Molecular dynamics simulation using GROMACS

Jane R. Allison

Preamble

In this workshop, you will use the GROMACS simulation package to set up, run and analyse a molecular dynamics (MD) simulation of a peptide.

One common problem of MD simulations is a lack of conformational sampling. Biological processes often happen on relatively long time-scales (μs - s) compared to the length of simulations (ns - μs). Not only do simulations seldom sample all possible conformations of the molecules, they may also be biased by the structure from which the simulation is initiated. Moreover, force fields are not a 'true' representation of how atoms behave, and, as you have learnt in the lectures, the MD method itself also includes a number of approximations!

The goal of this workshop, in addition to teaching you how to set up and run a simulation, is to enable you to get a sense of the degree of variation that can be expected between force fields as well as between simulations run using the same force field.

Aβ

The peptide that you will simulate is Amyloidβ_{1-42} (Aβ_{1-42}). This peptide is one of many variants of Aβ, a family of peptides that are made by cleavage of the amyloid precursor protein (APP). The normal function of Aβ peptides is not known, but they are of great interest as they are the main component of the amyloid plaques found in the brains of patients suffering from Alzheimer's disease. The two most common isoforms of Aβ are Aβ_{1-40} and Aβ_{1-42}. Aβ_{1-42} is the most fibrillogenic, and its production is upregulated in early-onset Alzheimer's disease.

Aβ_{1-42} is generally thought to be intrinsically unstructured in solution, meaning that it does not have a single, unique fold. It therefore cannot be crystallised, so most structural knowledge comes from NMR and from MD simulations. The two NMR structures of Aβ_{1-42} show high levels of α-helical structure [6, 5](Figure 1), but this may be a result of the NMR structure determination protocol, that requires a single structure to satisfy all of the experimental data. Several MD studies, by comparison, suggest that Aβ_{1-42} populates multiple discrete conformational states [2, 3]. NMR-guided MD simulations identified more structured regions [4].

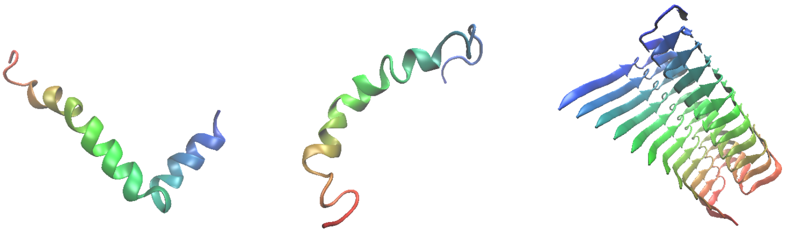

Figure 1: Examples of Aβ structures (left, centre) in solution and (right) in a fibril.

In comparison, the fibrillar form of Aβ found in amyloid plaques comprises predominantly β-sheet structure (Figure 1), indicating that the peptide must undergo substantial structural changes during the fibrillation process. During this process, Aβ forms soluble oligomers. It is now thought that these are the causative agents in the development of Alzheimer's disease, and that the fibrillar form is a sequestration mechanism.

Force fields

In yesterday's lecture on force fields, Tom Collier explained the general format of atomic-level force fields. There are many different force fields available, even just for biological molecules such as proteins. Each force field may use a slightly different set of mathematical terms to describe the atoms and how they interact, and may choose different values of the parameters of these terms.

The way in which these parameter values are optimised may also differ between force fields, for instance in terms of the reference data (quantum mechanical calculations vs experimental data, structural vs thermodynamic data), the fitting method (manual, machine learning), and the ultimate goal (a transferable force field for simulating a variety of related molecules in a range of situations vs a force field for a specific molecule and situation).

Finally, each force field is typically designed to work with or include a particular solvent (e.g. water) model, and a particular set of choices about how the simulations are run (distance to which non-bonded interactions are explicitly calculated, treatment of long-range electrostatic interactions, etc), which Alan Mark discussed in his lecture this morning.

For all of these reasons, the behaviour of the molecules you are interested in can depend on your choice of force field, and on how you run the simulations. This is particularly true for an unstructured molecule like Aβ_{1-42}.

CHARMM, AMBER, GROMOS and OPLS represent the four main families of biomolecular force fields. In each case we have chosen one of the most recent versions of the force field.

CHARMM36m and AMBER ff99sb-ILDN are force fields that are designed specifically for unstructured peptides and proteins. The other force fields are intended to be general to all structural states.

You will all start both of your simulations from the same structure (Model 27 from the ensemble of NMR solution structures with PDB ID 1Z0Q). We will collate the results from everyone in the workshop so as to investigate how much influence the choice of force field has on the resulting conformational ensemble.

Warnings

As a general guideline, the settings and parameters that were used to develop and test the force field that you are using should be used for the simulations. This is because both the force fields and the MD simulation procedure are approximate descriptions of the 'real' behaviour of atoms (an indeed of their sub-atomic constitents), and there is much cancellation of these approximations between the force field and the way in which the simulation is run. In particular, aspects of the simulation procedure such as the choice of thermostsat and barostat, the treatment of long-range electrostatic interactions (i.e. PME vs reaction field) and the cut-off schemes used to monitor which atoms are considered neighbours, and the parameters of these settings, can be *very* different between different force fields. A simulation using the settings appropriate for another force field may still run, i.e. the program will not crash, but the results may be meaningless and you may not necessarily be able to tell unless you have quantitative experimental data to compare to. The best advice we can give here is to always read the publication(s) in which the force field was originally described.

