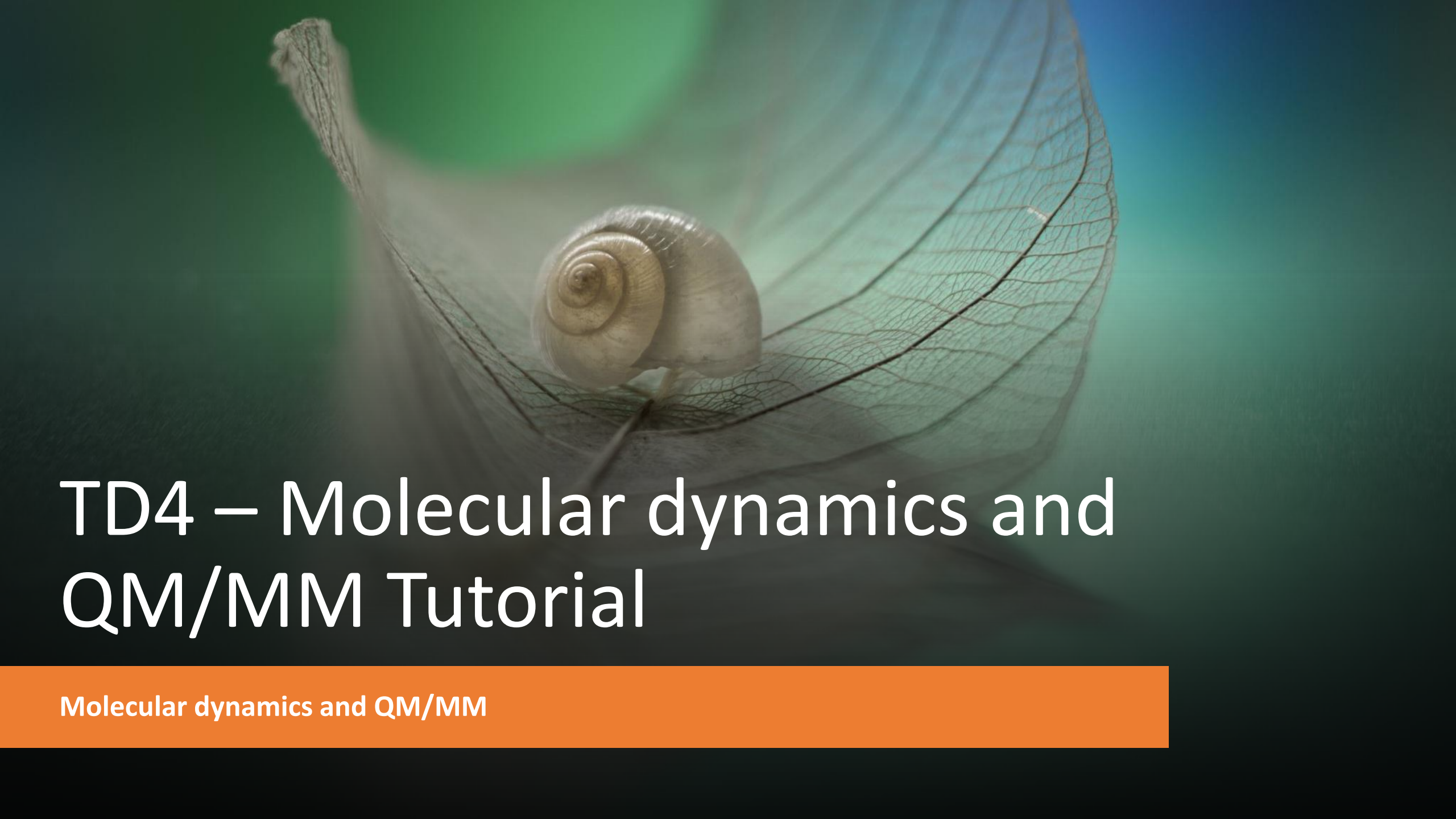


TD4 – Molecular dynamics and QM/MM Tutorial

Molecular dynamics and QM/MM

Create a folder with your name and copy the files needed for the tutorial

```
source .bashrc
mkdir josene
cd josene
cp ../tp4_files/* .
tar -xzvf 1yrf.tar.gz
cd 1yrf
cp inputs/* .
ls
l
```

How to submit a calculation to the queue

```
sbatch run_gromacs.job md
```

run_gromacs.job is the submission script.
where “md” is the input name that you want to submit to the queue.
It is convenient to submit the following calculations to the queue:
Energy minimization (em), thermalization (nvt), equilibration (npt) and production run (md)

Check the queue:

```
squeue -u tpsessions
```

Or simply:

```
sq
```

Watch your process:

```
watch squeue
```

`ctrl + c` (to stop watching)

Force fields

- **AMBER**
- CFF
- CHARMM
- COSMOS-NMR
- CVFF
- ECEPP
- GROMOS
- IFF
- MMFF
- MM2
- OPLS
- QCFF
- UFF
- ...

MD Software

- AMBER
- CHARMM
- COSMOS
- CP2K
- **GROMACS**
- GROMOS
- LAMMPS
- SPARTAN
- TINKER
- ...

en.wikipedia.org/wiki/Force_field_(chemistry)
en.wikipedia.org/wiki/Molecular_mechanics

Structure of a MD Program

FAST. FLEXIBLE. FREE.
GROMACS

Navigation

Quick search

Go

Welcome to GROMACS

A free and open-source software suite for high-performance molecular dynamics and output analysis.

New to GROMACS:

- Try the introduction tutorial.
- Watch the GROMACS webinar
- Download the current GROMACS version here.
- Have a look at documentation page to know more how to install and use GROMACS.
- Do you have any questions, have a look at the user discussions on GROMACS forums.

News:

"GROMACS 2023.3 is available." Read here for an overview of the patch release notes - 19 October 2023

GROMACS polls aim to understand if a feature or a tool has zero users or not. Any feature takes efforts during general refactoring. When it is unused and untested, it might end up broken for many releases causing a damage in the faith in the project. For this reason is important to let developers know if a tool/feature is used. Fill the poll on GROMACS analysis tools 3 October 2023

Interesed in alchemical transformation using AWH, try the tutorial on solvation free energy using AWH or in performing a membrane-protein simulation, check the tutorial introduction to membrane-protein simulation -27 July 2023

https://www.gromacs.org/

Installation

GROMACS must be installed on your machine. You can install it following the instructions of the GROMACS manual.

