Molecular Dynamics Simulations of BPTI in Vacuum

MSc in Bioinformatics and Medical Informatics

November 27, 2014

Fall Semester 2014-2015

Objective : The main objective of this practical is to provide an overview of classical Molecular Dynamics (MD) simulations and Normal Mode Analysis (NMA) by examining the protein called bovine pancreatic trypsin inhibitor (BPTI) within the framework of the CHARMM program.

1 Introduction

One of the principal tools in the theoretical study of biomolecules is the method of MD. It is a computational method which calculates the time dependent behaviour of a molecular system. MD methods are used to describe a complex molecular system in terms of a realistic atomic model, with an aim to understand and predict macroscopic properties based on detailed knowledge on an atomic scale.

MD simulations solve Newton's equations of motion for a system of N interacting atoms:

$$m_i \frac{\partial^2 r_i}{\partial t^2} = F_i, \quad i = 1, \ldots, N. \tag{1}$$

The forces are the negative derivatives of the potential function $V(r_1, r_2, r_3, \ldots, r_N)$:

$$F_i = -\frac{\partial V}{\partial r_i}. \tag{2}$$

The equations are solved simultaneously in small time steps. The system is followed for sometime, taking care that the temperature and pressure remain at the required values and the coordinates are written to an output file at regular intervals. The coordinates as a function of time represent a trajectory of the system. After initial changes, the system will usually reach an equillibrium state. By averaging over an equilibrium trajectory many macroscopic properties can be extracted from an output file.

2 Potential energy function

Theoretical studies of biological molecules permit the study of the relationships between structure, function and dynamics at the atomic level. Since biological systems involve

many atoms, quantum mechanical treatment of these atoms is not feasible. The usual way to solve them is to use empirical potential energy functions which are computationally less expensive, but involve numerous approximations leading to certain limitations. Current generation force fields provide a reasonable good compromise between accuracy and computational efficiency. They are often calibrated to experimental results and quantum mechanical calculations of small model compounds. Among the most commonly used potential energy functions are the AMBER, CHARMM, GROMOS and OPLS/AMBER force fields. One of the most important limitation of the empirical force fields is that no drastic changes in the electronic structure are allowed. i.e. no events like bond making or breaking can be modeled.

3 Setting up and running molecular dynamics simulations

To begin a molecular dynamics simulation, you must choose an initial configuration of the system, a starting point, or t = 0. Most often, in simulations of biomolecules, an X-ray crystal structure or an NMR structure is obtained from the Protein data bank (http://www.rcsb.org) and used as the initial structure. It is also possible to use a theoretical structure developed by homology modelling. The choice of the initial configuration must be done carefully as this can influence the quality of the simulation.

The MD simulations involves 4 main steps and these will be discussed below:

3.1 Energy minimization

The potential energy function of a biomolecular system is a very complex and multidimensional landscape. It has one deepest point, the global minimum and a very large number of local minima. The goal of the energy minimization is to find a local minimum. The energy at this local minimum may be much higher than the energy of the global minimum. Performing an energy minimization will guarantee the removal of any strong van der Waals interations that may exist which might otherwise lead to local structural distortion and result in an unstable simulation.

3.2 Heating

The initial velocities at low temperature are assigned to each atom of the system and Newton's equations of motion are integrated to propagate the system in time. During the heating phase, initial velocities are assigned at a low temperature and the simulation is started with periodically assigning new velocities at a slightly higher temperature and letting the simulation continue. This step is repeated until the desired temperature is reached.

3.3 Equilibration

Once the heating process is over and the desired temperature is reached, the simulation is continued and during this phase, properties such as structure, pressure, temperature and the energy are monitored. The point of the equilibration phase is to run the simulation until these properties become stable with respect to time. If in the process, the temperature increases or decreases significantly, the velocities are scaled such that the temperature returns to its desired value.

3.4 Production and Anlaysis

The final step of the simulation is the production phase, wherein the system is simulated for the time length required from several hundred ps to ns or more. During this process coordinates of the system at different times are stored in the form of trajectories. These are then used for calculations of mean energy, root mean square (RMS) fluctuations between structures etc. From MD simulations, time dependent properties such as correlation functions can also be calculated and these in turn can be related to spectroscopic measurements.

4 Normal mode analysis

Normal Mode Analysis (NMA) is a classical technique for studying the vibrational and thermal properties of various molecular structures at the atomic level. Although this technique is widely used for molecular systems consisting of a small number of atoms, performing NMA on large-scale systems is computational challenging. Mathematically, the motion of the molecule is often described by a second order ordinary differential equation

$$\ddot{q} + F_q = 0, \tag{3}$$

where the matrix F_q is a force constant matrix derived from the second derivative of the potential with respect to the Cartesian coordinates. The standard procedure for solving this equation is to diagonalize the matrix F_q by computing its eigenvalues and eigenvectors. Each eigenvector is often referred to as a normal mode with certain vibrational frequency. The frequency is determined by the eigenvalue. The overall dynamics of the molecular system can be described by a superposition of a number of linearly independent normal modes.

5 CHARMM

CHARMM is a general purpose molecular mechanics and molecular dynamics simulation program. The program contains a comprehensive analysis facility which enables the user to compare structures, evaluate energies, calculate time series and correlation functions. The program can treat molecules varying from few number of atoms to large proteins and DNA molecules. The potential energy function and the data structures used in CHARMM will be discussed below.

