NAMD-VMD tutorial (update June 2017)

Method · June 2017

DOI: 10.13140/RG.2.2.33910.60482

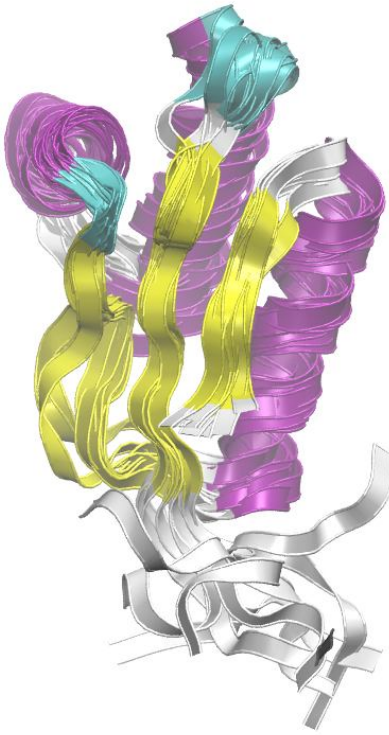
1 author:

Andreas Kukol
University of Hertfordshire
78 PUBLICATIONS **1,961** CITATIONS

SEE PROFILE

Molecular dynamics simulation of protein "folding" using VMD/NAMD

"*Certainly no subject or field is making more progress on so many fronts at the present moment, than biology, and if we were to name the most powerful assumption of all, which leads one on and on in an attempt to understand life it is that all things are made of atoms, and that everything that living things do can be understood in terms of jigglings and wigglings of atoms.*" (Richard Feynman, 1963)



Andreas Kukol
University of Hertfordshire
School of Life and Medical Sciences

(version June 2017 for VMD 1.93 and NAMD 2.12)

Aims and Objectives

To become familiar with molecular dynamics simulations of biomolecules and protein folding.

Skills

- Prepare protein structure files (pdb-file) for molecular dynamics (MD) simulation
- Create a solvated protein in the computer
- Add counterions
- Perform energy minimisation
- Perform position restraint MD simulations
- Perform unrestrained MD simulations
- Analyse the trajectory visually using molecular graphics programs (e.g. VMD)
- Quantitative analysis of the trajectory (root mean square deviations, hydrogen bonds etc.)

Please read THESE instructions carefully <u>before</u> the practical. Good to read is also the popular review by Sansom (2010).

Summary of Questions to be answered in your logbook:

1) Write an abstract of the AB2/AB3 practical (max. 250 words).
2) How many water molecules were added ?
3) How many sodium and chloride ions have been added ?
4) Based on those numbers, what net charge of the protein can you infer ?
5) What is the the cut-off distance for the calculation of both electrostatic and van der Waals interactions (in Å) ?
6) What is the time increment of the simulation (with correct units) (refer to page 4 if uncertain) ?
7) How many steps are performed in the position constraint simulation ?
8) How often are the coordinates of the system written to the trajectory (.dcd) file (note both the steps and the actual time interval) ?
9) What is the total simulation time specified (position constraint) ?
10) What is the reason that the ions disappear and re-appear at the edges of the box ?
11) After which time do you observe the start of folding ? At which time does the folding speed increase substantially ?
12) What trend do you observe in the number of hydrogen bonds during the "protein folding" process ?
13) Report the average number of hydrogen bonds and the standard deviation (sd) for this number as $n_{hb} \pm$ sd. Take care to report the correct number of significant figures for $n_{hb.}$
14) How many salt bridges were identified in your simulation trajectory ?

Conventions for this manual:

Text providing background information is indicated with this symbol:

Questions, which you should answer in your logbook are indicated like this:

Menu commands are shown in italics

`Commands typed at the command prompt are shown like this`

`Input files are shown in boldface`.

1. Introduction

Molecular dynamics (MD) simulations are a research method in computational biochemistry that yields the thermal fluctations of atoms in a molecule as well as the relative postions of molecules and atoms in dependence of time. Advances in the price/performance ratio of computer hardware has moved biomolecular simulations from the realm of supercomputers to desktop workstations. At the same time the accuracy of simulations compared to experimental observations has increased due to more accurate forcefields (see below), accelerated sampling methods and the use of explicit solvent in simulations. Biomolecular simulations are used to gain insight into ligand binding, enzymatic activities, signalling mechanisms and protein folding (Dodson et al, 2008). Additionally simulations are valuable tools for the refinement of electron microscopy, x-ray, NMR or other spectroscopic data in order to obtain more accurate molecular structures.

The ingredients of MD simulations are the atomic coordinates of the molecule, usually in the format of a pdb-file, and a definition of the atom types, bonds and angles between atoms as well as the number of molecules in the simulation system. This definition is usually called the topology. The interactions between the atoms in large biomolecules are treated by the principles of classical mechanics, i.e the molecule is represented in the computer as a set of spheres connected by springs. The potential energy V is given by classical physics-based equations, such as:

$$V = \sum_{bonds} k_i^{bond}(r_i - r_{i,0})^2 + \sum_{bondangles} k_i^{angle}(\alpha_i - \alpha_{i,0})^2 + \sum_{torsionangles} k_i^{torsion}(1 - \cos(n_i\phi_i - \phi_{i,0})) + \sum_{pairs-ij} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + \sum_{pairs-ij} \frac{1}{4\pi\varepsilon_r\varepsilon_0}\frac{Q_iQ_j}{r_{ij}} \quad (1)$$

Each summation term from left to right describes the bonds, bond angles, torsion angles, von der Waals interactions and electrostatic interactions as depicted graphically in figure 1 below.

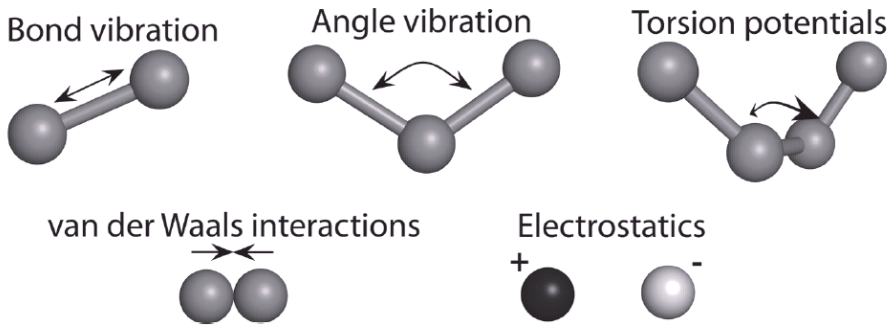


Figure 1: Interactions between atoms included in the forcefield (Lindahl, 2015)

The form of the equation like equation (**1**) and the parameters k^{bond}, k^{angle}, $k^{torsion}$, A, B are called the force field of the simulation. It is this force field and the way in which van der Waals and electrostatic interactions are treated that determines the accuracy of the simulation. The most widely used forcefields are AMBER, CHARMM, OPLS and GROMOS. The GROMOS forcefield differs from the rest in the way hydrogen atoms are treated. In the GROMOS forcefield non-polar hydrogen atoms are subsumed into their adjacent carbon atom, e.g. a three particle H-C-H group becomes a single CH2 particle. This type of forcefield is called a united-atom forcefield and is computationally more efficient that all-atom force fields, because the number of atoms is reduced with a small sacrifice in terms of accuracy. In this practical you will use an all-atom forcefield of the CHARMM variant, namely CHARMM36 (version Feb. 2012 as distributed with VMD version 1.93).

Once the molecular system is accurately described with coordinates, topology and force field, the actual simulation consists of a large number of time steps:

```
Loop
    For each particle: calculate force
    For each particle: update coordinates
    Increment time
Until maximum number of timesteps is reached.
```

The time is usually incremented in steps of one to three fs (femtoseconds). That means for simulations of a reasonable amount of time we need to carry out a large number of time steps. Simulation of proteins *in vaccuo* have been performed for 100 ps (picoseconds) on a Cray X-MP supercomputer (Karplus & Petsko, 1990), but in this practical you will attempt to go further with a 10,000 ps simulation of a small protein *in water*.