Alternatively, if you are using Ubuntu OS, you can simply execute the following command in a terminal:

```
sudo apt-get install gromacs
```

You can verify that GROMACS is indeed installed on your computer by typing in a terminal :

```
gmx
```

You should see the version of GROMACS that has been installed. On my computer I see

```
:-) GROMACS - gmx, 2023 (-:

Executable:   /usr/bin/gmx
Data prefix:  /usr

(...)
```

https://www.compchems.com/how-to-install-gromacs/#2-install-prerequisites

How to run a molecular dynamics using GROMACS

https://www.compchems.com/how-to-run-a-molecular-dynamics-simulation-using-gromacs/

The input files

To run the simulations using GROMACS, we need the following files:

- **configuration file** (.gro) containing the initial positions of the atoms and the box dimensions
- **topology file** (.top) specifying the location of the force field files (.itp)
- **input file** (.mdp) containing the parameters of the simulation (e.g. temperature, timestep)
- **force field parameters** (.itp) containing the information about the FF chosen (distances, angles, dihedrals…)

The configuration file (.gro)

A .gro file contains the initial positions and name of all the atoms of a simulation, as well as the box size and can be read by GROMACS. Its structure is the following:

```
Name of the system
number-of-atoms
residue-number residue-name atom-name atom-number atom-positions (x3) # first atom
residue-number residue-name atom-name atom-number atom-positions (x3) # second atom
residue-number residue-name atom-name atom-number atom-positions (x3) # third atom
(...)
residue-number residue-name atom-name atom-number atom-positions (x3) # penultimate atom
residue-number residue-name atom-name atom-number atom-positions (x3) # last atom
box-size (x3)
```

https://gromacstutorials.github.io/doc-sphinx/build/html/tutorials/createconfgro.html#create-conf-label

The topology file (.top)

The topology file contains information about the interactions of the different atoms and molecules. The topol.top file looks like that:

```
#include "ff/forcefield.itp"
#include "ff/h2o.itp"
#include "ff/na.itp"
#include "ff/so4.itp"

[ System ]
Na2SO4 solution

[ Molecules ]
SO4 6
Na 12
SOL 701
```

https://gromacstutorials.github.io/doc-sphinx/build/html/tutorials/bulksolution.html

The input file (.mdp)

The input file contains instructions about the simulation, such as:

- the number of steps to perform,
- the thermostat to be used (e.g. Langevin, Berendsen),
- the cut-off for the interactions (e.g. Lennard-Jones),
- the molecular dynamics integrator (e.g. steep-decent, molecular dynamics).

```
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
; Temperature coupling is on
tcoupl                   = V-rescale             ; 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
```

https://gromacstutorials.github.io/doc-sphinx/build/html/tutorials/bulksolution.html

The force field parameters file (.itp)

The forcefield.itp file is used to define basic combination rules, atom types, bond types, angle types….

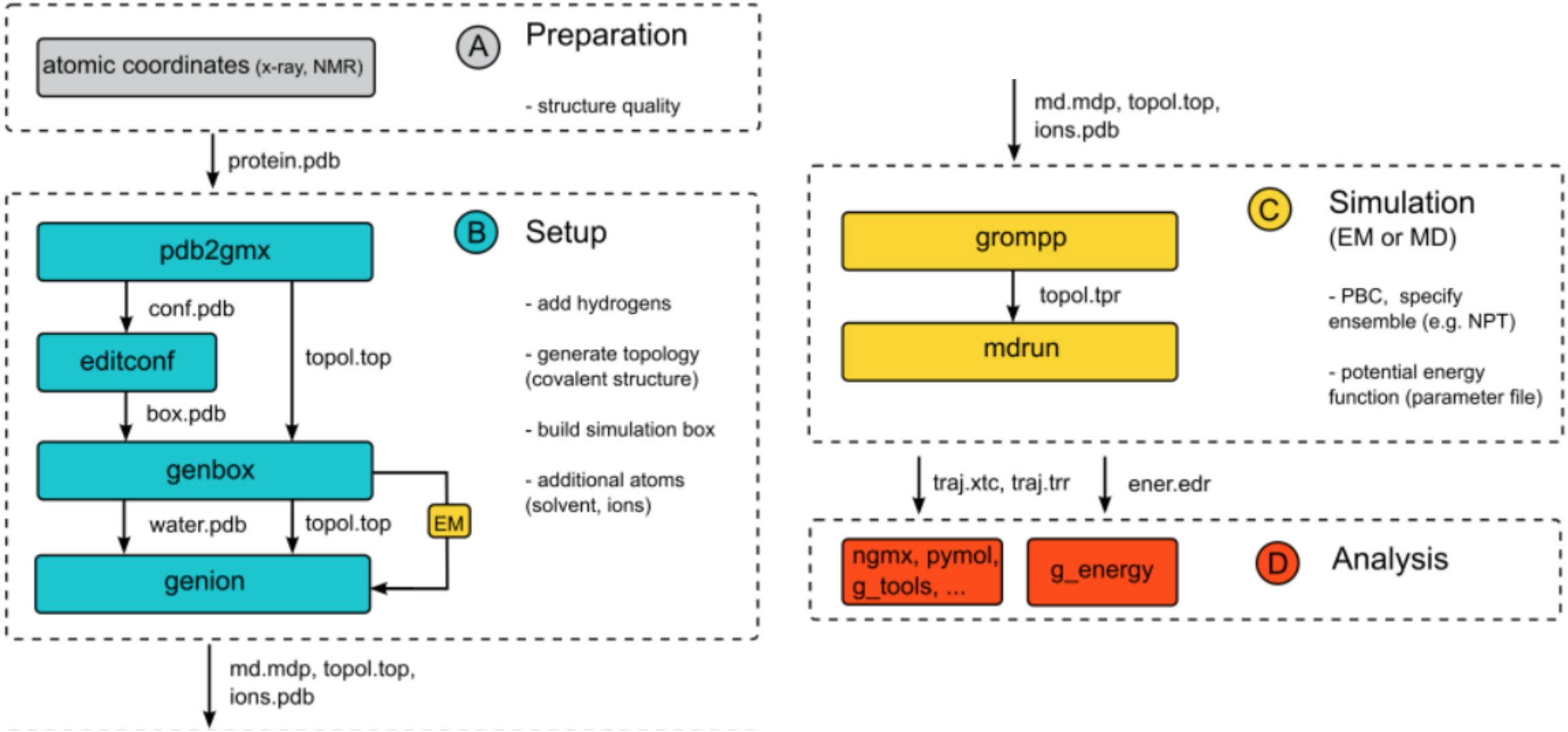
```
[ defaults ]
; nbfunc  comb-rule  gen-pairs  fudgeLJ  fudgeQQ
  1       2          no         1.0      0.833

[ atomtypes ]
; name  at.num  mass      charge  ptype  sigma    epsilon
  Na    11      22.9900   1.0000  A      0.23100  0.45000
  OS     8      15.9994  -1.0000  A      0.38600  0.12
  SO    16      32.0600   2.0000  A      0.35500  1.0465
  HW     1       1.0079   0.5270  A      0.00000  0.00000
  OW     8      15.9994   0.0000  A      0.31650  0.77323
  MW     0       0.0000  -1.0540  D      0.00000  0.00000

[ bondtypes ]
; i   j   func  b0    kb
  SO  OS  1     0.15  3.7656e4

[ angletypes ]
; i   j   k   func  theta  k0
  OS  SO  OS  1     109.5  520
```

https://gromacstutorials.github.io/doc-sphinx/build/html/tutorials/createtopoltop.html

