSF file). To visualize the results of a simulation, you need to merge both a structure and a coordinate file into the same molecule.

3 Now, start loading the simulation results into a second molecule. Staying in the Molecule File Browser, select New Molecule from the menu at the top of the form. Browse for the file `ubiquitin.psf`, then click on the Load button. You have just created a new molecule with a structure but no coordinates.

4 Notice that the menu at the top of the window now says 1: ubiquitin.psf. This ensures that the next file that you load will be added to that molecule (molecule ID 1). Now, browse for the file `ubiquitin-equilibrated.coor` and click on Load again. This will load the coordinates and merge them with the previously loaded structure information.

You should now see two superimposed and unaligned molecules. One is the original crystal ubiquitin. The other one is surrounded by a cube full of water molecules.

2.2 Using the Main Window

You will now give names to your molecules so that you can identify them later.

1 Double-click on the first molecule in the Main window molecule listing. The Rename Molecule dialog box should pop up. Type in `crystal`. Do the same thing for the second molecule, and call it `simulation`.

At this point, your Main window should look like Fig. 9. In front of the molecule name, there are four letters that you can use to manipulate your molecules.

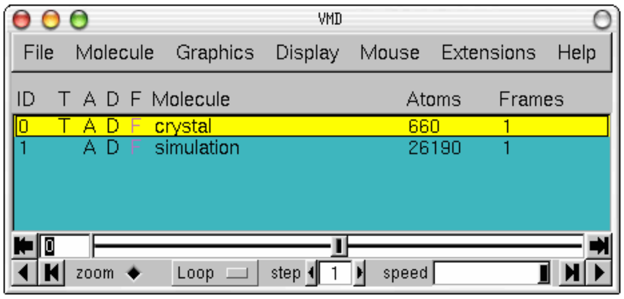


Figure 9: Main window with initial and final coordinates loaded into two separate molecules.

F stands for "Fixed," meaning that that molecule won't move when you move the scene around. When the F is black, the molecule is fixed; when it is grayed out, the molecule is mobile.

2 Double-click on the F to the left of a molecule description in the Main form. Then, try translating the scene with the mouse while one molecule is fixed. Do it again with both molecules fixed.

3 When you are done, unfix both molecules and select the Display → Reset View menu item to correctly reposition the two molecules relative to each other.

4 Next, double-click on the D of one molecule. The D stands for "Displayed" and when the D is grayed out it means that the molecule is hidden. You can control the visibility of all your molecules by double-clicking here.

5 When you are done, make sure that only the `crystal` molecule is "Displayed" and that none of the molecules are "Fixed."

6 Finally, double-click under the T column to the left of the `crystal` molecule. The letter T should now appear in front of it. This makes it the unique "top" molecule. Making a molecule top makes it a target for scripting commands.

2.3 Tcl Scripting Basics and TkCon

VMD includes support for the Tcl/Tk scripting language. This section will attempt to provide the minimal amount of scripting that you need to know to use very useful features.

> **The Tcl/Tk scripting language.** Tcl is a rich language that contains many other features and commands than those seen above, in addition to the typical conditional and looping expressions. Tk is an extension to Tcl that permits the writing of user interfaces with windows and buttons, *etc.* More information and docs about the Tcl/Tk language can be found at http://www.tcl.tk/doc.

To execute Tcl commands, you will be using a convenient text console called TkCon.

1 Select the Extensions → tkcon menu item. A console window should appear with a prompt (Fig. 10). You can now start entering Tcl/Tk commands in it.

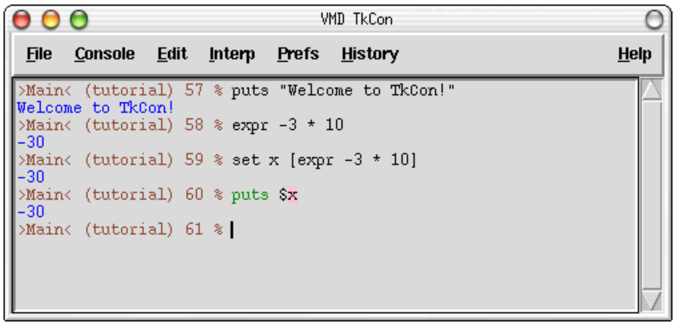


Figure 10: The TkCon console.

You will initially focus on the very basic of Tcl/Tk. Here are Tcl's `set` and `get` commands:

`set` *variable value*	– sets the value of *variable*
`puts $`*variable*	– prints out the value of *variable*

2 Try the following commands:

```
set x 10
puts "the value of x is:  $x"
set text "some text"
puts "the value of text is:  $text."
```

As you can see, `$`*variable* refers to the value of *variable*.

Here is a command that performs mathematical operations:

`expr` *expression* – evaluates a mathematical expression

3 Try experimenting with the `expr` command:

```
expr 3 - 8
set x 10
expr - 3 * $x
```

One of the most important aspects of Tcl is that you can embed Tcl commands into others by using brackets. A bracketed expression will automatically be substituted by the return value of the expression inside the brackets:

`[`*expr.*`]` – represents the result of the expression inside the brackets

4 Create some commands using brackets and test them. Here is an example:

```
set result [ expr -3 * $x ]
puts $result
```

2.4 The `atomselect` Command

You can edit atomic properties using VMD's `atomselect` command. The following examples will show you how.

1 After making sure that the `crystal` molecule is the top molecule (if not, double-click on T). Open the Representations window using the Graphics → Representations... menu item.