The use of solvated systems in MD simulation requires the introduction of another concept: periodic boundary conditions. When a protein molecule is enclosed in a box of water

molecules, the water molecules at the box edges would experience vacuum as shown in figure 2a. In order to avoid this problem, the box is repeated indefintely in space as shown in figure 2b. Obviously the box needs to be big enough so that the protein cannot see its mirror image in a neighboring box.

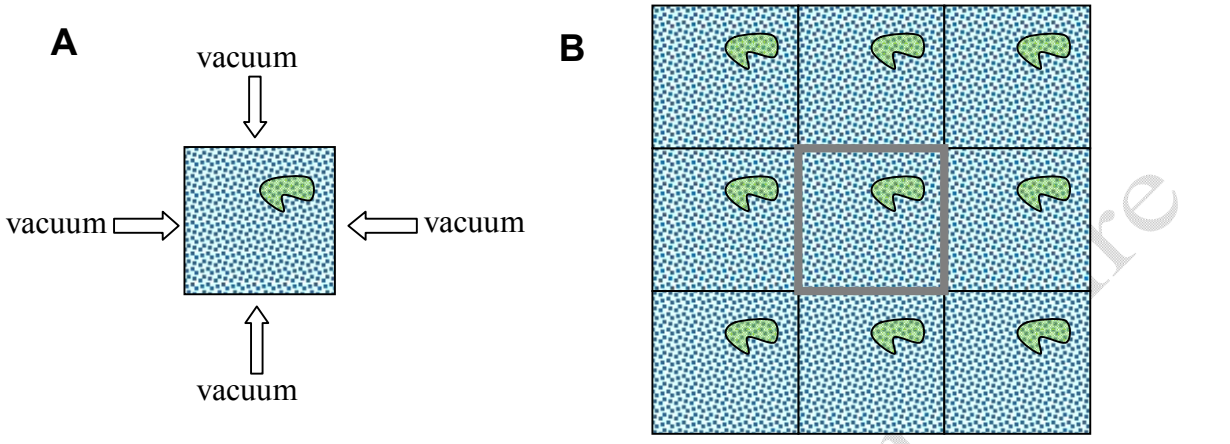


Figure 2: a) a water box in vaccum, b) with periodic boundary conditions

Another procedure encountered in molecular simulations is energy minimisation (EM). In EM the coordinates of the system are varied in order to find a minimum of the potential energy V or mathematically: find the coordinates x_i so that the first derivative of the potential energy becomes zero:

$$\frac{\partial V}{\partial x_i} = 0 \qquad (2)$$

This is similar to school maths, when you were asked to find the minimum or maximum of a function V, e.g. $V = (x-3)^2$ with respect to the x-coordinate. For biomolecules the potential energy function is much more complicated and there are thousands of coordinates, thus the computer applies a numerical algorithm called 'steepest descent'. A common issue is that most EM algorithms are unable to find a *global* minimum but only a *local* minimum as illustrated in figure 3. MD simulations followed by EM are better suited to find the global energy minimum, but EM is required before you can start an MD simulation.

In this practical you will investigate the ribosomal protein L7/L12. It is a 12 kDa protein that is part of the so-called stalk of the large ribosomal subunit. The protein L7/L12 consists of two domains, namely the N-terminal domain responsible for achoring the protein to the ribosome and the C-terminal domain that is involved in interaction with the translation

factor. We will focus here on the C-terminal domain, the structure of which is known from nuclear magnetic resonance (NMR) spectroscopy (PDB identifier: 1RQS).

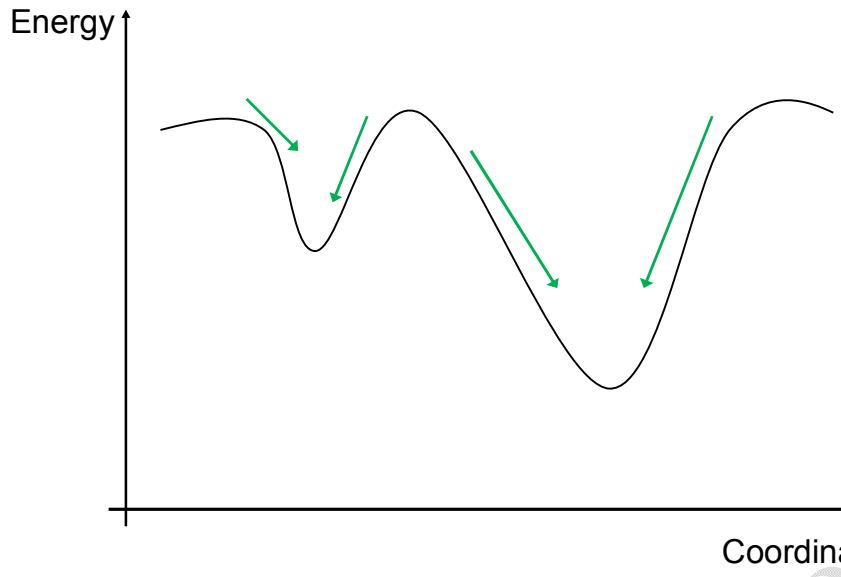


Figure 3: A schematic plot of energy against a coordinate. Energy minimisation follows the green arrows, thus dependent on the starting position it is not always possible to reach the global energy minimum.

Protein folding is a slow process that can take any time from microseconds to minutes, which is beyond the scope of MD simulations. Therefore we apply a trick and simulate protein unfolding at a high temperature, which can be seen as the reverse of protein folding (Day & Daggett, 2007).

2 Materials and Methods

- The protein structure of the C-terminal domain of the ribosomal protein L7/L12 as determined by NMR-spectroscopy (Bocharov et al, 2004).
- The VMD graphical user interface (Humphrey et al, 1996).
- STRIDE secondary structure prediction (as distributed with VMD) (Frishman & Argos, 1995).
- The molecular dynamics simulation software NAMD, which was developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign (Phillips et al, 2005).

3 Set-up of the simulation

For the simulation you will use the NAMD software. You are already familiar with VMD, which provides all the necessary tools for setting-up, running and analysing a NAMD simulation. The tasks required are summarised in the flow-chart shown in figure 4. It starts with a pdb-file and ends with the execution of a NAMD simulation.

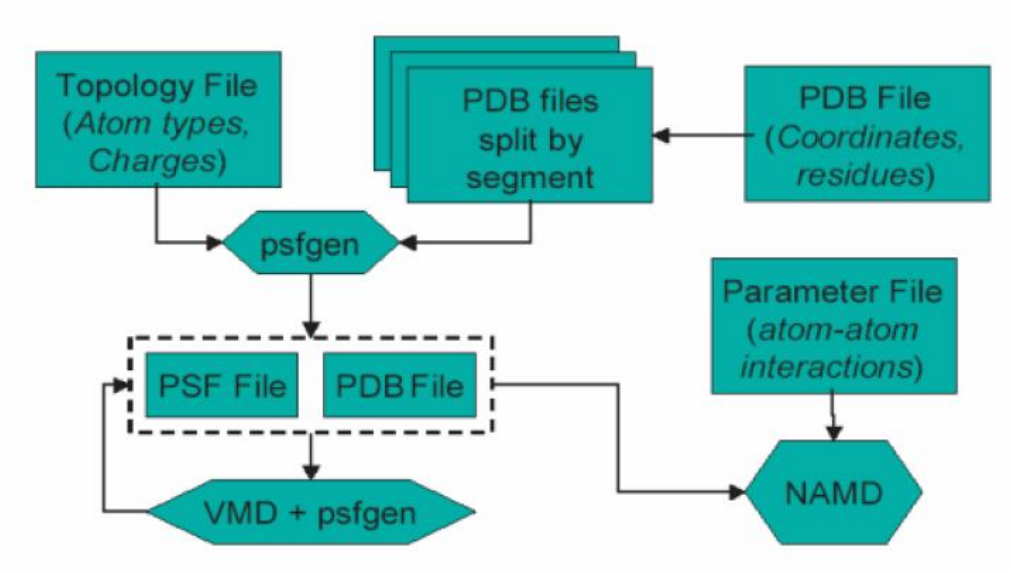


Figure 4: The steps required to perform a simulation with NAMD starting from a pdb-file. In your case the pdb-file has only one chain/segment. This image was taken from the NAMD tutorial (Phillips et al, 2012).