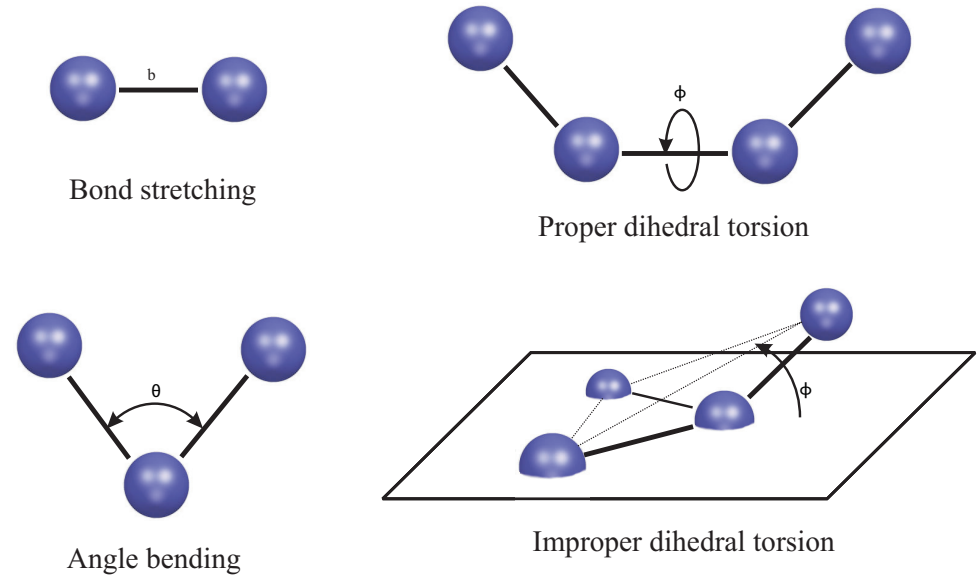

Figure 1: Schematic representation of the bonded interaction terms contributing to the force field: bond stretching, angle bending, proper and improper dihedrals.

5.1 CHARMM force field

The force field description of the interatomic forces is split into two categories: the bonded terms and the non-bonded terms. The bonded terms regroup simple covalent binding as well as the more complex hybridization and π-orbital effects, these are the bonds, angles, dihedrals and improper-dihedrals terms. These terms are schematically drawn in Fig. 1. The non-bonded terms describe the van der Waals forces and the electrostatic interactions between the atoms. The different terms will now be presented in more detail.

Bond stretching

The bond stretching term describes the forces acting between two covalently bonded atoms. The potential is assumed to be harmonic:

$$V_b = k_b(b - b_0)^2 \tag{4}$$

where b is the distance between the two atoms. Two parameters characterize each bonded interaction: b_0 the average distance between them and a force constant k_b.

Angle bending

The angle bending terms describes the force originating from the deformation of the valence angles between three covalently bonded atoms. The angle bending term is described using a harmonic potential:

$$V_\theta = k_\theta(\theta - \theta_0)^2 \tag{5}$$

where θ is the angle between three atoms. There again two parameters characterize each angle in the system: the reference angle θ_0 and a force constant k_θ.

Torsional terms

The torsional terms are weaker than the bond stretching and angle bending terms. They describe the barriers to rotations existing between four bonded atoms. There are two type of torsional terms: proper and improper dihedrals. Proper torsional potentials are described by a cosine function:

$$V_\phi = k_\phi[1 + \cos(n\phi - \delta)], \qquad n = 1, 2, 3, 4, 6 \tag{6}$$

where ϕ is the angle between the planes formed by the first and the last three of the four atoms. Three parameters characterize this interaction: δ sets the minimum energy angle, k_ϕ is a force constant and n is the periodicity.

The improper dihedral term is designed both to maintain chirality about a tetrahedral heavy atom and to maintain planarity about certain atoms. The potential is described by a harmonic function:

$$V_\omega = k_\omega(\omega - \omega_0)^2 \tag{7}$$

where ω is the angle between the plane formed by the central atom and two peripheral atoms and the plane formed by the peripheral atoms (see Fig 1).

van der Waals interations

Van der Waals interactions and electrostatic interactions are non-bonded interactions, i.e., they act between atoms which are not covalently bonded together.

The van der Waals force acts on atoms in close proximity. It is strongly repulsive at short range and weakly attractive at medium range. The interaction is described by a Lennard-Jones potential:

$$V_{VdW} = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{8}$$

where r is the distance between two atoms. It is parameterized by σ: the collision parameter (the separation for which the energy is zero) and ϵ the depth of the potential well. The Lennard-Jones potential is represented in Fig. 22.

Electrostatic interactions

Finally, the long distance electrostatic interaction between two atoms is described by Coulomb's law:

$$V_{Elec} = \frac{q_1 q_2}{4\pi\epsilon_0 r_{12}} \tag{9}$$

where q_1 and q_2 are the charges of both atoms and r_{12} the distance between them. ϵ_0 is the electric susceptibility of vacuum.

