How to set up a protein simulation

Session 3 - Equilibration

Practical session: equilibration using AMBER

In this session you will work through an example equilibration protocol for 1kyn using the energy minimisation and molecular dynamics module of AMBER10 (*sander*) .

1. In the directory **~/amber/session3** you will find template input and job submission scripts for this part of the workshop.
2. Copy the coordinate (.crd) and parameter (.top) files for 1kyn that you created in the last session into this directory.
3. The first step is to perform an energy minimisation of the solvent, while restraining the solute (protein) to its initial coordinates. This will remove any solvent-solvent and solvent-solute clashes created during the construction process.

The input file for this job is **eq1.in** (provided):

```
#Equilibration step 1: restrained minimisation
 &cntrl
  imin=1, maxcyc=1000, ncyc=100, drms=0.1, ntr=1,
  restraintmask="!:WAT,NA,CL,Na+,Cl-,IP,IM",  restraint_wt=50
 /
```

Key points to note:

Imin=1	Tells sander to do minimisation, not MD
maxcyc=1000	The maximum number of minimisation steps
ncyc=100	Start with 100 steps of steepest descent before switching to conjugate gradient
drms=0.1	The minimisation is terminated when the rms gradient reaches 0.1 kcal/mol/angstrom. This is not a very highly energy minimised state, but

there is no point in going for a lower gradient when pretty soon we will begin shaking things up with MD.

ntr=1 Position restraints will be used

restraintmask Note this selects everything that is NOT a solvent atom (identified by their typical residue names) – if you had something in your solvent that had a name other than one of those listed here, you would need to add it.

restraint_wt This is a high value of 50 kcal/mol/angstrom^2, which will tightly restrain each atom of the solute to its reference (initial) position.

```
#Equilibration step 1

sander -O -i eq1.in -o eq1.out -inf eq1.inf -c 1kyn.crd \

  -r 1kyn.eq1 -ref 1kyn.crd -p 1kyn.top
```

The job is submitted using the submission script **eq1.sub** (provided):

Key points to note:

-O Tells sander to overwrite any existing output files

-o eq1.out At the end will contain full information about how the minimisation progressed

-inf eq1.inf Will contain frequently-updated information about the status of the job as it progresses .

-r 1kyn.eq1 The energy minimised coordinates will be written to this file

-c 1kyn.crd These are the initial/starting coordinates. As you go through the equilibration steps you will need to ensure that the “-c” flag always corresponds to the outputted coordinates from the previous step of the equilibration.

-ref 1kyn.crd The reference coordinates are those used to define the position restraints. Here we use the starting coordinates as the reference coordinates too, but amber allows us to do differently,if we want to.

To submit the job, type:

```
% mpisub 1 eq1.sub
```

The job will be run on 1 node of magnum, which contains 8 cpus. Answer "y" to the prompts (unless something obviously looks wrong) and the job will be submitted. It should take about 1 minute to run. You can follow the progress of the job by looking at eq1.inf (you can also do "tail eq1.out", but due to the vagaries of filesystem I/O buffering, this file is not always up-to-date).

4. When the job has finished, use VMD to check what has happened. If you load both initial (**1kyn.crd**) and final (**1kyn.eq1**) coordinates, you can see what has moved, and what has been restrained.

5. The next step is to remove the restraints and allow everything to energy minimise. This will (should!) clean up any poor geometry or clashes in the protein.

   Prepare a new input file as follows:

   a. Open **eq1.in** for editing using your favourite editor
   b. Amend the title line appropriately
   c. Change ntr=1 to ntr=0, so position restraints are removed
   d. Remove the "restraint_mask=..." keyword
   e. Save as **eq2.in**

   Prepare a new job submission script as follows:

   a. Open **eq1.sub** for editing using your favourite editor
   b. Amend the title line appropriately
   c. The initial coordinates "-c" for this run are the final ones from the last run – i.e., 1kyn.eq1.
   d. This time there are no restraints, so the "-ref..." section can be deleted (this is in fact optional, the program won't complain if this is left in, it will just ignore it)
   e. We need new names for the input and output files – substitute "eq2" for "eq1" (file naming strategies are a matter of personal preference, this is just one suggestion).
   f. No need to change the "-p..." section – the coordinates may be different this time, but the parameters are unchanged.
   g. Save the new file as **eq2.sub**.

Submit the new job in the same manner as you submitted the last. It should take about 1 minute to run.

6. When the job has finished, again use VMD to check what has happened. You should notice small but significant changes to the protein coordinates this time.