All steps can be carried out with the VMD graphical user interface, except that the NAMD simulation will be started from the command line. The parameter files for various simulation tasks are provided on StudyNet. You will need a text editor, such as NotePad to view and modify these files.

3.1 Generate a Protein Structure File (psf-file)

The structure of the C-terminal domain of the ribosomal protein L7/L12 has been obtained by solution-state NMR spectroscopy. Obtain the pdb-file from the RCSB Protein Databank; the structure has the pdb-ID 1RQS. Once you have identified the database record for 1RQS you need to click on `Download – PDB Format` and save the file 1rqs.pdb into the folder, where you will carry out the simulation. It is best to create a new folder (for example 'ab2-3') on the U-drive, so the files do not get deleted when you log out.

The file 1rqs.pdb contains 20 individual structures as it is often the case with NMR-determined stuctures. These are 20 slightly different conformers that all satisfy the structural restraints derived from the NMR experiment. Copy the ATOM records of the best

representative conformer (model 9) into a new text file and save it as `riboprotein.pdb`.

The pdb-file contains the atom coordinates, but we need to tell the computer, what type of atoms are present in the pdb-file, if they carry any charge, how the atoms are connected and if there is free or restricted rotation around bonds. All this information is captured in the psf-file and for proteins it can be generated automatically by pfsgen.

Start VMD and open the 'TK console' window (Menu *Extensions – TK console*). In the TK console window go to the folder you want to work in (where you saved riboprotein.pdb), for example:

```
% cd u:/ab2-3
```

Note that the forward slash is used to separate folders (this is a LINUX convention, Windows uses '\'). You may check the active folder with `'pwd'`.

Load riboprotein.pdb into VMD via the menu *File -New Molecule*. Then open the automatic psf-builder: *Extensions – Modelling – Automatic PSF Builder*. Change the output basename to 'rbp', click on 'Load input files', then 'Guess and split chains'. The picture on screen after the last click is shown in figure 5. Then click on 'Create Chains'. There are many warnings displayed in the console window, but you may check that the files 'rbp_formatted_autopsf.psf' and 'rbp_formatted_autopsf.pdb' have indeed been generated. The next steps are required to place the protein into the centre of a water box.

3.2 Solvate the system

Close the 'AutoPSF' window and delete all molecules from the 'VMD Main' window, by selecting each entry in the list and using the menu *Molecule – Delete Molecule*. Then load first rbp_formatted_autopsf.psf and 'rbp_formatted_autopsf.pdb' **into** the psf (make sure 'Load files for rbp_formatted_autopsf.psf is selected). Now we need to place the protein in a box of water. To start with, the protein molecule must be placed at the centre of the coordinate system; this is done form the 'VMD TK Console' window (note the American spelling of 'center' vs English 'centre'):

```
% set all [atomselect top all]
% measure center $all  (this outputs the current x, y, z coordinates of the centre)
% $all moveby [vecinvert [measure center $all]]
% measure center $all  (this checks, if the move was successful)
```

As you can see from the last command, the coordinates of the centre are close to zero, thus the protein has been centred. These commands are part of the Tcl/TK programming language.

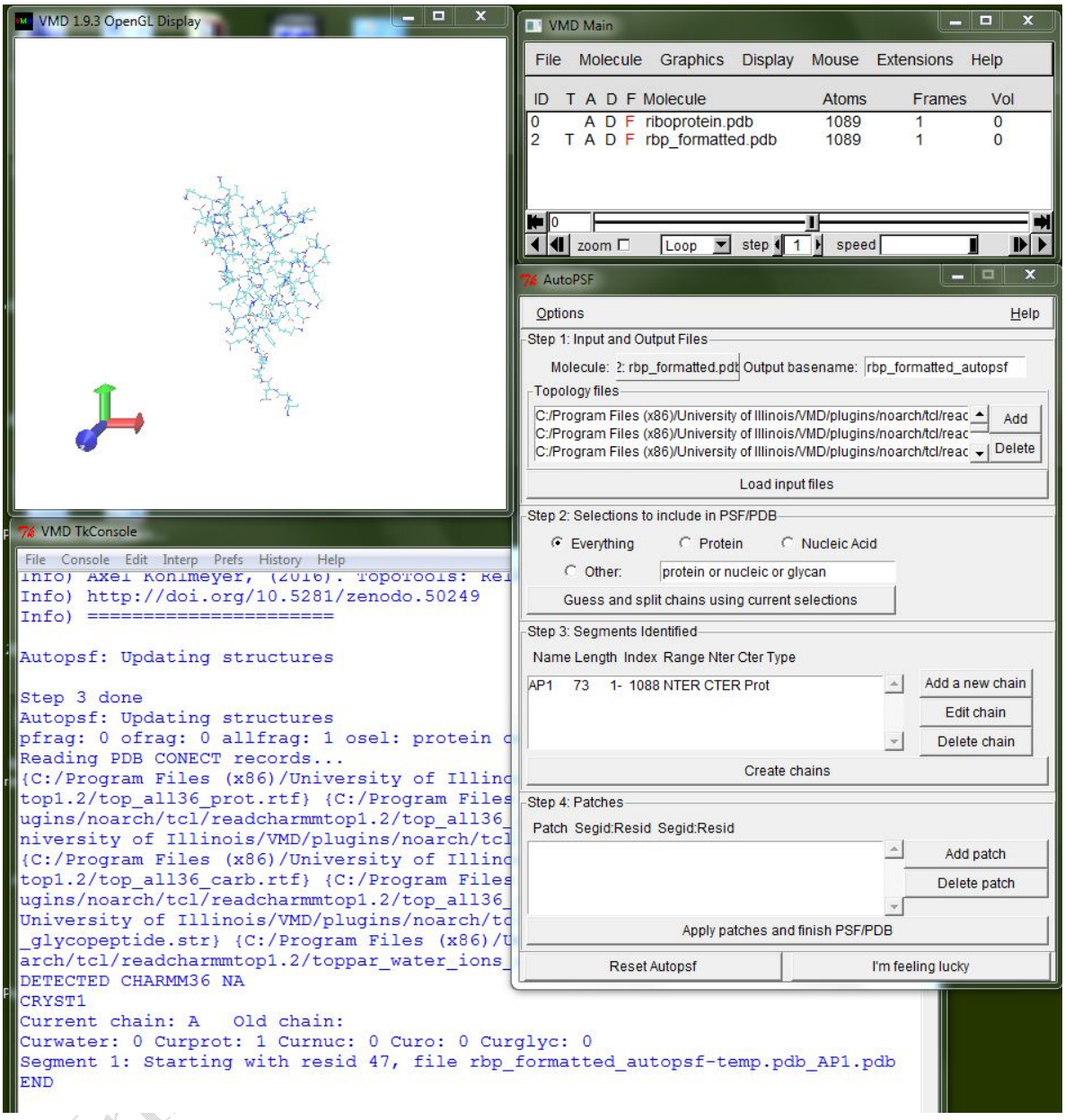


Figure 5: The screenshot after performing all steps including 'Guess and split chains using current selections'.

We are now ready to solvate the protein using the solvation box tool of VMD. Menu *Extensions – Modelling – Add Solvation Box* brings up a new window. In this window change the output to 'rbp_water' and all 'Box Padding' values to 10 Å. This value specifies the distance between the protein and the edge of the box; as a size of a water molecule is approximately 3 Å there will be about three layers of water molecules between the protein and the box edge. Finally click on solvate. How many water molecules were added ? The

following Tcl/Tk commands may help you to answer this question. The output is the number of atoms belonging to water molecules:

```
% set wat [atomselect top water]
% $wat num
```

3.3 Neutralise the system

Proteins carry charges and the net charge of a protein is the sum of negative and positive charges contributed by the side-chains of amino acid residues (Asp, Glu, Arg, Lys, His) and the N- and C-terminus. The net charge may not be zero for the riboprotein, but any system of molecules in a test tube has a zero net charge. Leaving a charge in your simulation system would create unrealistic conditions, thus any charge in your system must be neutralised by adding counterions. At the same time we will take the opportunity to set a realistic salt concentration of 0.1 M NaCl. This step is performed with the add ions tool: *Extensions – Modelling – Add Ions*. In this window set the 'Ion placement mode' to 'Neutralise and set the NaCl concentration to' 0.10 mol/L as shown in figure 6.

