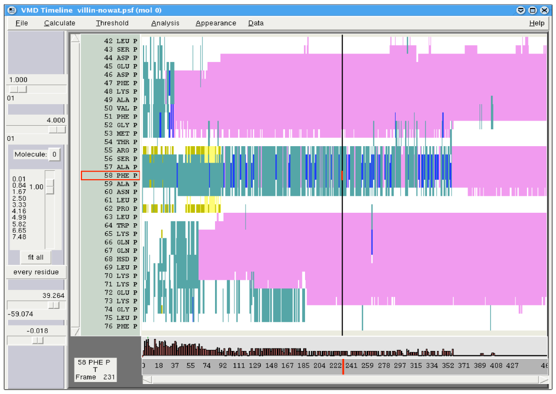
The University of Illinois at Urbana-Champaign
NIH Resource for Macromolecular Modeling and Bioinformatics
Beckman Institute
Computational Biophysics Workshop

Timeline: a VMD plugin for trajectory analysis



Author:
Barry Isralewitz

March 28, 2011

A current version of this tutorial is available at
http://www.ks.uiuc.edu/Training/Tutorials/

Contents

1 Introduction

This tutorial demonstrates how to use the VMD plugin `Timeline` to analyze and identify events in molecular dynamics (MD) trajectories. Timeline creates an interactive 2D box-plot – time vs. structural component – that can show detailed structural events of an entire system over an entire MD trajectory. Events in the trajectory appear as patterns in the 2D plot. The plugin provides several built-in analysis methods, and the means to define new analysis methods. Timeline can read and write data sets, allowing external analysis and plotting with other software packages. Timeline includes features to help analysis of long trajectories and trajectories with large structures.

In the main 2D box-plot graph, users identify events by looking for patterns of changing values of the analyzed parameter. The user can visually identify regions of interest – rapidly changing structure values, clusters of broken bonds, differences between stable and non-stable values, and similar. The user can explore the resulting structures by tracing the mouse cursor ("scrubbing") over the identified areas. The structure is highlighted and the trajectory is moved in time to track the highlight.

If you have any questions or comments on this tutorial, please email the TCB Tutorial mailing list at tutorial-l@ks.uiuc.edu. The mailing list is archived at http://www.ks.uiuc.edu/Training/Tutorials/mailing_list/tutorial-l/.

1.1 Getting Started

If you downloaded the tutorial from the web, the files that you will be needing can be found in a directory called `timeline-tutorial-files`. If you received the files through a workshop, they can be found in the same directory under the path ∼`/Workshop/timeline-tutorial`.

- **Unix/Mac OS X Users:** In a Terminal window type:

  ```
  cd <path to timeline-tutorial-files directory>
  ```

  You can list the content of this directory, by using the command `ls`.

- **Windows Users:** Navigate to the `timeline-tutorial-files` directory using Windows Explorer.

This will place you in the directory containing all the necessary files. In the figure below, you can see the structure of this directory.

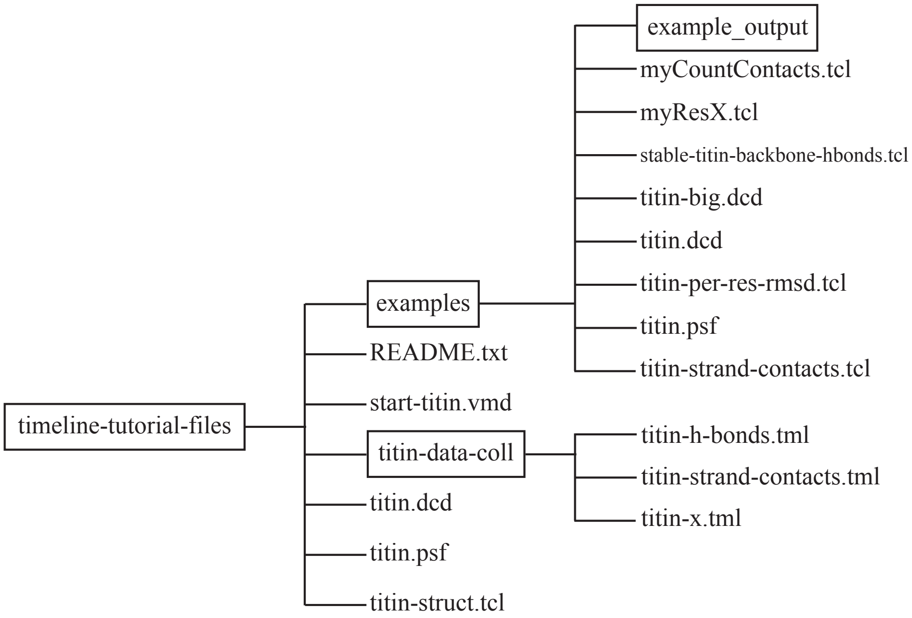


Figure 1: Directory structure for tutorial exercises. Output for file-writing exercises is provided in the "example_output" subdirectory.

1.2 Required Programs

The following programs are required for this tutorial:

- **VMD:** The tutorial assumes that you already have a working knowledge of VMD, which is available at http://www.ks.uiuc.edu/Research/vmd/ (for all platforms)
    - The VMD tutorial is available at http://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/
- a text editor of your choice (we offer a few easy-to-use recommendations):
    - UNIX: nedit (www.nedit.org)
    - Windows XP: WordPad (included with OS). We recommend using WordPad as opposed to NotePad for this tutorial. However, please ensure that you save any files in this tutorial as `.txt` format files, as opposed to `.rtf` or `.doc` files, and to use the file extensions specified in the exercise, e.g. `.tml`.
    - Mac OS X: Smultron (smultron.sourceforge.net), TextEdit (included with OS)

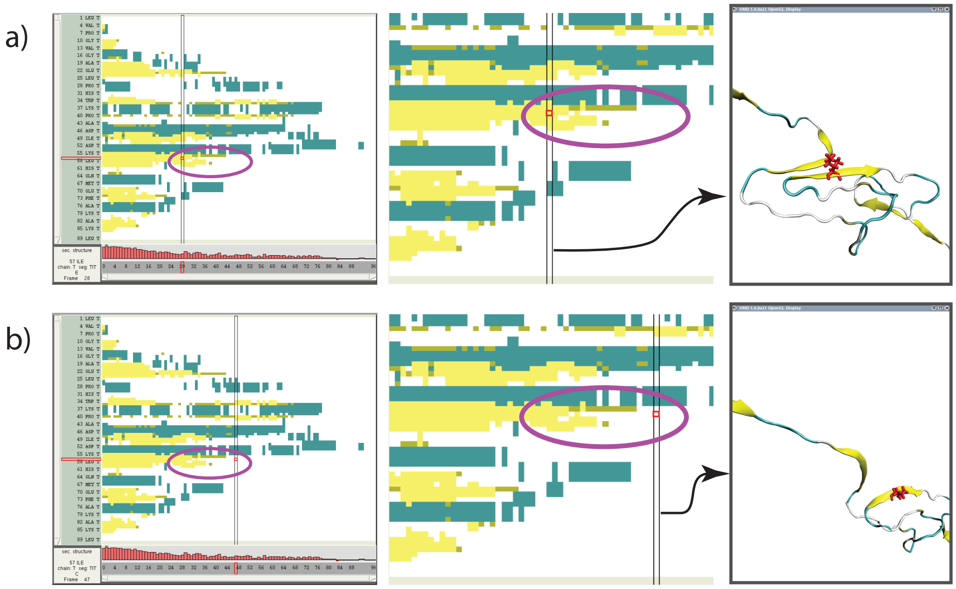


Figure 2: Exploring a secondary structure event. Row a) shows the system before the local structure change, row b) shows the system afterwards. The left panels show the entire data plot. The middle panels shows a zoomed-in part of the data plot, and the right panels shows the 3D structure of the selected trajectory frame. The vertical bar indicates the current frame. The purple oval, added for this figure, shows the rough area of the identified event. See text for additional details.

- a command prompt, such as a terminal in UNIX, Terminal.app in Mac OS X, or the DOS command prompt in Windows. Windows users can obtain a command prompt by clicking Start → Programs → Accessories → Command Prompt.