> **The PDB B-factor field.** The "B" field of a PDB file typically stores the "temperature factor" for a crystal structure and is read into VMD's "Beta" field. Since we are not currently interested in this information, we can recycle this field to store our own numerical values. VMD has a "Beta" coloring mode, which you will soon use, and which colors atoms according to their B-factors. Thus, by replacing the Beta values for various atoms, you can control the color in which they are drawn. This is very useful when you want to show a property of the system that you have computed and which is not supported out-of-the-box by VMD.

2 Select the molecule called "crystal" in the pop-up menu at the top of the form. Type in `protein` as the atom selection, change its coloring method to Beta and its drawing method to VdW. Your molecule should now appear as a mostly red and green assembly of spheres.

You will now learn a very important Tcl command in VMD:

`atomselect` *molid selection* – creates a new atom selection

The first argument to `atomselect` is the molecule ID (shown to the very left of the Main form), the second argument is a textual atom selection like what you have been using to describe graphical reps in unit 1. The selection returned by `atomselect` is itself a command which you will learn to use.

3 Type `set crystal [atomselect top "all"]` in the TkCon window. This creates a selection containing all the atoms in the molecule and assigns it to the variable `all`. Instead of a molecule ID (which is a number), we have used the shortcut "`top`" to refer to the top molecule.

The result of `atomselect` is a function. Thus, `$crystal` is now a function that performs actions on the contents of the "all" selection.

4 Type `$crystal num`. Passing `num` to an atom selection returns the number of atoms in that selection. Check that this number matches the number of atoms for that molecule (as read from the Main form).

5 Type `$crystal set beta 0`. This resets the "beta" field (which is being displayed) to be zero for all atoms. As you do this, you should observe that the atoms on your screen will suddenly change to a uniform color (since they all have the same beta values).

Atom selections are just references to the atoms in the original molecule. When you change a property (*e.g.* beta value) of some atoms through a selection, that change is reflected in all the other selections that contain those atoms.

6 Now, type `set sel [atomselect top "hydrophobic"]`. This creates a selection containing all the hydrophobic residues.

7 Let's label all hydrophobic atoms by setting their beta values to 1. You should know how to do this now: `$sel set beta 1`. If the colors in the OpenGL Display do not get updated, click on the Apply button at the bottom of the Representations form.

Examples of atomic properties. You can get and set many atomic properties using atom selections, including segment, residue and atom names, position (x, y and z), charge, mass, occupancy and radius, just to name a few.

8 You will now change a physical property of the atoms to further illustrate the distribution of hydrophobic residues. Type `$crystal set radius 1.0` to make all the atoms smaller and easier to see through, and then `$sel set radius 1.5` to make the hydrophobic residues larger. The radius field affects the way that some representations (*e.g.,* VdW, CPK) are drawn.

You have now created a visual state that clearly distinguishes which parts of the protein are hydrophobic and which are hydrophilic. If you have followed the instructions correctly, your protein should resemble Fig. 11.

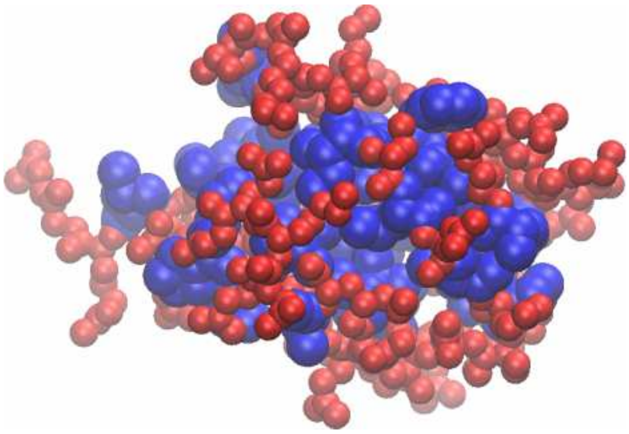
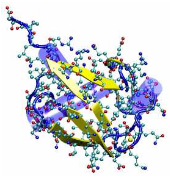


Figure 11: Ubiquitin in the VdW representation, colored according to the hydrophobicity of its residues.



Hydrophobic residues. As you probably noticed in your rendering of ubiquitin, the hydrophobic residues are almost exclusively contained in the inner core of the protein. This is typical for small water-soluble proteins. As the protein folds, the hydrophylic residues will have a tendency to stay at the water interface, while the hydrophobic residues are pushed together and play a structural role. This helps the protein achieve proper folding and increases it's stability.

Atom selections are useful not only for setting atomic data, but also for getting information. Let's say that you wish to communicate which residues are hydrophobic, all you need to do is to create a hydrophobic selection and use `get` instead of `set`.

9 Use `get` with your hydrophobic `sel` atom selection:

```
$sel get resname
```

But there is a problem! Each residue contain many atoms, resulting in multiple repeated entries. Can you think of a way to circumvent this? We know that each amino-acid residue have the same backbone atoms. If you pick only one of these atoms per residue, each residue will be present only once in your selection.

10 Let's try this solution. Each residue has one and only one α-carbon (`name CA` = `alpha`):

```
set sel [atomselect top "hydrophobic and alpha"]
$sel get resname
```

Aha! It worked.

11 You can also get multiple properties simultaneously. Try the following:

```
$sel get resid
$sel get {resname resid}
$sel get {x y z}
```

2.5 Aligning Two Molecules

To properly visualize how ubiquitin has evolved during the simulation, you must first make sure to spatially superimpose your two molecules as best as you can. VMD offers some convenient commands to do this. You will now color the two molecules differently to visually distinguish them.