So finally, the equation for the potential energy describing the force field can be written:

$$\begin{aligned} V &= \sum_{bonds} k_B(b - b_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2 \\ &\quad + \sum_{\substack{proper \\ dihedrals}} k_\phi[1 + \cos(n\phi - \delta)] + \sum_{\substack{improper \\ dihedrals}} k_\omega(\omega - \omega_0)^2 \end{aligned} \tag{10}$$

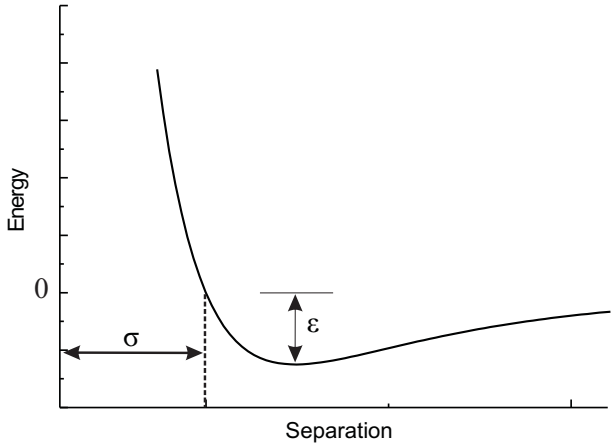

Figure 2: The Lennard-Jones potential. The collision parameter, σ, is shown along with the well depth, ϵ.

$$+\sum_{\substack{i,j\\i<j}} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12}-\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]+\sum_{\substack{i,j\\i<j}}\frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

5.2 Data Structures

The data structures include information about the molecule, its chemical composition, its chemical connectivity, certain atomic properties, internal coordinates for the energy function etc. This information for a particular class of molecule, e.g. protein and nucleic acids, is contained in the topolgy and parameter files.

5.2.1 Residue topology file (RTF)

This file (`top.chm`) contains the information of the residues which are used to build large molecules. It contains the atom type, mass, hydrogen bond donors and acceptors as well as atoms partial charges in particular residue for each and every atom in the system.

5.2.2 Parameter file (PARAM)

This file (`par.chm`) is associated with the RTF file as it contains all the necessary parameters for calculating the energy of the molecule. These include equilibrium bond distances, angles for bond stretching, angle bending and dihedral angle terms in the potential energy function as well as the force constants and the Leonnard Jones parameters.

5.2.3 Protein structure file (PSF)

It is the most fundamental data structure for CHARMM. It is generated for a specific molecule and it is the concatenation of the information contained in the RTF file. It provides detailed information on the composition and the connectivity of the atoms in the molecule of interest. It gives the total number of bonds and provides information as to which atoms connect to form a particular bond. For all the bonds, angles, the torsion

and the dihedral angles are listed. The PSF file must be specified before any calculation is performed. It contains the molecular topology, but not any information regarding bond lengths, angles etc. These informations are provided by the above topology and parameter files.

5.2.4 Coordinate file (CRD)

The coordinate file contains the Cartesian coordinates of all atoms in the system. These are mostly obtained by the X-ray or the NMR experiments. Missing coordinates can be built within CHARMM using the internal coordinate facility. In addition, the hydrogens which are not present in the X-ray crystal structure can also be built in by CHARMM by using the module HBUILD.

6 Exercise

During the course of this practical, you will need several input and data files. All files are located on flory in the ∼zoe/Praktika/pract2011/ directory. They are read-only: you are allowed to read and copy them, but you cannot modify them. Since you will have to modify several input files and create new ones, you need to copy the content of the ∼zoe/Praktika/pract2011/ directory to your 'home directory' (in which you can do what you want). To do so, go to this directory and copy all its content using the following commands:

```
cd ∼zoe/Praktika/
scp -r pract2011 yourmachinename:∼/
```

where ∼ denotes your home directory. The `-r *` options of the command `cp` means that you copy all files (recursively) from the current directory and from all its sub-directories (`-r`). Once the command is done, you can type `ls` in your home directory to list its content. You should see 3 new directories called `dyna`, `initial`, and `norm`.

In grey boxes, you may find questions which will guide through this practical and which you are asked to answer in your report.

6.1 Energy minimization

The first step will be to set up a suitable structure of BPTI for performing molecular dynamics and for normal mode calculations. Such a procedure usually involves energy minimization of a set of atomic coordinates. Several minimization methods will be studied in this part.

6.1.1 Initial Coordinates

Download the best resolution structure of BPTI from the Protein Data Bank located at `www.rcsb.org`. You can do this by using the `Search` field. In the search field, enter the word `BPTI`. Then go through the different structures and find the one with the highest resolution, i.e. the smallest value in the resolution field. The resolution represents to what

degree of error the crystallographers obtained the structure. Therefore, the smaller the error, the higher the resolution and consequently the better the structure.

Download the structure in your `initial` directory. After downloading, you can view the content of the `.pdb` file with the molecular graphics program called VMD. For instance, type the following command: `vmd bpti.pdb` and you get a graphical window showing the molecule in 'wireframe' mode, i.e. atoms are displayed as dots and the chemical bonds as lines. You can see that the H atoms have not been added to the protein and crystal waters yet. CHARMM will do this. Since the process of adding the H atoms is rather tedious, it has already been done for you. In the next sections you will use the structure files produced by CHARMM (BPTI with hydrogens).

6.1.2 Analysis of the Structure and Parameter Files

This section of the practical uses files in your `~/initial` directory. Go to this directory by typing the command: `cd ~/initial` The protein coordinate files are called `bpti.crd`, `bpti.psf`, and `bpti.pdb`. Have a look at the `.pdb` file now using VMD. The `.crd` and `.psf` formats are used by CHARMM. If you use a simple text editor such as `nedit` to look inside the `.crd` and `.pdb` files, you will see they both contain the x,y,z coordinates of each atom, and they identify residues of the protein and the water molecules.

