PCLab 2 – Molecular dynamic

From preparation to production

Note: you will need some basic Linux knowledge for this practical since all the tools that will be used are command lines tools.

External useful link :

- GROMACS tutorial: http://www.mdtutorials.com/gmx/lysozyme/index.html
- Second GROMACS tutorial (maybe more detailed): http://www.strodel.info/index_files/lecture/html/tutorial.html
- Linux basic commands: https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners

It is well admitted that prolines are "alpha-helices breakers", which means that they will alter the conformation of an Alpha helix.
We will try to understand why prolines can "break" a helix with molecular dynamics simulation. While this PCLab will be focused on Running a MD, the next one will be focused on analyzing MD results and fully answering this question.

The system used for this PCLab is the nucleosome-interacting C-terminal alpha-helix (PDB ID 6UCH) which is an alpha helix that was obtained by NMR. The advantage of this taking a liquid state NMR structure is that the system was studied in solution and the helix should be stable in solution (so stable with MD as well).
Only the amino aicds belonging to the helix was keps and the amino acid numbering restarted from 1.
The GLN12 was replaced by a Proline (cf. Fig 1). We will not study the stability of the mutated helix with molecular dynamics simulation (both simulation would be too long, but the MD files will be provided in the next practical).

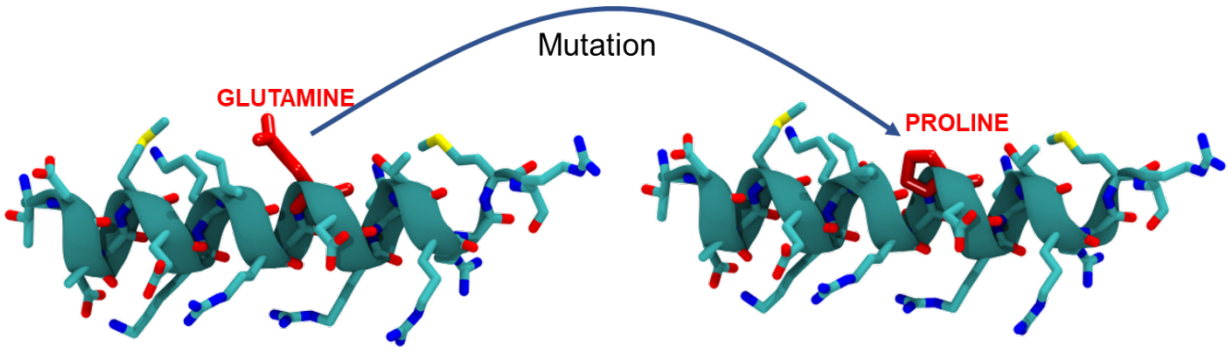

Fig 1 - Position of the proline in the nucleosome-interacting C-terminal alpha helix

FILES for this practical :

- **helixPRO.pdb** → the mutated alpha Helix (Q12P, GLN 12 mutated into a PRO).
- **mdp/** → Folder that contains all parameters files for ionization, minimization, equilibration, and production.

0. Before starting

Command (or action) to execute

Question to answer

0.1 GROMACS pipeline

Open source and completely free for both academic and industrial, GROMACS is one of the most used molecular modelling software.
MD with GROMACS is iterative, which means that before running the calculation, you have to generate all parameters, topology, etc..