1 Make sure that both molecules are "Displayed" (by double-clicking on D in the Main form).

2 In the Representations window, change the rep for the `crystal` molecule (remember to select it from the menu at the top of the window). Set its coloring method to ColorID, its color to 0 blue and its drawing method to Tube.

3 You can apply changes to different molecules in the Graphical Representations window by selecting the target molecule from the Selected Molecule menu at the very top of the window. Choose the same representation for the molecule called `simulation` (ColorID and Tube), but set its color to 1 red instead.

4 Create two atom selections containing the protein backbone α-carbons (using the `alpha` atom selection macro) for each molecule by typing in the TkCon window:

```
set alpha1 [atomselect 0 "alpha"]
set alpha2 [atomselect 1 "alpha"]
```

Here we have selected the α-carbons because they are more stable than the protein's floppy side-chains and give a good indication of the protein's spatial conformation.

Even though the two molecules are very different (the crystal structure has no hydrogens), the selections that you created both contain the exact same atoms. VMD provides a command – `measure fit` – for finding the best fit between two selections corresponding to the same atoms:

`measure fit` *atomsel1 atomsel2* – calculates the best fit matrix

5 Find the transformation matrix `M` that will best map the first selection onto the second by typing:

```
set M [measure fit $alpha1 $alpha2]
```

6 Apply the matrix you just found to the entire initial molecule (you had previously defined an atom selection called `crystal` for this), by typing:

```
$crystal move $M
```

In the OpenGL window, the two molecules should now be aligned based on the positions of their α-carbons. It is to be noticed that some parts of the molecule are well-aligned, whereas other dangling ends and loops seem more floppy (they have moved more during the equilibration). You can realign the molecules by repeating the alignment process using the alpha carbons for the the rigid residues only. Try it! The final result should resemble Fig. 12.

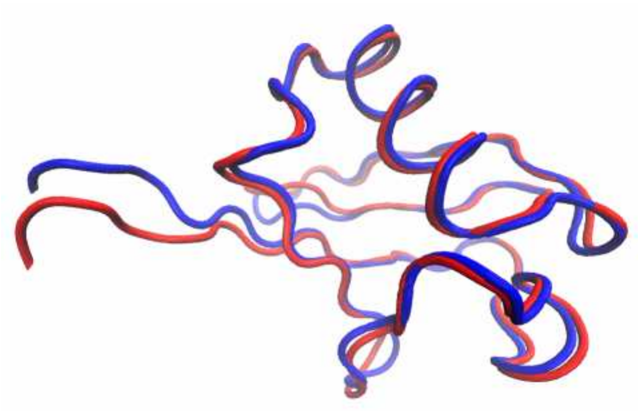


Figure 12: Initial and final states of ubiquitin after spatial alignment.

2.6 Using Color to Display the Deviation

Once the two molecules are aligned, you can compare their atoms' positions. To do this, you will use a pre-written script (since it is slightly complex). Feel free to have a look at the script if you feel adventurous; as you become experienced, writing such scripts will become trivial. To run a script from a file, you can use Tcl's `source` command:

`source` *file* – runs a script from a text file

1 The script is in the `coloring.tcl` file. It will change the `crystal` molecule's beta values to reflect the displacement (in Å) of each atom of the molecule after equilibration. Let's run the script by typing:

```
source coloring.tcl
```

2 To see the new beta values, you need to set the Coloring Style of the `crystal` molecule to Beta, and hide the `simulation` molecule (by double-clicking on D).

The `crystal` molecule should now be colored according to the total displacement that occurred during the simulation. To have a better idea of what went on, we would like to adjust the color scale in a useful way.

3 In the Graphical Representations window, choose the Trajectory tab. Under the Color Scale Range label, you can set the min and max values used for the beta values color scale. Make sure that the right representation (of the `crystal` molecule) is selected, then enter 0 and 5 and click on Set. This sets the color range to scale between 0 and 5 Å.

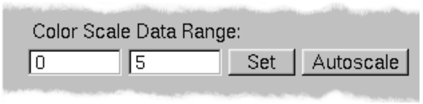


Figure 13: The Color Range controls in the Representations window.

4 Now, bring up the Color Controls window (Fig. 14) by choosing the Graphics → Colors... menu item from the VMD Main window. Then select the Color Scale tab.

5 In the color scale Method menu, select the BWR color scheme. This will show residues with a low displacement (0 Å) in blue, those with a high displacement (5 Å) in red, and those in between as white.

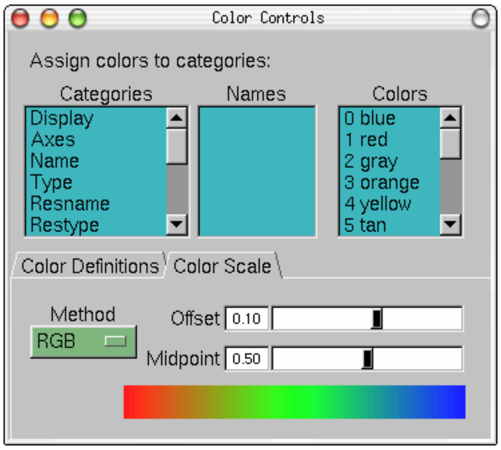


Figure 14: The Color Controls window showing the Color Scale tab.

6 You can now also adjust the color scale Midpoint to shift the level of atomic displacement that will be assigned to white. Try assigning a midpoint of `0.1`.