The `.psf` file is the protein structure file. It contains information needed by CHARMM to run the simulation. Each atom is associated with its mass, partial charge, atom reference number (in the topology file `top.inp`), and residue. Please ask your supervisor to explain these files, and the `top.inp` and `par.inp` files needed by CHARMM in more detail.

1. How many amino acids and atoms are in BPTI, how many atoms are hydrogen?
2. Which amino acids are in BPTI, are they hydrophillic or hydrophobic? Where are the residues 3 and 22 located in the protein and why?
3. How many CYS amino acids are present, what is their residue number and what is their respective distance (C_α-C_α in Å)?
4. How are the amino acids connected to each other? Name properties of this bond and explain their origin!
5. How many and which parameters are needed to describe the protein backbone structure?

Di-Sulfide Bonds: Inspect the `top.inp` and `par.inp` files

6. What is the partial charge of the hydrogen (HS/HG1) and the sulfur (S/SG) atom in CYS?
7. What are the parameters for the sulfur-hydrogen (S-HS) and the sulfur-sulfur (SM-SM) bond ? Which of these two bonds is stronger ?

6.1.3 Minimize Structure

Now that you have studied these files, you can do an energy minimization of the protein. This should remove any unwanted high-energy interactions that may have been present in the crystal structure. The CHARMM input file for doing this is `emin.chm`. This file is called a script, i.e. a succession of commands that CHARMM will run sequentially. It contains some of the most commonly used CHARMM commands and also several comments that will help you understand better what are the steps in creating a CHARMM input file. In a CHARMM script comments start with an exclamation mark (!). Their only role is to give explanatory information about the CHARMM commands that follow them. When CHARMM is run, the comments are simply ignored while all the rest is executed. Please read through the comments in your input `.chm` files so you can understand each CHARMM command.

Several minimizations will be performed with a few changes in the input files. For each minimization, you will create a new input file and number it. The output file will have the same number to avoid writting over previous results. For instance the first minimization will be done with a file called `emin1.chm`; running CHARMM will give an output file named `emin1.out`. The next minimization will be conducted with `emin2.chm` that will produce `emin2.out`, and so on. You should also number accordingly the files that are written by the CHARMM scripts (e.g. the coordinate files, their names are specified at the end of the CHARMM scripts).

The energy minimization steps are initally 50 steps of steepest descent (sd) method and 50 steps of the finer Newton-Raphson (abnr) method. First copy `emin.chm` into `emin1.chm` using the `cp` command, and edit the file to change the names of the coordinates files that CHARMM will write (see at the end of the file). To run `emin1.chm` in CHARMM, type:

```
charmm < emin1.chm > emin1.out &
```

The CHARMM output is written to `emin1.out`. Follow the output as CHARMM runs by typing `tail -f emin1.out`. To stop the tail, type `Cntrl C`. At the end of the run, type:

```
grep 'MINI>' emin1.out > emin1.dat
```

to look at how the energy has changed during the simulation. This can be plotted using XMGRACE (see notes on how to use XMGRACE). The second column will be the number of time steps (dimensionless), the third the total energy (in kcal.mol^{-1}), and the fourth ΔE (in kcal.mol^{-1}).

1. Plot the total energy vs. minimization step! Has the minimization converged?

If the energy hasn't converged, create a new input file `emin2.chm` and edit it to increase the number of minimization steps (Hint: we suggest 500 SD and 1000 ABNR). Run this new script with a similar command as above; output file will be `emin2.out` and the result of the `grep` command will be `emin2.dat`.

2. Examine the `.pdb` files before and after energy minimization using VMD. Notice any differences ? (Hint 1: calculate the RMSD using VMD, Hint 2: first load the `.psf` and then the `.pdb` file into VMD.)

3. What is the greatest gradient during any of the minimizations? When does it appear and why ?

4. Try to imagine a few structures for which the CHARMM force-field would give an energy that tends to ∞.

Once a segment of a protein has been generated in CHARMM, it may be manipulated. A general way to do this is to use the `PATCH` command in a CHARMM script. It allows, for instance, the addition of disulfide bridges, the protonation state of a titratible residue to be changed, or to make a histidine heme crosslink. Now you will repeat the energy minimization for BPTI with added disulfide bridges. To put in disulfide bridges, create an `emin3.chm` file and insert the following command just above the minimization call:

`PATCH DISU 1BPI` **residue1** `1BPI` **residue2**.

Actually these commands are already in your input file, but so far they were commented out (i.e. not executed). You need to find the CYS residues where the disulfide bridges will be located and insert their residue numbers in place of **residue1** and **residue2**. (Hint: look for relatively close CYS residues by checking the coordinates in the `.crd` file or by visual inspection using VMD; CYS should be coloured yellow, and there should be three bridges.)