1.3 Timeline in action: examine events in titin domain extension

This subsection provides a short run-through of of Timeline features to provide a general idea of how the plugin is used. Later sections will introduce the display and user interface in greater detail. Here, we look at an MD trajectory of the I91 domain of the muscle protein titin. In the simulation trajectory, an external force is applied to the I91 domain with the Steered Molecular Dynamics method: one terminus of the domain is fixed and a harmonic restraint moving with constant velocity is applied to the other terminus, causing the domain to extend, unfold, and unravel. The secondary structure changes greatly as the domain unfolds; examining these changes will give us some initial insight into how the domain architecture responds to applied force.

With the titin I91 extension trajectory loaded into VMD, the user starts Timeline by selecting Extension → Analysis → Timeline from the main VMD window, then generates the secondary structure plot by selecting Calculate → Calc. Sec. Structure in the VMD Timeline window.

We examine trajectory events which appear as changes in value in the 2D plot. As the user moves the highlight cursor around the area of an event in the 2D plot, she can explore the behavior of the 3D structure. Figure 2 illustrates examining an event in a Timeline plot of per-residue secondary structure. By moving the cursor around the event area indicated with a purple oval, the user observes a beta strand, depicted in yellow, losing its secondary structure as it peels away from another beta strand. Note that the purple oval is added for the purposes of this tutorial, the user must normally visually identify the event area. Here the event to notice is the abrupt transition of a horizontal region of yellow to white: part of the domain losing the beta strand character (yellow) it had since the start of the trajectory and becoming random coil (white) as the domain extension proceeds. A user would scrub the mouse highlight cursor around the area of the purple oval, would produce the state shown in Figure 2a, before the local structure change, and examine the steps needed to reach Figure 2b, after the local structure change has taken place. To "scrub", click down the left mouse button with the cursor in the area of interest in the 2D graph, then move the cursor around the area of interest with the button still depressed – while splitting attention between the 2D graph, the changing 3D structure, and the changing numerical values displayed in the Highlight Details panel (described below).

In Figure 3, a plot of salt bridge lifetimes during the titin domain extension trajectory shown in Figure 2 is displayed, illustrating moving the highlight cursor to examine a salt bridge breaking event. This plot is per-selection; each selection is the pair of residues involved in a salt bridge. Only residue pairs that form a salt bridge for at least one frame of the trajectory are listed. A highlighted salt bridge is seen breaking between Figure 3a and 3b. The Timeline highlighted representation can be manipulated like any other VMD representation; here coloring of the highlight was changed from the default ColorID 1/red to ResType to show the different residue types of the salt bridge partners, and easily track salt bridge partners as they move apart.

In Figure 4 Timeline's Threshold Count plot is used to help find a suitable color scaling for the plot shown (per-residue Room Mean Square Fluctuation (RMSF)). The Threshold Count is the number of residues or selections in each frame with a value within a given range. As the Threshold Count range controls are adjusted as shown in Figure 4b, a pattern with multiple peaks appears (Figure 4c), this range is then used to change the color scale, with the result shown in Figure 4d. The Threshold Count sliders allow the users to very quickly scan through threshold ranges, without taking the time to redraw the entire 2D plot, while still giving an overview of the entire trajectory.

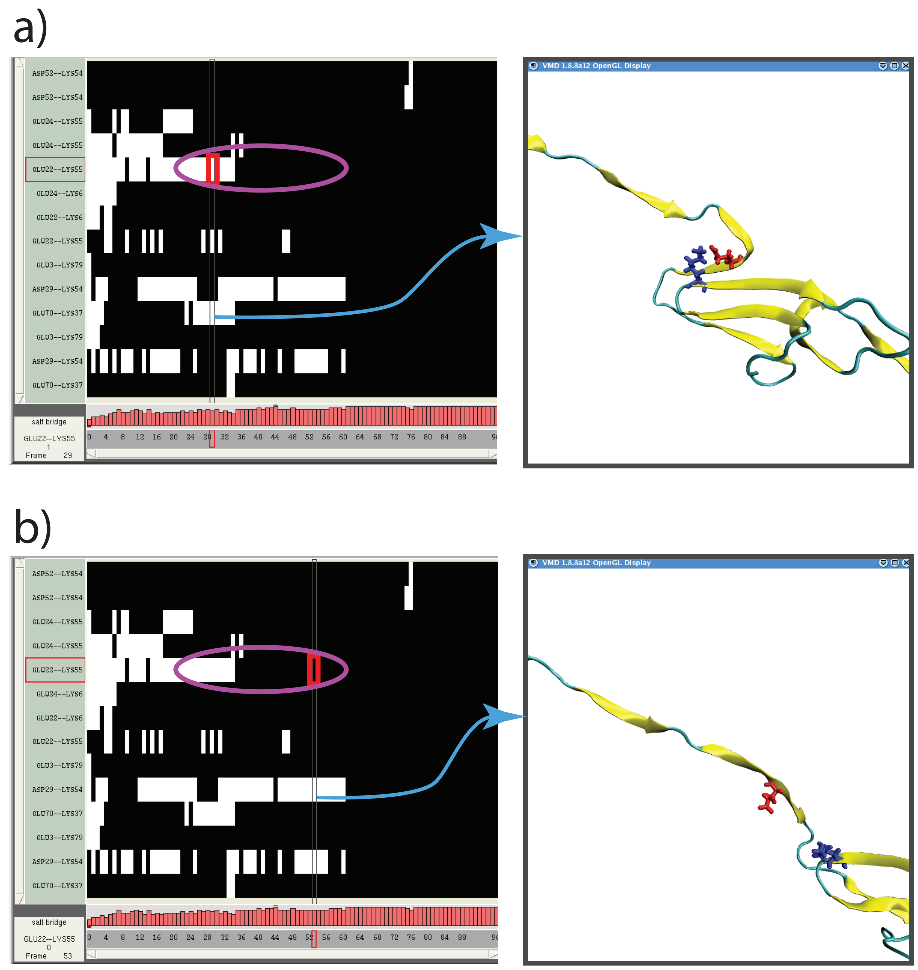


Figure 3: Exploring a salt bridge breaking event. A Timeline plot of the salt bridges found throughout a trajectory. The structural elements in the vertical axis are selection groups, with each group being the pair of residues involved in a salt bridge. The left panels show the data plot, and the right panels show the 3D structure of the selected trajectory frame. The vertical bar indicates the current frame. The purple oval, added to this figure, shows the rough area of the identified event. Row a) shows the system before the salt bridge breaks, row b) shows the system after the salt bridge breaks. See text for additional details.

1.4 Per-residue vs. per-selection calculations

Timeline plots data about a list of structural elements vs. time. There are two ways of defining the list of structural elements to be analyzed in Timeline.