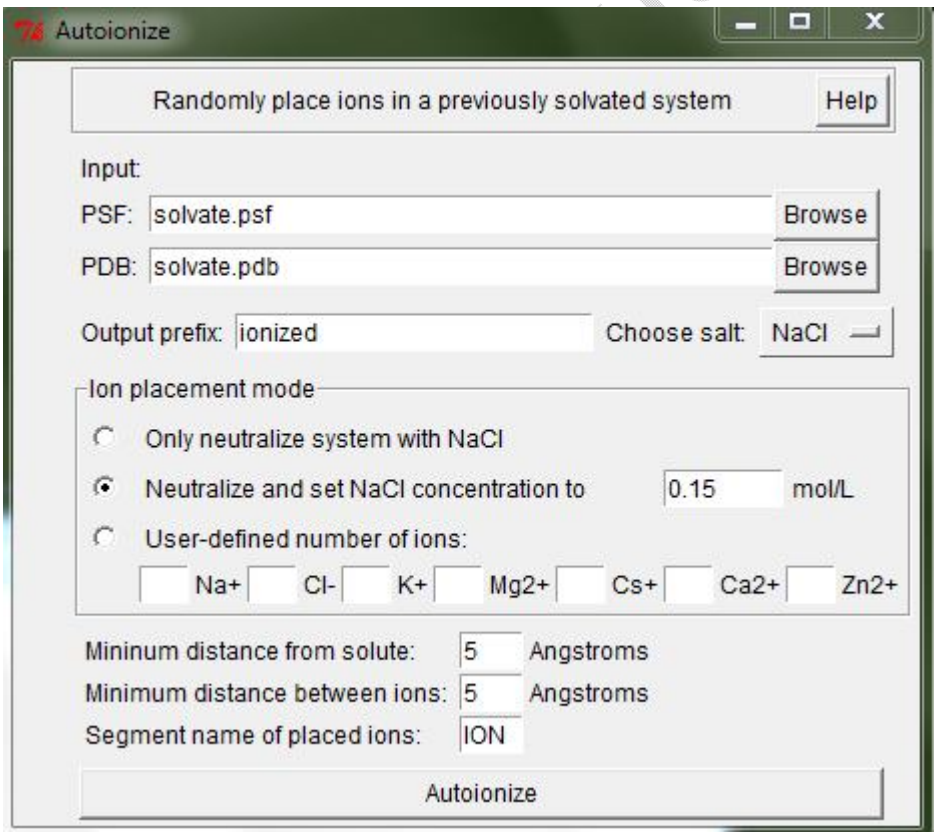


Figure 6: The settings of the Autoionize tool.

? Finally click on 'Autoionize'. How many sodium and chloride ions have been added ? You could check those numbers using Tcl/TK commands. Alternatively, you may use the

Windows Explorer, go to your working folder and open ionized.pdb in Notepad. At the end of the file you will find the sodium and chloride ions that have been added. Based on those numbers, what net charge of the protein can you infer ?

4 Energy minimisation and molecular dynamics simulation

Before the start of the actual unfolding simulation over 10,000 ps, also called the 'production run', we must perform a number of preparatory simulation steps starting with an energy minimisation (EM) followed by equilibration of the water molecules around the protein molecule. Without this equilibration the water molecules would bump into the protein and distort the structure at the start of the simulation. Yet the structure is based on NMR experimental data and should not be distorted artificially. In order to equilibrate the water, we keep all protein atoms fixed in space, while water molecules and ions will be free to move. This is called *position constraint MD*, whereby the postions of the protein atoms are constraint to the positions from the NMR structure. For this purpose you will use NAMD from the Windows command line, while all options that apply to the EM and MD simulation are specified in text files with the extension '.namd'.

4.1 Energy minimisation (EM)

For the first energy minimisation we need to obtain some information about the size of the simulation box and the coordinates of the centre. You need to note this information into your logbook and write it into the namd-file. In VMD load the file ionized.pdb, if not already loaded and type the following commands into the TKConsole:

```
% set all [atomselect top all]
% measure minmax $all
```

The output should look similar to that:

```
{-23.841999053955078 -37.52799987792969 -21.398000717163086}
{24.74799919128418 27.742000579833984 21.966999053955078}
```

These are the coordinates of the bottom left and top right corner of the simulation box. For these coordinates we can get the dimension (size) of the simulation box along the x-coordinate as: $x_{dim} = 24.75 – (-23.84) = 48.59$ and y_{dim} and z_{dim} in the same way. The centre of the simulation box is obtained with:

```
% measure center $all
```

```
0.5538268685340881 -4.7834625244140625 0.3787666857242584
```

Note these numbers down as well (to five significant figures). Now you are ready to prepare the file 'minimize.namd' specifying the energy minimisation of the system. Obtain the file from StudyNet and insert the required information as indicated in the file and save the file into your working folder. Then open the Windows Command Prompt, go to your working folder and start the energy minimisation (Windows uses the backward slash '\'):

```
> u:              (for example)
> cd \ab2-3       (for example)
> c:\NAMD_2.12_Win64-multicore\namd2 +p4 minimize.namd > minimize.log
```

(Note that the location of the namd2 program may be different on your computer.)

After less than one minute the minimisation has completed, open the file minimize.log with Notepad to verify that everything went to completion without errors.

4.2 Position constraint MD

For the position constraint MD we must tell NAMD, which atoms of the system should be constrained and to which coordinates they should be constrained. We will constrain all non-hydrogen atoms of the protein to the positions obtained from energy minimisation (the output of the previous step, emin.pdb). In order to tell NAMD the atoms that should be constrained we need to create a pdb-file that contains the number 1.00 in the beta-factor column. In order to do this, rename emin.coor to emin.pdb at the Windows Command Prompt:

```
> rename emin.coor emin.pdb
```

Then restart VMD (or delete all molecules) and open emin.pdb and the TKConsole. In the TKConsole type:

```
% set allprotein [atomselect top protein]
% set fix [atomselect top "protein not hydrogen"]
% $allprotein set beta 0
% $fix set beta 1
% [atomselect top all] writepdb fixsystem.pdb
```

Open fixsystem.pdb with Notepad and verify that the penultimate column contains 1.00 for non-hydrogen protein atoms. You are now ready to run the position constraint MD simulation:

```
> c:\NAMD_2.10_Win64-multicore\namd2 +p4 sim_fixprot.namd > simfix.log
```

While the simulation is running (≈ 1-2 hours) answer the following questions with regards to the position constraint MD simulation in your logbook. The online NAMD user's guide may be helpful (http://www.ks.uiuc.edu/Research/namd/2.9/ug/ug.html).

? What is the value of the cut-off distance for the calculation of both electrostatic and van der Waals interactions (in Å) ?

What is the time increment of the simulation (with correct units) (refer to page 4 if uncertain) ?

How many steps are performed in the position constraint simulation ?

How often are the coordinates of the system written to the trajectory (.dcd) file (note both the steps and the actual time interval) ?

4.3 Free MD simulation (production run)

Once the previous simulation has completed you are ready to run the unconstraint MD simulation at high temperature in order to study the unfolding of the riboprotein. The output of the previous simulation sim_fixprot.pdb is used to to run a long simulation at a temperature of 550K. Save the configuration file sim_free.namd from StudyNet into your working directory and start the simulation as previously described. While the simulation is running you may analyse the content of sim_free.namd. Note that there is no pressure control, thus the simulation is carried out at fixed volumen, the NVT ensemble ('canonical ensemble' in statistical mechanics), in order to avoid the likely evaporation of water at this high temperature.

? What is the total simulation time specified ?

You may notice that only a very short simulation is performed. For the next practical you will be provided with a complete trajectory over 10,000 ps, that would take a couple of days to run on a typical dual core CPU.