MD of a protein in water environment

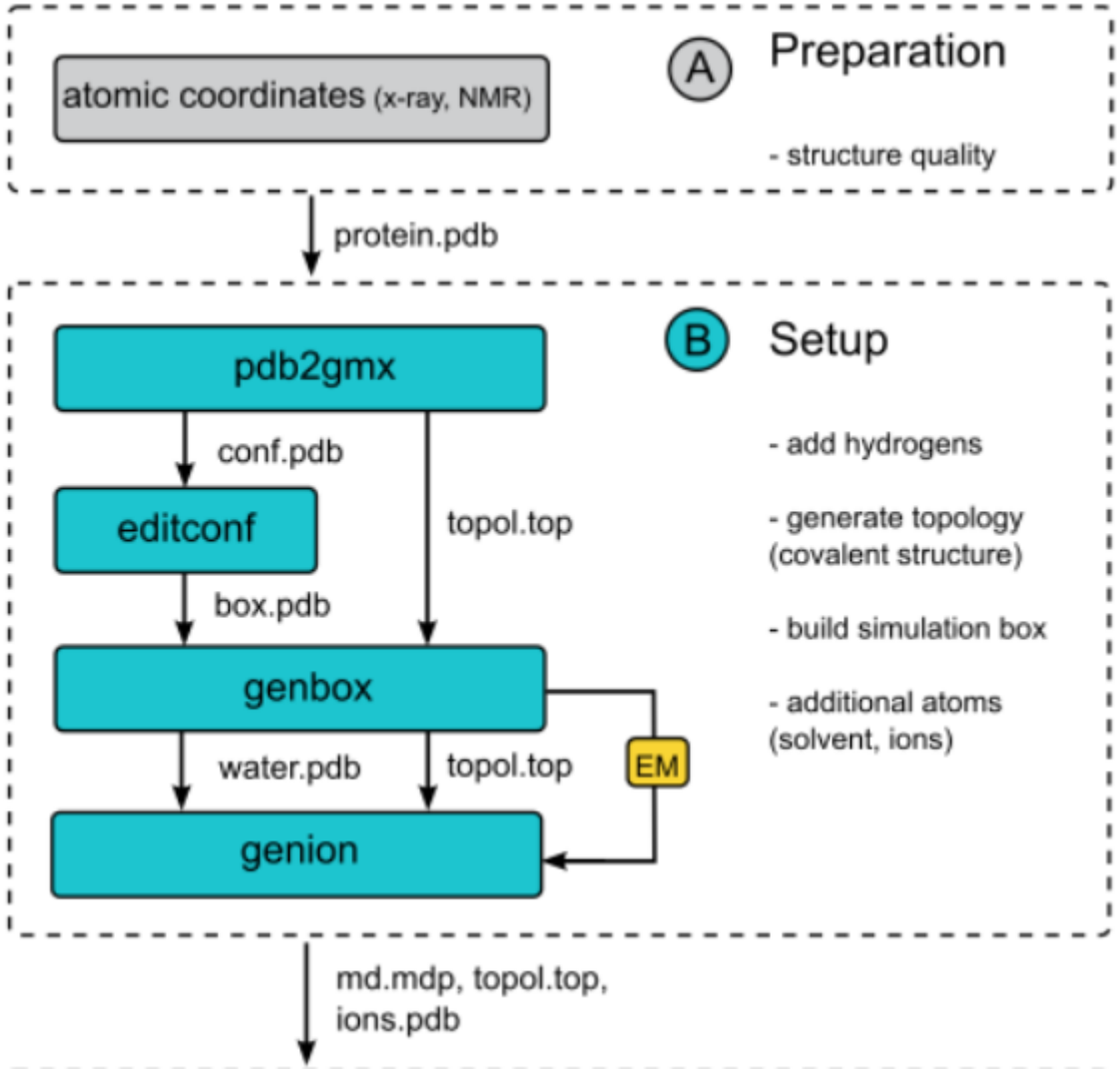
available at:
https://www.compchems.com/gromacs-tutorial-molecular-dynamics-simulation-of-a-protein-in-water-environment/#background-for-molecular-simulation