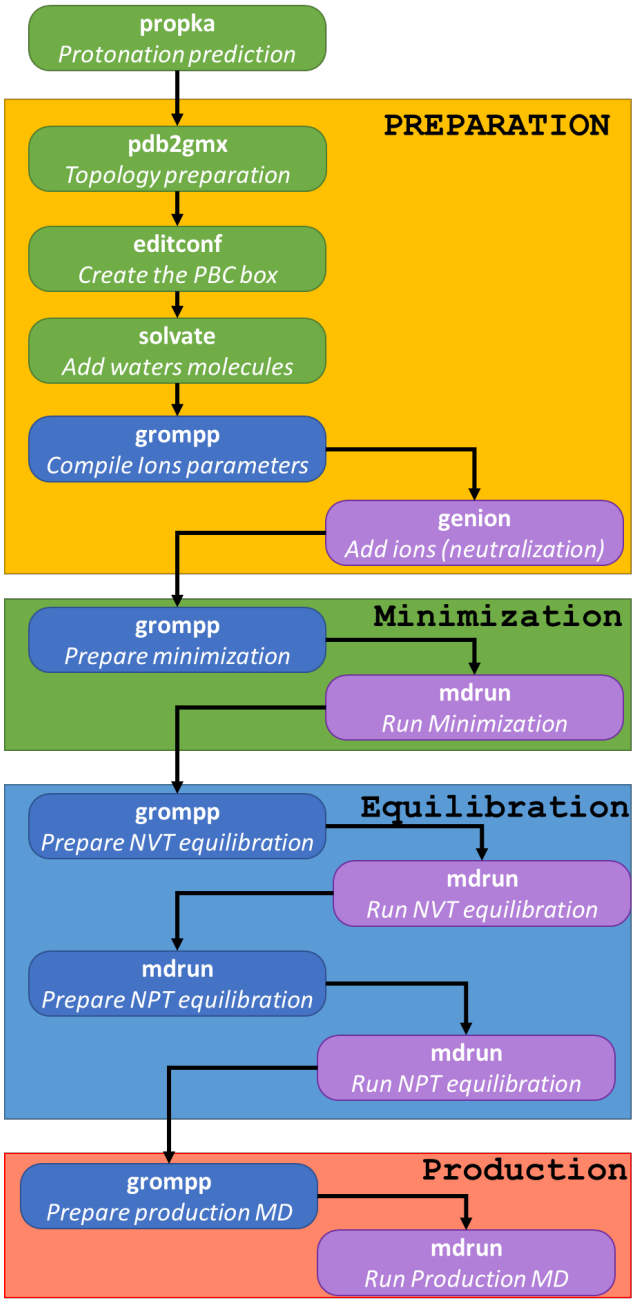

Fig 2 - GROMACS working pipeline. Reproduced and adapted from http://www.strodel.info/index_files/lecture/html/setup.html (originaly made by Oliver Schillinger). You can check his Gromacs tutorial if you need more explanation on some part, it's a very good one!

0.2 Genion and mdrun execution

In GROMACS, you need to "configure" a calculation before running it (*this phase is also called compiling)*. You need to gather all the parameters for the simulation, contained in a `mdp` file, the structure of your system of course and the topology that describe your system (how many bonds, which atoms are connected together….).

***Note about MDP files**: You don't have to put all parameters, if you don't specify some parameters GROMACS will use the default value for each non-specified parameters*

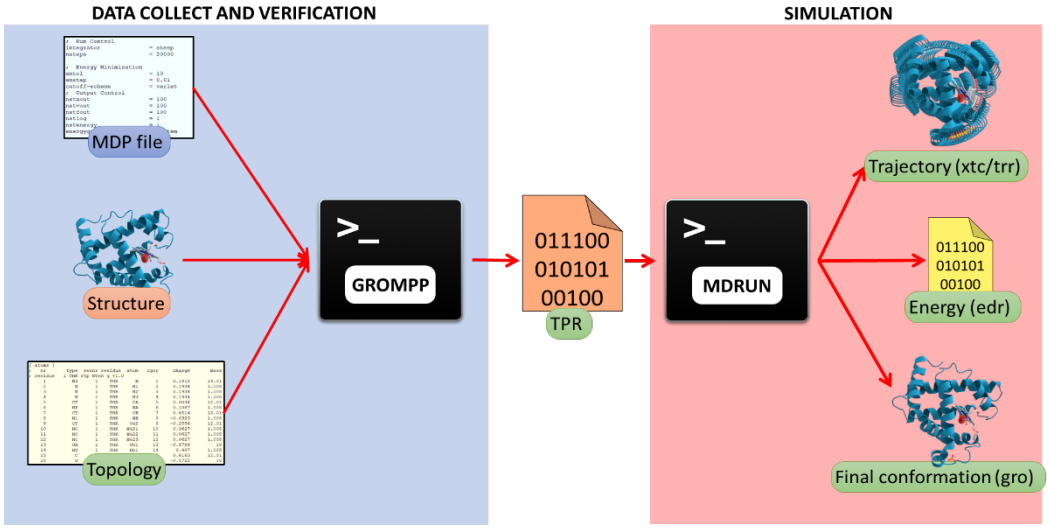

Fig 3 - Representative diagram of the preparation and execution of a simulation with GROMACS.

The software that gathers and checks the file before running the MD is called **grompp** and it will generate a **tpr** file (for **T**opology **P**ortable **R**un file). This is a binary file (*you can't read it with a regular text editor*) and it contains every parameter for the simulation, the initial coordinates, velocities, topology, restraints…..

This file is then given to MDRUN that will execute the MD based on the content of the tpr file. This workflow is described in Fig 3.

1. Preparation: protonation, solvatation, neutralization and minimization.

1.1 Protonation.

The first step (which is not mandatory, but better if you want to be more biological relevant), is the prediction of the protonation state. You can use the software "`propka`" to predict the protonation state of your amino acids, but for today, we will use a webserver instead: `PDB2PQR` (since we will not need to install the software).

1. Go to http://nbcr-222.ucsd.edu/pdb2pqr_2.1.1/ and upload our working structure: helixPRO.pdb [tick "upload a PDB file" →choose a file → navigate to your working folder, click on helixPRO.pdb → open"].
2. Choose the force field
   For this practical, we will use the **Amber** force field. So better keep the same force field for everything!

3. Output naming shame
   Remember: in the force field, every amino acids are already parametrized, for each of their atoms. If you add or remove atoms, the software will realize that something is different and will not be able to process the file (or it will just use the "standard" protonation state and ignore the modification you've made.
   To bypass this issue, protonated and unprotonated amino acids were parametrized as well but recorded with a different amino acid name (eg: unprotonated TYR is TYM, protonated ASP is ASH….)
   >Since we will use **Amber** for the simulation, use this output naming scheme.
4. Available Options
   tick the "Add/keep chain IDs in the PQR file" option
5. pKa Options
   let the pH at 7 (neutral) and be sure that "use PROPKA" is ticked.
6. Click on Run!

The structure is small, you should have the results within 60 seconds, and you will have 3 **output files**.

- helixPRO.propka → pKa and protonation state for every amino acid than can be charged
- helixPRO.pqr → PDB output (PQR is a slightly modified version of the PDB, but the format stay the same)
- helixPRO.in → Input file for Electrostatic map calculation. Not needed for this course.