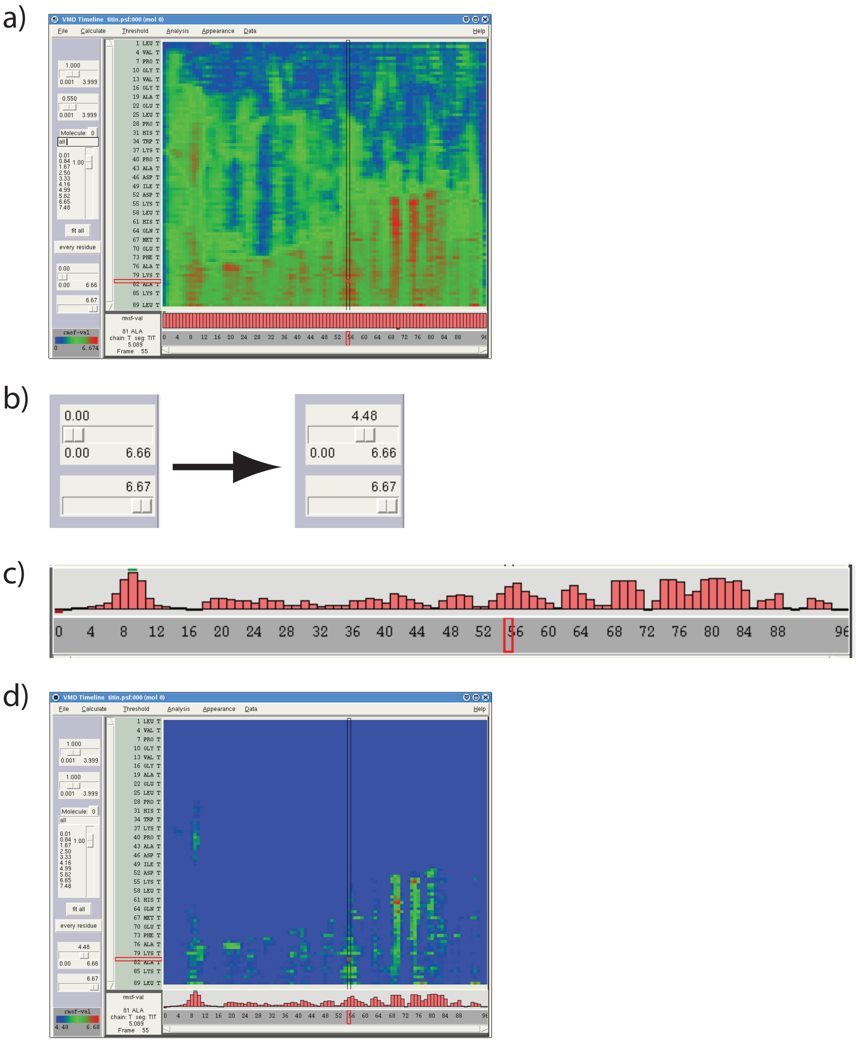

a)
VMD Timeline titin.psf:000 (mol 0)
File Calculate Threshold Analysis Appearance Data Help
b)
0.00
0.00 6.66
6.67
4.48
0.00 6.66
6.67
c)
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 96
d)
VMD Timeline titin.psf:000 (mol 0)
File Calculate Threshold Analysis Appearance Data Help

Figure 4: Using the threshold count to find a color scaling range. The per-residue Root Mean Square Fluctuation (RMSF) plot in a) has color scale set to the default, i.e., full, data range. Adjusting the Threshold Count range sliders in b) shows multiple peaks in the for the displayed range in c). In d), the 2D plot coloring scale of has been set to the range that was found in c), revealing which structures contribute to the threshold count.

Per-residue: a list of residues; the protein and nucleotide residues in the user-defined selection. The default selection is “all”, so this defaults to being a list of all residues in the current molecule.

Per-selection: a list of any VMD selections. For example, a list of salt bridge pairs, a list of the segments in a large protein complex, a list of of favorite structures in a molecule. These may be defined in built-in functions, listed explicitly, or defined algorithmically in user scripts.

2 Interface and controls

This general description of the Timeline interface should help with navigating in Section 3, "Trajectory Analysis".

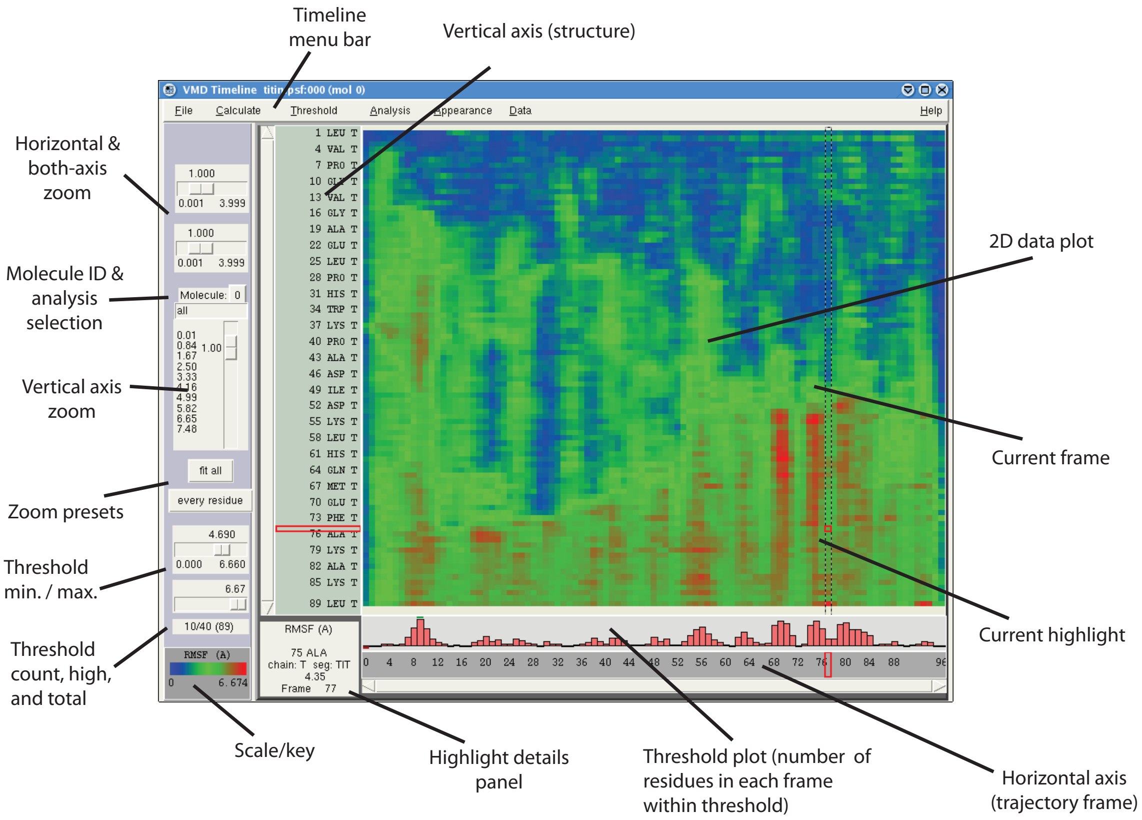


Figure 5: The Timeline 2D data plot. Main interface features are indicated.

See a closer view of parts of the interface in Figure 6, and descriptions of these in Table 1. A diagram of the Highlight Details panel is in Figure 7. The top-level Timeline menus are described in Table 2.