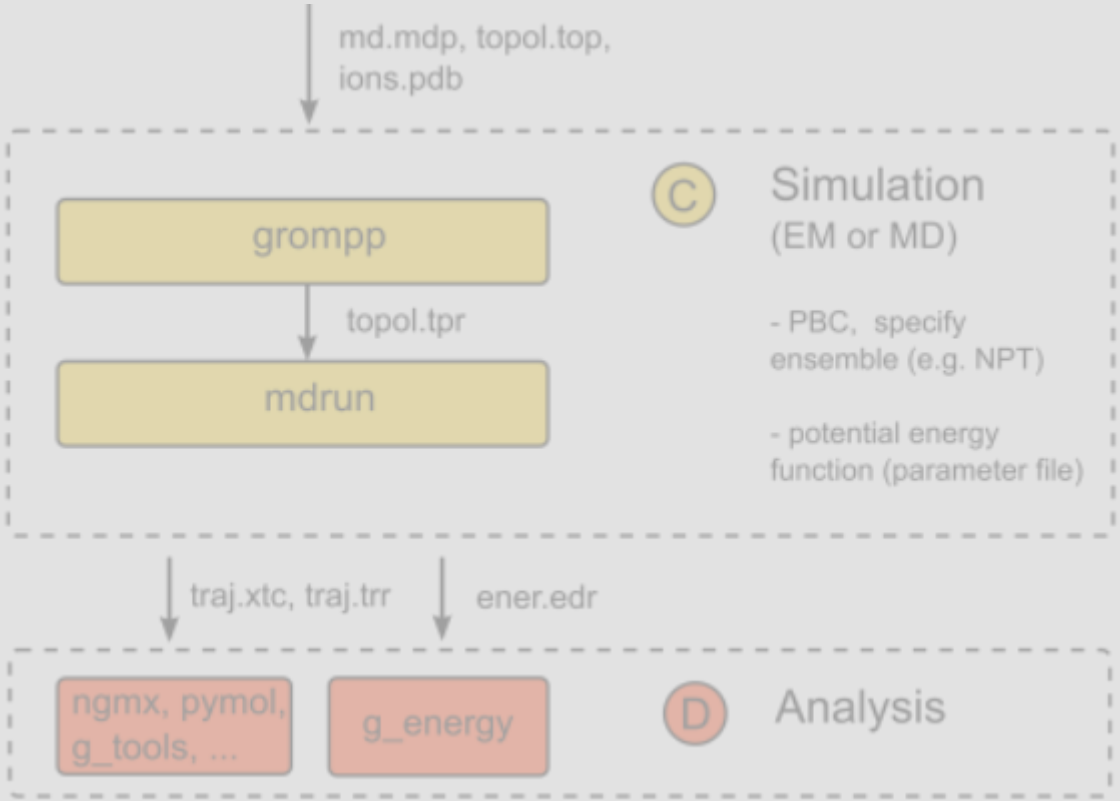
MD workflow



Interactive workflow: https://manual.gromacs.org/current/user-guide/flow.html

System preparation workflow

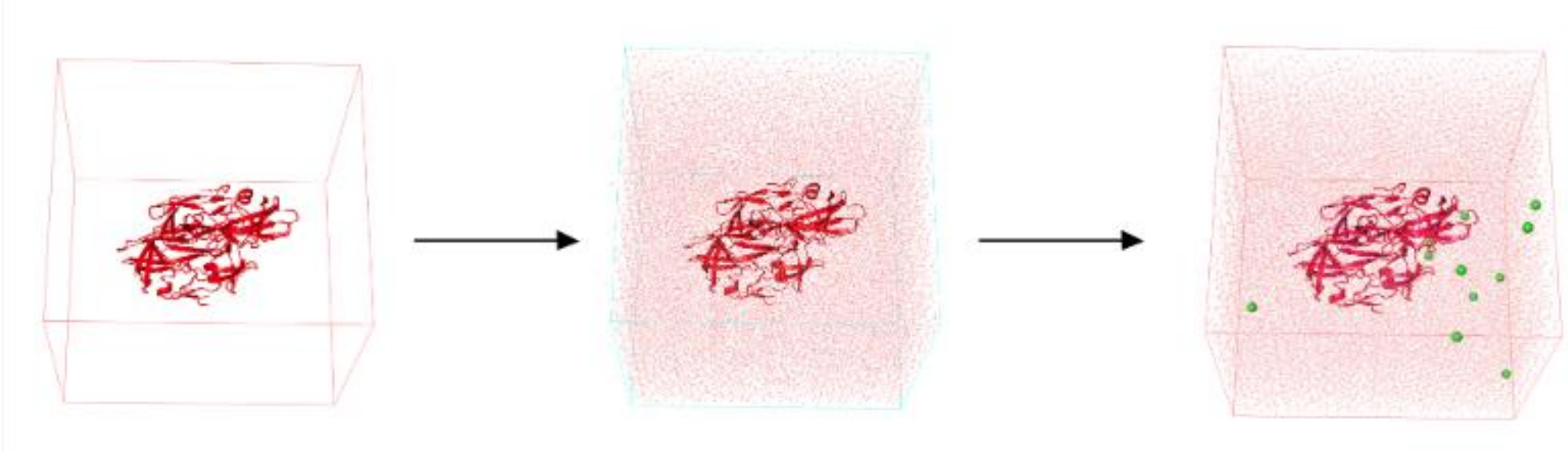




System preparation workflow

The **system preparation** phase is composed by three steps.

1. Create a simulation box
2. Solvate the system (e.g. using water)
3. Neutralize the overall system using counterions (Na^+ and Cl^-)

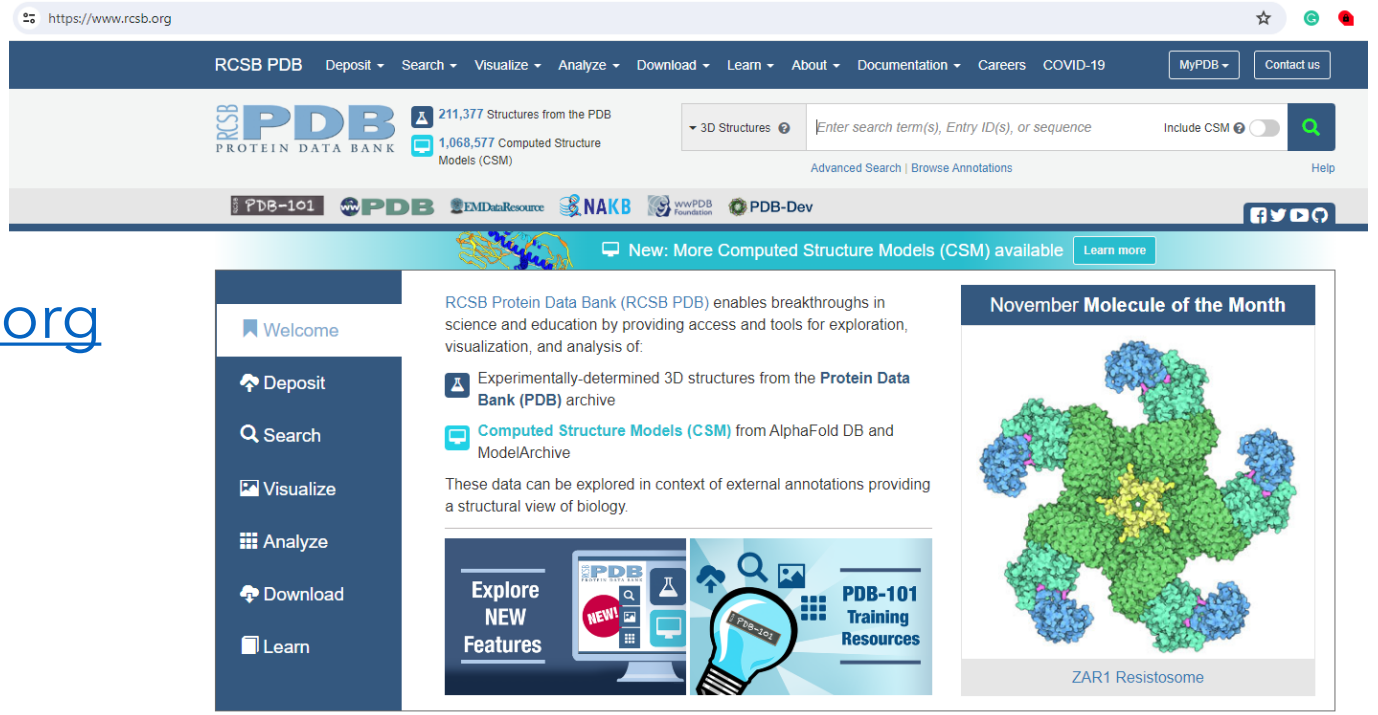


https://www.compchems.com/how-to-run-a-molecular-dynamics-simulation-using-gromacs/#what-is-gromacs

Getting the structure

a) A good source is the Protein Data Bank:
wget https://files.rcsb.org/download/1YRF.pdb

https://www.rcsb.org



b) Visualize the structure (jmol, VMD, Pymol…)