(*End of first part*)

5 Data Analysis

For this part you are provided with a 10,000 ps long simulation trajectory sim_free.dcd and the corresponding log-file, simfree.log

5.1 Visual analysis

Start VMD and use the TKConsole to go to your working directory. Open the file emin.pdb. Then use Menu *Display – Orthographic*. With Menu Graphics – Representations you may change to the display to your liking, e.g. as shown in the following figure 7.

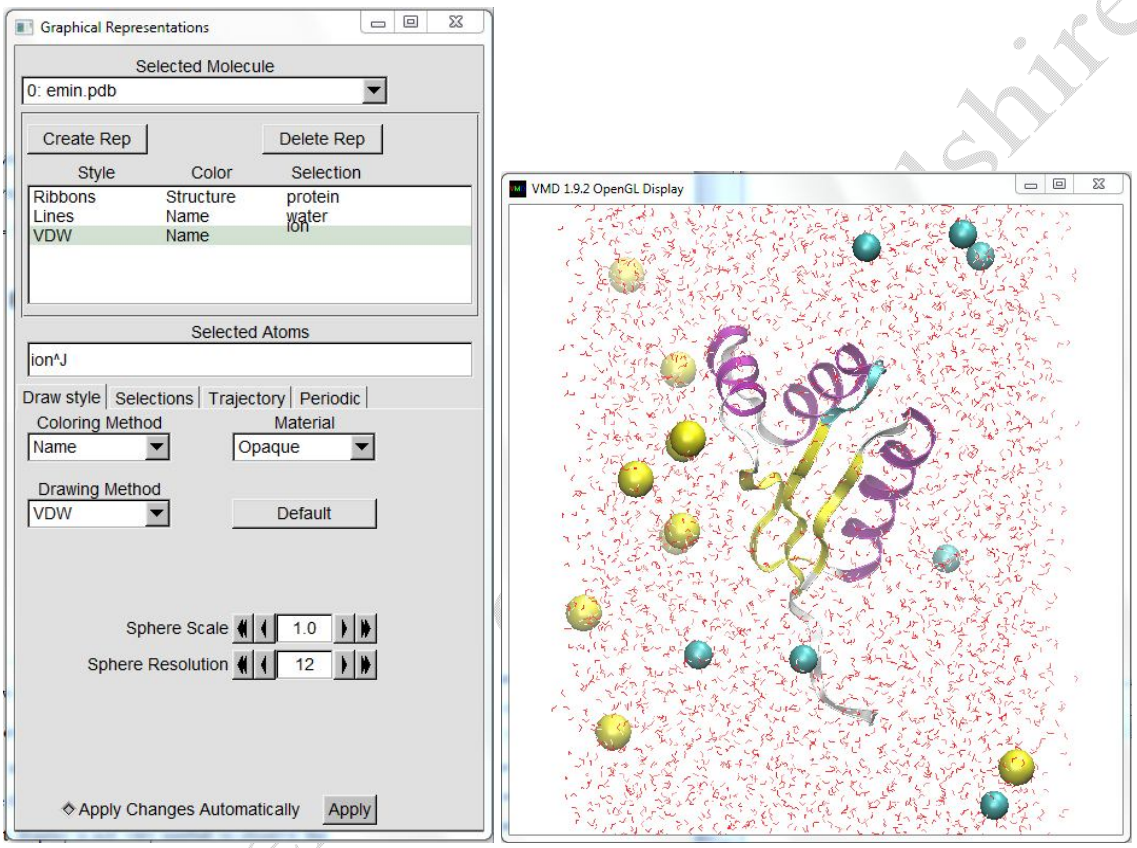


Figure 7: The settings of VMD (left) to creat the view shown on the right.

Then activate the molecule in the main window and select Menu *File – Load data into molecule*. Browse to `sim_fixprot.dcd` and load it into VMD. About 1000 frames are loaded. By using the player controls at the bottom of the 'VMD Main' window you should verify that the position constraint simulation was successful, by observing that only the ions and water molecules change position, while the position of the protein atoms remain constant. What is the reason that the ions disappear and re-appear at the edges of the box ?

Now delete the molecule: *Molecule – Delete Molecule* and load *sim_fixprot.pdb*, the final output of the position constraint simulation. Load into this molecule the long 10,000 ps trajectory, *sim_free.dcd*. This should contain 5000 frames.

Now you are ready to observe the unfolding of the ribosomal protein L7/L12 by playing the trajectory forwards or the folding by playing the trajectory backwards. The current display is not very usefull to observe changes of the protein structure, because the molecule undergoes diffusional and rotational motions. In order to eliminate rotation and diffusion we will align all frames of the trajectory to the starting position. Open *Extensions – Analysis – RMSD Trajectory Tool* and select 'Backbone' under 'Selection Modifiers', then click on 'ALIGN'. After a short wait, close the tool window and play the trajectory again; now we can focus on the internal motions of the protein.

5.2 Quantitative analysis

5.2.1 Root mean square deviation

The root mean square deviation (RMSD) between two structures is a measure of overall structural similarity and calculated as a mass weighted average over all N backbone atoms between molecule 0 and molecule 1:

$$RMSD = \sqrt{\frac{1}{M}\sum_{i=1}^{N} m_i (r_i^0 - r_i^1)^2}$$

where $r_i = (x,y,z)$ is the vector of the coordinates and m_i the mass of each atom. A backbone RMSD around 0.2 nm (2 Å) is indicative of small structural fluctuations, while an RMSD > 0.3 nm (3 Å) indicates a conformational change. Molecule 0 is the first frame of the trajectory (the reference frame), while molecule 1 corresponds to all the sucessive frames in the trajectory.

The RMSD Trajectory tool can be used to plot the RMSD of each frame with reference to the first frame, by checking 'Plot' in the 'Trajectory' section. Keep the 'Backbone' option checked. The RMSD plot is shown in figure 8. This is the default unweighted RMSD.

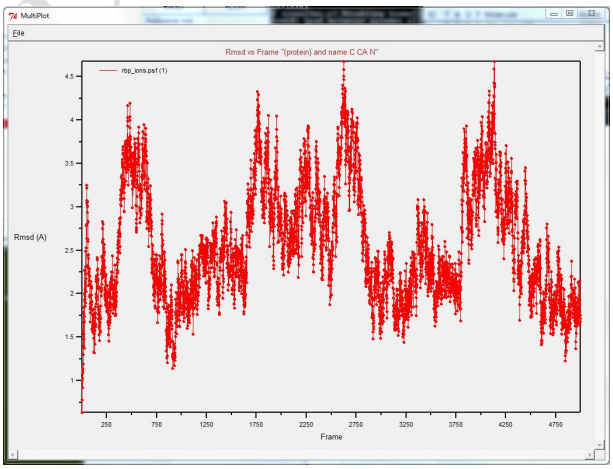
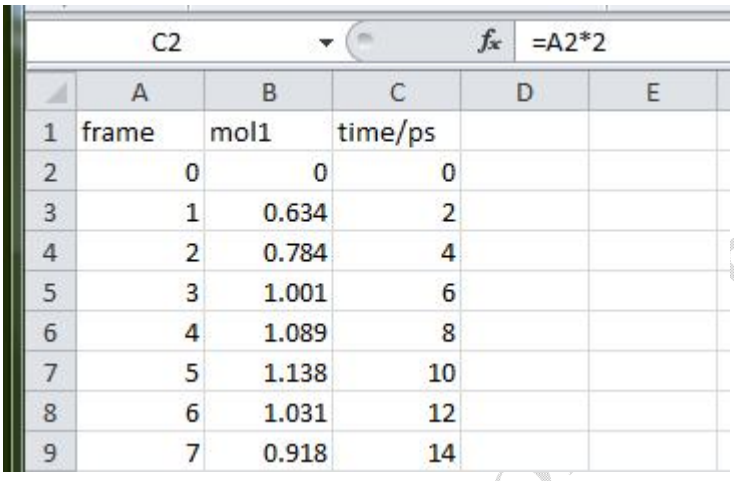


Figure 8: RMSD plot against frame number generated by the RMSD trajectory tool. 5000 frames were generated from a 10000 ps simulation, thus the time difference between two consecutive frames is 2 ps.