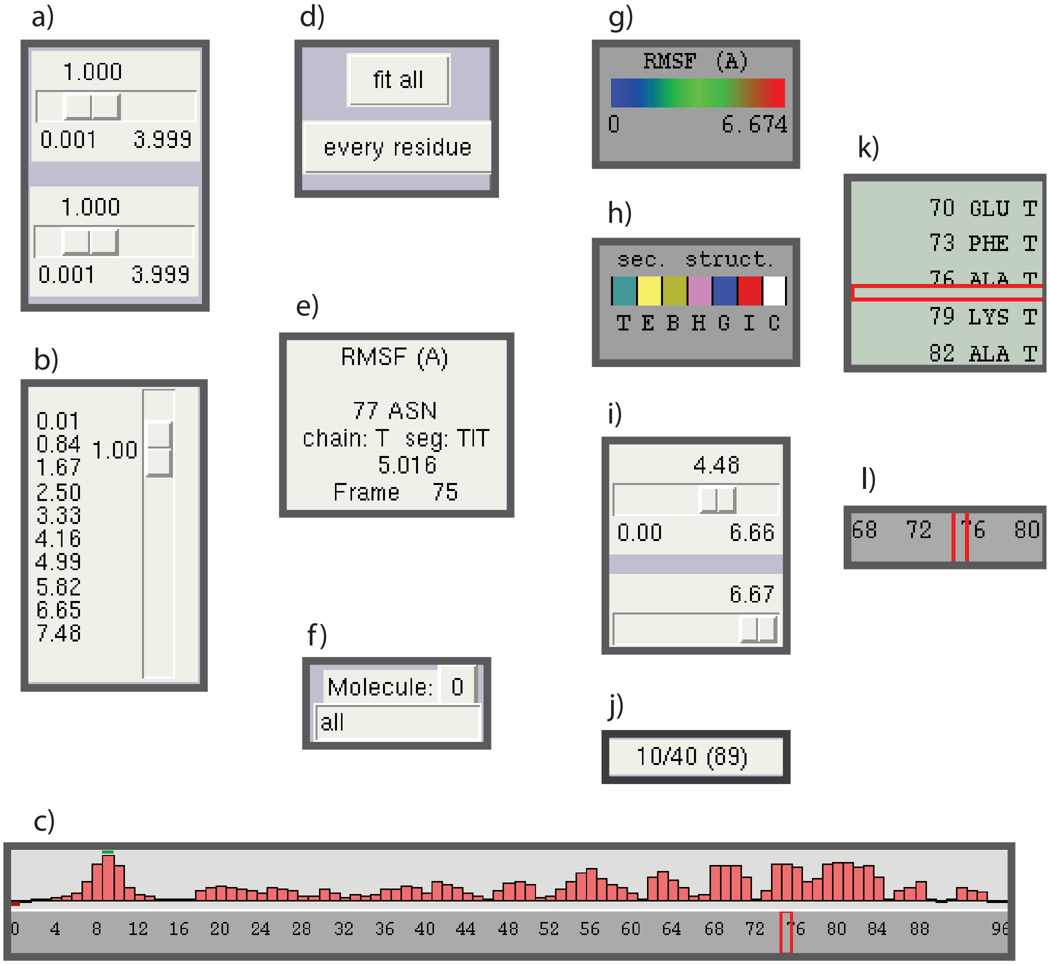
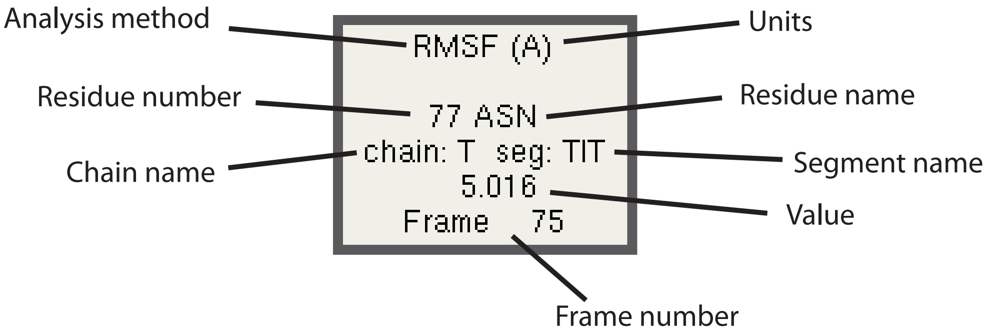


Figure 6: Parts of the Timeline interface. a) Horizontal and both-axis zoom sliders, b) Vertical axis zoom slider, c) Threshold count plot, d) Fixed-zoom buttons, e) Highlight Details panel, f) Molecule ID and Analysis selection, g) Key/Scale, h) Key Scale (secondary structure), i) Threshold minimum/maximum, j) Threshold count, high, total k) Current structure highlight, l) Current frame highlight



RMSF (A)
77 ASN
chain: T seg: TIT
5.016
Frame 75

Figure 7: Highlight Details panel.

Table 1: Timeline main interface features

Interface element	Description	See Figure
Current highlight selection cursor	This 2D highlight cursor determines the structure that is highlighted in the 3D molecule view, and which is described in the Highlight Details panel. Left mouse button click or drag to change the current highlight selection.	5
Horizontal axis zoom slider.	Scales the 2D data plot horizontally. Useful for changing time range viewed without changing structure range.	6a
Both-axis zoom slider.	Scales both axes simultaneously. Combine with resizing Timeline window to fit a larger trajectory or structure in less screen area.	6a
Vertical axis zoom slider.	Scales the 2D data plot vertically. Useful for changing range of structure data displayed, without changing time range.	6b
Fixed-zoom buttons.	The fit all button scales the entire loaded trajectory to display within the default window size. The every residue button scales the vertical axis to show an axis label for every residue.	6d
Molecule ID	Specifies the `molid` of the molecule to analyze.	6f
Analysis selection	A VMD selection string, specifies what part of the current molecule to use for per-residue analysis. Default is "`all`".	6f
Threshold count plot	Graphs number of items (residues or selections) in each frame that are within the threshold range for each frame. The max value is normalized to the height of the graph. First instance of max. (min.) value marked with a green (red) line.	6c
Key/Scale	Displays the analysis method, units, and numerical range of currently displayed colors.	6g
Key/Scale (sec. struct.)	When displaying Secondary Structure, displays the structure coloring codes. The one-letter code descriptions are listed in Help → Structure codes... and in Figure 9.	6h, 9
2D Data Plot	This is a plot of structural elements (residues or selections) vs. time. Exploring this plot for changes, transitions, and visually-detectable correlations helps the user find events in the trajectory.	5
Threshold min./max.	Set the range for Threshold plot. The Threshold plot will change as you drag these sliders. The limits of the sliders are adjusted for the range of the loaded data set.	6i
Threshold count, high, and total	Shows the threshold count for the currently highlighted frame, the highest threshold count in the trajectory, and the total number of items in the structure.	6j
Current frame bar	The long vertical black bar indicates the current frame, and helps to visually locate the highlight cursor.	5
Current frame highlight	Highlight box in the horizontal axis to indicate current frame	6l
Current structure highlight	Highlight box in the vertical axis to indicate current structure	6k
Mouse zoom-in/zoom-out.	Zoom in with right-button click-and-drag; this creates a rectangle selection, the contents of which will expand to fill the 2D plot. Zoom out with right click, or with the zoom slider and fixed zoom controls.	
Scroll Bars	Use to move around the 2D data block, whenever either scale is zoomed in beyond the "fit all" size. Moves thorugh the structure and the trajectory.	5
Current structure highlight	The long vertical black bar indicates the current frame; this along with highlights in the horizontal and vertical axes labels helps visually locate the highlight cursor.	5
Highlight Details panel	Description of the residue/selection beneath the 2D cursor. The current residue number, residue name, chain, segment, and frame, data value for the selected residue, and current analysis method.	6e, 7

Table 2: Timeline Top-level Menus

Title	Description
File	Allows saving and loading formatted data sets, and printing of the 2D data plot.
Calculate	Provides several built-in calculation options, access to user data fields, and user-defined per-residue functions.
Threshold	Allows direct entry of values for threshold graph, and clearing/recalculating the graph.
Analysis	Allows a user to enter the user-defined per-residue function, and to set the frame range to which to apply the chosen analysis method.
Appearance	Sets the value range for scaling on the 2D data plot, the shading pallette (greyscale or rainbow), and other user interface details.
Data	File list of the current data collection, for quick browsing between .tml files in a designated directory.

3 Trajectory analysis: titin domain extension

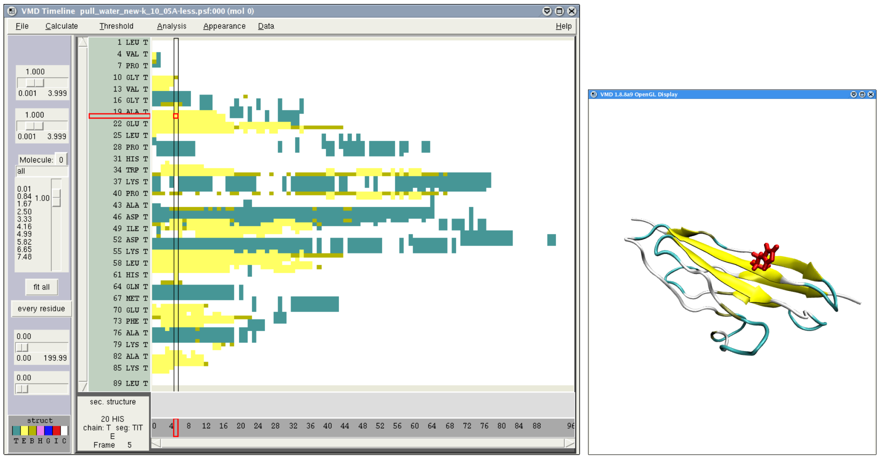
To use Timeline to identify events in a trajectory, we must choose which parameters to examine, perform the required analyses, and explore the resulting 2D data sets. We use as an example the titin I91 extension trajectory introduced above, and start with examining secondary structure, to get an overall sense of the architectural changes that take place during forced extension. Then we will look at geometric fluctuations during the trajectory, which will both illustrate how to use some of the Timeline tools and provide some hints as to when and where important events are taking place in the I91 extension. Then we will look at the hydrogen bond breaking between the β-strands of the domain's β-sheets, to get more insight into how the protein architecture handles applied force.