You have now seen how to manipulate continuous color scales on custom data. Look at your molecule (it should resemble Fig. 15). You should now be able to determine which parts of ubiquitin are stable (blue) and which are floppy (red). In general, the dangling ends and loops are flexible, whereas the helices and β sheets are much more rigid.

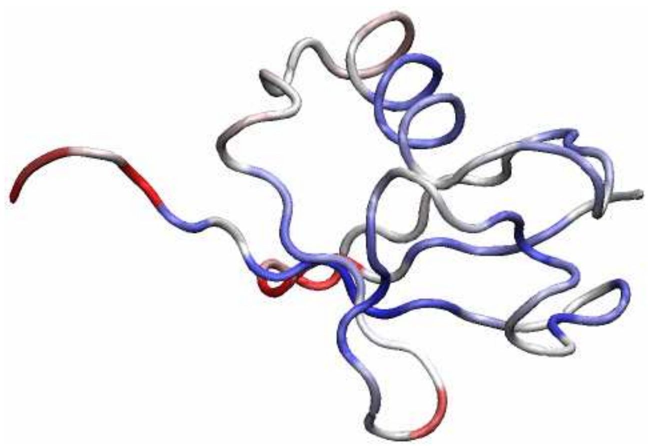


Figure 15: Ubiquitin colored according to displacement after a 1 ns equilibration.

3 Trajectories, Macros and Labels

In this unit, you will learn how to load trajectories, create macros, place labels on atoms and bonds, and calculate the RMSD of a trajectory using a simple tcl scripts. At the end, you should determine if the ubiquitin system is equilibrated by looking at RMSD plots.

3.1 Loading trajectories

You will now learn to load the time evolving coordinates of a system, called *trajectories*. You will be able to see a movie of your system.

Trajectory files are normally binary files that contain several sets of coordinates for the system. Each set of coordinates corresponds to one *frame* in time. An example of trajectory files are DCD files. The trajectory files do not contain the information of the system contained in the protein structure (PSF) file, so we need to load first the parameter file, and then add the trajectory data to this file, as explained in Unit 2.

1 Start a new VMD session.

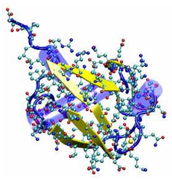
2 Load the PSF file of the system `ubiquitin.psf`, as done in Unit 2.

3 In the Molecule File Browser window, click on the Browse button, making sure that `ubiquitin.psf` is selected on the menu. Browse for `pulling.dcd`, click OK and click on the Load button again. You will be able to see the frames as they are loaded into the molecule.

4 After the trajectory finishes loading, you will be will be looking at the last frame of your trajectory. To go to the beginning of the trajectory, you will use one of the Animation Tools, that will be explained extensively later in the tutorial. In the Main menu, click on the  button in the lower left.

5 Choose Graphics → Representations menu item. In the Drawing Method pulldown menu, select Tube, and in the Selected Atoms window, type `protein`.

6 In the same menu item, create another representation by clicking on the Create Rep button. In the Drawing Method pulldown menu, pick Lines, and in the Selected Atoms window, type `water`. For now, turn off this representation by double-clicking on it.



Elastic properties of ubiquitin. Ubiquitin has many functions in the cell. It is currently believed that some of these functions depend on the elastic properties of ubiquitin.

Only certain proteins are known to show elastic properties. Examples of these molecules are titin, which acts like a “spring” in the muscle. The elastic properties of these molecules are due to hydrogen bonding between residues in β strands of this molecules, like the ones in ubiquitin.

The trajectory you just loaded is a simulation of an AFM (Atomic Force Microscopy) experiment pulling on a single ubiquitin molecule. We will look at the behavior of the protein as it unfolds while being pulled form one end.

In Molecular Dynamics, before starting a simulation like the one shown for the AFM experiment, one first needs to perform the equilibration of the protein. You will have a chance to look at such an equilibration trajectory later in this tutorial.

In the next section, you will learn how to create representations with a useful feature called *macros*. Once you create some representations relevant to this trajectory, the following section will teach you how to use the Animation Tools to look at trajectories.

3.2 Macros

You will now create similar representations to the ones you learned in Unit 1. While creating these representations, you will learn macros. A macro is text that represents a selection. It is useful to create macros when you use certain selections often. Macros are created with the `atomselect` command you learned in unit 2.

`atomselect macro` *name selection* – creates a macro for *selection*

Ubiquitin has five β sheets, and they play a major role in the unfolding of the protein. To create a macro for these sheets:

1 Open the tkcon window by choosing the Main → Extensions → tkcon.

2 In the tkcon window, type:

```
atomselect macro bsheet1 {protein and resid 2 to 6 }
```

This will create a macro for the first β sheet, which includes residues 2 to 6.

For the other sheets, you will find out which residues belong to them by using the sequence viewer introduced in Unit 1, and then create similar macros:

3 Choose the Extensions → sequence menu item.

As you learned in Unit 1, the second color column corresponds to structural features of the protein. The sections in yellow correspond to β sheets.

4 With the mouse, click and drag to highlight the second β sheet. This action will create a representation in the Graphics → Representations window.

5 In the Graphics → Representations window, click on the new representation. The text corresponding to the selection is displayed on the Selected Atoms window. You should have `( chain U and resid 12 13 14 15 16 )`.

6 Use this text to create a new macro by typing in the tkcon:

```
atomselect macro bsheet2 { chain U and resid 12 13 14 15 16 }
```

This will create a macro called `bsheet2` that will contain residues 12 to 16 of `chain U`, which corresponds to the protein.