Download "helixPRO.pqr" and save it in your working directory (*right-click on the link → save as)*

Q1: Open the file helixPRO.pqr with a Text Editor (like `gedit` for Linux, `TextEdit` for MacOS, `Notepad` for Windows but not Word or OpenOffice!), and observe the differences with "helixPRO.pdb".

a. What information can you see in the header (line starting with "REMARK")?

b. What type of atoms were added on each amino acid?

Note: for this peptide (small protein), the protonation state is standard at pH 7. You can start the process again but changing the pH to 1 to see the differences (ASP will be changed to ASH with one extra hydrogen atom, and GLU in GLH with one extra hydrogen atom as well), but do not use the pH1 file for calculation.

1.2 Conversion in GROMACS Format

Now that you have prepared your structure for GROMACS, it is time to convert it to the GROMACS format.

GROMACS don't take ".pqr" format, but remember, pqr format is almost identical to PDB format. So the first thing to do:

- Rename **helixPRO.pqr** to **helixPRO_proto.pdb** (change the extension, say yes if you have a confirmation window)

Now convert **helixPRO_proto.pdb** to **helixPRO_processed.gro** using pdb2gmx *("gro" format is specific to GROMACS, but the structure generated can be opened in most of molecular viewer softwares).*

a. Open a terminal (CTRL+T on Linux)
b. Navigate into your working directory (command `cd` followed by the folder name)
c. Run the command
```
gmx pdb2gmx -f helixPRO_proto.pdb -o helixPRO_processed.gro -ignh
```
d. When prompted, choose the **AMBER99SB-ILDN protein** Force Field by pressing "**6**" then press the touch "Enter" to validate
e. Choose the "TIP3P" water model (press "**1**" then "Enter")

Q2. Let's have a look at the generated output and files

a. From the output on the terminal, what is the total charge of the system?

b. From the output on the terminal, how many dihedrals, impropers dihedral, angles, bonds, and pairs there are?

c. How many files were produced?

d. Do you have an idea of what are they?

1.3 Creation of the simulation box

Now that the protein was converted and prepared for GROMACS (with a specific force field), it is time to create a simulation box. Since we have a long and thin peptide, we will use a cubic box so if the helix rotates, it will still have the "area" to move without too much border effects.

Generate the box with this command:

```
gmx editconf -f  helixPRO_processed.gro -o helixPRO_newbox.gro -d 0.8 -bt cubic
```

inputs:
- GROMACS Converted structure file
- Boxsize
- Box type

Outputs:
- Structure with box dimention

Q3. Let's have a look at outputs…

a. Here the minimum limit between any protein's atoms and one face of the box is 0.8nm (8Å). Do you think it is enough?

b. What is the box volume (check for "new box volume")

1.4 Solvatation

It is now time to file this box with water! Use this command :

```
gmx solvate -cp helixPRO_newbox.gro -cs spc216.gro -o helixPRO_solv.gro -p topol.top
```

Inputs
- Structure with box dimention encoded
- Water box model
- Topology

Outputs
- Structure box filled with solvent
- New topology with water information

You might ask “what is spc216.gro?” You can “normally” open this file with VMD with this comand:

```
vmd ~miniconda3/pkgs/GROMACS-2019.5-hc9558a2_0/share/GROMACS/top/spc216.gro
```

In case you have a different version of GROMACS or this command doesn’t work… Here’s the concent of spc216.gro in Fig 4.