5. What can you conclude from the two different calculations, i.e. BPTI with and without disulfide bridges ?

Repeat the energy minimization of BPTI in vacuum with these different variants on minimization method (don't forget to create a new input file for each minimization):

- 500 steps of Newton-Raphson (`abnr`) and then 100 steps of steepest-descent (`sd`)
- 600 steps of `sd`
- 600 steps of `abnr`
- 100 steps of `sd` and 500 steps of `abnr`

6. For each minimization, plot the total energy vs number of time steps. Which of the four variants is better and why ?

7. Generate a colour picture of one of the disulfide bridges.

8. Plot total energy in $\mathbf{kJ.mol^{-1}}$ vs number of time steps.

9. Does the sign of the total energy have any significance ? Why ?

10. It's also possible to 'solvate' the protein very easily using an implicit solvent model. To do this, simply change in `emin.chm` the keyword `CDIEL` to `RDIEL` and change `elec atom fshift` to `elec atom shift`. This changes the way the electrostatic energy of the system is computed. `CDIEL` means $E_{el} \propto 1/r$ whereas `RDIEL` means $E_{el} \propto 1/r^2$ (E_{el} is the electrostatic energy, r is the distance between two atoms; the $1/r^2$ dependends originates in the assumtion, that $\epsilon \propto r$). Now repeat the minimization with the solvated system. What can you conclude from the two different calculations, i.e. BPTI in vacuum or in solvent ?

11. Plot the different energy terms, e.g. VdW, electrostatic. (Hint: you will need to type `grep 'MINI EXTERN>' emin.out > emin-extern.dat` to obtain these energies.)

6.2 Molecular Dynamics Simulations

6.2.1 Introduction

In this section you will perform a molecular dynamics (MD) simulation of BPTI in vacuum, in the microcanonical (NVE) ensemble. The simulation will be analysed and the dynamics of the protein explored by plotting time series as a function of simulation time, for different structural parts of the protein. The simulation will consist of four parts:

1. heating up the system from the energy-minimized structure at 0 K (Kelvin) to the desired temperature for collecting results (e.g. 300 K).

2. initial equilibration with velocity rescaling. Velocity rescaling is required in addition to the heating to keep the molecule at the temperature where you want to collect results.

3. second equilibration without velocity rescaling. This is done to relax the system before collecting data.

4. and finally a longer run to collect data.

Each of you will simulate at a different temperature, so comparisons in the dynamics of the protein at a range of temperatures can be made. All the files you need are in your `~/dyna/` directory.

6.2.2 Assign Inital Velocities and Heating Dynamics

You will start the dynamics runs using the energy minimized structure of BPTI with the disulfide bridges, so you must also copy `eminNUMBER.crd` and `eminNUMBER.pdb` across to your new subdirectory. (NUMBER refers to the number of the best minimized structure of BPTI with disulfide bridges.)

The first step in the process is to heat up the system from 0 K to the temperature you want to simulate at. Each of you should have been assigned a different temperature (between 270 and 310 K). Please ask your supervisor.

The appropriate CHARMM input file is `heat.chm`. Look through it and read carefully the comments. You should be heating up for a total of 20 ps (`nstep 20000`), with heating every 0.3 ps (`ihtfrq 300`) by 5 K (`teminc 5.0`) at a time step of 0.001 ps (`tstep 0.001`). During the heating, the velocities of all of the atoms are scaled to the desired temperature (`finalt`). Make sure you remember to change `finalt` to your temperature. To run the simulation, type:

```
charmm < heat.chm > heat.out &
```

and, as before, follow the simulation by 'tailing' `heat.out`. At the end, type:

```
grep 'DYNA>' heat.out > heat.dat
```

The file `heat.dat` will contain the number of time steps (dimensionless), simulation time (in ps) total energy, kinetic energy (KE), potential energy (PE) (all in kcal.mol^{-1}), and temperature (in K) in the second to seventh columns respectively.

Have a look at the protein using VMD. Compare the structures of the hot protein (i.e. heated structure) with the cold protein (minimized or crystal structure).

1. What limits the simulation time step to 1 fs. How could this be improved?
2. Why is the protein slowly heated in short intervals during the heating phase and not at once at the beginning. Why is there only one heating step every 300 simulation time steps.
3. Examine and plot using XMGRACE the total energy, potential and kinetic energy, and temperature versus simulation time. Notice anything unusual in your plots ?
4. Can you relate the kinetic energy and the temperature to each other? (Hint: Inspect the CHARMM output-file and look at the parameters just after the dynamics command.)

6.2.3 Equilibration Dynamics and Rescaling Velocities

After heating the protein so quickly the structure may be a little unstable, and in further simulations, the temperature may drop too low; you still need to control the temperature. To do this, you will simulate with some periodic velocity rescaling to the desired temperature. The input file for this is `equil1.chm`. Remember again to change the `firstt` and

`finalt` (first and final temperature) to your value. Notice `ihtfrq` (heating frequency) is now 0 and `ieqfrq` (frequency of scaling to desired temperature) is set to 1000 (1 ps). The velocity will be rescaled every 1 ps to your temperature and you are simulating for a total time of 20 ps. Run the simulation by typing:

```
charmm < equil1.chm > equil1.out &
```

and examine the output as before.

Next, you will have to write your own input file for the next equilibration step. This will be done without any velocity scaling or heating. You will run a 20 ps simulation (20000 steps) and then check the output as before to see that the system is stable. This input file will be very similar to `equil1.chm`, just a few parameters will need to be changed. An example of the input file is `equil2.chm`, but don't cheat !

1. Are the temperature and energy stable ? Did the system reach thermodynamic equilibrium ?
2. Print out the same plots as in the first equilibration; total energy, potential energy, and temperature.

6.2.4 Production Dynamics

If the system has equilibrated, you are ready to start the analysis run ! Now you will simulate for 50 ps, with no heating or temperature scaling of course, and write out the trajectory file `prod.dcd`. See if you can write the input file yourself. An example is the `prod.chm` file.