System preparation workflow : Initial setup

1. Clean the pdb file by deleting water and ligands from the pdb:

```
grep -v HETATM 1yrf.pdb > 1yrf_clean.pdb
```

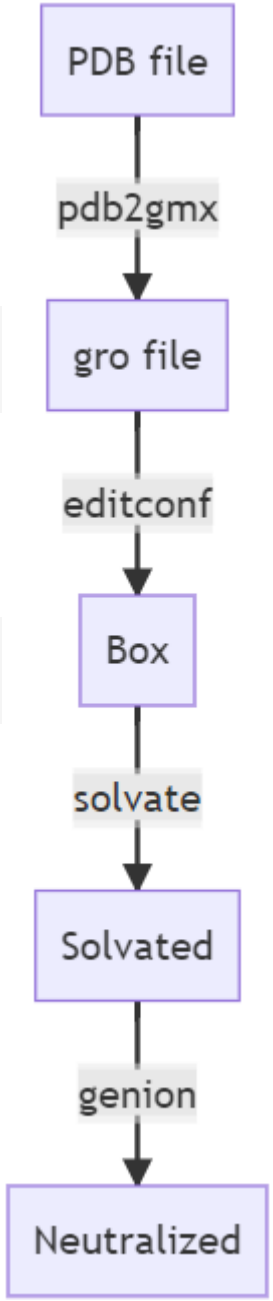
2. Create a topology of your protein

```
gmx pdb2gmx -f 1yrf_clean.pdb -o 1yrf.gro -water tip3p -ignh
```

For this tutorial select option 6 (**AMBER99SB-ILDN**)

This will generates three files:

`1yrf.gro` : structure
`topol.top` : topology
`postre.itp` : include topology (bond lengths, angles…)



System preparation workflow

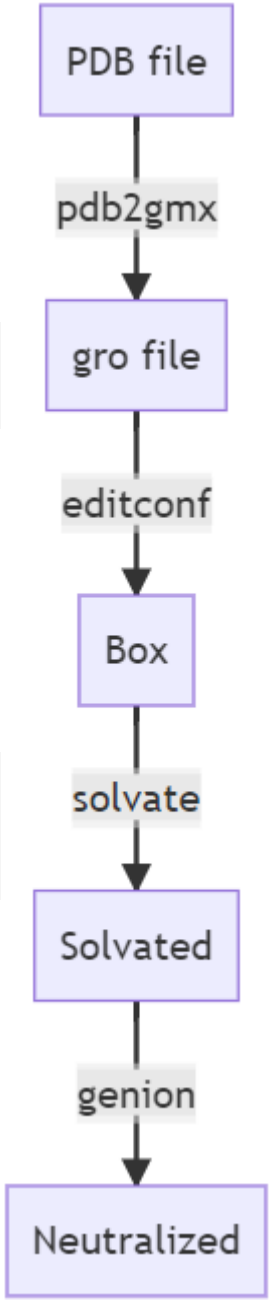
3. Create a simulation box for the protein:

```
gmx_mpi editconf -f 1yrf.gro -o 1yrf_box.gro -c -d 1.0 -bt cubic
```

4. Solvate the system

```
gmx solvate -cp 1yrf_box.gro -cs spc216.gro -o 1yrf_solv.gro -p
topol.top
```

Visualize the solvated box using Pymol, VMD or jmol
`1yrf_solv.gro` : solvated structure in the box

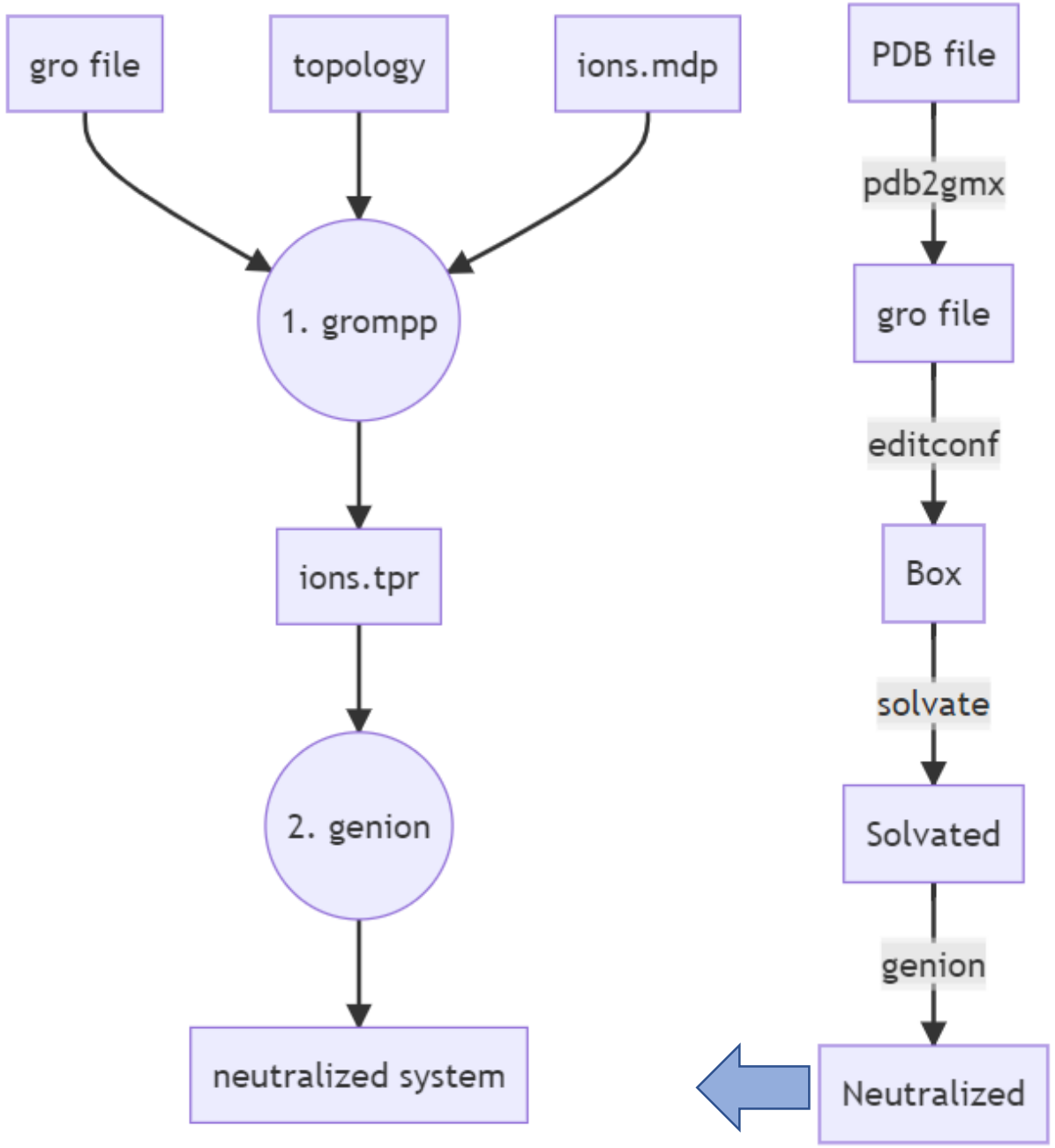


System preparation workflow

5. Neutralize the system by adding ions:

We need the following **input file**:
ions.mdp

To generate ions.tpr prior adding the ions.