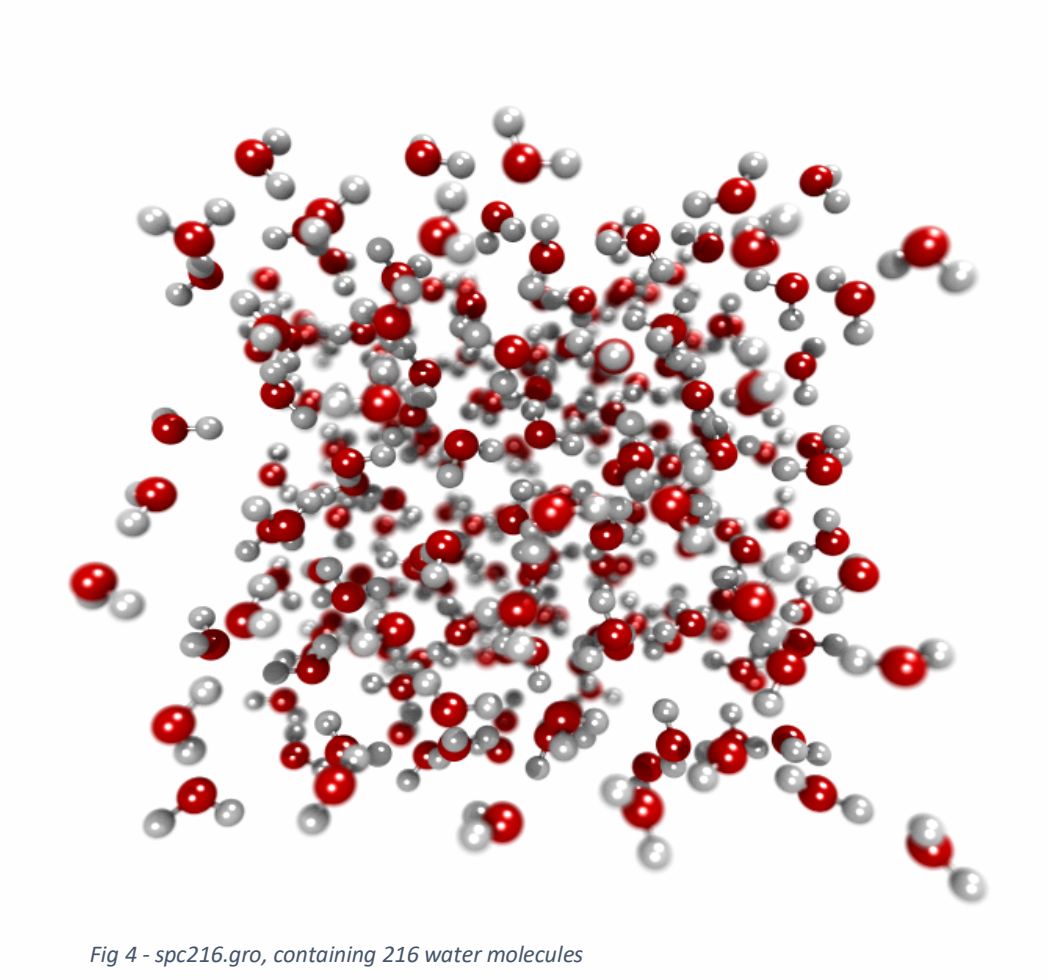

Fig 4 - spc216.gro, containing 216 water molecules

Q4. Solvatation.

a. Why GROMACS need this small water molecule box, `spc216.gro` (*check lecture 5 again if you don't remember)*?

b. From the terminal outputs: how many atoms solvent molecules were firstly added?

c. From the terminal outputs: How many water molecules do you have at the end

d. Why some water molecules were removed (*check lecture 5 again if you don't remember)*?

1.5 Ionization

The software that adds ion needs a "tpr" file. So we will use the software grompp to generate it (see Fig 3). To generate the tpr file write in your terminal:

```
gmx grompp -f mdp/ions.mdp -c helixPRO_solv.gro -p topol.top -o ions.tpr --maxwarn 1
```

Arguments:

- -f → mdp file containing all parameters
- -c → Input structure (c was used for "continuous")
- -p → Input topology
- -o → Output TPR file
- --maxwarn → Maximum warning authorized.

Inputs
- Genion parameters
- Structure with solvated box
- Topology

Outputs
- Compiled TPR file

Note:

- We have just generated the file "ions.tpr".
- grompp is mostly used to prepare the file for MD. It has a "checking" algorithm to evaluate a potential error that you can get. The argument `--maxwarn 1` is used to ignore the warning that you can have during this process. Here the warning is about electrostatic: the system net charge is not 0 so if you simulate a nonneutral charge system you can have severe artifacts. But during the ionization you will not simulate anything, just replace randomly some water molecules by ions. So you can ignore this warning

Now use genion to replace water molecules by ions.

```
gmx genion -s ions.tpr -o helixPRO_solv_ions.gro -p topol.top -pname NA -nname CL -neutral
```

When prompt, choose what you want to replace by ions : Solvent of course 😊! The group is 13 (or 12). Write "13" then press "Enter".

Arguments:

- -s → Topology Portable Run file
- -o → output file, it will be called
- -p → New topology that will be created (since you replace some water molecules with ions, you have to recreate the topology)
- -pname → Type of Positive ion (here $NA^{+)}$
- -nname → Type of Negative ion (here CL^{-})
- -neutral → We want to neutralize the system, this option calculates the number of necessary ions to neutralize the system (otherwise you have to specify the number of each ion by yourself).

Inputs
- Compiled TPR file
- Ions type
- Concentration (or neutral)

Outputs
- Solvated system with ions
- New topology with ions

Q5. Ionization

a. How many ions were added? What type of ions is it?

b. what is the number of water molecules that have been replaced?

c. What is the random seed used?

d. Compare the random seed with someone else (*or execute the* `grompp` *and* `genion` *command once again)*. Are the replaced water molecules the same with another random seed?

e. What can you do to have the same water molecules replaced at every execution (which means “reproducible”)?

f. What is the total number of atoms in your system now (you can also check this value by opening helixPRO_solv_ions.gro in VMD)?

1.6 Minimization

Since we have now finished the preparation of the simulation box, we can minimize it to remove the potential clashes and optimize the geometry of all molecules.

All parameters for MD calculation are stored in a "em.mdp" file. You can open this file in a text editor or refer to Box 3 (page 19).

Compile and generate the TPR file with grompp

```
gmx grompp -f mdp/em.mdp -c helixPRO_solv_ions.gro -p topol.top -o em.tpr
```

And run the minimization

```
gmx mdrun -v -deffnm em -nt 4
```

for details about input/output for Grompp and MDRUN, see Fig 3

Arguments:

- `-v`: "verbose", it will give you realtime feedback on the terminal
- `-deffnm`: basename for your input and output file (basename = file name without extension. Here the base name will be "em", the input is "em.tpr" and outputs: "em.trr"…
- `-nt`: number for cores for calculation. PCLab computers have 4 cores.

Q6. Minimization

a. From the MDP file: Which minimization algorithm is used? When the *steepest descent* steps are performed?

b. How many steps did you need to reach the minima?

c. What is the convergence criteria here (force, GRMS…)?

d. What is the starting potential energy?

e. What is the final potential energy?