7 Create similar macros for the other three β sheets for later use using the sequence extension.

Once a macro is created, you can refer to it both in the tkcon, and in the Representations selections.

Macros you have created and other macros that come with VMD can be seen in the Selections tab of the Graphical Representations window. The macros are listed in the Singlewords window. Clicking on a macro will show its definition in the Macro Definition window. Double-clicking on it will select it and put its definition on the Selected Atoms form.

You will now create a representation with the first and fifth β sheets:

8 In the Graphics → Representations menu item, click on the Create Rep button.

9 In the Selected Atoms window, erase the text that appears there.

10 Click in the Selections tab. Browse in the Singlewords form until you find your newly created macros.

11 Double-click on `bsheet1`, click the button or and then double-click on `bsheet5` (Fig. 16). Then, click on the Apply button.

12 In the Draw Style tab, choose a Cartoon representation for this selection and color it yellow. You should now see the β sheets.

13 Now, create a similar representation with the other three β sheets. Do this by clicking on the Create Rep button. Now, in the Selected Atoms form, type:
`bsheet2 or bsheet3 or bsheet4`
Typing them directly works too!

14 In the Graphical Representations window, choose the representation for `protein`. In the Atom Selections window, substitute `protein` for `protein and not betasheet`.

As you can see, macros can be very useful. When saving your work in a saved state, macros are included in the saved state file.

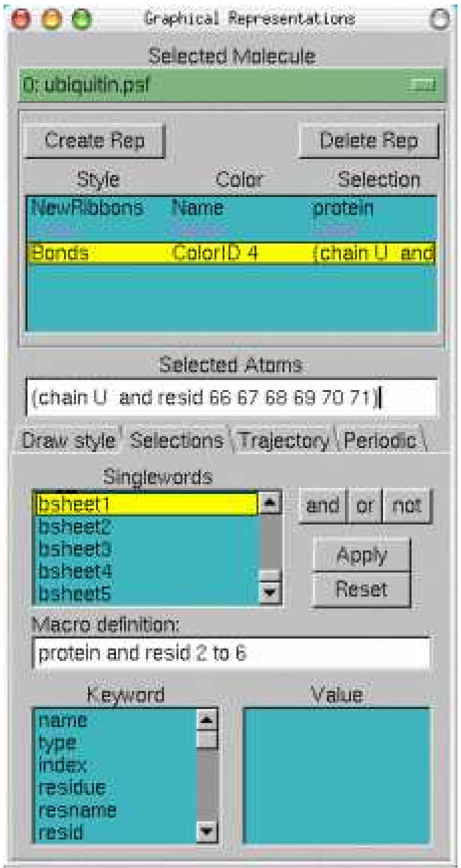
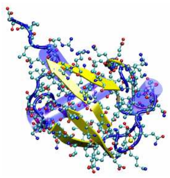


Figure 16: Macros are listed in the Selections tab of the Graphical Representations Menu.

Macros. You can create several macros and store them in a VMD preferences file `.vmdrc` in your home directory (Windows uses the file `vmd.rc`). VMD will look for this file upon startup and will recognize all your macros. For more information of the VMD startup files, refer to the VMD user manual.

To finish this section, you will create a very interesting representation, that shows a key feature of the trajectory we are looking at. This is an H-bonds representation.



Hydrogen Bonds. The H-bonds give stability to proteins in several ways. The H-bonds between β sheet backbone atoms are a trademark of elastic proteins; they give these proteins stability and their formation and rupture under mechanical stress confers them their elastic properties.

15 Create a representation with the selection `betasheet and backbone`, choose the Hbonds drawing method, and color it red using the `Color ID` Drawing Method. In the options, set Distance Cutoff to 3.2, Angle Cutoff to 30 and Line Thickness to 5.

You can now appreciate the most important features of ubiquitin for this unfolding trajectory. Your protein should now look similar to the one in Fig. 17.

16 Save a VMD state of this session, so if you want to come back and keep on working in this tutorial, you don't have to work through those representations again. Do this with the File → Save State... menu item, as done in Unit 1.

Reusing saved states. When you have a saved state for a system, you can use it later for viewing different coordinates (PDB or a different trajectory), if these have the same PSF file. You can do this by following these steps:

- Load your saved state file: File → Load State...
- Select the file you saved before (i.e. `nice-ubiquitin.vmd`)
- Go to Molecule → Delete Frames.... You need to delete all frames currently loaded.The default options will do that. Click on the Delete button.
- Load your new file `yournewtrajectory.dcd` into the PSF in File → Load Data Into Molecule.

3.3 Main Menu Animation Tools

Now that you have nice representations, you will be able to observe features of your trajectory. The Animation Tools help you do that. The Main Menu includes all the Animation Tools you need for navigating through your trajectories. They are located at the bottom of the Menu (Fig. 18).

1 Try using the ⏭ button to jump to the end of the trajectory and go back to the beginning with the ⏮ button. You can see the final and initial states of the trajectory, that correspond to the unfolded and folded states of the protein.