System preparation workflow

5. Generate the ions.tpr file:

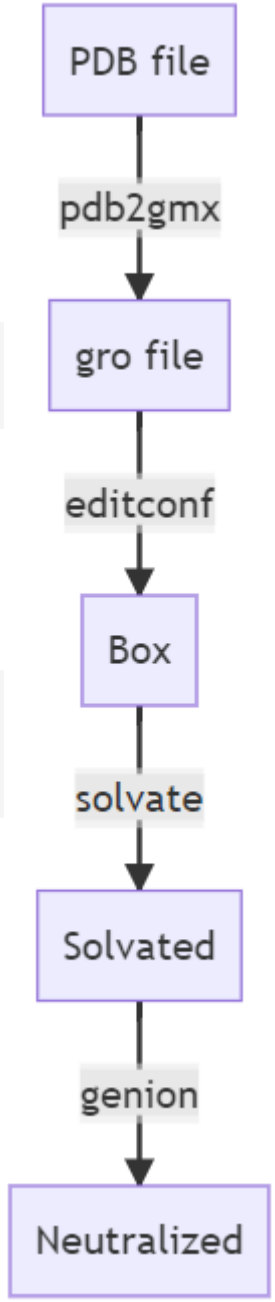
```
gmx grompp -f ions.mdp -c 1yrf_solv.gro -p topol.top -o ions.tpr
```

6. Use *genion* module to include the ions:

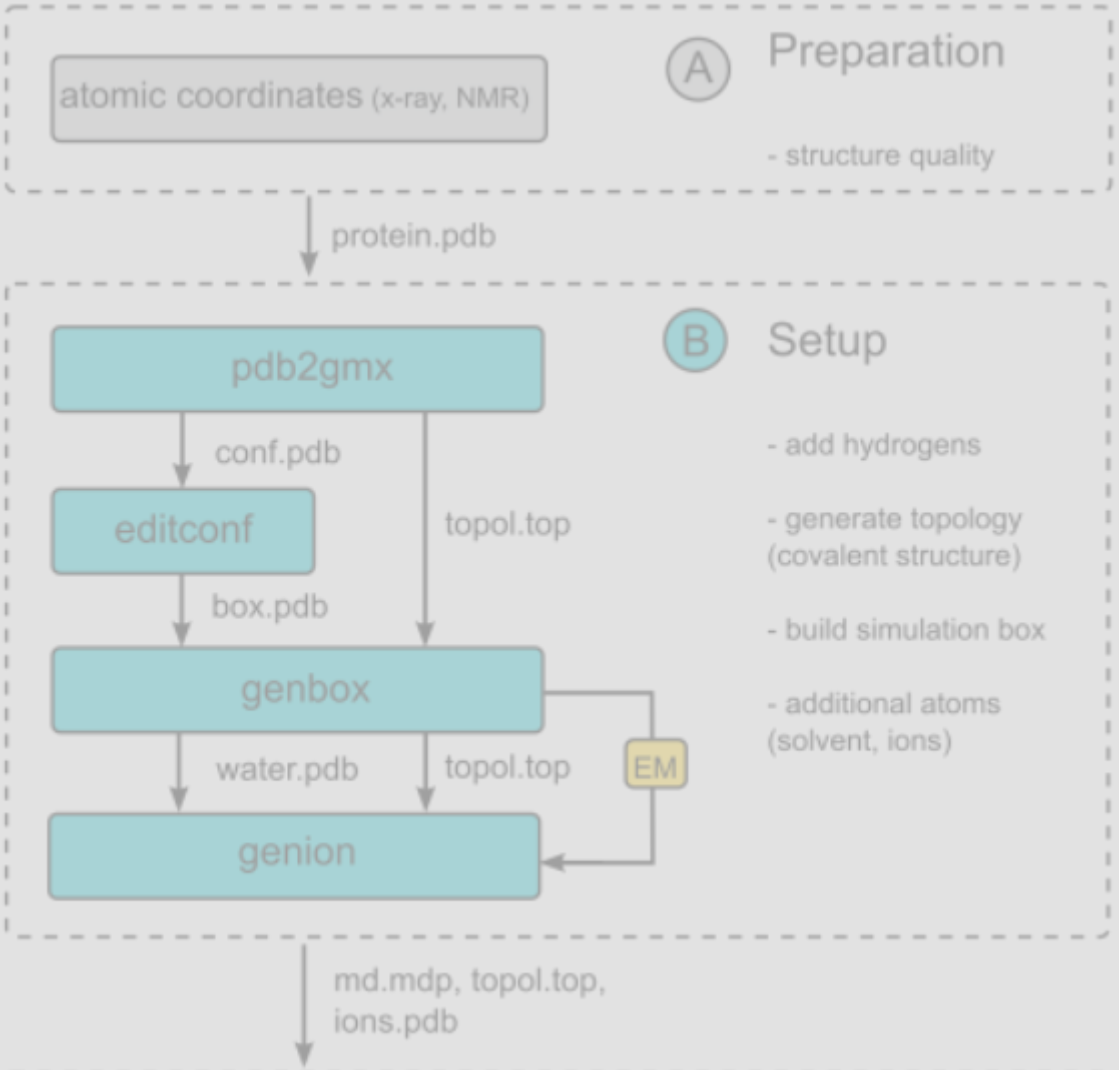
```
gmx genion -s ions.tpr -o 1yrf_ions.gro -p topol.top -pname NA -
nname CL -neutral -conc 0.15
```

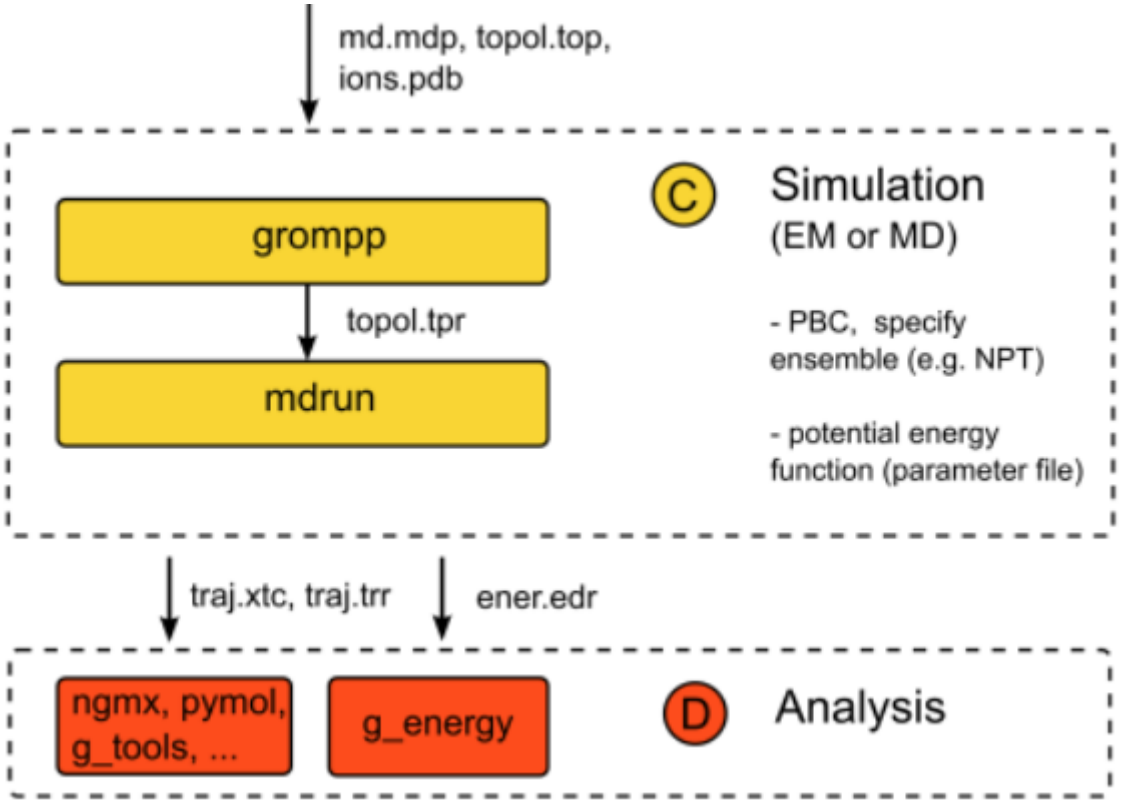
GROMACS will ask to select which molecules will be replaced by ions.
Select:

```
Group 13 (SOL) to replace solvent molecules by Na and Cl ions.
The resulting file is called 1yrf_ions.gro
```