Additionally, when using GROMACS to run MD simulations, one has also to be careful about the version of the software that is used. Not only are the binary files produced by different versions of the software incompatible with one another, there have also been substantial changes to which algorithms are implemented, and how they are implemented, between e.g. versions 3.3, 4.0, 4.6 (avoid 4.5) and 5.x.

There are too many details to cover in the limited space of this workshop - force fields and simulation settings are complex and it takes time to get to know all the options - but these papers [7, 8, 9, 11, 12, 13, 15] are a good starting point to read up on comparisons between different force fields and how simulations using them should be run, and on how differences between different versions of GROMACS can radically affect the outcome of simulations even when they appear to have been run in a consistent manner.

Instructions

Setup

You should already have a directory `Workshop2` within the `MolSim2017-master` directory that you created yesterday. If for some reason you do not have this, go to `https://github.com/arohl/MolSim2017`, click the green 'Clone or download' button, choose 'Download ZIP', and save the zip archive to an appropriate location on your computer. Extract its contents by typing:

```
unzip MolSim2017-master.zip
```

This should give you two sub-directories, `Workshop1` and `Workshop2`. The latter contains the files for today's workshop, specifically, the input files that you will need to set up, run and analyse your simulations using GROMACS. Some of these files begin with the name of the force field to which they are specific. If not, they are general to any force field.

Before you begin, make two separate sub-directories within your `Workshop2` directory whose names correspond to the two force fields that you will use. As you go through the steps outlined below for each force field, you will work within the corresponding directory, copying in the files that you need as you proceed.

Software

GROMACS If you are using one of the desktop computers provided, GROMACS version 5.1.4 is already installed on your Virtual Machine. If you are using your own laptop, you should already have installed it yourself.

GROMACS provides a number of programs for setting up and analysing simulations, as well as 'mdrun', the engine that runs the MD simulation. To run a GROMACS program, you first type `gmx` followed by a space and then the name of the program. For information about what a program does and the options that can be passed to it, type `gmx progname -h`, where 'progname' is the name of the program you would like to know about.

VMD You will use VMD to view the results of the simulation, and to carry out some of the analysis.

gnuplot You will use gnuplot to make graphs showing some of the values you calculate from your simulations.

Preparation for MD simulation

System

In order to run a MD simulation, you first need to decide what you will simulate and at what level of detail. Here, these choices have been made for you: you will simulate the Aβ_{1-42} peptide in water, using an atomic-level representation.

Coordinates

Before starting a simulation, you need to assign coordinates to all of the atoms in the system. For biological molecules, these are often obtained from the Protein Data Bank (PDB), `https://www.rcsb.org/pdb/home/home.do`. This is a repository where the atomic coordinates of proteins and other biological molecules obtained from X-ray or neutron diffraction, NMR, and, more recently, cryo-electron microscopy, are deposited.

You can search the PDB for the molecule that you are interested in by its PDB ID (a unique code assigned to every set of coordinates deposited in the PDB), the author of the paper describing the structure, the name of the macromolecule or its sequence or ligands.

In this case, you don't need to search the PDB, as we have already identified which set of coordinates you will use. You will start your simulation from Model 27 of the ensemble of structures determined using NMR data with PDB ID 1Z0Q. If you would like to see this entry in the PDB, go to the web address given above and type '1Z0Q' into the search field. The 'Download Files' tab towards the top right of the page allows you to download the coordinates in PDB format, a standard file format for coordinate information.

You already have the 1Z0Q PDB file in the `MolSim2017Workshop2-master` directory. Open it in a text editor. You will see that there is a lot of information about the source of the protein, how the structure was determined, and the quality of the structure.

Scroll down or search for the line that beings `MODEL 1` (line 549). This is where the coordinate information begins. There is a line for each atom, e.g.:

```
ATOM 1 N ASP A 1 -20.053 -5.788 1.939 1.00 0.00 N
```

The columns after 'ATOM' correspond to the atom number, atom type, amino acid residue type, chain (for a multi-subunit protein), amino acid residue number, the x, y and z coordinates, and then two columns that are only relevant for X-ray crystal structures (occupancy, temperature factor), and lastly the element type.

If you scroll down further, or search for 'MODEL', you will see the coordinates for the other models that make up the NMR ensemble (here there are 30 models). Structures solved using X-ray diffraction of crystals typically only comprise one set of coordinates, as crystals are (relatively) rigid, but NMR data is an average over an ensemble of molecules in solution, which can be more or less dynamic, so that it is not appropriate and often not possible to define a single structure that exactly fits all of the experimental data, thus multiple model structures are usually generated.

You will be starting your simulation from the coordinates of Model 27. We have already extracted the coordinates for just this model into a separate file, `M27.pdb`. You can view these coordinates using VMD. Open VMD by typing `vmd`. In the 'VMD Main' window, click 'File → New Molecule'. Click 'Browse' to find and select `M27.pdb` and click 'Load'. You should now see the peptide in the display.