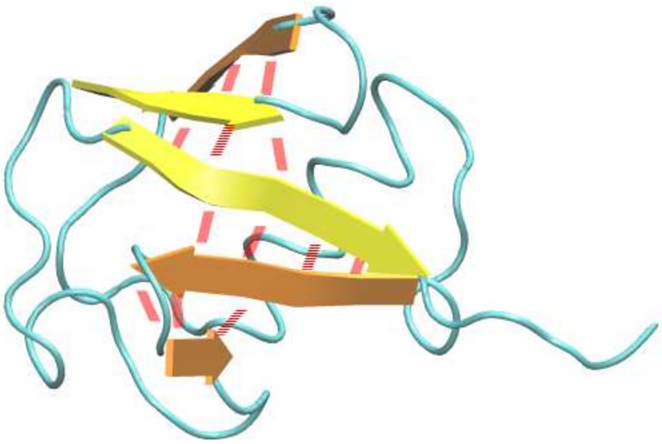


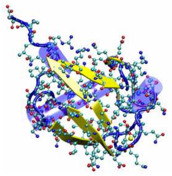
Figure 17: Ubiquitin with key secondary structure features highlighted

Figure 18: Animation Tools

2 Turn on again the water representation. Choose the Main → Graphics → Representations menu item. Double-clicking on the representation with the text `water` will turn it on.

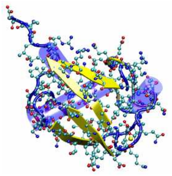
You can click on the slider  and drag it back and forth to navigate through your trajectory. You can stop at anytime you want, or go at the speed you need. This is helpful when you are looking at a trajectory and want to spot the time when something interesting happens.

3 Using the slider, observe the behavior of the water around the protein at the beginning of the trajectory.



Minimizing free energy. Note how the shape of the water box changes form a cube to a sphere. To minimize free energy, the water adopts the configuration that exposes less surface to the vacuum, that is, a sphere! Check how fast the water comes to this configuration (Each frame step corresponds to 10 ps.)

4 Now, remove the Water Representation from the Representations form by double clicking on it to be able to give the protein a closer look. Slide through the trajectory to look at the protein unfold. Do you notice any features?



Breaking H-Bonds. Look at the two yellow β sheets in the middle. They are connected to each other with H-bonds. Note how they are the first β sheets to separate, this means their H-bonds are the first to break. We will talk some more about the relevance of this phenomenon later.

On the lower part of the Animation tools, you will find all the tools necessary to play an animation without using the slider. This is done with the Play buttons, that go forward and backward.

5 Play the trajectory backwards. Do you think this is the way the protein would fold in nature?

There are two ways to change the speed of your animation. You can adjust the speed of the play using the Speed Slider. You can also adjust the step size. This is done using the Step Window. If this step is set to 3, the animation will show every 3rd frame, so it will make it faster.

6 Make the step 5, and play the trajectory. Note that it plays faster, but it also looks less smooth than before. However, this can come handy if you are looking at long trajectories.

NOTE: The Animation Tools you learned cycle through the frames of the Top molecule, but apply to all Active molecules.

Looping styles. When playing animations, you can choose between 3 looping styles in the Style Chooser. These are "Once", "Loop" and "Rock". A nice trick:

- Set the step number to the total number of frames in your trajectory (100).
- Set the style to "Rock".
- Set the speed to lowest.
- Play.

You can see now a comparison between the first and the last frame of your trajectory. This trick is better than clicking on the Start and End buttons.

3.4 Labels

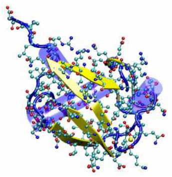
In VMD, you can place labels to get information on a particular selection. We will now make use of those labels for fun and profit. Labels are selected with the mouse. In this example, we will cover labels that can be placed on atoms and bonds, although angle and dihedral labeling are also possible.

1 Choose the Mouse → Labels → Atoms menu item. The mouse is now set to "Display Label for Atom" mode. You can now click on any atom on your molecule and a label will be placed into this atom. Clicking again on it will erase the label.

We will now try the same for bonds.

2 Choose the Mouse → Labels → Bonds menu item. This selects the "Display Label for Bond" mode.

You will make a VDW Representation for the α carbon of Lysine 48 and of the C terminus. In the pulling simulation, the former is kept fixed, and the latter is pulled at constant force of 500 pN.



Residue K48. As you learned in Unit 1, polyubiquitin chains can be linked by a connection between the C terminus of one ubiquitin molecule and residue K48 of the next. The simulation then mimic the effect of pulling on the C terminus with this kind of linkage.

To find out the index of these atoms:

3 Make a selection including these two atoms, by typing in the tkcon:
`set sel [atomselect top "resid 48 76 and name CA"]`

4 Get the indices:
`$sel get index`

This command should give the indices `770 1242`.

> **PDB and VMD atom numbering.** Note that the atom number of these atoms in the pdb file is 771 and 1243. VMD starts counting the `index` from zero, so the text in the Representation should be the numbers that VMD understands. This is only the case for index, since VMD does not read them from the PDB file. Other keywords, such as residue, are consistent with the PDB file.

5 Create a VDW Representation with selection `index 770 1242`.

6 Now that you can see them, click on both atoms (one after the other). You should get a line connecting the two atoms. The number appearing next to the line is the distance between the two atoms in angstroms (Fig. 19).