3.1 Secondary structure during titin domain extension

We examine how the secondary structure of the domain changes as the force extension takes place. We can see visually by animating the 3D trajectory that the domain begins as a β-sandwich then unravels as the force extension takes place, but there is more to the story; we can make a 2D data plot to examine the fate of each β-strand.

- Start a new VMD session. Launch VMD by:
  - **Unix/Mac OS X Users:** typing `vmd` in a Terminal window.
  - **Windows Users:** using the Start menu.
- Load in the PSF file for titin I91 domain. In the VMD Main window, select File → New Molecule, then in the Molecule File Browser window, with Load files for: New Molecule selected, navigate to the `timeline-tutorial-files` directory and select `titin.psf`.
- Load in the titin extension trajectory. In the Molecule File Browser window, with Load files for: 0: titin.psf selected, open titin.dcd, which is also in the `timeline-tutorial-files` directory. The trajectory shows a steered molecular dynamics extension of the I91 domain of the giant muscle protein titin, solvated in a water droplet with a non-periodic boundary.
- Adjust the VMD Main time slider to frame 0. In the VMD Main window, select Graphics → Representations... to display the Graphical Representations window. In the Graphical Representations window, edit the default `all` selection to `not water`. Set the Coloring Method to Secondary Structure and Drawing Method to New Cartoon.
- Open the Timeline plugin window. In the VMD Main window, select Extension → Analysis → Timeline.
- Display secondary structure for all frames, all residues in the trajectory. In the VMD Timeline window, select Calculate → Calc. Sec. Structure

- Scrub the cursor and explore the 2D data plot. With left mouse depressed move the 2D cursor around the 2D data plot. Note how the 3D highlight (a red Licorice representation) moves around the molecule, highlighting whatever is currently under the 2D cursor.



sec. structure

20 HIS
chain: T seg: TIT
E
Frame 5

Figure 8: Secondary structure for all residues of I27 for all frames of the loaded trajectory. The structure corresponding to the red highlight rectangle in the Timeline (left) data panel is shown in the 3D view (right panel). See Figure 9 for a magnified view of the color key.

- Get detailed information about the trajectory. Details about the residue beneath the 2D cursor are shown in the details box to left and below the box plot. The current residue number, residue name, chain, segment, and frame, as well as the data value for the selected residue, and the current analysis method. These change as you move the cursor.

- Zoom and scroll around the data plot. Right-mouse-click-and-drag to draw a green-outline box to define the area to zoom into. Right-click will zoom out. The three slider controls on the right side of the Timeline window also control zoom: the vertical slider zooms vertically, the horizontal slider zoom horizontally, the central zoom slider scales the whole data block at once. The fit all button scales both axes to fit within the standard size of the Timeline window. The every residue button changes the vertical scale to display numbering for every vertical element (residues or selections (covered below, in Section 4.3)).

- Examine color scale key: The color key for secondary structure attributes is labeled "sec. structure", and displays the one-letter structure codes used by STRIDE software. The one-letter code descriptions are listed in Help → Structure codes..., and in Figure 9. Most relevant here, 'E' (yellow) corresponds to 'Extended conformation', the main component of beta

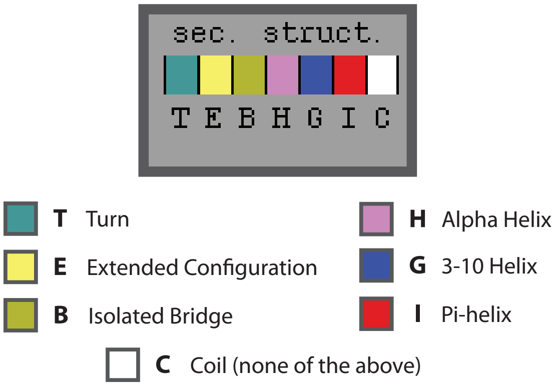


Figure 9: Color key for secondary structure plots.

sheets; 'T'(aqua) corresponds to "Turn", another beta sheet component, and 'C' (white) stands for "Coil" (random coil, no structure). Note that secondary structure is a special case for coloring; for all other cases of calculations, the color scale shows a numerical range.

- Identify the time frames when each of the individual β-strands begins to lose secondary structure, and when it has completely lost secondary structure.

3.2 Thresholding and changing appearance: examining RMSF

There are two related ways to help examine how the values of a data set are distributed: first, by using the Threshold Count tool; second, by changing the color scaling range of the 2D plot. The Threshold Count tool adds up, for each frame, the number of residues or selections that fall within a given range, then plots these numbers for all frames. The plot is dynamically updated as the Threshold Min and Max values are changed. Here we review how to do this for the Root Mean Square Fluctuation data set examined in Figure 4, then use the results from this to set the color scaling range, in order to identify which structures are involved in the fluctuation events.

- Display root mean square fluctuation for all frames, all residues in the trajectory. In the VMD Timeline window, select Calculate → Calc. RMSF

- Use the thresholding tools. Adjust the Threshold Plot by changing the Threshold Min and Threshold Max bounds sliders. (Type in values here with select Threshold → Set bounds... Set them to 4.48 and 6.67 and note the repeating pattern of peaks. This thresholding can change much more quickly than we can adjust the 2D plot (especially for a large plot) and can track changes hard to total up by eye. The Threshold plot updates as the range is changed, to allow quick exploration of the plots. Now apply the threshold bounds values we set to the coloring of the whole 2D plot: select Appearance→ Set scaling... and enter `4.48` and `6.67` as bottom and top values. The result should resemble Figure 4c. Scrub the highlight cursor over the transitions in the resultant plot to see what structures are involved in the fluctuation increases.

More on changing appearance. Perform other changes to the Timeline plot, to see features that may be helpful when working with different data sets or different structures:

- Toggle display of one letter vs. three letter residue codes in the vertical plot labels by selecting Appearance → Show 1-letter codes .
- With the Analysis Selection set to `all`, or to any interesting sub-selection, make an RMSF plot as before with Calculate → Calc. RMSF. Change the 2D Plot color scale with Appearance → Color scale → Rainbow and Appearance → Color scale → Grayscale.
- For more experimentation with this specific trajectory, source a TCL script to load the example system in the state shown in Figures 2 and 3, with cartoon representation and coloring showing native secondary structure. To this, start a new VMD session, change directory using the VMD Tk terminal to `timeline-tutorial-files`, then enter into the VMD Tk terminal `source start-titin.vmd`. Note that this file sources `titin-struct.vmd` in the same directory. The latter Tcl script imposes the known native secondary structure of the published structure, in order to show more accurate beta-strands at trajectory start compared to that determined from the built-in secondary structure algorithm (STRIDE).

3.3 Examine backbone hydrogen bonds during titin extension

Now we will look at the hydrogen bond breaking pattern during the I91 extension trajectory. Since we are looking at the protein architecture, we will examine the bonds between the partner β-strands that make up the two β-sheets which form the I91 domain β-sandwich.

- Change the Analysis Selection from `all` to `name N HN O C`, to select only the atoms involved in inter-β-stand hydrogen bonds.
- Calculate the hydrogen bonds that can be formed with the current selection, in Calculate → Calc. H-bonds.... Enter a Bond distance cutoff of 3.5 and an Angle cutoff (deg) of 45, then click the Calculate button . The result should resemble Figure 10.