To change how the molecule looks, click 'Graphics → Representations'. Try different options in the ‘Draw style’ tab. You can overlay different representations by clicking ‘Create Rep’ to create a replicate of the molecule and drawing each replicate differently.

To change your view of the molecule, hover the mouse over the display area, hold down the left mouse button, and move the mouse around. The molecule should rotate.

PDB format is one standard coordinate file format, but there are many others. The GROMACS simulation software that we will use today has its own coordinate file format, which has the file suffix `.gro`. You will convert this PDB format file into `.gro` format as part of the first step of setting up the simulation.

Parameters

As well as coordinates, you also need to provide the parameters of the force field that you will use to describe how the atoms interact with one another. The GROMACS software distribution includes force field files for several commonly-used force fields.

Information about which force field has been chosen, and the parameter values for the molecule(s) that are to be simulated, are assembled in a ‘topology’ file (`.top`).

Both the creation of the topology file, and the conversion of the PDB format coordinates to `.gro` format, are carried out by the GROMACS program `pdb2gmx`.

File Conversion

Run `pdb2gmx` by typing:

```
gmx pdb2gmx -f M27.pdb -o M27.gro -ignh
```

The ‘-ignh’ option tells the program to ignore the hydrogen atoms. These can cause problems in the coordinate conversion and parameter assignment as there are many different nomenclatures used to name hydrogen atoms. Luckily, as X-ray structures do not contain coordinates for hydrogen atoms, there are robust procedures for assigning coordinates of hydrogen atoms according to the local geometry, so we will take advantage of those and ignore the hydrogen atoms in our NMR model structure.

You will be asked which force field you would like to use. Type the number corresponding to the force field you have been assigned and hit ‘Enter’.

You will then be asked which solvent model you would like to use. Each force field is compatible with one or a few solvent models, and it’s important to choose the right one. Look in Table 1 to find the water model that is compatible with the force field you are using (note that in some cases, other water models might also be appropriate).

The `pdb2gmx` program writes a lot of information to the screen about what it is doing. It is a good idea to read through this information, as it helps

Table 1: Force fields and corresponding water models

Force field	Water model
CHARMM27	TIP3P
CHARMM36m	TIP3P
AMBER ff99sb	TIP3P
AMBER ff99sb-ILDN	TIP3P
GROMOS 54A7	SPC
OPLS	TIP3P

you to understand the processes taking place and is also a good way to pick up any potential problems or errors before you run your simulation. Ask the demonstrators if there is anything you don't understand in this output.

You should now have three new files:

`M27.gro` The coordinates of Model 27 in `.gro` format.

`topol.top` The topology file, containing the force field parameters.

`posre.itp` A position restraint file, which we will use to restrain the positions of the heavy atoms during the initial stages of the simulation.

Open all three files and make sure you understand their contents.

Simulation Box

Before proceeding further, the size of the simulation box needs to be increased so that there is sufficient distance from the peptide to the edge of the box such that its periodic image is never within the cut-off distance to which non-bonded interactions are calculated. For an unstructured molecule, the box size needs to account for potential changes in the structure that might make the molecule larger.

For all of the force fields listed in Table 1, the longest cut-off distance for the non-bonded interactions is 1.4 nm, so that will form the basis of the box size. The GROMACS program `editconf` can change the box size, and will also centre the peptide in the box. Run `editconf` using the following command:

```
gmx_d editconf -f M27.gro -o M27_box.gro -bt cubic -d 1.4 -c
```

Of the options passed to `editconf`, `-bt cubic` assigns a cubic box, `-d 1.4` specifies that the distance from the peptide to the edge of the box should be 1.4 nm in all directions, and `-c` centres the peptide in the box.

You should now have a new `.gro` file, `M27_box.gro`. The box size is specified at the end of the file. Compare the box size in this file to the box size in the original `M27.gro` file.

Energy minimisation

The coordinates deposited in the PDB, or obtained from elsewhere, should be chemically reasonable, and hopefully also biologically relevant, but may not agree exactly with the ideal geometries specified by the force field. It is therefore good practise to minimise the energy of the system according to the force field so that the system is at a stable point or a minimum on the potential energy surface. Beginning the dynamics from here has the advantage that the net force on each atom vanishes.

There are many different algorithms for finding an energy minimum, each of which has different pros and cons. Two methods that are commonly used prior to running MD simulations are steepest-descent and conjugate-gradient.

The steepest descent method uses the first derivative of the energy function to determine the direction towards the minimum. It is not particularly efficient, but it is robust. It is therefore often used to minimize initially when the structure is far from the energy minimum.

More efficient minimization can be obtained using conjugate gradients or the Newton-Raphson algorithms. The conjugate gradient technique uses information from previous first derivatives, whereas the Newton-Raphson method also uses the second derivative, i.e. the curvature, to predict where along the gradient the function will change direction.

You will first minimise the energy of the peptide using the steepest descent method. The input file that tells `mdrun` how to do this is `min_sd.mdp`. Open it in your text editor. The items that are most relevant to the energy minimisation are:

`integrator = steep` Specifies that the steepest descent method will be used.