1. Analyse the output from the final run as before, i.e. plot energies and temperature versus time.
2. Was the system stable during the entire run ?
3. Generate histogram plots for the total energy and temperature using XmGrace (data¿ transformation¿ histograms). What is their functional form? (Hint: This is discussed in any textbook about Thermodynamics.)
4. Is the total energy conserved? Why?

6.2.5 Analysis of Trajectories

You will calculate time series of different parts of the protein. This gives an indication of how much the protein structure has changed from the start to the end of the simulations, and which parts of the protein have changed the most.

The input files for this are as follows:

1. `prot-rms.chm` calculates the RMSD of the whole protein. The output will be written to `prot-rms.out`.

2. `bbone-rms.chm` calculates the RMSD of the backbone atoms only. (Remember the backbone atoms can be seen in `top.inp`). The output will be written to `bbone-rms.out`.

3. `schain-rms.chm` calculates the RMSD of the side chain atoms only. (The side chain atoms can be seen in `top.inp` too. Side chain = whole protein - backbone). The output will be written to `schain-rms.out`.

All scripts produce a `.dat` file automatically. `correl.chm` calculates the time series of any parts of a protein e.g. bonds, angles. You will calculate for the CYS 14 – CYS 38 disulfide bridge the time series of: the S-S distance, the C-S-S angle, and the C-S-S-C dihedral angle. The output will be written to `correl.out`.

These input files are even more complicated than the others! We have set them up for you to run as they are. Please run each of them using e.g.

```
charmm < prot-rms.chm > prot-rms.out &
```

2. Plot out all time series vs time. The output files will consist of: column 1 = trajectory time steps, column 2 = RMSD or ANGLE or ....

3. How do the RMSD of the backbone and sidechains compare to the whole protein ?

4. Compare with the other student's results. How do the above results vary with temperature?

5. Visualize the trajectory of your simulation (see VMD notes). Which parts of the protein are more flexible ?

6. Plot out all time series vs time. The output files will consist of: column 1 = simulation time (in ps), column 2 = time series of S-S distance, column 3 = time series of C-S-S angle, column 4 = time series of C-S-S-C dihedral.

7. What is the average value and standard deviation of the S-S distance, the C-S-S angle, and the C-S-S-C dihedral angle ? Compare with the other student's results.

6.3 Normal Mode Analysis

In this section you will be examining the vibrations of parts of BPTI using normal mode analysis. All relevant files are located in your ∼`/norm` directory.

The first stage in examing the vibrations is to compute the normal modes on an energy-minimized structure of BPTI. We have already done this for you because it takes a lot of time. You can examine the input file we used, called `run.chm` and its output file (unfortunately it is called `bpti-modes.out`; you should rename it to `run.out`). All of the

possible 3N (where N is the number of atoms in BPTI) normal modes were computed; however, only the low-frequency modes are interesting.

In order to see what motions these normal modes represent, you have to superimpose the normal modes on BPTI and create trajectories. Edit the `bpti-modes.chm` file and change "bpti-superNUMBER.dcd" (near the end) to e.g. "bpti-super1.dcd" (NUMBER will be changed again later on). Run CHARMM to create the trajectories from the normal modes:

```
charmm < bpti-modes.chm > bpti-modes.out &
```

The number of trajectories created depends on the number of modes you want to analyze. Here, CHARMM writes out only the 3 lowest modes and one high-frequency mode, for comparison. The corresponding trajectories are `bpti-norm1.dcd`, `bpti-norm2.dcd`, `bpti-norm3.dcd`, and `bpti-norm4.dcd` respectively. These modes can be viewed using VMD.

The combined effect of all three low-frequency modes have been written out in the `bpti-superNUMBER.dcd` file. Now view this file using VMD.

You can also change the 'temperature' of these vibrations. In the `bpti-modes.chm` file, find the line starting with `write unit 11 traj mode ...` and alter the value of the keyword `temp`. Choose a new value for NUMBER, so that you don't overwrite the previous trajectory file. Then repeat the process as before. It's only necessary to view `bpti-superNUMBER.dcd` with VMD for this exercise.

1. Why does one has to minimize the potential energy prior to a NMA and what is the result of the energy minimization?
2. What is the value of the first eigen frequencies and why?
3. What is the maximum frequency and how does it compare to the maximal possible frequency?
4. Why are the low-frequency modes more interesting ? Which parts of the protein are involved in these modes ?
5. Which parts of the protein move in the high frequency modes?
6. What happens as you lower or raise the temperature ?
7. What is the difference between a MD and a NM trajectory? Which technique is more appropriate for high temperatures and why?

How to use XmGrace

You will be using XmGrace on your PC to produce plots of your data. This is a short guide outlining the basics of XmGrace. First of all, to start XmGrace, type `xmgrace` in the directory where you want to plot the data.

Reading in data

1. Using the mouse, at the top click on `Data`, then under file click on `Import` and follow the arrow with your mouse to `ASCII`. You should now see a `Read sets` pop-up window.

2. Inside this window, scroll through the `Files` section on the right hand side using your mouse. This lists all of the files in the directory you are running XmGrace in. Highlight the file that contains the data you want to plot.

3. Click on `Load as block data`,

4. Hit `OK` at the bottom of the `Read sets` pop-up window. A new pop-up window, `Edit block data`, should appear.

5. Inside this window, you will need to select which column you want the x-axis of the plot to be by clicking on `X from column: index` and changing the value of `index` to your column number. You also need to select the y-axis in the same way.