2. Equilibration

When simulating, it is important to match the experimental conditions. Most of the time, experiments are done at constant temperature and pressure, so your simulations have to be done at constant temperature and pressure.

The equilibration process aims to bring "energy" to your system. Right now, you don't have any velocity, temperature or pressure. It is like your system is "freeze". It is now time to "unfreeze" it!
You usually equilibrate your system in 2 steps :

- **NVT**: first you bring the temperature to your system (remember from the lecture: temperature is linked to velocity, so you will also add velocities on each atom)
- **NPT**: Once the temperature is equilibrated, you can equilibrate the pressure.

You can also equilibrate the temperature and the pressure in 1 step, but you increase the chance that your system crash…

2.1 NVT equilibration

All the parameters for NVT equilibration are store in "nvt_300.mdp" (or refer to Box 4 - nvt_300.mdp page 20)

You can have the description of all those parameters with the online documentation (http://manual.gromacs.org/documentation/2019/user-guide/mdp-options.html) but I would like to emphasize on several parameters :

- the "integrator" is the algorithm that you use. Here it is MD (for Molecular Dynamic), which is a Leap-frog algorithm (`md-vv` for Velocity Verlet)
- For more details on `Nonbonded settings` (neighbor search), please refer to 4.1 Two extra things about MD (extra from the lectures): (page 16).
- Same for `PME`, please read 4.1 Two extra things about MD (extra from the lectures): (page 16).
- Look at "`Velocity generation`": it is set to "`yes`", and the temperature is 300K (around 27°C). This is also here where you set-up the random seed for MD (`gen_seed`). -1 is random.
- The timestep here is 2fs.
- The line "`define = -DPOSRES`" is the definition of restraint forces. You can check "4.2 Restraints in GROMACS. Page 16

Commands :
Compiling and preparing the input md file :

```
gmx grompp -f mdp/nvt_300.mdp -c em.gro -r em.gro -p topol.top -o nvt_300.tpr
```

New argument:

- `-r`: restrain file: this file contains the coordinate of the atoms that are restraint during the equilibration.

Runing the equilibration:

```
gmx mdrun -deffnm nvt_300 -v -nt 4
```

Q7. NVT Production :

a. From the mdp file, which thermostat was used?

b. How many steps are there for the equilibration?

c. how much is that in simulation time?

d. If the timestep was 4fs, what would have been the simulation time then?

e. How much time did it take to finish the NVT equilibration?

f. Do you understand the difference between “Core t (s)” and “Wall t (s)” ?

g. What is the performance of the simulation? Do you understand what the unit means?

Just for comparison, those calculations can be done on GPU (graphic card) if GROMACS was compiled for GPU usage, on 8 cores (AMD Ryzen 7 3700X) + Nvidia RTX 2070 SUPER, the performance was of 471ns/day for this system ….

2.1 NPT Equilibration

All the parameters for NVT equilibration are stored in “npt.mdp” (or refer to Box 5 - npt.mdp page 21).
On the MDP file, you can see that this time, we do not generate velocity (we keep the velocity that was generated during the NVT equilibration).

Same as for NVT Equilibration, you will use the output structure from the NVT this time (nvt_300.gro), to restart from the last step of the NVT equilibration but this time you will equilibrate the pressure
Note: If you compare “`em.gro`” and “`nvt_300.gro`” you will see 3 extra columns. Those columns are the velocities for each atom in each direction (X, Y, Z). This is all you need to restart or continue a molecular dynamics simulation.

Commands :

Compiling and preparing the input md file :

```
gmx grompp -f mdp/npt.mdp -c nvt_300.gro -r nvt_300.gro -p topol.top -o npt.tpr
```

Runing the equilibration:

```
gmx mdrun -deffnm npt -v -nt 4
```

Q8. NPT Production:

a. Look the mdp file: Which barostat was used for the equilibration?

b. Are the atoms restraint (*see 4.2 Restraints in GROMACS.*) ?

c. Which atoms are restraint ?

3. Production

Production is the final step. You can remove all restraints and let the system explore its conformational space. Some small changes have been made in the MDP files (see Box 6 – md_prod.mdp page 22).

Commands :

```
gmx grompp -f mdp/md_prod.mdp -c npt.gro -r npt.gro -p topol.top -o md_helixPRO.tpr
```

Running the simulation

```
gmx mdrun -v -deffnm md_helixPRO -nt 4
```

Q9 – MD Production:

a. From the MDP file: Which thermostat and barostat is used ? Why ?

b. From the MDP file: are the atoms still restraint ?

c. When the calculation will be finished?

d. Calculate roughly the performance (ns/day)

NOTE:
The calculation is taking too much time, so for the next practical (analysing the results), I will generate all the MD files for you.

4. Appendix

4.1 Two extra things about MD (extra from the lectures):

Neighbor search

Remember from the lecture: to reduce the calculation of the non-bonded interactions you will use a cutoff distance, and you will ignore every atom far away from this cutoff distance.
Of course, you need to know which atoms are "neighbor" (which means which atoms an under this cutoff distance). So for every atom, you have to calculate the distance with every other atom to get this neighbor list. This is of course costly.
In MD, this neighbor list is calculated every *X* steps, *X* to be defined by the user (since two atoms will not move much under a certain number of steps).