`nsteps` Specifies how many energy minimisation steps to take.

`emtol` Tolerance - the minimisation is converged when the maximum force is smaller than this value (kJ·mol·nm^{-1}).

`emstep` Initial step size (nm).

The remainder of the options will be explained later in the tutorial.

Before the energy minimisation can be carried out (or indeed before running any simulation), it is first necessary to prepare a binary input file for `mdrun` that combines the coordinate and parameter information with the instructions in the `.mdp` file. This is done using the program `grompp`:

```
gmx_d grompp -f min_sd.mdp -c M27_box.gro -p topol.top -o M27_minsd.tpr
```

Then run the energy minimisation by typing:

```
gmx_d mdrun -deffnm M27_minsd
```

This should produce a number of output files that start with 'M27_minsd'. The -deffnm command specifies a generic file name for the input and output files:

M27_minsd.edr Energy trajectory; contains the system energies at each step of the minimisation.

M27_minsd.xtc Coordinate trajectory; contains the system coordinates at each step of the minimisation.

M27_minsd.gro Final coordinates at the end of the minimisation.

M27_minsd.log Log file; contains detailed output of the minimisation process.

It is good practise to look at both what is printed to the screen and also at the log file. At the start of this file, GROMACS prints the values of all the parameters that could be set, not just the ones that are specified in the input .mdp file. Take a look - there are a lot of parameters! Not all of these are relevant for each type of simulation. For some, the default values are fine, for others, you need to think more carefully about which value you use. Some of these choices will be discussed during this tutorial.

The second energy minimisation step will use the conjugate gradient method. Compare the input files min_sd.mdp and min_cg.mdp. What differences do you see?

Prepare for the second minimisation by typing:

```
gmx_d grompp -f min_cg.mdp -c M27_minsd.gro -p topol.top -o M27_mincg.tpr
```

and then run the minimisation by typing:

```
gmx_d mdrun -deffnm M27_mincg
```

This should produce a second set of output files like those listed above, but with names that start with 'M27_mincg'.

It is good practise to keep track of the coordinates and energies as you prepare a simulation, as well as afterwards. To view how the coordinates of the peptide change during the energy minimisation procedure, open or go to VMD. Click 'File → New molecule' and load the following files into the same molecule by choosing each in turn: M27_newbox.gro, M27_minsd.xtc, M27_mincg.xtc. You should see the molecule moving slightly as the last two files load.

To analyse the energy during the minimisation (or during a simulation), you first need to use the program energy to extract the energy values of interest from the energy trajectory:

```
gmx_d energy -f M27_minsd.edr -s M27_minsd.tpr -xvg none
```

Type in the number corresponding to the potential energy and hit 'Enter' twice. A summary of the average potential energy and its drift during the steepest descent minimisation is printed to screen, and you should now have a new file energy.xvg. Rename this file to M27_minsd_ene.dat.

Now run energy again, this time providing the .edr and .tpr files from the conjugate gradient energy minimisation. Rename the energy.xvg output file to M27_mincg_ene.dat.

Plot the energies from both minimisation runs by typing:

```
gnuplot plot_emin.gnu
```

View the resulting graph (minimisation_energy.pdf). What happens to the energy during each phase of the minimisation?

Solvate

You may have noticed that up until now, you have only dealt with the peptide. The next step is to fill the box with solvent molecules. This is done by copy/pasting a box of pre-equilibrated solvent (water) molecules (spc216.gro, used even for other water models, and distributed with GROMACS) enough times to cover the entire simulation box, then deleting any molecules that fall outside the box or overlap with peptide atoms.

```
gmx_d solvate -cp M27_mincg.gro -cs spc216.gro -p topol.top -o M27_solv.gro
```

Open the topology file, topol.top, and move to the end of the file. There is now an additional line specifying the number of solvent molecules.

Open M27_solv.gro in VMD to view the peptide surrounded by water molecules.

Energy minimisation

It's a good idea to run an energy minimisation each time a major change is made to the system. After adding solvent, not only are there a lot more molecules present, but it is possible that some of the water molecules are not positioned optimally with respect to the protein.

```
gmx_d grompp -f min_sd.mdp -c M27_solv.gro -p topol.top -o M27_solv_minsd.tpr
gmx_d mdrun -deffnm M27_solv_minsd
```

Add ions

You may have noticed in during the minimisation steps that grompp complained that the total charge of the system is -3. It's a good idea to neutralise the total charge of the system before running a simulation. This is done by adding counter ions. Additionally, you may sometimes wish to add further ions to e.g. mimic the experimental salt concentration.

The addition of ions requires a preparation step using grompp. As no simulation will actually be run, a dummy input file can be used.

```
gmx_d grompp -f ions.mdp -c M27_solv_minsd.gro -p topol.top -o M27_ions.tpr

gmx_d genion -s M27_ions.tpr -p topol.top -o M27_ions.gro -np 3 -pname
NA
```

genion will ask you to select a continuous group of solvent molecules. Type the number corresponding to 'SOL' and hit 'Enter'.

Open the topology file, topol.top. The number of solvent molecules should now be reduced by three, and there should be three NA ions added to the end.