- The value of data in the 2D plot is either 0 or 1. Set the Threshold Plot range from 0 to 1 to see the number of formed hydrogen bonds in each frame. (Although the auto-ranging is already set from 0 to 1, you must still adjust the threshold minimum control once to display the Threshold Plot).
- Zoom in the plot to examine individual hydrogen bonds. Click on the atoms in the 3D view to display the names and numbers of the residues involved.
- Use the hydrogen bond plot, and the fit all button, to look for overall patterns of hydrogen bonds breaking. Look for H-bond breaking — the end of a white horizontal line — clustered around the same time, and check the associated structures. While using a copy of a secondary structure plot such as Figure 8 for reference, note which β-strands the bond breakings are associated with, and how they relate to the beginning/ending times of secondary structure loss (unraveling) of individual β-strands.
- Load in a version of the I91 extension trajectory with four times as many frames (four times the temporal detail):
  navigate to to the `examples` directory and load `titin-big.dcd`
- We will examine a set of inter-β-strand hydrogen bonds that are stable during equilibration of I91. Change directory to the `examples` directory and type
  `source stable-titin-backbone-hbonds.tcl` in the Tk console. This will add two vmd atom macros: `backboneHSel` and `backboneOSel`.
- Change the analysis selection to `backboneHSel or backboneOSel`.
- Select Calculate → Calc. H-bonds... and click the Calculate button.
- Scrub through the trajectory with the highlight cursor, using the Threshold count tool to note how the hydrogen bond count has the largest drop between frames 11 and 18. Note where these breaking H-bonds are in the 2D view, as in the previous example.
- Add both `CPK` and `Hbond` graphic representations as for selection `backboneHSel or backboneOSel`. Note the hydrogen bonds ending in the 2D graph between frames 11 and 18, and how they correspond to the hydrogen breaking in the 3D view, as shown in Figure 11. These simultaneous hydrogen bond breaking seen in Figure 11c is what is responsible for the extension force peak seen in Figure 11a. Under forced extension, I91 architecture cannot allow further unraveling until these bonds are broken.

3.4 Using Data Files and Data Collections

These features help with saving and loading completed calculations (so you won't have to spend time calculating them again), and with creating and browsing sets of pre-computed calculations.

- Export and Import the calculated data. Select File → Write Data File... and save as `test-rmsf.tml`. Clear the plot with Calculate → Clear data, then read the data file back in with File → Load Data File..., and specify the `test-rmsf.tml` you just saved.

- Use a data collection: With the titin trajectory loaded, select Data → Set Data Collection directory... and select `titin-data-coll` in the dialog box, which is in the `timeline-tutorial-files` directory.

- The Data menu is now populated with the .tml files in that directory, select each to load that calculation data. Select Data → titin-strand-contacts.tml, then Data → titin-h-bonds.tml, etc. Loading each data set will reset the coloring range of the 2D plot to the full range of each loaded set.

- Try making a new directory, save two or more .tml files to it, then display the contents via the Data menu.

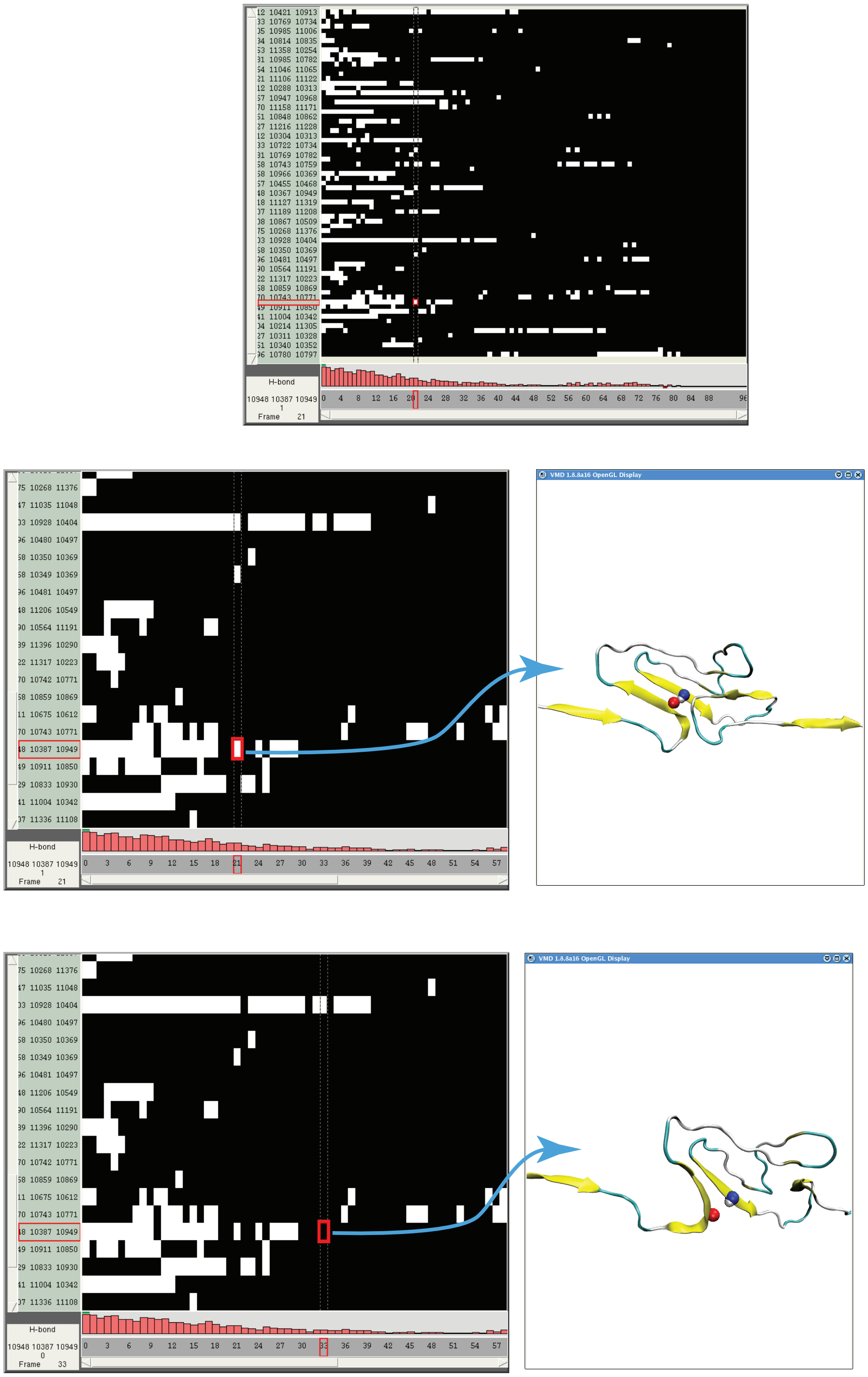


Figure 10: Exploring hydrogen bonds during titin I91 extension. The top panel shows hydrogen bond for all residues of I91 for all frames of the loaded trajectory. In the middle and bottom rows, structure corresponding to the red highlight rectangle in the zoomed-in Timeline 2D data plot (left panel) is shown in the 3D view (right panel). Hydrogen bonds are plotted as white when formed, and black when broken. The middle row shows highlighted a single, unbroken backbone hydrogen bond. The bottom row shows highlighted the same backbone hydrogen bond, now broken, at a later time frame.