Simulation workflow

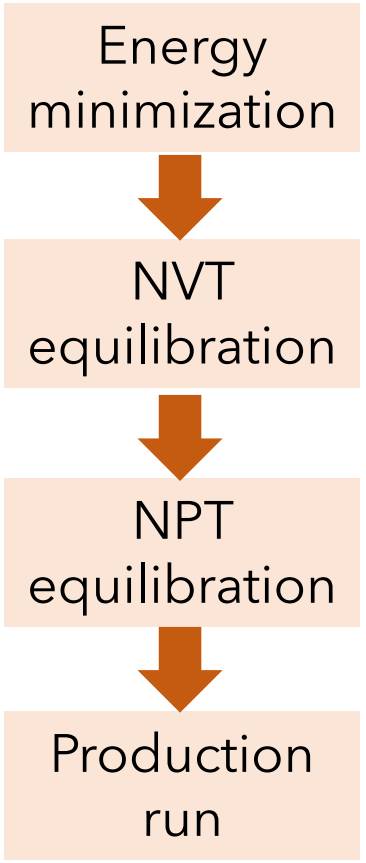




Simulation workflow

The simulation procedure involves multiple steps:

1. **Energy minimization**
2. **NVT equilibration**
3. **NPT equilibration**
4. **Production run**



For each one of these runs, we need an ***mdp* input file** with the simulation parameters (input folder).

.mdp templates are discussed here https://www.compchems.com/gromacs-mdp-file-parameters/#mdp-files

Simulation workflow

Minimize the potential energy of the system by adjusting the atomic coordinates to avoid steric clashes.

7. Energy minimization

Generates the *.tpr* file:

```
gmx grompp -f minim.mdp -c 1yrf_ions.gro -p topol.top -o em.tpr
```

Run the simulation:

```
gmx mdrun -v -deffnm em
```

It will generates the following files:

em.gro: containing the final structure
em.log: information about the run
em.trr: trajectory file of the system
em.edr: energy file useful for the analysis

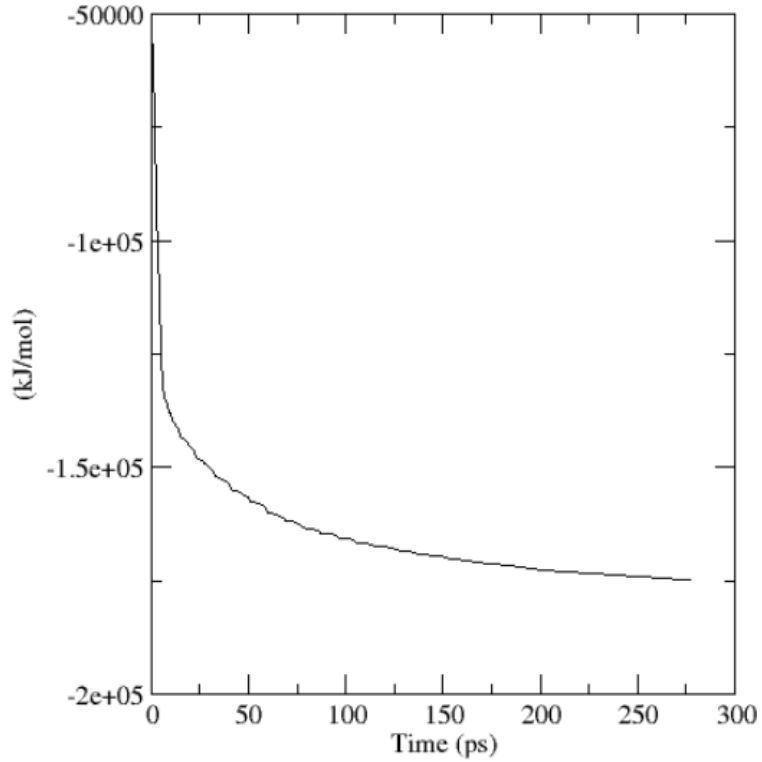
Simulation workflow: minimization analysis

Observe how the energy gets minimized along the simulation:

```
gmx energy -f em.edr -o potential.xvg
```

At the prompt, choose:
10 0
to select the Potential energy.

This will generate the file potential.xvg that we can plot to follow the energy minimization.



Simulation workflow

8. NVT Equilibration

Equilibrates the ions and the solvent around the protein.
Use a thermostat to equilibrate the system at constant temperature.

Generates the *.tpr* file:

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

Run the simulation:

```
gmx mdrun -v -deffnm nvt
```

The simulation time is defined in the nvt.mdp file.
In this case, we are going to run for 100 ps using the V-rescale thermostat

It will generates several files as before.
To visualize the trajectories, you need to load: nvt.gro + nvt.trr in pymol, VMD…