Make index file

You have now added all the components to your system. To make it easier to refer to specific groups of atoms or molecules, it's a good idea to make an index file:

```
gmx_d make_ndx -f M27_ions.gro -o index.ndx
```

Type 'q' and hit enter to finish this procedure. The groups of atoms that are automatically defined are printined onto the screen. You can also open index.ndx in a plain text editor to see the lists of atoms that make up each group.

Energy minimisation

There is just one last step to carry out before starting the first phase of the MD simulation: a final energy minimisation.

```
gmx_d grompp -f min_sd.mdp -c M27_ions.gro -p topol.top -o M27_ions_minsd.tpr

gmx_d mdrun -deffnm M27_ions_minsd
```

Running MD simulations

Initialisation and heating

To initiate a MD simulation, each atom needs to be assigned a velocity. Given that you (usually) have no prior knowledge about the velocity of each atom, the velocities are assigned at random from e.g. a Maxwell-Boltzmann distribution such that the total linear momentum of the system is zero.

To avoid atoms being assigned large velocities whose directions would result in large changes in the atomic positions incompatible with the force field after the first step of the MD simultion, the velocities should be assigned at a low temperature, e.g. 50-60 K, and the system slowly heated to the desired simulation temperature (often ≈300 K for biological systems). Alternatively, but less safely, the velocities may be assigned at the final simulation temperature, and the atoms restrained to their initial positions during the early phases of the simulation.

Here you will assign the velocities at 50 K, equilibrate the system briefly (for 10 ps) at 50 K to allow the velocities to become correlated, then heat the system from 50 K to 298 K over 210 ps, followed by a second brief equilibration at 298 K for 40 ps. This process will be run in the NVT, or canonical, ensemble. You will apply position restraints to the protein atoms during this entire procedure, so that its structure is maintained while allowing the water molecules to relax.

The MD input file for running this simulation is `nvt_heat.mdp`. Open it in a plain text editor and take a look at the instructions and parameters being passed to the MD engine. There are some changes and additions compared to the `.mdp` file used to energy mininimise the system, namely:

`-DPOSRES` Tells `grompp` to include `posre.itp` into topol.top so that position restraints are applied. The force constants for the harmonic function that keeps each atom in place are defined in the final three columns of `posre.itp`.

`integrator = md` Specifies that an MD simulation will be run using the leap-frog integrator.

`dt` The integration time step. Values greater than 2 fs result in loss of energy conservation due to the approximations inherent in numerical integration of the equations of motion.

`Bond constraints` Bond lengths are now constrained using the LINCS algorithm. Constraints mean that the bond lengths are no longer flexible, and are reset after each integration time step using an iterative algorithm.

`Neighbour searching` While some of these parameters were present in the energy minimisation files, they were not explained there. The parameters in this section determine the maximum cutoff distance to which pairwise interactions are calculated (`rlist`), how often to update the list of neighbouring (i.e. within the cutoff) atoms (`nstlist`), and the methods that are used for searching for neighbouring atoms (`grid`) and for monitoring when atoms cross the cutoff (`Verlet`).

`van der Waals` These are parameters specific to calculating the van der Waals interactions, including the distance to which van der Waals interactions are evaluated (`rvdw`), and whether to use an unmodified Lennard-Jones potential or use a switch or shift function to ensure that the van der Waals interactions are zero at the cut-off distance.

`Electrostatics` These are parameters governing how long-range electrostatic interactions are calculated. Typical choices are PME (particle mesh Ewald) or Reaction Field - the optimal choice will depend on the force field. The distance to which coulombic interactions are explicitly calculated (`rcoulomb`) is also specified.

`tcoupl` We are now using temperature coupling, i.e. a thermostat, to control the temperature of the simulation so that we are in the NVT ensemble. Again, the method that is used and the choices of the parameter values may differ between force fields and also between situations (e.g. some thermostats are better suited to the early stages of a simulation where the temperature is still equilibrating). Here we are using the Velocity rescale method, which does exactly what the name implies, to maintain a temperature of 298 K. The solute and the solvent (including ions) are coupled separately.

`gen_vel` This tells the MD engine to generate velocities for each atom, in this case at 50 K.

`annealing` This section is where the changes in temperature are controlled. Both the time points for temperature changes and the associated temperatures are specified.

Once you are satisfied that you understand the settings in the `.mdp` file, run this stage of the MD simulation:

```
gmx_d grompp -f nvt_heat.mdp -c M27_ions_minsd.gro -p topol.top
-o M27_nvt_heat.tpr
gmx_d mdrun -deffnm M27_nvt_heat
```

This may take some time to run.

Before proceeding further, it's a good idea to check that the heating process worked. This can be done by using the `energy` program to extract the system temperature during the simulation.

```
gmx_d energy -f M27_nvt_heat.edr -s M27_nvt_heat.tpr -xvg none
```

Type in the number corresponding to the temperature and hit 'Enter' twice. A summary of the average temperature and its drift during the NVT heating phase is printed to screen, and you should now have a new file `energy.xvg`.

Rename this file to M27_nvt_heat_temp.dat and plot it by typing:

```
gnuplot plot_temp.gnu
```

and open the resulting pdf file. Does the temperature behave as you expected?