Since we want to study the process of protein folding by reversing the trajectory, we will import the RMSD data into a spreadsheet program (e.g. Microsoft Excel, Gnumeric) and reverse the data. Close the plot window, remove the checkmark from 'Plot', place a checkmark next to 'Save' and click on RMSD again. The program has generated a file called 'trajrmsd.dat' in your working folder. In Microsoft Excel open the file `trajrmsd.dat`. Remember to select 'all files' in the file type drop-down box. Automatically the 'Text Import Wizard' comes up; here you select 'Fixed Width' and 'Next – Next – Finish'. Replace NA by zero and then you need to generate a proper time by taking into account that frames occur at an interval of 2 ps/frame (= 10000 ps/5000 frames) as shown in figure 9.



	A	B	C	D	E
1	frame	mol1	time/ps		
2	0	0	0		
3	1	0.634	2		
4	2	0.784	4		
5	3	1.001	6		
6	4	1.089	8		
7	5	1.138	10		
8	6	1.031	12		
9	7	0.918	14		

Figure 9: The imported RMSD data and the conversion of frame number into time in column C

Then overwrite the content of column A with column C, copy column C followed by 'Paste Values' into column A.

Because we want to analyse 'protein folding' we apply a trick and reverse the data in column B. To do this copy column A into column C. Then select columns B and C and use the *Data – Sort* function choosing 'Sort by Column C' and order 'Largest to Smallest'.

In the next step create a plot of the RMSD in dependence of time, by selecting columns A and B and creating a 'Scatter Plot' of the type 'Scatter with Straight Lines". An example is shown in figure 10. In this particluar example, the RMSD decreases from 8 to 2 Å and then rapidly to 0 Å. The results for your simulation trajectory may be different, as each simulation is a 'single molecule' experiment that is subject to large stochastic variations.

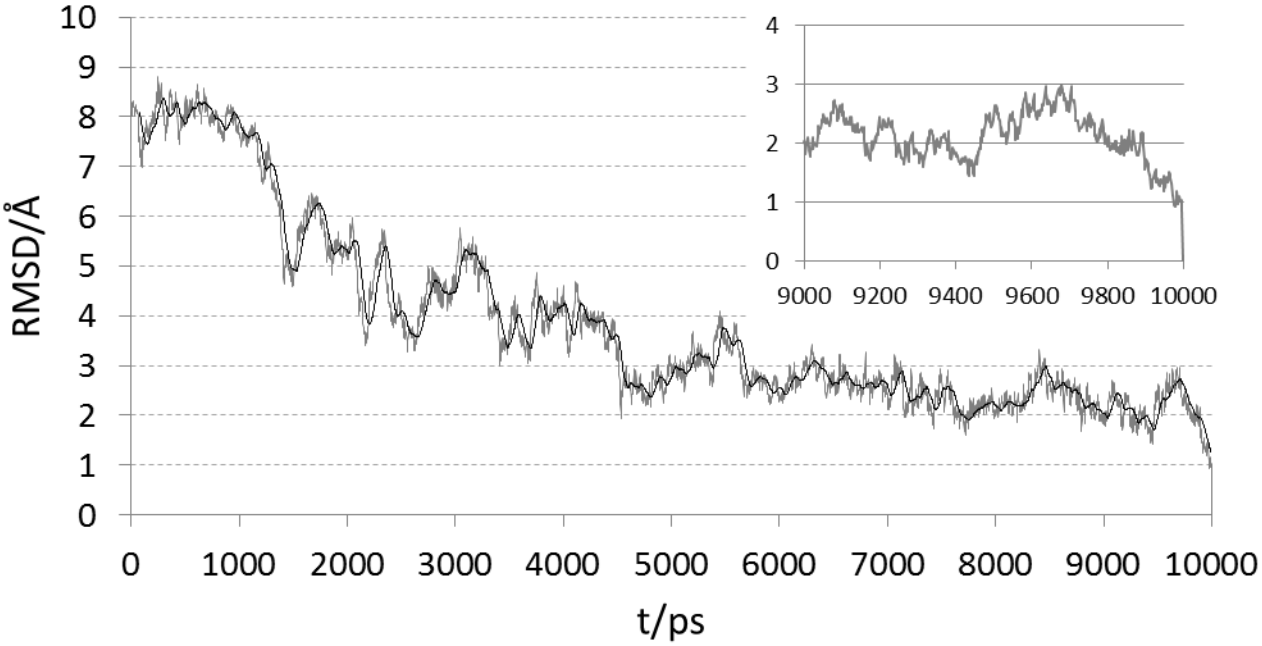


Figure 10: Backbone RMSD of the "folding" trajectory of the C-terminal domain of the L7/L12 ribosomal protein. The dark line is a moving average over a window of 30 data points. The inset shows the last 1000 ps of the 'folding trajectory.

? After which time do you observe the start of folding ? At which time does the folding speed increase substantially ? You may go back to VMD and play the trajectory in order to visually confirm the structural changes observed in the RMSD plot.

5.2.2 Formation of hydrogen bonds

In order to calculate the number of hydrogen bonds during the trajectory, use *Extensions – Analysis – Hydrogen bonds*.

Change 'Donor-Acceptor distance' to 3.5 Å, 'Angle cutoff' to 30° and click on 'Find hydrogen bonds'. After some time, a plot similar to the one in figure 11 is generated.

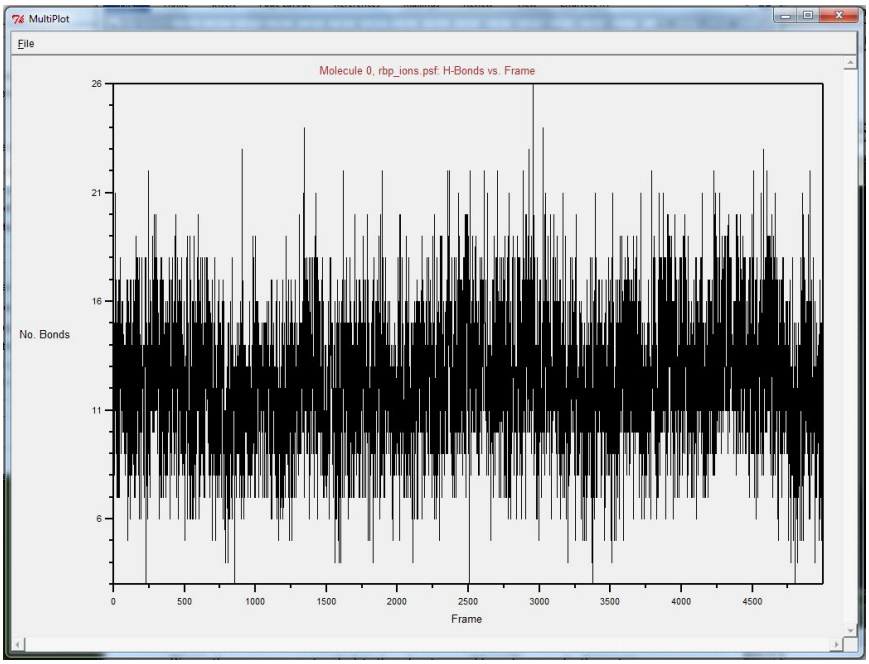


Figure 11: The number of hydrogen bonds plotted against frame number.

Export the data from the Multiplot window by selecting *File – Export to ASCII vectors*. Then import that data into as 'space delimited' ino Excel and reverse the time. What trend do you observe in the number of hydrogen bonds during the "protein folding" process ? Report the average number of hydrogen bonds and the standard deviation (sd) for this number as $n_{hb} \pm$ sd. Take care to report the correct number of significant figures for n_{hb}; the standard deviation will inform you what is a reasonable number of significant figures. Standard deviations are usually reported with one or two significant figures.
Example: 14.123 ± 1.2 is incorrect, 14.1 ± 1.2 or 14 ± 1 are correct.