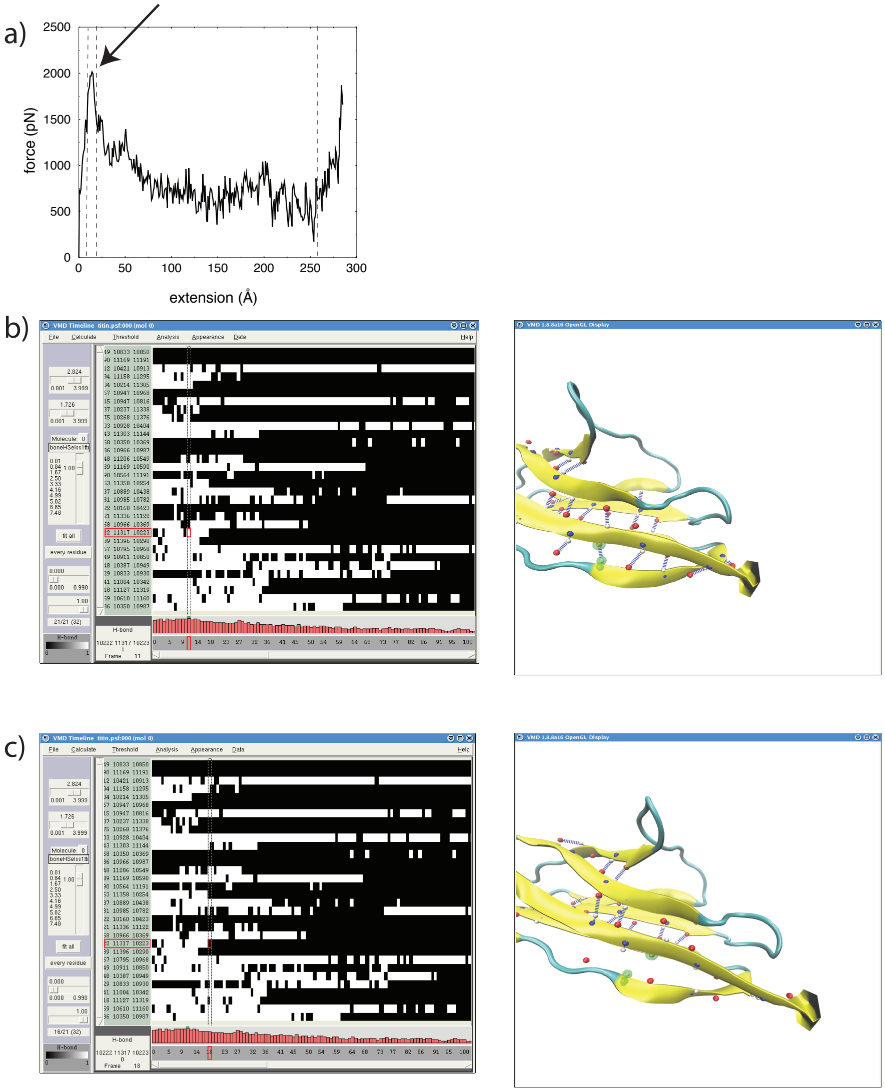


Figure 11: Clustered bond breaking during I91 extension. a) Force-extension curve of I91 extension, early force peak is marked with an arrow b) Before clustered bond breaking c) After clustered bond breaking. In b) and c) , structure corresponding to the red highlight rectangle in the zoomed-in Timeline 2D data plot (left panel) is shown in the 3D view (right panel). Hydrogen bonds are plotted as white when formed, and black when broken. The selected hydrogen bond is highlighted in green in the 3D plots.

4 Analysis Scripts

4.1 Simple user-defined all-residue scripts

The Calculate → Calc. User Defined (Per.-Res.) command allows you to call the name of a user-defined tcl procedure which will be run on every residue. Name the function in `Analysis → Define every residue function`; prepend "::" to indicate the global namespace. The named Tcl procedure will be provided with three VMD selections to work with: `resCompleteSel` (the current residue or selection group), `resAtomSel` (a single atom for each protein or nucleic residue), and `proteinNucSel` (`"protein or nucleic"` VMD selection for the whole molecule).

Example 1:

Type into the VMD Tk console:

```
proc ::myResX {resAtomSel  resCompleteSel  proteinNucSel} {
    return [$resAtomSel get x]
}
```

Example 2: Type into the VMD Tk console:

```
proc  ::myCountContacts {resAtomSel  resCompleteSel  proteinNucSel} {
    return [llength [lindex [measure contacts 4.0 $resCompleteSel \
             $proteinNucSel ] 0]]
}
```

The above files can be sourced from the VMD Tk Console:
`cd timeline-tutorial-files`, `cd examples`,then `source myResX.tcl`
and
`source myCountContacts.tcl` .
The functions will remain available for the rest of the VMD session; .

4.2 Writing a .tcl script to generate a per-residue .tml data file

The Timeline plugin can save data to `.tml` files, which provide per-residue or per-selection data for a given molecule and trajectory. This subsection shows how to generate a per-residue `.tml` file using Tcl code. These can be generated in a non-interactive VMD session, running on one or more processors, to enable long calculation jobs.

Use your text editor to create a tcl file in the the `timeline-tutorial-files` to contain the data calculation script. Name it: `titin-per-res-rmsd.tcl`. We'll include some simple commands, and two tcl procedures: the user-specfied analysis `myRmsdBatchCalc`, and the plugin-provided `::timeline::writeDataFileHeader`

See sample scripts, including the completed version of `titin-per-res-rmsd.tcl`, in the `examples` directory, which is in the `timeline-tutorial-files` directory.

- Require the Timeline Tcl package (will load if not already present; this is required when VMD is started in text-only mode)

```
package require timeline 2.0
```

- Add commands to load trajectory data. For a newly-started VMD session, the following will load the trajectory as molecule 0, so we'll set that as the molecule to analyze.

```
mol new titin.psf type psf first 0 last -1 step 1 filebonds 1 \
        autobonds 1 waitfor all
mol addfile titin.dcd type dcd first 0 last -1 step 1 filebonds 1 \
        autobonds 1 waitfor all

set myMolid 0
```

- Set the file name to which you will write data:

```
set outFilename titin-per-res-rmsd.tml
```

- Begin a tcl procedure to analyze and write the analysis data

```
proc myRmsdBatchCalc {filename molid} {
```

- Since we are making a per-residue data file, `usesFreeSelection` is set to 0. (See the follwinng subsection for more on "Free Selection".)

```
  # 0 for per-residue calcs
  # 1 for per-selection calcs
  set usesFreeSelection 0
```

- Set all the following values, which will satisfy all user-set header data Here, we set set the chain and the number of residues we examine manually. This molecule has only 1 chain; for molecules with more chains we would have to generate data for all chains.

```
  # include units in title
  set dataTitle "res. RMSD (A)"
  set firstFrame 0
  set lastFrame 96
  # the sample molecule is a 1-chain, 1-segment molecule, so
  # we will do a simple loop over residue numbers
  set theChain "T"
  set theSeg "TIT"
  set firstRes 1
  set lastRes 89

  set numFrames [expr $lastFrame - $firstFrame + 1]
```

- Set the number of items (referred to as *selection groups*)

```
  set numSelectionGroups [expr $lastRes-$firstRes + 1]
```

- Check the filename and open the file for writing

```
if {$filename == ""  } {
    die "usage: myRMSDBatchCalc FILENAME MOLID\n FILENAME\
           cannot be an empty string."
   }

set outDataFile [open $filename w]
```

- Now write out the `.tml` data file header.

```
::timeline::writeDataFileHeader $outDataFile $molid $dataTitle \
     $numFrames $numSelectionGroups $usesFreeSelection
```

- Now perform the actual analysis calculations for the whole trajectory, looping over residues and frames, and output the data. For this example, we calculate the RMSD of each individual residue for each frame of the simulation vs. its initial (frame 0) configuration

```
  set chain $theChain
  set seg $theSeg
  for {set r $firstRes} {$r <= $lastRes} {incr r} {
    set sela [atomselect $molid "resid $r"]
    set selb [atomselect $molid "resid $r"]
    $sela frame 0
    for {set f $firstFrame} {$f<=$lastFrame} {incr f} {
      $selb frame $f
      display update
      set val [measure rmsd $sela $selb]
      set resid $r
      puts $outDataFile "$resid $chain $seg $f $val"
    }
  }
```

- Now that all the data has been written, close the data file and end the procedure.

```
  close $outDataFile
  return
}
```

- Now we call the procedure we just entered.

  ```
  myRmsdBatchCalc $outFilename $myMolid
  ```

- Generate the `.tml` file by running the `.tcl` script in VMD, in either an interactive session using the `source` command or at a command prompt:

  ```
  vmd -dispdev text -eofexit < titin-per-res-rmsd.tcl > titin-per-res-rmsd.log
  ```

  .

- To view the data in Timeline, load the corresponding trajectory into an interactive VMD session, then load the .tml file you generated into Timeline, in either of these ways:
  - Use Timeline menu File → Load Data File to load a single file
  - or via Timeline menu Data → Set Collection Directory when you have placed the `.tml` file placed in a directory with other `.tml` files to form a Data Collection for this trajectory.

- An example of the output Timeline data file `titin-per-res-rmsd.tml`, which is in the `example_output` directory in the `examples` directory.