Equilibration

Having heated and briefly equilibrated the system, the next step is to equilibrate it further in the NpT ensemble. This ensemble is generally more representative of the conditions under which 'real world' experiments are carried out. The input file for this phase is npt_eq.mdp. Open it in a plain text editor and take a look at the instructions and parameters being passed to the MD engine. There is one key change compared to nvt_heat.mdp, namely:

pcoupl The pressure is now coupled to a barostat. Here we use isotropic coupling, but for certain systems, such as a lipid bilayer, semi-isotropic coupling should be used. The choice of barostat depends on the system simulated, the phase of the simulation and the force field, and the values of the accompanying parameters depend on the barostat. Here, the Berendsen barostat is used to maintain a pressure of 1 bar, and the isothermal compressibility is an approximate value for a protein in water.

Once you are satisfied that you understand the settings in the .mdp file, run this stage of the MD simulation:

```
gmx_d grompp -f npt_eq.mdp -c M27_nvt_heat.gro -t M27_nvt_heat.cpt -p
topol.top -o M27_npt_eq.tpr
gmx_d mdrun -deffnm M27_npt_eq
```

Notice that the cpt file is now additionally passed to grompp. This file includes the atomic velocities from the end of the last simulation, which are required to continue the dynamics.

This simulation may take some time to run.

An interesting property to monitor during this phase of the simulation is the volume of the simulation box, as this can change as the system adjust its pressure. Do this using the energy program:

```
gmx_d energy -f M27_npt_eq.edr -s M27_npt_eq.tpr -xvg none
```

Type in the number corresponding to the pressure and hit 'Enter' twice. A summary of the average pressure and its drift during the NpT equilibration phase is printed to screen, and you should now have a new file energy.xvg. Rename this file to M27_npt_eq_press.dat and plot it by typing:

```
gnuplot plot_press.gnu
```

and open the resulting pdf file. How does the pressure change during the equilibration? What does this tell you about the initial volume of the simulation box?

Production run

You are now ready to run a 'production' MD simulation, i.e. from now on, you are collecting results! The input file for running this simulation is essentially the same as the one for the previous step. Open npt_eq.mdp and npt_prod.mdp in a text editor and see if you can spot the difference.

Finally, start your production run:

```
gmx_d grompp -f npt_prod.mdp -c M27_npt_eq.gro -t M27_npt_eq.cpt
-p topol.top -o M27_npt_prod.tpr
gmx_d mdrun -deffnm M27_npt_prod
```

This may take some time to run. While it is running, read ahead to make sure you understand what is coming next.

Analysis of MD simulations

Running a simulation is not the end of the story - the analysis is key to extracting useful information from the simulation, as well as checking that nothing anomalous occurred.

RMSD

The first quantity that is often calculated from a simulation is the atom-positional root-mean-square deviation, or RMSD. This gives a single value at each time point during the simulation trajectory that describes how much the coordinates of some set of atoms have deviated from the some reference set of coordinates. This requires that the coordinates at each step during the simulation are superimposed as well as possible onto the reference coordinates. The residual deviation is then calculated and reported. Thus there are three key choices that need to be made: which set of coordinates are used as a reference, which group of atoms are superimposed, and for which group of atoms the residual deviation is claculated and reported.

Here, you will use the coordinates at the start of the production NpT simulation as the reference state. You will superimpose the Cα atoms of the peptide onto their initial positions, and then calculate the RMSD of two groups of atoms: a) the Cα atoms; b) all atoms of the peptide. This will report on how much the backbone and the side chains of the peptide move with respect to their initial positions, respectively.

```
gmx rms -s M27_npt_prod.tpr -f M27_npt_prod.xtc -n index.ndx
-o M27_npt_prod_rmsdCA.xvg -m M27_npt_prod_rmsdCA.xpm -xvg none
```

You will be asked to choose two groups of atoms. For the first (group for least squares fit), select the number corresponding to 'C-alpha'. For the second (group for RMSD calculation), select the number corresponding to 'C-alpha' again.

Now run the `rms` program again, this time giving the output file the name `M27_npt_prot_rmsdALL.xvg`, and selecting the number corresponding to 'Protein' for the second group.

Rename each of the `.xvg` output files to have the suffix `.dat`, and plot the RMSD time-series using gnuplot:

```
gnuplot plot_rmsd.gnu
```

Do the relative magnitudes of the all-atom and Cα-atom RMSD values make sense? Compare the RMSD values that you get for each force field that you are using, and to those obtained by others who have used different force fields. Are the values similar for each force field?

RMSF

The RMSD reports on the overall deviation of the selected group of atoms from their initial positions. It is often useful to know which atoms are moving the most. This can be illustrated using the root-mean-square fluctuation of each atom relative to its average position during the simulation. In the case of a peptide or protein, it is often helpful to summarise this information at a residue level.

```
gmx rmsf -f M27_npt_prod.xtc -s M27_npt_prod.tpr -n index.ndx
-o M27_npt_prod_RMSF.xvg -xvg none -res
```

You will be asked to choose a group of atoms for the root-mean-square calculation - select the number corresponding to 'Protein' again.