6. Hit `Accept` at the bottom of the `Edit block data` window. Hit `Cancel` at the bottom of the `Read sets` window.

7. Your data should be inside xmgrace ready for plotting !

Simple plotting of data

1. To see your data quickly (if it isn't already plotted), hit the A_S button, located on the left hand side of `xmgrace`, below the `Draw` button. You data should now be plotted.

Adding axis labels

1. Click at the top on `Plot` and pull down to `Axis properties`. An `Axes` pop-up window should appear.

2. Inside the window, type your x-axis label inside the `Axis label:` box. If you want to write a superscript, e.g. kcal.mol^{-1}, type `kcal mol\S-1\N`. If you want to write a subscript, e.g. E_i, type `E\si\N`. To get an angstrom symbol, type `\cE\C`.

3. Hit `Apply` at the bottom of the window.

4. At the top of this window, click on `Edit: X axis` to change from x-axis to y-axis labeling.

5. Type your y-axis label inside the `Axis label:` box.

6. Hit `Apply`, then `Close` at the bottom of the window.

Adding a graph label

1. Double-click at the top of the graph, but still inside the white part of the `xmgrace` screen. A `Graph Appearance` window should appear.

2. Inside the window, type your title into the box marked `Title:`. You can also add a smaller subtitle in the `Subtitle` box below.

3. Hit `Apply`, then `Close` at the bottom of the window.

Checking on data statistics

1. At the top, click on `Edit`, then pull down the menu to `Data sets`. A `Data set properties` pop-up window should appear. The set are written at the top, you normally should only have one set. Click on the set to highlight it black. The data statistics of this set will be written below. The number of data points is written in `Length:`. The numbers in `min` are your lowest-value x-axis and y-axis data points. The points which have these values are next to `min` in `at`. The numbers in `max` are your highest-value x-axis and y-axis data points. Again `at` to the right of this indicates the points which have these values.

2. To close this window, click on `Close` in the bottom-left-hand corner.

3. If you want to keep this window open but it's in the way, you can move it around by clicking and holding in the title bar at the top of the window.

To delete a graph without leaving XmGrace

1. Click on `Plot`, then pull down to `Set appearance`. A `Set Appearance` pop-up window should appear, select the data set you want to delete. This is described above.

2. To kill completely: in the `Data sets:` box, click and hold with the right mouse button and slide down to `Kill data`.

3. To hide: follow the steps above except release the mouse on `Hide`.

4. To show: follow the steps above except release the mouse on `Show`.

Printing graphs

1. At the top of the window pull down the `File` menu to `Print setup`. A `Device setup` window should appear.

2. Type in the box for `Print command`, `lpr` for black and white printing. For colour, type in `lpr -Pdcolor`.

3. Click on `Size:` and change the paper size from `Letter` to `A4`.

4. Hit `Accept`.

5. Again pull down the `File` menu to `Print`. The file will now be printed !

Manipulating data sets

For example: use this if you want to convert your energy data from kcal.mol^{-1} to kJ.mol^{-1}, or convert trajectory time steps into simulation time.

1. Click on the `Data` menu and drag the menu down to `Transformations`. Then drag horizontally from `Transformations` to `Evaulate expression`. A pop-up window, `Evaluate expression` should appear.

2. The boxes `Set:` contains all sets that are available. Click on the sets you want to alter, if they aren't already highlighted black.

3. In the `Formula:` box, type in the equation for transforming the data.
   For example: if you want to convert from kcal.mol^{-1} to kJ.mol^{-1} type: `y=y*4.184`. Your energy data should be in the y-axis, and you are multiplying the y-axis data by the conversion factor of 4.184.
   For example: if you want to convert from trajectory time steps to simulation time, type: `x=x*0.1`. Your time step data should be in the x-axis, and you are multiplying the x-axis data by the trajectory time step in ps. This will give the simulation time in ps.

4. Hit `Accept` at the bottom of the window when you are finished.

5. You may need to hit autoscale $\mathtt{A}_S$ or rescale your axes manually to see your transformed data.

And finally...

To exit XmGrace, click on `Exit`, then `OK` in the following pop-up window !

How to use VMD

You will use VMD to view protein structures and movies of MD simulations and normal modes analysis. The basics will be outlined below.

First of all, to start vmd on your local PC, type `vmd` in the directory where the files you want to view are located. A black window should appear overlayed by a small window with buttons. This is the main menu.

General information

1. To close any pop-up window, click on the button with the blue writing at the top of the window.

2. Look for the `vmd console` window, it gives you information on what VMD is doing. Any errors will be written here.

Reading in pdb files

1. Click on the `Molecule` button in the main menu. A pop-up window titled `molecule` should appear.

2. Click on the `Load From Files` button. Another pop-up window titled `files` should appear. Highlight yellow the `Molecule File Types` you want to view by clicking with the mouse. For example, for viewing a pdb file, click on `pdb only`.

3. Next to `Structure` click on the `Select pdb` button. A pop-up window called `Select molecular structure file` should appear.

4. Click on the name of the file you want to view, then hit `Open`. The window should disappear.

5. Back in the `files` window, the file you selected should be displayed in the white box. Now click on `Load Molecule` to read in your structure. The window should disappear, leaving you with your structure in the black window.

