

## Setting up a protein simulation

### Session 3 - Equilibration

#### **Practical session: equilibration using GROMACS**

In this session you will work through an example equilibration protocol for 1kyn performing energy minimisation and molecular dynamics using GROMACS3. The module that runs simulations is within GROMACS is called *mdrun*.

In the directory `~/gromacs/session3` you will find the various parameter and command files required for this part of the workshop.

Copy the coordinate (**system.gro**) file that you created in the last session into this directory. We have copied the other files to this directory for you.

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.

Before we make the input file for the energy minimisation, let's take a look at the command file; the **equil1.mdp** file.

Key points to note are:

1. The `DPOSRES_PROT` flag tells *mdrun* to apply positional restraints to the protein (this is done by reading the `posres.itp` file), which is sourced in the top file.
2. The integrator is set to 'steep' this tells *mdrun* that we wish to perform a steepest descents energy minimisation. The other option would be 'cg' to perform a conjugate gradients energy minimisation. If you have time at the end of the workshop, please take some time to look up the difference between the two minimisation algorithms, using online resources.
3. The number of steps is set to 100, with a tolerance of 100. The minimisation will stop either when the 100 steps have been completed, or when the calculation has converge; whichever occurs first.

We will use the *grompp* program to create our input file. As with any GROMACS program, the '-h' flag will provide help and a description of the various options available. So let's try this first. Do:

```
% grompp -h
```

Please read through the output text. This should give you an idea of what the different flag mean.

Ok, let's create our input file now. Do:

```
% grompp -c system.gro -p system.top -f equil1.mdp -o  
equil1.tpr -maxwarn 5 -np 8
```

The **-np 8** option is telling grompp to create an input file that can be run on 8 processors. Please note this may not be appropriate for the machine on which you are running your simulation. If you are unsure about this step, please ask your systems administrator for advice.

Now we are ready to run the energy minimisation. To run the calculation we will use a job submission script; **equil1.jif**. Take a look at this file. It should look like this:

```
mdrun-mpi -s equil1.tpr -v
```

To submit the job, do:

```
% mpisub-mpi 1 equil1.jif
```

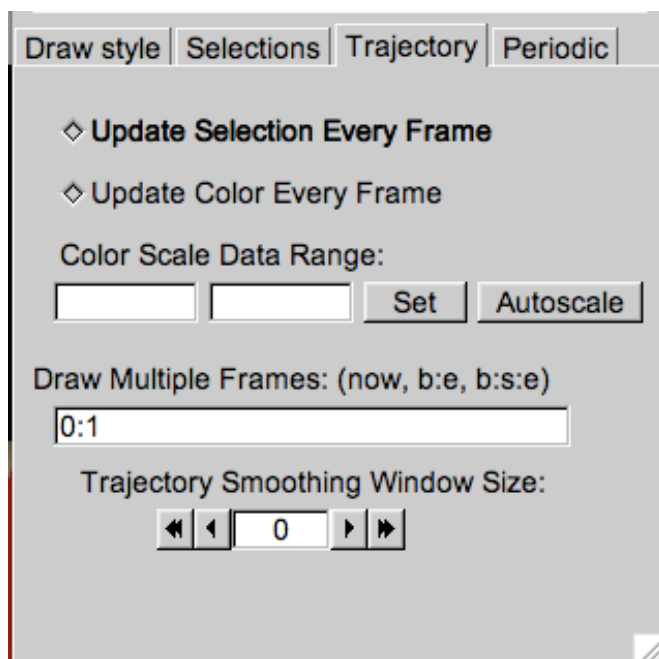
you can see the status of your job by doing:

```
% qstat
```

So once your energy minimisation has run to completion, take a look at the resulting coordinate file; the **confout.gro** in *vmd*. If you load in both the original structure and the minimised one, this will allow you to see what has moved and what is restrained:

```
% vmd gro confout.gro
```

(hint: from within the graphics menu, go to **representations>trajectory** then enter **0:1** in the **draw multiple frames** box to display both structures at once. Also, colour by timestep to make things slightly easier to visualise)



If all is well, we are ready to progress with the next stage of the equilibration procedure. We will now remove all positional restraints and energy minimise the whole system. This step should (!) clean up any poor geometry or clashes in the protein.