Rename the output file to have the suffix `.dat`, and plot the RMSF values using gnuplot:

```
gnuplot plot_rmsf.gnu
```

Which residues move the most/least during the simulation? Can you rationalise this based on where they are in the sequence, or in the initial structure?

Secondary Structure

The secondary structure content during the simulation provides insight into the types of structures that a protein or peptide forms. While this can be calculated using the GROMACS do_dssp program, this requires installation of the separate dssp program, which can be problematic. Thus here, we will use VMD to calculate the secondary structure content instead.

Open VMD, and load the following two files into the same molecule: M27_npt_eq.gro and M27_npt_prod.xtc. You should now be able to see a movie of your peptide during the simulation. You may wish to view only the peptide component - to do so, open the 'Graphical Representations' window by clicking 'Graphics → Representations' and type protein in the 'Selected Atoms' box. You may wish to change the Drawing Method and/or the Coloring Method.

To calculate the secondary structure, you will use the script calcSS.tcl, which was provided with the other files for this tutorial. In VMD, click 'Extensions → Tk Console'. This opens a window that looks a little like a terminal window. Many of the usual command line tools work here, but commands in tcl can also be entered directly.

Type pwd to determine where you are, and navigate to the directory where the 'calcSS.tcl' script is. To run the script, type source calcSS.tcl.

Go back to your terminal window. You should now have a set of files with names like helixPercent.dat. Plot these by typing gnuplot plot_ss.gnu. Open the resulting pdf file (M27_NpT_prod_secStruct.pdf). This shows the percentage occupancy of each type of secondary structure during the simulation.

Which type of secondary structure is most dominant? How does this compare to the secondary structure of the structure that your simulation started from?

Conformational clustering

An MD simulation can sample many thousands of structures, which can be difficult to visualise. One method for extracting the most populated structures is conformational clustering, which groups similar structures together and reports the central structure of each cluster. The RMSD between two structures is one way to define how similar they are.

Before clustering the structures sampled during the simulation according to their RMSD from one another, it's a good idea to visualise the pairwise RMSD values. When you calculated the RMSD earlier, you made an output file M27_npt_prod_rmsdCA.xpm which you have not yet used. This file contains a matrix of pairwise RMSD values between each structure sampled during the simulation. To view this matrix, first convert it to an .eps file:

```
gmx xpm2ps -f M27_npt_prod_rmsdCA.xpm -o M27_npt_prod_rmsdCA.eps -rainbow
blue
```

and then convert the eps file into a pdf file:

```
epstopdf M27_npt_prod_rmsdCA.eps
```

Open the pdf file. Can you figure out what it is showing? What is the maximum RMSD between pairs of structures sampled during the simulation? You should see some pale or even dark blue squares along the diagonal. These indicate that similar structures were being sampled at that point during the simulation.

In order to cluster the structures sampled during the simulation, you will need to provide a cut-off value of the RMSD: pairs of structures with RMSD values less than this cut-off will be considered similar. Choose a value corresponding to the colour of the squares visible in the RMSD matrix. Note that there is no right or wrong choice for this; the smaller the value, the more clusters (with fewer members) you will obtain. The ideal number of clusters and their populations will depend on the scale of the conformational changes that take place and that are of interest. If you are not sure about this choice, talk to one of the demonstrators.

An example of how to cluster the conformations sampled during your simulation is:

```
gmx cluster -f M27_npt_prod.xtc -s M27_npt_prod.tpr -n index.ndx
-o M27_npt_prod_cluster.xpm -g M27_npt_prod_cluster.log
-clid M27_npt_prod_clustID.xvg -cl M27_npt_prod_clusters.pdb
-cutoff 0.25 -nst 100 -method gromos -minstruct 100 -xvg none
```

You will be asked to select a group for carrying out the least squares fit and RMSD - choose 'C-alpha' - and for the output - choose 'Protein' (only the peptide coordinates will be printed).

Here we have chosen the 'gromos' clustering method. This method counts the number of neighbours within the cut-off, takes the structure with largest number of neighbours and all its neighbours as a cluster, and eliminates it from the pool of structures. This process is then repeated for all the remaining structures. There are several other methods available too.

Other choices included in the command above include the cut-off RMSD (0.25 nm) - you should change this if you selected a different value - and the minimum number of structures that need to be in a cluster before it is included in the results (100). You may wish to increase this to reduce the number of clusters that are reported, so as to focus only on the most populated clusters. Information about how many clusters were found, and how many structures are in each, is printed to the command line and is also in the log file (M27_npt_prod_cluster.log). You may need to run the clustering program several times until you get a result you are happy with (e.g. manageable number of clusters).

Once you are satisifed with the number of clusters, you can view the central structure of each cluster using VMD. Open VMD and load the file M27_npt_prod_clusters.pdb.

Open the Graphical Representations tab and choose an appropriate Drawing Method. Under Colouring Method, choose 'Trajectory' and then 'Timestep'. In the 'Trajectory' tab, in the 'Draw Multiple Frames' box, type `0:x`, where `x` is the number of frames you have loaded (you can see this in the 'VMD Main' window, and it is the same as the number of clusters. If you don't like the colouring, then in the 'VMD Main' window, click on 'Graphics → Colors', and use the 'Color Scale' tab to change the colour scale 'Method'.

