

Setting up a protein simulation

Session 2 - Adding water and ions

Practical session: setting up the full system using GROMACS

In this session you will learn how to create a topology for your protein and how to add water and ions to your system using GROMACS3 utilities. We will be using **the pdb2gmx, genbox and genion** utilities.

In the directory `~/gromacs/session2` you will find the corrected protein coordinate file; **1kyn_correct.pdb** and the other files you will need to set up your system.

First of all let's create a topology for our protein. This defines which atom is bonded to which and allows us to assign atom types, angles etc. This is very straightforward to do in GROMACS using the *pdb2gmx* utility. As with all GROMACS programs, the 'h' flag will provide a description of the options available. So do this first to learn about what we will be doing:

```
% pdb2gmx -h
```

You should see that the `-ignh` option will allow us to ignore any hydrogen atoms in the pdb. Although we have used WHAT IF in session1 to assign ionisation states already, we will now do this using the automatic protocol in *genion*. The `-inter` option allows us to decide the ionisation states of sidechains interactively, so this option would allow you retain the ionisation states assigned by WHAT IF. We don't have time to do this today, but please do explore this option if you are using GROMACS for your research as subtle changes in ionisation state can have profound structural implications!

Let's create our topology:

```
% pdb2gmx -f 1kyn_correct.pdb -ignh  
(we will use the G43a1 force field)
```

Please make a note of the charge on your protein. The output file is called **conf.gro**.

Before proceeding to the next stage we must edit the **topol.top** file manually. First of all delete the line that reads:

```
#include "ffG43a1.itp"
```

Secondly change
'Protein_A' to 'Protein'

So that the first few lines now look like:

```

;
;   File 'topol.top' was generated
;   By user: onbekend (0)
;   On host: onbekend
;   At date: Mon Oct 26 16:52:57 2009
;
;   This is your topology file
;   DUMMY<cr>
;
; Include forcefield parameters

[ moleculetype ]
; Name          nrexcl
Protein         3

[ atoms ]
;  nr          type  resnr  residue  atom  cgnr      charge      mass  typeB  chargeB  massB
;  1           NL    1      ILE     N     1         0.129      14.0067 ; qtot 0.129
;  2           H     1      ILE     H1    1         0.248      1.008   ; qtot 0.377
;  3           H     1      ILE     H2    1         0.248      1.008   ; qtot 0.625
;  4           H     1      ILE     H3    1         0.248      1.008   ; qtot 0.873
;  5           OH    1      ILE     O     1         0.129      13.010   ; qtot 1.002

```

Delete the line

; Include Position restraint file

and everything below it, so that the end of the file looks like:

```

2315  2313  2328  2316      2  gi_2
2319  2318  2321  2320      2  gi_1
2319  2325  2322  2321      2  gi_1
2321  2324  2323  2322      2  gi_1
2321  2327  2326  2325      2  gi_1
2328  2315  2330  2329      2  gi_1

```

Now rename the file to **protein.itp**:

```
%mv topol.top protein.itp
```

Before we can add water to the system, we must place the protein inside a box. We will use a cube (10 x 10 x 10 nm)- remember GROMACS works in nm NOT angstroms:

```
% editconf -f conf.gro -box 8 8 8 -o boxed.gro
```

The output file is called **boxed.gro**. You can visualise this in *vmd*. Check you have a box by using the periodic option under `graphics>representations>periodic`. Now replicate the box in the x,y and z dimensions. If all is in order, we are ready to add water to the box containing protein. Do:

```
% genbox -cp boxed.gro -cs spc216.gro -o solvated.gro
```

Please make a note of how many water molecules are added. Please note that we have added spc waters (many water models exist, each with strengths and weakness, the one used must be know to work well with your protein force-field).

Ok, before going any further let's add the number of water added to our **system.top** file. This file keeps account of how many molecules of each we have in the system. Water molecules are called 'SOL' in GROMACS. So for example if 3077 water molecules were added to the system edit your **system.top** file to reflect this addition:

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

#ifdef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
; i funct      fcx      fcy      fcz
  1   1      1000      1000      1000
#endif

; Include generic topology for ions
#include "ions.itp"

[ system ]
; Name
Protein

[ molecules ]
; Compound      #mols
Protein          1
SOL              32077
```

Check to see all is in order using *vmd*. Visual inspection is very important as most large errors can be identified simply by looking at your system.

Ok, so let's add some ions to neutralise our system. We will need to create a tpr file first. To do this use the *grompp* program (we will explore this program in more detail in session 3):

```
% grompp -c solvated.gro -p system.top -f equil1.mdp -o
genion.tpr
```

The output file should be called **genion.tpr**. Now let's add 23 Cl⁻ ions to neutralise the system. Do:

```
% genion -s genion.tpr -nn 23 -o system.gro
```

When *genion* ask for a continuous group of solvent molecule, please select **SOL**.

Ok, so the ions have been added to the system by replacing 23 water molecules. In GROMACS the Cl⁻ ions are called **CL-**, so let's edit the **system.top** file to reflect the changes. We need to take 23 off the number of water molecules and to add a line to show the addition of 23 **CL-** ions.

So the end of the **system.top** file should look like:

```
; Include generic topology for ions
#include "ions.itp"

[ system ]
; Name
Protein

[ molecules ]
; Compound      #mols
Protein          1
SOL              32054
CL-              23
```

Please check in *vmd* that all is ok.

If so, you are now ready to proceed to session3!