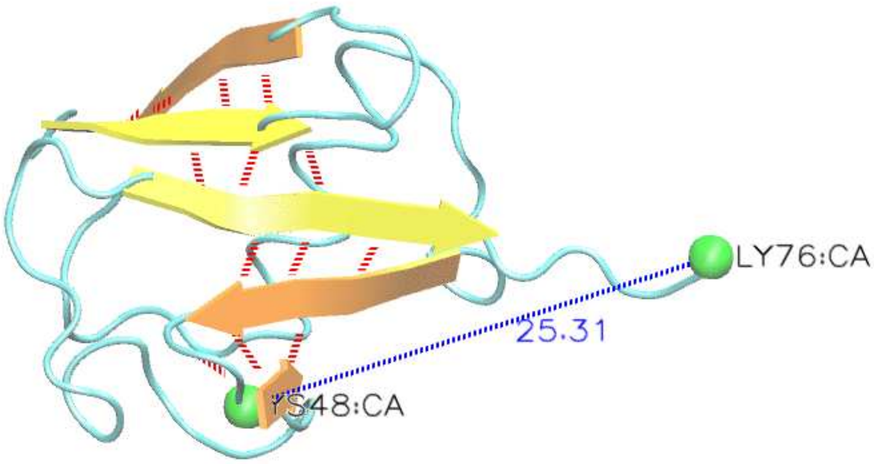


Figure 19: Bond selection of fixed and pulled atom in simulation. Both atoms selected display labels in black. The bond is shown in blue, with the value of the distance between the atoms in angstroms displayed.

The value of the distance corresponds to the current frame.

Labels. The shortcut keys for labels are 1: Atoms and 2: Bonds. You can use these instead of the Mouse menu. Be sure the Open GL display window is active when using these shortcuts.

.

7 There are more things you can do with labels, in the Graphics → Label menu item. In the left side of the window (Fig. 20), there is a pull-down menu where you can choose the type of label (Atoms, Bonds, Angles, Dihedrals). For now, keep it in Atoms. You can see the list of atoms for which you made a label.

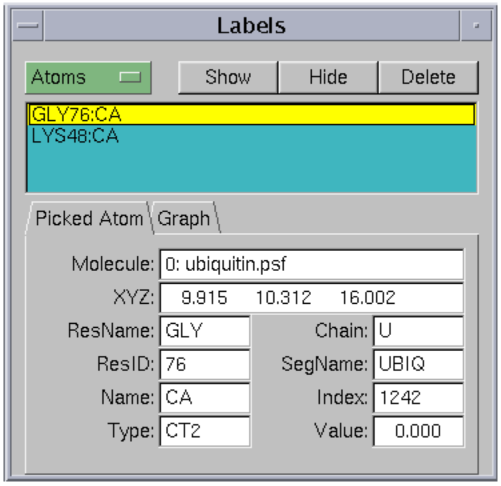


Figure 20: Label window.

8 Click on one of the atoms. You can see all the information of the atom displayed. You can delete, hide, or show the label by clicking on these buttons.

Note that this information is useful to make selections. The information about the atom corresponds to the current frame, and is updated as the frame is changed.

9 Now, in the Label window, choose the label type Bonds. Note that the information given corresponds to only the first atom in the bond, but the number in the Value field corresponds to the length of the bond in angstroms. Click on the Graph tab. Select the bond you labeled between atoms 770 and 1242. Click on the Graph button. This will create the plot of the distance between these two atoms over time. You can also save this data to a file by clicking on the Save button.

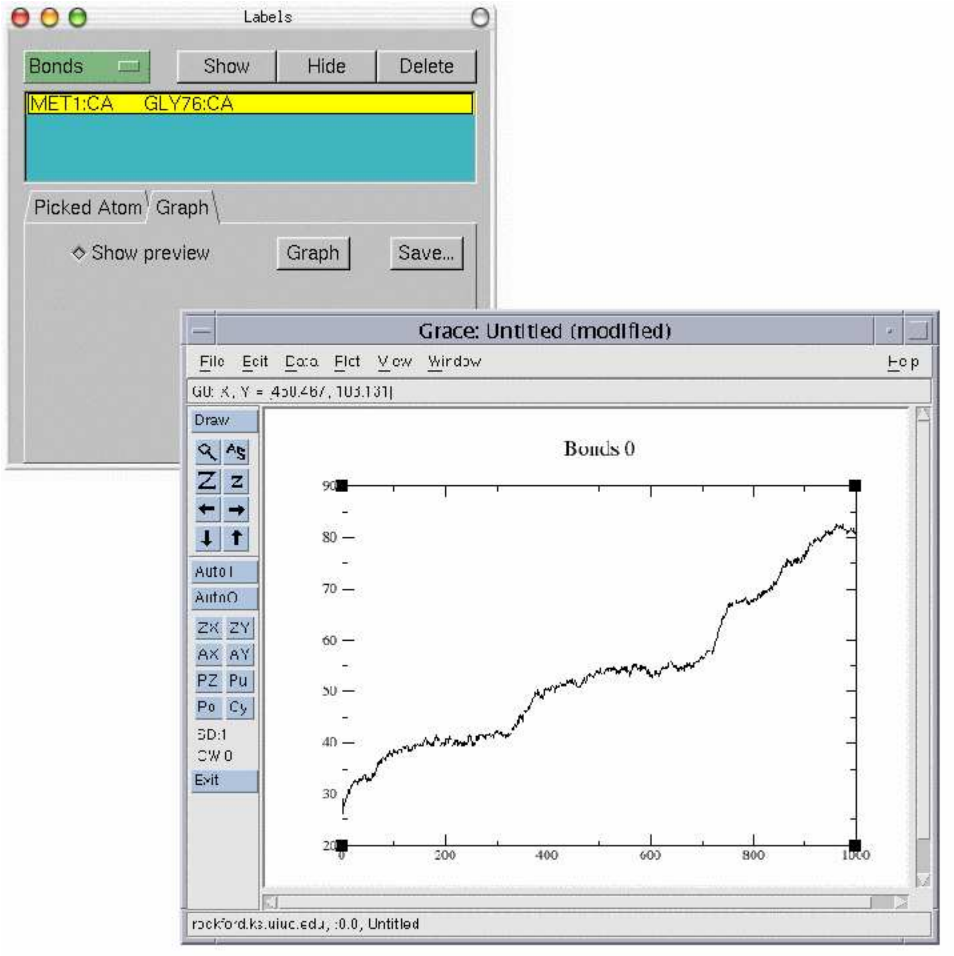
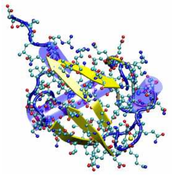


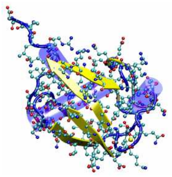
Figure 21: Label window Plot of selected bond over time created with the Graph button.