PME (Particle Mesh Ewald)

Ignoring electrostatic potential beyond the cutoff value can lead to severe errors. That's why, after the cutoff values, another algorithm is used to calculate the electrostatic potential.
Based on Ewald's method, it consists of adding a virtual grid to the system, then for each point of the grid, interpolate their charge. The electrostatic potential is then calculated in the reciprocal space, after a Fourier transform, and the forces interpolated back to atoms in real space, whose positions are updated.

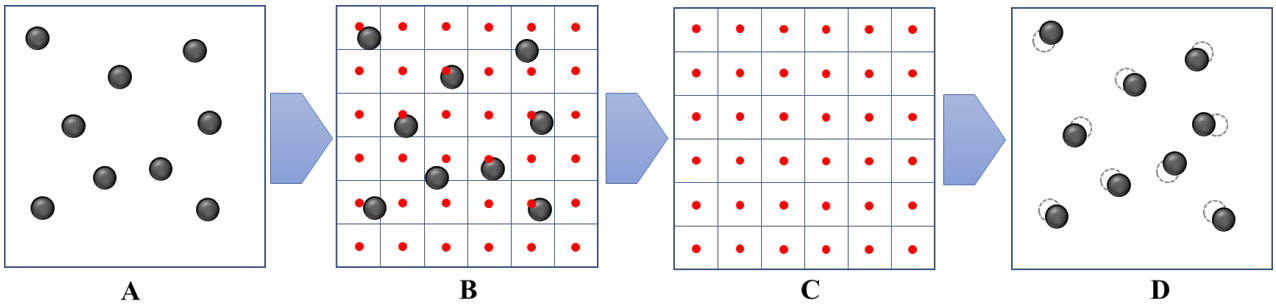

Fig 5 - PME approach on a two-dimensional network. (A) Charged particle system. (B) Charges are interpolated on a two-dimensional grid. (C) Potential and forces are evaluated at each point of the grid in reciprocal space with a Fourier transform. (D) The forces are interpolated back (inverse Fourier transform) to the particles, whose positions are then updated. Adapted from Christophe Chipot's lectures (French) (http://ecole.modelisation.free.fr/cours_chipot.pdf)

4.2 Restraints in GROMACS.

During the equilibration, you increase the temperature by scaling the velocities. If the velocities are too high, your system can crash (lead to an explosion). To avoid that kind of artifact, you have to apply restraints to all atom's positions.
The restraint is a Force applied on each restrained atoms, to remove the movement as much as possible. (*cf. lecture 5, slide 80).* You will have a "shaking" effect on all atoms.

But how does it work with Gromacs?

After you used pdb2gmx, a file called "posre.itp" (for position restraint) is generated. If you open it with a text editor you will find this :

Box 1 – head of posre.itp

```
; In this topology include file, you will find position restraint
; entries for all the heavy atoms in your original pdb file.
; This means that all the protons which were added by pdb2gmx are
; not restrained.

[ position_restraints ]
; atom  type      fx      fy      fz
     1     1  1000  1000  1000
     5     1  1000  1000  1000
     7     1  1000  1000  1000
     9     1  1000  1000  1000
    13     1  1000  1000  1000
    15     1  1000  1000  1000
```

The header is quite informative: you will find restraints for all heavy atoms (*in MD, heavy atom = all non-hydrogen atoms).*
Then you have the field [position_restrains] with 5 columns.

- The atom number
- The type (*internal coding type for Gromacs*)
- The restraint force applied on X, Y, and Z (in $kJ.mol^{-1}.nm^{-2}$)

Now, look the end of the topology file (topol.top)

Box 2 - End of topol.top

```
; Include Position restraint file
#ifdef POSRES
#include "posre.itp"
#endif

; Include water topology
#include "amber99sb-ildn.ff/tip3p.itp"

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

The 4 first lines mean **"If you define a variable called POSRES, you include the restraint file".**

If you look at the position beginning of equilibration parameters files (`nvt_300.mdp` and `npt.mdp`) you will see the line "`define = -DPOSRES`". In those files, you say to the MD program "Define a Variable called `POSRES`" (`-D`, in the beginning, is an argument and therefore is not considered in the name of the variable).
It means that during equilibration, the file `posre.itp` will be integrated and you will have restraint forces applied on the atoms present in `posre.itp`.

Note :

The `posre.itp` file was created at the beginning of the process. You only have heavy atoms of the protein. Not the water. But in `topol.top` you have those lines :

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

It means that if you add "`define = -DPOSRES_WATER`" in the mdp files, the water will be restraint.

But maybe you have seen that we don't have this line on our MDP file… That's right! Water is free to move in our equilibration :-).

This is one way of doing the equilibration, some people like to restraint the water as well, but in that case, it requires more steps and we don't have the time during this practical.

4.3 Some advice on the MD pipeline

If you have a big or complex system (with a lot of atoms, or non-standard molecules like modified amino acids or other molecules), the equilibration may be a critical step and lead you to a crash. Do not hesitate to run multiple equilibration phase (like for the temperature: first at 150K, then 300K) with restraints on water molecules. You can also extend equilibration to 500ps for example.

Then before the production, you can have a "preproduction" phase when you remove restraints only on a water molecule, then on the full system for the production.

You will get something like this :

1. Preparation
2. Conversion
3. Box
4. Solvatation
5. Neutralization
6. Equilibration NVT 150K – 500ps
   Restraints on Protein + water
7. Equilibration NVT 300K – 500ps
   Restraints on Protein + water
8. Equilibration NPT – 500ps
   Restraints on Protein + water
9. Preproduction – 2ns
   Restraints on Protein only
10. Production - as long as you want.

But also remember that there is no universal recipe for MD, it always depends on your input system.

4.4 MDP files *(parameter files)*

A. Minimization parameters

Box 3 - em.mdp