We will use the output from the previous step (**confout.gro**), as input for this calculation. To avoid confusion with the output from the new calculation let's rename **confout.gro** to **equil1.gro**:

```
% mv confout.gro equil1.gro
```

Now we will use the **equil2.mdp** command file. Take a look at the contents of the file. What is the difference between this file and the **equil1.mdp** file?

You will notice that the DPOSRES\_PROT flag has disappeared; this is because we are not applying positional restraints at all to the system in this energy minimisation.

So let's create the input file, as before do:

```
% grompp -c equil1.gro -p system.top -f equil2.mdp -o
equil2.tpr -maxwarn 5 -np 8
```

Then run the minimisation by creating a new job submission script called **equil2.jif** that looks like this:

```
mdrun-mpi -s equil2.tpr -v
```

submit the script as previously:

```
% mpisub 1 equil2.jif
```

Once again view the minimised and input systems in *vmd*.

To avoid confusion in future steps let's rename the **confout.gro** to **equil2.gro**:

```
% mv confout.gro equil2.gro
```

If all is in order we are ready to run some molecular dynamics to start heating up the system. Firstly we simulate the process of heating the system up to 300 K.

We will use the **md1.mdp** command file. Please take a look at the file. You should see that the integrator has now changed from 'steep' to 'md'. This is to tell *mdrun* that we will be performing molecular dynamics and not an energy minimisation. The **-DPOSRES\_PROT** flag is set once again; this is to apply positional restraints to the protein.

Other points to note:

- 1- the **nsteps** (number of steps) is set to 20000, the **dt** (integration timestep) is set to 2fs. This gives us a simulation time of  $20000 \times 2 = 40\text{ps}$ .
- 2- This is a constant pressure and temperature simulation. The initial temperature is 50 K (**gen-temp**), this will be raised to 300 K (**ref -t**) during the simulation. The pressure will be maintained at 1 atm (**ref-p**). One of the aims of this step is to equilibrate the solvent box to fill any voids created that may have been created during the process of system setup.
- 3- Additional information about the parameters in the mdp file can be found in the GROMACS manual (freely available online). Please do read through this section carefully if you plan to use GROMACS for your research, remember the output is only as accurate as the input!

Let's create the input file and run the simulation.

```
% grompp -c equil2.gro -p system.top -f md1.mdp -o  
equil3.tpr -np 8
```

and now create a new job submission script **called equil3.jif**, that looks like:

```
mdrun-mpi -s equil3.tpr -v
```

Now submit the job.

Ok, once this has run, check the output in *vmd*.

Again, to avoid confusion let's rename the output file to **equil3.gro**.

Ok, so now we will run another short MD equilibration step, this time be restraining just the backbone of the protein, i.e. allowing the side chains to relax. We need a new positional restraints file for this step. To create this file we will need to first create an index file. Do:

```
% make_ndx -f equil3.gro
```

then hit 'q' to save and exit.

Now let's create the positional restraints file. Do:

```
% genrestr -f equil3.gro -n index.ndx -o ca.itp
```

select the **C-alpha** option.

We now have our new positional restraints file; **ca.itp**

We will now edit the **system.top** file so it points to the **ca.itp** file and NOT the **posre.itp** file:

```
;
; Include forcefield parameters
#include "ffG43a1.itp"

; Include chain topologies
#include "protein.itp"

#include "ca.itp"
#endif

; Include water topology
#include "spc.itp"
```

Now let's create the input file:

```
% grompp -c equil3.gro -p system.top -f md1.mdp -o  
equil4.tpr -np 8
```

Create a job submission script called `equil4.jif`, that looks like:

```
mdrun-mpi -s equil4.tpr -v
```

Submit the job:

```
% mpisub 1 equil4.jif
```

Once again rename the output file:

```
% cp confout.gro equil4.gro
```

Once this has finished running, check the output in *vmd*. If all is in order, we are ready to proceed with the final equilibration step. This will be a short molecular dynamics run without any positional restraints.

Let's create the input file:

```
% grompp -c equil4.gro -p system.top -f md2.mdp -o  
equil5.tpr -np 8
```

And run the final step using a job submission script called **`equil5.jif`** that looks like:

```
mdrun-mpi -s equil5.tpr -v
```

submit:

```
% mpisub 1 equil5.jif
```

Once this has finished running you have a equilibrated system ready for a production run simulation!