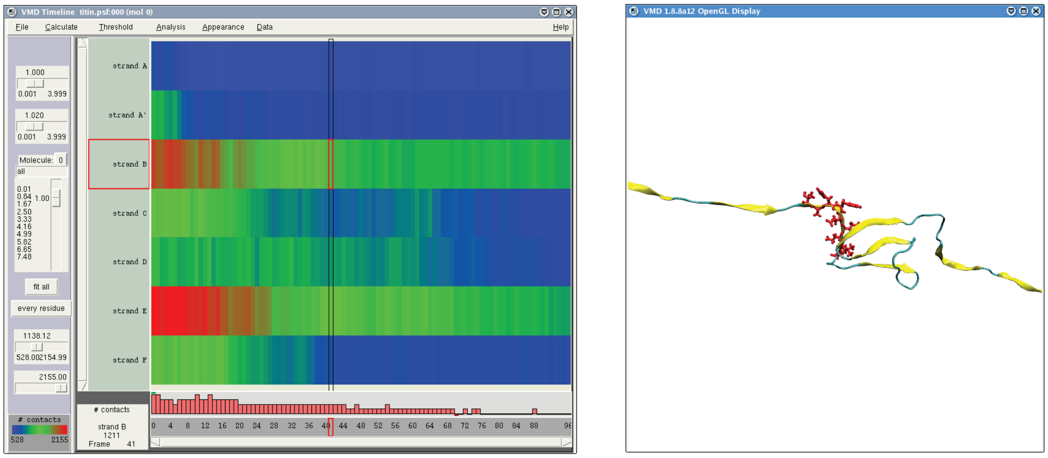


Figure 12: Protein contacts count, per beta strand, for titin I91 domain, over all frames of the loaded trajectory. When selected in the 2D view (left panel), the whole strand is highlighted in the 3D view. (right panel). The values in the 2D view are the total protein contacts for all the residues in each strand.

4.3 Writing a .tcl script to generate a per-selection .tml data file

This subsection shows how to generate a per-selection `.tml` file using Tcl code. These can be generated in a non-interactive VMD session, running on one or more processors, to enable long calculation jobs.

See a per-selection plot in Figure 12. Note that the selection of a single element in the 2D plot selects a large selection of the molecule; here all the residues in beta-strand B.

- Create a tcl file to contain the data calculation script. For example: `titin-strand-contacts.tcl`. We'll include in this file some simple commands, and two tcl procedures: the user-specfied analysis `myUserDataWrite`, and the plugin-provided `::timeline::writeDataFileHeader`.
- See sample scripts, including the completed version of `titin-strand-contacts.tcl`, in the `examples` directory in `timeline-tutorial-files`.
- Require the Timeline Tcl package (will load if not already present; this is required when VMD is started in text-only mode)

```
package require timeline 2.0
```

- Set the file name of the .tml file you want to produce, and the target molecule number.

```
set theFilename titin-strand-contacts.tml
set theMolid 0
```

- Add commands to load the trajectory data. For a newly-started VMD session, the following will load trajectory as molecule 0:

```
# load the file
mol new titin.psf type psf first 0 last -1 step 1 filebonds 1 \
        autobonds 1 waitfor all
mol addfile titin.dcd type dcd first 0 last -1 step 1 filebonds 1 \
        autobonds 1 waitfor all
```

- Start defining the analysis procedure:

```
proc myBatchDataWrite {filename molid} {
```

- Since we are making a per-selection data file, `usesFreeSelection` is set to 1. "Free Selection" since the user can assign arbitrary atoms to each selection group. The atoms in a selection group are set at the initial frame of the loaded trajectory; the atoms in a selection group are not recalculated for later frames in the trajectory.

```
# 0 for per-residue calcs
# 1 for per-selection calcs
set usesFreeSelection 1
```

- Set all the following data, which will satisfy the required header data except for numSelectionGroups.

```
set dataTitle "# contacts"
set firstFrame 0
set lastFrame 96

set numFrames [expr $lastFrame - $firstFrame + 1]
```

- Set the number of selection groups. For each selection group, there is a VMD selection string, and a human-readable label.

```
  set labeledSels  [list {"strand A" "resid 4 to 7"}\
          {"strand A'" "resid 11 to 15"} \
          {"strand B" "resid 18 to 26"}  \
          {"strand C" "resid 32 to 36"}  \
          {"strand D" "resid 47 to 52"}  \
          {"strand E" "resid 55 to 61"}  \
          {"strand F" "resid 69 to 75"}  \
          {"strand G" "resid 78 to 88"}]
  set numSelectionGroups [llength $labeledSels]
```

- Set data filename to output and open the file for writing

```
  if {$filename == ""  } {
    die "usage: myBatchDataWrite FILENAME MOLID"
   }

  set outDataFile [open $filename w]
```

- Analyze the trajectory. For this tutorial, we have manually specified selections, and so here we do know the number of selection groups. In other scripts, selection groups are chosen during analysis, e.g. for a hydrogen bond display script, each selection group is the set of atoms involved in a hydrogen bond, and the number of selection groups is the total number of such groups detected across frame in the loaded trajectory, which changes with each trajectory.

```
  # Calculation takes place here
  # Also, assigns the freeSelLabel and freeSelString for
  # each selection group.
```

```
set protSel [atomselect $molid "protein"]
set i 0
foreach {selPair} $labeledSels {
  foreach {label seltext} $selPair {}
  puts "starting sel. label: >>$label<<  seltext: >$seltext<"
  set trajdat(freeSelLabel,$i) $label
  set trajdat(freeSelString,$i) $seltext
  set sel [atomselect $molid $seltext]
  for {set trajFrame $firstFrame} {$trajFrame <= $lastFrame} \
         {incr  trajFrame} {
    $sel frame $trajFrame
    $protSel frame $trajFrame

    # Calculation...
    set d  [llength [lindex [measure contacts 4.0 $sel $protSel] 0]]

    set trajdat($trajFrame,$i) $d
    puts "$seltext, $trajFrame $i $d"
  }
  $sel delete
  incr i
}
```

- Set the default value for any unset selection/frame combinations, in case we missed any:

```
set defaultVal 0
```

- Here we write the file header, several lines of formatted comments at the start of the file to identify the file version, number of entries, some data labels, etc.. It is the same for all .tml calculation files, and is provided by the plugin. See the manual for details of the current header format, a current version of Timeline should be backward compatible with all older .tml file formats.

```
::timeline::writeDataFileHeader $outDataFile $molid $dataTitle $numFrames \
     $numSelectionGroups $usesFreeSelection
```

- ...write out the calculated data ...

```
for {set i 0} {$i<$numSelectionGroups} {incr i} {
  puts $outDataFile "freeSelLabel $i $trajdat(freeSelLabel,$i)"
  puts $outDataFile "freeSelString $i $trajdat(freeSelString,$i)"
  for {set trajFrame 0} {$trajFrame < $numFrames} {incr  trajFrame} {
    if {![info exists trajdat($trajFrame,$i)]} {
       set v $defaultVal
    } else {
       set v $trajdat($trajFrame,$i)
    }
    puts $outDataFile "$trajFrame $i $v"
  }
}
```

- ...then close the data file, and end the procedure.

```
  close $outDataFile
  return
}
```

- Now call the calculation procedure for molecule 0.

```
myBatchDataWrite $theFilename $theMolid
```

- Generate the `.tml` file by running the `.tcl` script in VMD, in either an interactive session with `source titin-strand-contacts.tcl` or in a non-windowed command line session as in
  `vmd -dispdev text -e titin-strand-contacts.tcl`
  or
  `vmd -dispdev text -eofexit < titin-strand-contacts.tcl > tsc.log`
  .
- You can read the resulting `tml` file into Timeline in two ways: via menu File → Load Data File to load a single file or via Data → Set Collection Directory when placed in a directory with other `.tml` files to form a Data Collection for this trajectory.
- An example of the output Timeline data file `titin-strand-contacts.tml`, which is in the `example_output` directory in the `examples` directory.
- Note how the number of contacts per strand changes as individual strands unravel. Compare this to the secondary structure 2D plot in Figure 8 and to the hydrogen bond-breaking patterns in Figure 10.

5 Acknowledgements

Development of this tutorial was supported by the National Institutes of Health (P41-RR005969 - Resource for Macromolecular Modeling and Bioinformatics).