```
;  Preprocessing
define                  = -DFLEXIBLE    ; defines to pass to the preprocessor

;  Run Control
integrator              = cg         ; Conjugate Gradient energy minimization
nstcgsteep              = 50         ; every 50 steps, do a steepest descent
nsteps                  = 1000          ; maximum number of steps to integrate

;  Energy Minimization
emtol                   = 1000          ; [kJ/mol/nm] minimization is converged when max force
is < emtol
emstep                  = 0.01          ; [nm] initial step-size
cutoff-scheme           = verlet
;  Output Control
nstxout                 = 100           ; [steps] freq to write coordinates to trajectory
nstvout                 = 100           ; [steps] freq to write velocities to trajectory
nstfout                 = 100           ; [steps] freq to write forces to trajectory
nstlog                  = 1             ; [steps] freq to write energies to log file
nstenergy               = 1             ; [steps] freq to write energies to energy file
energygrps              = System        ; group(s) to write to energy file

;  Neighbor Searching
nstlist                 = 100          ; [steps] freq to update neighbor list
ns_type                 = grid          ; method of updating neighbor list
pbc                     = xyz           ; periodic boundary conditions in all directions
rlist                   = 1.0           ; [nm] cut-off distance for the short-range neighbor
list

;  Electrostatics
coulombtype             = PME           ; Particle-Mesh Ewald electrostatics
rcoulomb                = 1.0           ; [nm] distance for Coulomb cut-off

;  VdW
vdwtype                 = cut-off       ; twin-range cut-off with rlist where rvdw >= rlist
rvdw                    = 1.0           ; [nm] distance for LJ cut-off
DispCorr                = Ener          ; apply long range dispersion corrections for energy

;  Ewald
fourierspacing          = 0.12          ; [nm] grid spacing for FFT grid when using PME
pme_order               = 4             ; interpolation order for PME, 4 = cubic
ewald_rtol              = 1e-5          ; relative strength of Ewald-shifted potential at
rcoulomb
```

B. NVT Equilibration

Box 4 - nvt_300.mdp

```
title                    = NVT Equilibration

define                   = -DPOSRES  ; position restrain the protein


; Run parameters
integrator               = md        ; leap-frog integrator
nsteps                   = 50000     ; 2 * 50000 = 100 ps
dt                       = 0.002     ; 2 fs

; Output control
nstxout                  = 500       ; save coordinates every 1.0 ps
nstvout                  = 500       ; save velocities every 1.0 ps
nstenergy                = 500       ; save energies every 1.0 ps
nstlog                   = 500       ; update log file every 1.0 ps

[..]

; Nonbonded settings
cutoff-scheme            = Verlet    ; Buffered neighbor searching
ns_type                  = grid      ; search neighboring grid cells
nstlist                  = 10        ; 100 fs, largely irrelevant with Verlet
rcoulomb                 = 1.0       ; short-range electrostatic cutoff (in nm)
rvdw                     = 1.0       ; short-range van der Waals cutoff (in nm)
DispCorr                 = EnerPres  ; account for cut-off vdW scheme


; Electrostatics
coulombtype              = PME       ; Particle Mesh Ewald for long-range electrostatics
pme_order                = 4         ; cubic interpolation
fourierspacing           = 0.16      ; grid spacing for FFT


; Temperature coupling is on
tcoupl                   = Berendsen             ; modified Berendsen thermostat
tc-grps                  = Protein Non-Protein   ; two coupling groups - more accurate
tau_t                    = 0.1     0.1           ; time constant, in ps
ref_t                    = 300     300           ; reference temperature, one for each group, in K

; Pressure coupling is off
pcoupl                   = no        ; no pressure coupling in NVT

; Periodic boundary conditions
pbc                      = xyz       ; 3-D PBC

; Velocity generation
gen_vel                  = yes       ; assign velocities from Maxwell distribution
gen_temp                 = 300       ; temperature for Maxwell distribution
gen_seed                 = -1        ; generate a random seed
```

C. NPT Equilibration

Box 5 - npt.mdp

```
; 7.3.2 Preprocessing
title                    = NPT Equilibration
define                   = -DPOSRES              ; defines to pass to the preprocessor

; 7.3.3 Run Control
integrator               = md                    ; md integrator
tinit                    = 0                     ; [ps] starting time for run
dt                       = 0.002                 ; [ps] time step for integration
nsteps                   = 50000               ; maximum number of steps to integrate, 0.002 * 50000 = 100 ps
comm_mode                = Linear                ; remove center of mass translation
nstcomm                  = 1                     ; [steps] frequency of mass motion removal
comm_grps                = Protein Non-Protein   ; group(s) for center of mass motion removal
cutoff-scheme            = verlet

; 7.3.8 Output Control
nstxout                  = 5000          ; [steps] freq to write coordinates to trajectory
nstvout                  = 5000          ; [steps] freq to write velocities to trajectory
nstfout                  = 5000          ; [steps] freq to write forces to trajectory
nstlog                   = 5000          ; [steps] freq to write energies to log file
nstenergy                = 5000           ; [steps] freq to write energies to energy file
nstxtcout                = 5000          ; [steps] freq to write coordinates to xtc trajectory
xtc_precision            = 1000          ; [real] precision to write xtc trajectory
xtc_grps                 = System        ; group(s) to write to xtc trajectory
energygrps               = System        ; group(s) to write to energy file