5.2.3 Secondary structure

With the tool *Extensions – Timeline* we can investigate changes of secondary structure during the simulation. **Since the calculation takes a long time (30+ mins), it is recommended to plan a break at this point.**

From the tool window select *Calculate – Calc. Sec. Struct.*

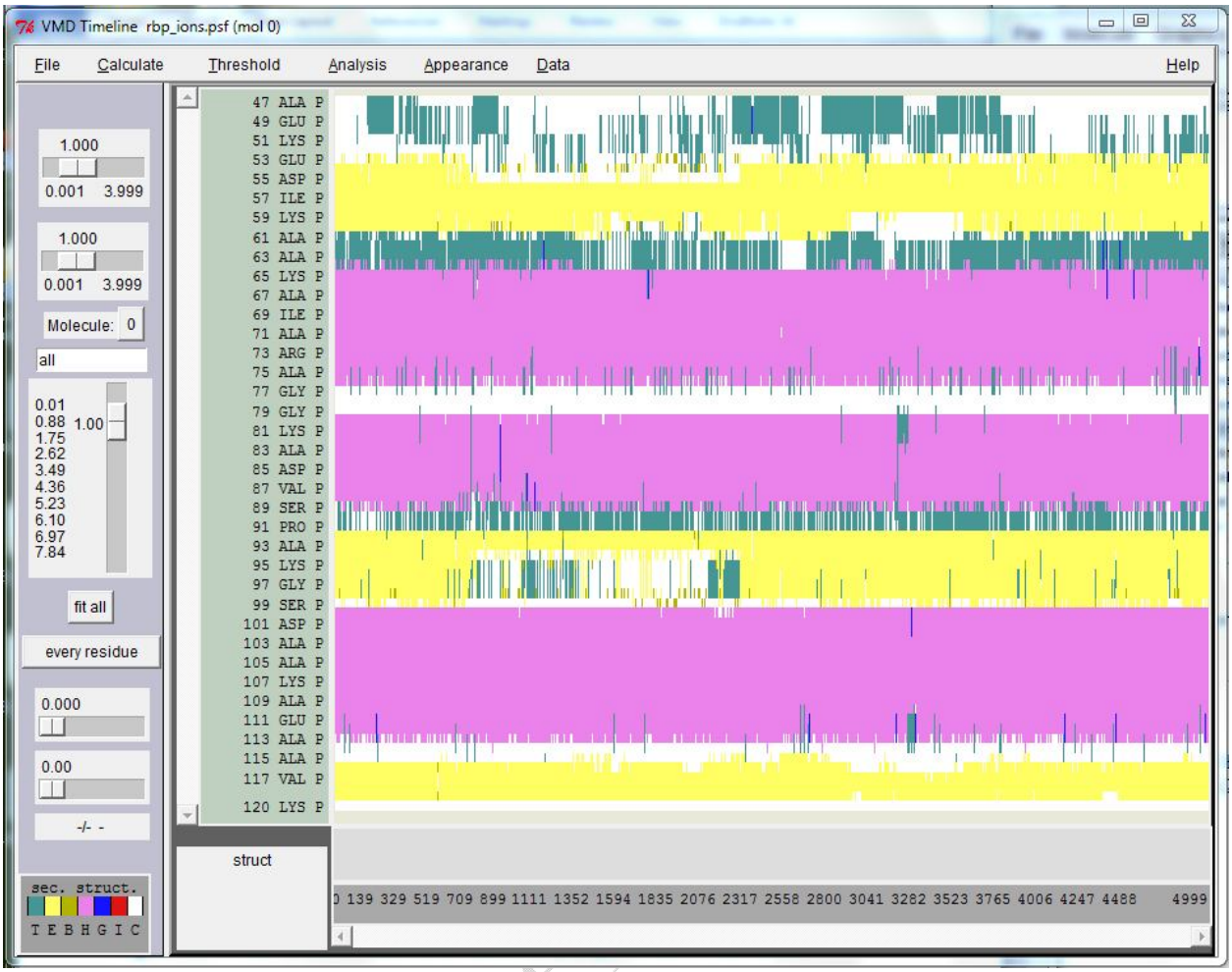


Figure 12: The secondary structure of each residue plotted against the frame number (T: turn, E: β-strand, B: isolated bridge, H: α-helix, G: 3-10 helix, I: π-helix, C: coil).

From the example shown in figure 12, you can see that the three α-helices remain fairly stable during the simulation, while the N-terminal β-strand secondary structure centred at Ile57 is reduced due to the formation of turns and coils in particular around frame 1500 (3000 ps, time refers to 'unfolding'). The second β-strand centred at Lys95 is even more substantially reduced around the same time. The C-terminal β-strand experiences some shortening around frame 3200 (6400 ps). The results from your own simulation trajectory may be completely different.

By clicking with mouse button on the frame number at the bottom, you show the corresponding point of the trajectory in the VMD graphical display window. By clicking on the sequence bar at the left, you can highlight the amino acid in red in the graphical display. The update of displays may take a while with the message 'VMD not responding', but eventually the display will update.

5.2.4 Salt bridges

With the *Extensions – Analysis – Salt Bridges* we can identify salt bridges as well as obtain for each salt bridge the distance over time.

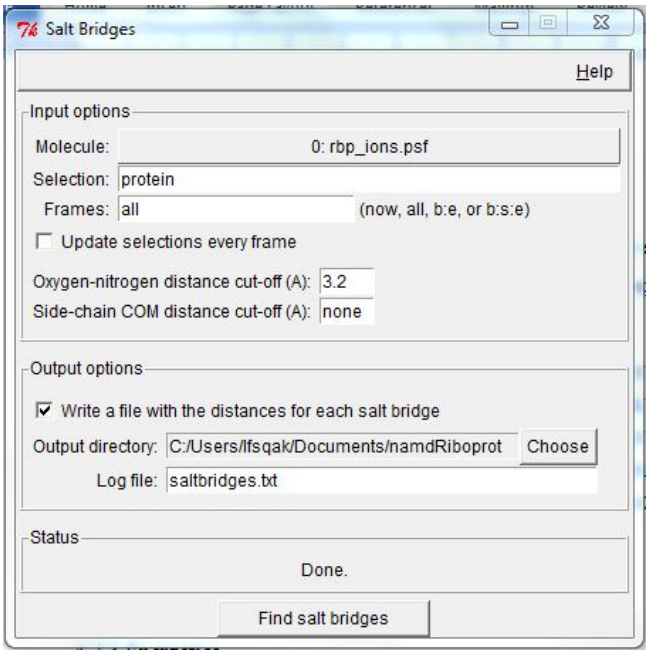


Figure 13: The settings of Salt Bridges tool. Note the settings in 'Output Options'; additionally 'Update selections every frame' was unchecked.

By setting the output options as shown in figure 13, the following salt bridges were identified:

GLU96-LYS95, ASP102-LYS95, GLU116-LYS107, ASP85-LYS95, GLU111-LYS108, GLU104-LYS108, GLU50-LYS100, GLU112-LYS70, GLU53-LYS51, ASP101-LYS100, GLU82-LYS81, GLU49-LYS100, GLU104-LYS100, GLU88-LYS65, ASP55-LYS120, ASP85-LYS84, GLU104-LYS107, GLU118-LYS120, GLU111-LYS107, GLU49-LYS51, ASP55-LYS51, GLU82-LYS95, GLU118-LYS59, GLU88-LYS84, GLU53-LYS120, GLU96-LYS120, ASP85-LYS81, GLU116-LYS59, GLU112-LYS108, GLU50-LYS51

Note that if at any point in the trajectory the distance was equal to or lower than the threshold of 3.2 Å, the residue pair would have been identified as a salt bridge. For the salt bridge Glu112-Lys70 the distance is plotted against 'unfolding' time in figure 14.