How different/similar are the central structures of each cluster? Did you expect more or less conformational change? Compare the results you get with each different force field, and to the results of others who used different force fields. How do the structures compare to the initial structure at the start of the simulation?

It's also interesting to see how the structure of the peptide transitions from one cluster to another during the simulation. To view a representation of this, along with the RMSD matrix that you viewed earlier, type:

```
gmx xpm2ps -f M27_npt_prod_cluster.xpm -o M27_npt_prod_cluster.eps
```

```
epstopdf M27_npt_prod_cluster.eps
```

and view the resulting `pdf` file. The upper half should be the same as the RMSD matrix you made earlier (but in black and white), and the lower half shows the transitions between clusters, each of which is coloured differently.

Does the peptide ever revisit a conformational cluster during the simulation? If so, the simulation is probably doing a good job of sampling the accessible conformational space. How long does it spend in each cluster? This should give you a sense of how long a simulation might need to be to fully sample conformational space.

References

[1] Frenkel, D. & Smit, B. *Understanding Molecular Simulation* Academic Press, San Diego, 2nd edition. (2002)

[2] Yang, M. & Teplow, D.B. Amyloid β-protein monomer folding: free energy surfaces reveal alloform specific differences. *J. Mol. Biol.* **384**:450-464 (2008)

[3] Sgourakis, N.G., Merced-Serrano, M., Boutsidis, C., Drineas, P., Du, Z., Wang, C., & Garcia, A. E. Atomic-level characterization of the ensemble of the Aβ(1-42) monomer in water using unbiased Molecular Dynamics simulations and spectral algorithms. *J. Mol. Biol.* **405**:570-583 (2011)

[4] Sgourakis, N.G., Yan, Y., McCallum, S., Wang, C., & Garcia, A.E. The Alzheimers peptides Aβ40 and 42 adopt distinct conformations in water: A combined MD / NMR study. *J. Mol. Biol.* **368**:1448-1457 (2007)

[5] Tomaselli, S., Esposito, V., Paolo, V., van Nuland, N.A.J., Bonvin, A.M.J.J., Guerrini, R., Tancredi, T., Temussi, P.A. & Picone, D. The α-to-β Conformational Transition of Alzheimer's Aβ-(142) Peptide in Aqueous Media is Reversible: A Step by Step Conformational Analysis Suggests the Location of β Conformation Seeding *ChemBioChem* **7**:1439-7633 (2006)

[6] Crescenzi, O., Tomaselli, S., Guerrini, R., Salvadori, S., D'Ursi, A.M., Temussi, P.A. & Picone, D. Solution structure of the Alzheimer amyloid beta-peptide (1-42) in an apolar microenvironment. Similarity with a virus fusion domain *Eur. J. Biochem.* **269**:5642-5648 2002

[7] Reisser, S., Poger, D., Stroet, M. & Mark, A.E. Real Cost of Speed: The Effect of a Time-Saving Multiple-Time-Stepping Algorithm on the Accuracy of Molecular Dynamics Simulations *J. Chem. Theory Comput.* **13**:23672372 2017

[8] Rauscher, S., Gapsys, V., Gajda, M.J., Zweckstetter, M., de Groot, B.L. & Grubmüller, H. Structural Ensembles of Intrinsically Disordered Proteins Depend Strongly on Force Field: A Comparison to Experiment *J. Chem. Theory Comput.* **11**:55135524 2015

[9] Guvench, O. & MacKerell A.D. Comparison of Protein Force Fields for Molecular Dynamics Simulations **443**:63-88 2008

[10] Bob Tadashi Wakabayashi *Anti-Foreignism and Western Learning in Early-Modern Japan* 1986: Harvard University Press.

[11] Martín-García, F., Papaleo, E., Gomez-Puertas, P., Boomsma, W. & Lindorff-Larsen, K. Comparing Molecular Dynamics Force Fields in the Essential Subspace *PLoS One* **10**:e0121114 2015

[12] Lindorff-Larsen, K., Maragakis, P., Piana, S., Eastwood, M.P., Dror, R.O. & Shaw, D.E. Systematic Validation of Protein Force Fields against Experimental Data. *PLoSOne* **7**:e32131 2012

[13] Piana, S., Lindorff-Larsen, K. & Shaw, D.E¿ How Robust Are Protein Folding Simulations with Respect to Force Field Parameterization? *Biophys. J.* **100**:L47-L49 2011

[14] Mercadante, D., Wagner, J.A., Aramburu, I.V., Lemke, E.A. & Gräter, F. Sampling Long- versus Short-Range Interactions Defines the Ability of Force Fields To Reproduce the Dynamics of Intrinsically Disordered Proteins *J. Chem. Theory Comput.* **13**:39643974 2017

[15] Man, V.H., Nguyen, P.H. & Derremaux, P. High-Resolution Structures of the Amyloid- 142 Dimers from the Comparison of Four Atomistic Force Fields *J. Phys. Chem. B* **121**:59775987 2017

Overview

Today you will investigate the conformational preferences of Aβ_{1-42} in solution by conducting MD simulations of a single monomer of Aβ_{1-42}. You have each been assigned two different force fields to use out of this of the six different force fields listed in Table 1.