; 7.3.9 Neighbor Searching
nstlist                  = 10             ; [steps] freq to update neighbor list
ns_type                  = grid          ; method of updating neighbor list
pbc                      = xyz           ; periodic boundary conditions in all directions
rlist                    = 1.0           ; [nm] cut-off distance for the short-range neighbor list

; 7.3.10 Electrostatics
coulombtype              = PME           ; Particle-Mesh Ewald electrostatics
rcoulomb                 = 1.0           ; [nm] distance for Coulomb cut-off

; 7.3.11 VdW
vdwtype                  = cut-off       ; twin-range cut-off with rlist where rvdw >= rlist
rvdw                     = 1.0           ; [nm] distance for LJ cut-off
DispCorr                 = EnerPres      ; apply long range dispersion corrections for energy

; 7.3.13 Ewald
fourierspacing           = 0.16          ; [nm] grid spacing for FFT grid when using PME
pme_order                = 4             ; interpolation order for PME, 4 = cubic
ewald_rtol               = 1e-5          ; relative strength of Ewald-shifted potential at rcoulomb

; 7.3.14 Temperature Coupling
tcoupl                   = Berendsen             ; modified Berendsen thermostat
tc-grps                  = Protein Non-Protein   ; two coupling groups - more accurate
tau_t                    = 0.1     0.1           ; time constant, in ps
ref_t                    = 300     300           ; reference temperature, one for each group, in K

; 7.3.15 Pressure Coupling
pcoupl                   = Berendsen                    ; pressure coupling where box vectors are variable
pcoupltype               = isotropic             ; pressure coupling in x-y-z directions
tau_p                    = 1.0                   ; [ps] time constant for coupling
compressibility          = 4.5e-5                ; [bar^-1] compressibility
ref_p                    = 1.0                   ; [bar] reference pressure for coupling
refcoord_scaling         = com

; 7.3.17 Velocity Generation
gen_vel                  = no            ; velocity generation turned off

; 7.3.18 Bonds
constraints              = all-bonds  ; convert all bonds to constraints
constraint_algorithm     = LINCS      ; LINear Constraint Solver
continuation             = yes        ; apply constraints to the start configuration
lincs_order              = 4          ; highest order in the expansion of the contraint coupling matrix
lincs_iter               = 1          ; number of iterations to correct for rotational lengthening
lincs_warnangle          = 30         ; [degrees] maximum angle that a bond can rotate before LINCS will complain
```

D. Production

Box 6 – md_prod.mdp

```
title             = production MD for protein in explicit water

; Run parameters
integrator        = md              ; leap-frog algorithm
nsteps            = 2500000        ; 0.002 * 2500000 = 5000 ps or 5 ns
dt                = 0.002           ; 2 fs

; Output control
nstxout           = 5000               ; save coordinates every 10 ps
nstvout           = 5000               ; save velocities every 10 ps
nstxtcout         = 5000             ; xtc compressed trajectory output every 10 ps
nstenergy         = 5000             ; save energies every 10 ps
nstlog            = 5000             ; update log file every 10 ps
nstcomm           = 1000               ; center of mass motion removal

; Bond parameters
constraint_algorithm = lincs         ; holonomic constraints
constraints          = all-bonds     ; all bonds (even heavy atom-H bonds) constrained
lincs_iter           = 1             ; accuracy of LINCS
lincs_order          = 4             ; also related to accuracy

; Neighborsearching
ns_type           = grid             ; search neighboring grid cells
nstlist           = 25               ; with Verlet lists the optimal nstlist is >= 10, with GPUs >= 20.
rlist             = 1.0              ; short-range neighborlist cutoff (in nm)
rcoulomb          = 1.0              ; short-range electrostatic cutoff (in nm)
rvdw              = 1.0              ; short-range van der Waals cutoff (in nm)
rlistlong         = 1.0              ; long-range neighborlist cutoff (in nm)
cutoff-scheme     = Verlet

; Electrostatics
coulombtype       = PME              ; Particle Mesh Ewald for long-range electrostatics
pme_order         = 4                ; cubic interpolation
fourierspacing    = 0.16             ; grid spacing for FFT

; Temperature coupling is on
tcoupl          = V-rescale               ; v-rescale is used now to have a canonical space
tc-grps         = Protein Non-Protein     ; two coupling groups - more accurate
tau_t           = 0.1    0.1              ; time constant, in ps
ref_t           = 300    300             ; reference temperature, one for each group,in K

; Pressure coupling is off
pcoupl          = Parrinello-Rahman     ; pressure coupling is on for NPT
pcoupltype      = isotropic             ; uniform scaling of box vectors
tau_p           = 2.0                   ; time constant, in ps
ref_p           = 1.0                   ; reference pressure, in bar
compressibility = 4.5e-5                ; isothermal compressibility of water, bar^-1

; Periodic boundary conditions
pbc             = xyz                   ; 3-D PBC

; Dispersion correction
DispCorr        = EnerPres              ; account for cut-off vdW scheme

; Velocity generation
gen_vel         = no                    ; Velocity generation is off
gen_temp        = 300                   ; reference temperature, for protein in K
```