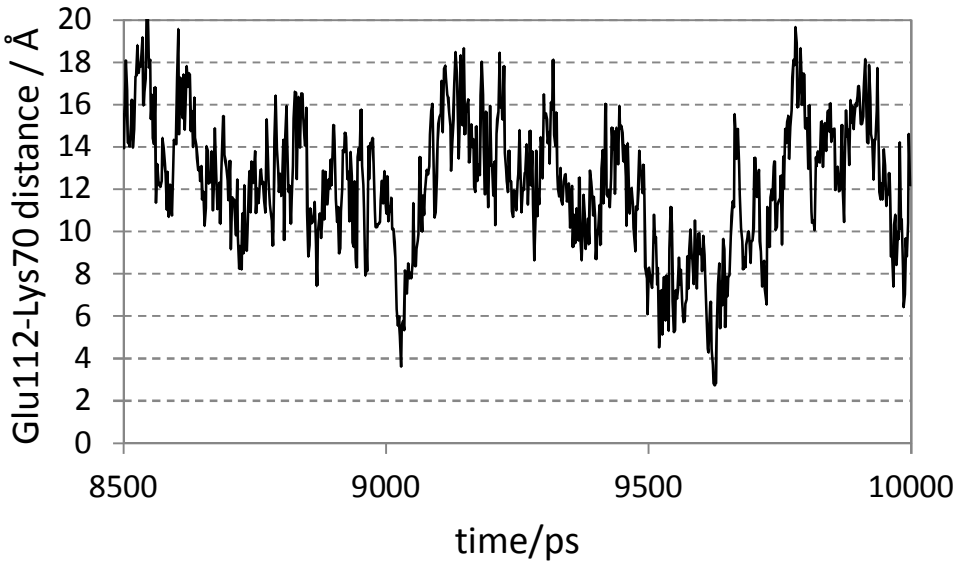


Figure 14: Distance between Glu112 and Lys17 plotted against time for a extract of the 'unfolding' trajectory.

? How many salt bridges were identified in your simulation trajectory ?

5.2.5 Energetics

In the following analysis you will plot the van der Waals and electrostatic internal energy of the protein in dependence of time. This analysis is performed by the NAMDEnergy tool: *Extensions – Analysis – NAMD Energy* with the settings shown in figure 15. Note that the parameter files will be inserted automatically.

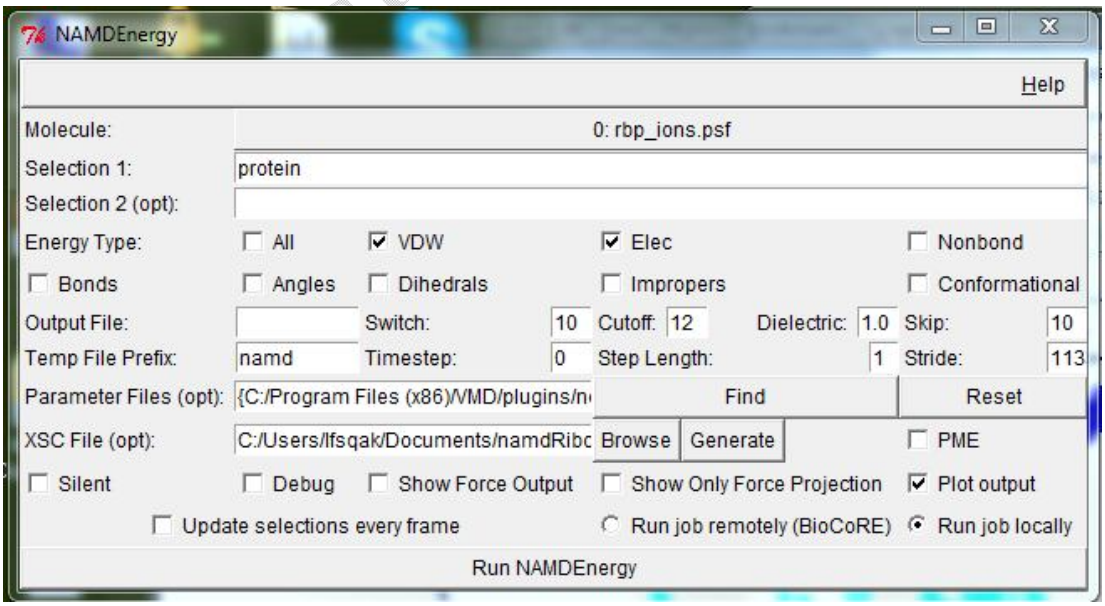


Figure 15: Calculation of protein van der Waals and electrostatic internal energy during protein folding as a function of simulation time.

From the resulting Multiplot window use *File – Export to ASCII matrix*. Then import the generated file multiplot.dat into Excel. The first column denotes the trajectory frame followed by total, electrostatic and van der Waals energy terms in kcal/mol. Reverse the time axis as described previously to represent the "folding" process and plot the electrostatic and van der Waals internal energy against time as shown for the example in figure 16.

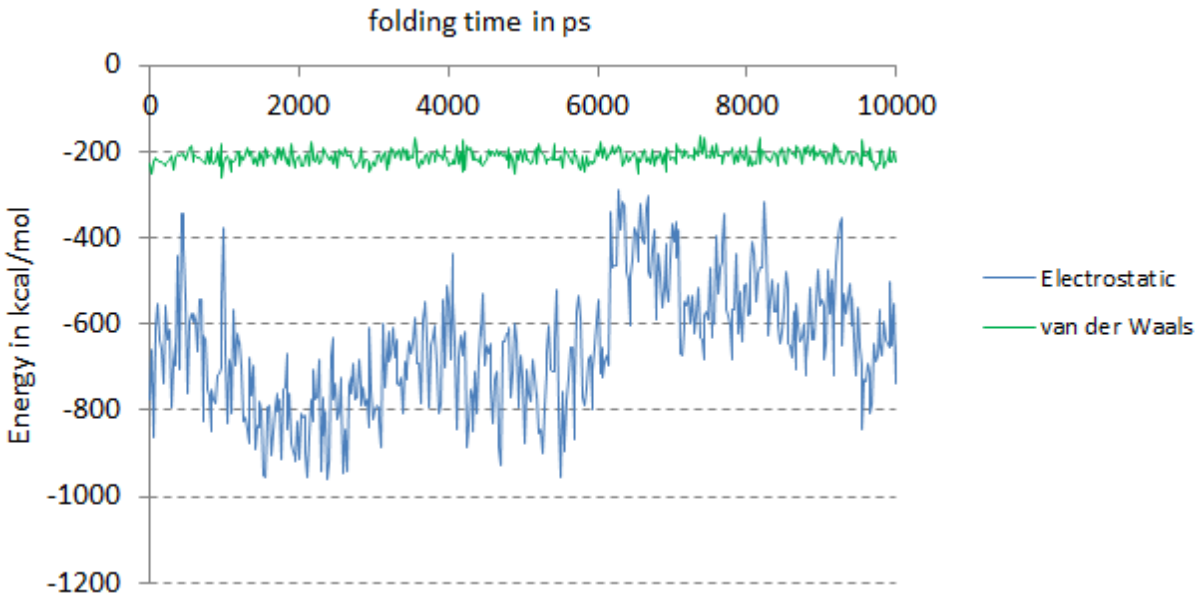


Figure 16: Protein internal energy terms plotted against folding time.

References

Bocharov EV, Sobol AG, Pavlov KV, Korzhnev DM, Jaravine VA, Gudkov AT, Arseniev AS (2004) From structure and dynamics of protein L7/L12 to molecular switching in ribosome. *Journal of Biological Chemistry* **279:** 17697-17706

Day R, Daggett V (2007) Direct observation of microscopic reversibility in single-molecule protein folding. *J Mol Biol* **366:** 677-686

Dodson GG, Lane DP, Verma CS (2008) Molecular simulations of protein dynamics: new windows on mechanisms in biology. *EMBO reports* **9:** 144-150

Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. *Proteins-Structure Function and Genetics* **23:** 566-579

Humphrey W, Dalke A, Schulten K (1996) VMD: Visual molecular dynamics. *Journal of molecular graphics & modelling* **14:** 33-38

Karplus M, Petsko GA (1990) Molecular dynamics simulations in biology. *Nature* **347:** 631-639

Lindahl E (2015) Molecular dynamics simulations. *Methods in molecular biology (Clifton, NJ* **1215:** 3-26

Phillips J, Isgro T, Sotomayor M, Villa E, Yu H, Tanner D, Liu Y. (2012) NAMD tutorial. University of Illinois at Urbana-Champaign.

Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K (2005) Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry* **26:** 1781-1802

Sansom C (2010) Model Molecules. *Chemistry World* **April:** 50-53