Ubiquitin unfolding. The plot displayed shows the length of the bond over time. You can think of it as a plot of the "length" of the molecule, as it is pulled at constant force.
Note that at the beginning of the curve, the curve is flattening, like if the pulling did not affect the structure of the molecule, and all of the sudden, the distance increases a lot. Take a look at the time this occurs: when the two yellow β sheets separated! This means that, those H-bonds that keep the β sheets together provide an initial resistance to the unfolding of the protein, and once they are broken, the protein unfolds in a more regular way. Try identifying what happens in the second jump. You will have identified key features of ubiquitin unfolding!

3.5 An Example Tcl Script: Calculating the RMSD of a trajectory

VMD is a powerful tool for MD analysis. In this section you will use tcl scripts to perform analysis of trajectories. You will load a new trajectory, the equilibration of the ubiquitin system. You will use a short script to calculate the RMSD of the protein during the MD run, and determine if the system is equilibrated and ready to simulate.



Root Mean Square Deviation. The Root Mean Squared Deviation (RMSD) is a numerical measure of the difference between two structures. It is defined as :

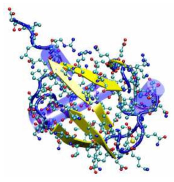
$$RMSD = \sqrt{\frac{\sum_{i=1}^{N_{atoms}} (r_i(t_1) - r_i(t_2))^2}{N_{atoms}}} \tag{1}$$

where N_{atoms} is the number of atoms whose positions are being compared, and $r_i(t)$ is the position of atom i at time t (or, if you are comparing two molecules, like in Unit 2, t labels the molecules).

1 Delete the current trajectory. Go to Molecule → Delete Frames.... The default values on this window will delete the whole trajectory. Click on the Delete button.

2 Load the equilibration trajectory into the psf file by going to the File → Load Data Into Molecule. The file is `equilibration.dcd`.

3 Turn on the water representation and take a look at the trajectory with the Animation tools that you learned before.



Equilibration. The trajectory you loaded is the equilibration of the ubiquitin and water system. In reality, this trajectory was simulated before the pulling trajectory you saw before. In MD, a system must be equilibrated first to have the stability necessary to observe other features related to the system and not to changes in the environment.

Is there anything you can say about this trajectory? Everything seems to be moving around randomly. Note how the water comes closer to the protein filling in the gaps, and the size of the water box decreases to obtain the right water density.

Now you will find out how to determine if the protein is equilibrated. One important factor to determine this is looking at the RMSD of a protein in a trajectory.

4 The script we are going to use is called `rmsd.tcl`. This is the script content:

```
set outfile [open rmsd.dat w]
set nf [molinfo top get numframes]
set frame0 [atomselect top "protein and backbone and noh" frame 0]
# rmsd calculation loop
for { set i 1 } { $i ≤ $nf } { incr i } {
 set sel [atomselect top "protein and backbone and noh" frame $i]
 $sel move [measure fit $sel $frame0]
 puts $outfile "[measure rmsd $sel $frame0]"
}
close $outfile
```

5 The script does the following:

- Open file `rmsd.dat` for writing
  `set outfile [open rmsd.dat w]`

- Get the number of frames in the trajectory and assign this value to the variable `nf`
  `set nf [molinfo top get numframes]`

- Select the first frame of the molecule to be the one other frames will compare to. The selection contains the atoms in the backbone of the protein, excluding hydrogens:
  `set frame0 [atomselect top "protein and backbone and noh" frame 0]`

- Text after # denotes comment.
  `# rmsd calculation loop`

- Loop over all frames in the trajectory:
  `for { set i 1 } { $i ≤ $nf } { incr i } {`

- Make the same selection as before for the current frame (the frame to be compared).
  `set sel [atomselect top "protein and backbone and noh" frame $i]`

- Calculate the matrix that will fit both selections. Apply this matrix to the second selection to align the molecules:
  `$sel move [measure fit $sel $frame0]`

- Calculate the RMSD value between these two selections, and write it to file:
  `puts $outfile "[measure rmsd $sel $frame0]"`

You can use the script for the system to test for equilibration.

6 Type `source rmsd.tcl` in the tkcon. This will perform all the commands in the script. The script will write a file `rmsd.dat` that will contain the value of the RMSD of the protein backbone against time.

Outside of VMD, you can use some plotting program to see this data. Examples of these are gnuplot, xmgrace, excel, Mathematica.

7 Use one of the above programs to plot the file `rmsd.dat` (For example, in Unix, you can type `xmgrace rmsd.dat` in a terminal). Can you see the RMSD curve flattening? This means your system is equilibrated!

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.

This ends the VMD tutorial. We hope that you learned a lot with it, and that you will make a great use of all the capabilities VMD has to offer.