7. Now the system is ready for molecular dynamics. Firstly we simulate the process of heating the system to 300K. The input file for sander is provided: **eq3.in**:

```
#Equilibration step 3: warm the system to 300K
 &cntrl
  imin=0, ntr=1, nstlim=10000, dt=0.002,
  ntb=2, ntp=1, ntt=3, gamma_ln=0.2, ntf=2, ntc=2,
  temp0=300, tempi=50,
  restraint_wt=25, restraintmask="!:WAT,NA,CL,Na+,Cl-,IP,IM",
 /
```

For full information about the parameters set here, see pages 19-50 of the AMBER10 manual (available as a pdf file on your desktop). Key points to note are:

nstlim, dt	This will be a 20 picosecond simulation – not a critical parameter, but indicative of the sorts of timescales required for this process. (dt is the timestep of the simulation, in this case 2fs and nstlim is the number of MD steps to be performed, therefore 10000 x 0.002 = 20ps)
ntr=1	We are back to restraining the solute, though not quite so tightly as in eq1.
ntb,ntp	This is a constant pressure simulation. One of the main aims is to equilibrate the solvent box to a density of 1 by correcting voids in the solvent distribution created during the model building process.
tempi/temp0	Initial velocities are assigned for a temperature of 50K, the coupling to the Langevin thermostat will gradually raise these to 300K

Use **eq1.sub** or **eq2.sub** as a guide to creating the new job submission script (**eq3.sub**). Remember we are back to using position restraints, so the "-ref" argument must be included, and note that the file used for the reference coordinates should be the same as that used for the starting coordinates ("-c" argument).

Submit the job in the usual way. It should take about 10 minutes to run.

When it has completed you will find it as useful to scroll through the output file (**eq3.out**) as examine the new coordinates (**1kyn.eq3**) using VMD. From the output file you should see the temperature of the system rising, and the density of the system increasing, through the simulation. Harking back to the presentation, you should see that effectively you have achieved both steps 3 and 4 of the suggested equilibration protocol in this MD simulation.

8. The next step is to equilibrate the side chains, in addition to the solvent. The input file for sander is provided – **eq4.in**:

```
#Equilibration step 4: MD on protein side chains
 &cntrl
  imin=0, ntr=1, nstlim=10000, dt=0.002,
  ntb=2, ntp=1, ntt=3, gamma_ln=0.2, temp0 = 300, ntf=2, ntc=2,
  irest=1, ntx=5
  restraint_wt=5,   restraintmask="@CA,N,C,O",
 /
```

Comparing with **eq3.in**, and referring again to the Amber10 manual, you will see the key points are:

Irest/ntx	We will be using the final velocities from the previous simulation as the initial velocities in this one.
Restraintmask	Only protein main chain atoms are restrained.

Edit **eq3.sub** as required to generate **eq4.sub**, and submit the job in the usual way. It should take about 10 minutes to run.

9. The final stage of the equilibration process is unrestrained MD on the whole system. Harking back to the presentation you will recall that a) here we are looking as much at the process of *structural* equilibration as *energetic* equilibration and b) the time for this to occur cannot be reliably predicted in advance. A suggested input file is provided – **eq5.in**:

```
#Equilibration step 5: 100 ps of unrestrained dynamics
 &cntrl
  imin=0, ntpr=500, ntwx=500, ntwe=500,
  irest=1, ntx=5, gamma_ln=0.2, ntwr=10000, nstlim=50000,
  ntb=2, ntf=2, ntc=2, ntp=1, ntt=3, temp0=300, dt=0.002
 /
```

You will see the key points are:

ntwx/ntwe We will collect coordinates and energies every picoseconds to monitor the equilibration process.

nstlim This is a 100ps run – almost certainly far too short for proper equilibration, but as long as is realistic for this workshop.

Prepare **eq5.sub**, and submit the job, in the usual way. It should take about 35-45 minutes to run.

When it has completed you will find the file 'mdcrd' in your directory – the trajectory file. If you have time, you can use the AMBER trajectory analysis tool *ptraj* to generate rmd plots. A detailed discussion of trajectory analysis is the subject of another workshop, but for now, if you type:

```
% ptraj < ptraj1.in
```

Then:

```
% ptraj < ptraj2.in
```

You will find two files created : **rms_from_first.dat** and **rms_from_avg.dat** which give backbone rms deviations of each snapshot in the trajectory file from a) the starting structure

and b) the time-averaged structure. Plot these using *gnuplot* or *xmgrace* and see how they compare with the plots discussed in the presentation.