6. If you open up the `Molecule` window again, it will contain some statistics on the file you just opened. You can also open more files and display more than one structure. You can also choose which structures you want active (able to rotate, translate, scale), drawn (displayed), or fixed (not able to rotate, translate, scale) by clicking on the appropriate buttons.

Viewing your structure

1. You can begin rotating the structure by clicking and dragging in the black window.

2. If you want to scale or translate, click on the `Mouse` button in the main menu. A pop-up window titled `mouse` will appear. Click on the `View Mode` window and release your hold of the mouse button on what you want to select. Then in the

black window the mouse will perform this function. To go back to rotating, follow the same procedure.

Changing the display of your structure

1. If you are bored with the line drawing of your molecule, you can change the display by clicking on the `Graphics` button in the main window. A `graphics` pop-up window should appear.

2. Select the molecule you want to change by clicking in the `Selected Molecule` window.

3. To change the colour, click in `Coloring Method` and select the method for colouring. E.g., to colour all CYS residues the same (hopefully yellow !), select `ResName`.

4. To change the drawing method, click in `Drawing Method` and make your selection !

5. Once you have finished making changes, click on `Apply`.

Labeling residues and atoms

1. Click on the `Mouse` button in the main menu. In the `mouse` pop-up window choose `Pick` from `Object Mode` window then `Atoms`.

2. Click on the part you want to label, a label should be written in the display.

3. You can also see the labels more clearly by clicking on `Labels` in the `Main menu` window. A pop-up window `labels` should appear. The labels are displayed in the blue box.

4. To delete the labels, click again on the ones you want to delete.

RMSD fits and structure alignment

Warning: This procedure usually fails with newer versions of VMD. You can use a CHARMM script instead. Refer to e.g. the file `prot-rms.chm` for a guideline.

1. Type `vmdrms` command in `vmd comsole`. Two pop-up windows will appear.

2. In the pop-up window `RMS Alignment` by typing e.g. `all` in the pink box and clicking to `Align` button you can align two different molecules.

3. In the pop-up window `RMSD Calculator` type e.g. `all` in the pink box and click on `RMSD` button. The value of the RMSD will be printed at the bottom of this window.

4. RMSD fits is a little tricky !

Making a movie from a trajectory

1. First you need to read in your trajectory file and a psf file. Follow the directions given in **Reading in a pdb file**. However, instead of selecting `pdb only`, select `psf and dcd`.

2. Once you have read in the data, the movie should start automatically.

3. It will stop when it reaches the end. To play it again, in the `Main Menu`, click on `Animate`. An `animate` pop-up window should appear.

4. To play it again, you need to hit the rewind button.

5. As the movie is playing you can rotate your structure, change the colour and drawing style of your atoms, etc.

Some UNIX commands

Some basic UNIX commands will be listed here for you to refer to when required.

Changing directories

- `ls`
  Gives you the contents of the directory you are in.

- `ls -l`
  Gives you more information ie file permissions.

- `mkdir` *directory_name*
  Creates a directory called *directory_name* inside the directory where you type this command.

- `cd ..`
  Go back up one directory.

- `cd`
  Go to my home directory.

- `cd ~jsmith/initial`
  Change to directory `initial` in the home directory `jsmith`. The ~ refers to the home directory.

- `pwd`
  The location of the directory you are currently in, with the full path.

Copying files

- `cp ~jsmith/initial/ .`
  Copies all files from the directory `initial` from the home directory `jsmith` to the directory you are currently in `.`.

- `cp` *filename(s) /directory_path/directory_name*
  Copies a file selection from the directory where you type the command to any other directory. You must type `pwd` in the directory where you want to copy the files to to obtain the path to that directory.

- `cp` *filename1 filename2*
  Copies *filename1* to *filename2*. *filename1* is not deleted.

- `mv` *filename1 filename2*
  Moves *filename1* to *filename2*. *filename1* is deleted.

- `cp` *filename1 ..*
  Copies *filename1* to the directory above the current directory.

- `scp -rp` *files_to_copy your_login@destination_computer:directory_path/*
  Copies files from the computer you are type the command on, to another computer. For instance, to copy files from your PC to another one called neon:
  ```
  scp -rp * jsmith@neon:/users/jsmith/initial/
  ```
  Make sure you type the command on the computer where the files are located.

- `scp -rp` *your_login@source_computer:directory_path/file_name* .
  Copies files from another computer to the computer you type the command on. The files will be copied to the directory you typed the command in. Example:
  ```
  scp -rp jsmith@neon:/users/jsmith/initial/emin.pdb .
  ```
  This copies the file emin.pdb from neon to the directory on the computer where the command was typed (note the last dot in the command).

Viewing files

- `more` *your_file_name*
  You can view the file one page at a time. To go forward, type the space bar. To go back up, type b. To quit, type q.

- `less` *your_file_name*
  Similar to the `more` command.

- `vi` *your_file_name*
  The most widely-used terminal-based editor. A must!

- `nedit` *your_file_name*
  Editor with menus at the top of the window. Please use `nedit help` for more details.

Removing files

- `rm` *file_name*
  Will remove one file called *file_name*.

- `rm -r` *directory_name*
  Will remove the directory called *directory_name* and files and directories below it.

Checking running processes

- `top`
  Writes out all processes that are currently running on the computer you type the command on. It writes out their PID (process identification number), CPU, and RAM usage. It updates automatically.

- `kill` *PID*
  To kill a running process. Use `top` first to get the PID. If the process is not actually killed, use `kill -9` *PID* instead.