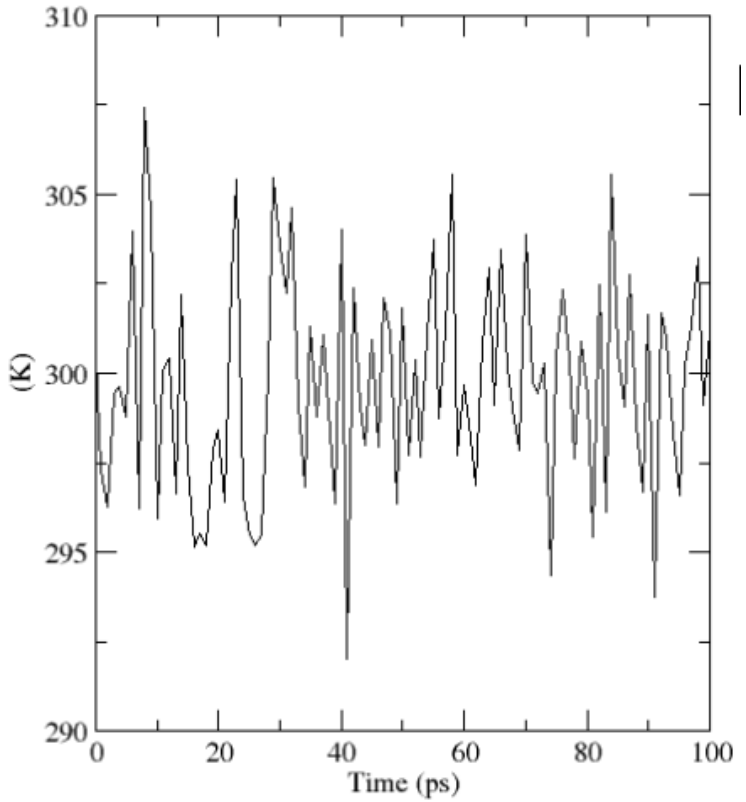
Simulation workflow: Thermalization analysis

Observe how the temperature changes with time:

```
gmx energy -f nvt.edr -o nvt_temperature.xvg
```

At the prompt, choose:
16 0
to select the temperature.

This will generate the file nvt_temperature.xvg
Observe if the temperature reached a plateau around 300 K.



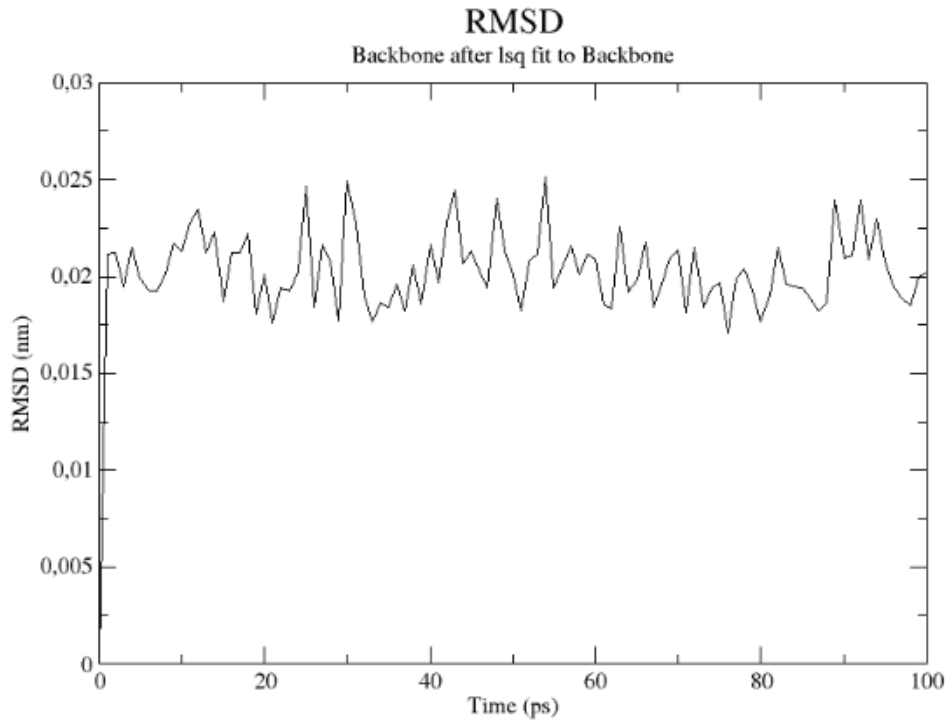
Simulation workflow: Thermalization analysis

To check if the system is properly equilibrated, we can check the RMSD value for the backbone to backbone of the protein.

```
printf "4 4" | gmx rms -f nvt.trr -s nvt.tpr -o nvt_rmsd.xvg
```

At the prompt, choose:
4 4

This will generate the file
`nvt_rmsd.xvg`
The RMSD quickly converges to a stable value signaling that the system has equilibrated at the desired temperature.



Simulation workflow

9. NPT Equilibration

Equilibrates the ions and the solvent around the protein.
Use a barostat to equilibrate the system at constant pressure.

Generates the *.tpr* file:

```
gmx grompp -f npt.mdp -c nvt.gro -r nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
```

Run the simulation:

```
gmx mdrun -v -deffnm npt
```

The simulation time is defined in the npt.mdp file.
In this case, we are going to run for 100 ps using the Parrinello-Rahman barostat.

It will generates several files as before.
To visualize the trajectories, you need to load: npt.gro + npt.trr in pymol, VMD…

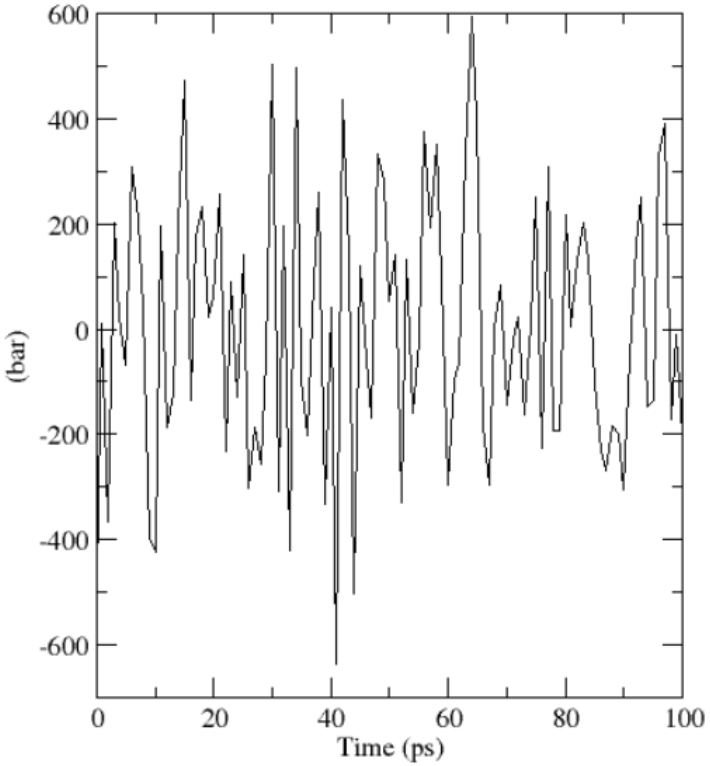
Simulation workflow: Equilibration analysis

Observe how the temperature changes with time:

```
gmx_mpi energy -f npt.edr -o npt_pressure.xvg
```

At the prompt, choose:
18 0
to select the pressure.

This will generate the file `npt_pressure.xvg`
Observe if the pressure fluctuates around 1 bar.Note that large pressure fluctuations are completely normal for nanoscale simulations of aqueous solutions.



Simulation workflow: Equilibration analysis

To check if the system is properly equilibrated, we can check the RMSD value for the backbone to backbone of the protein.

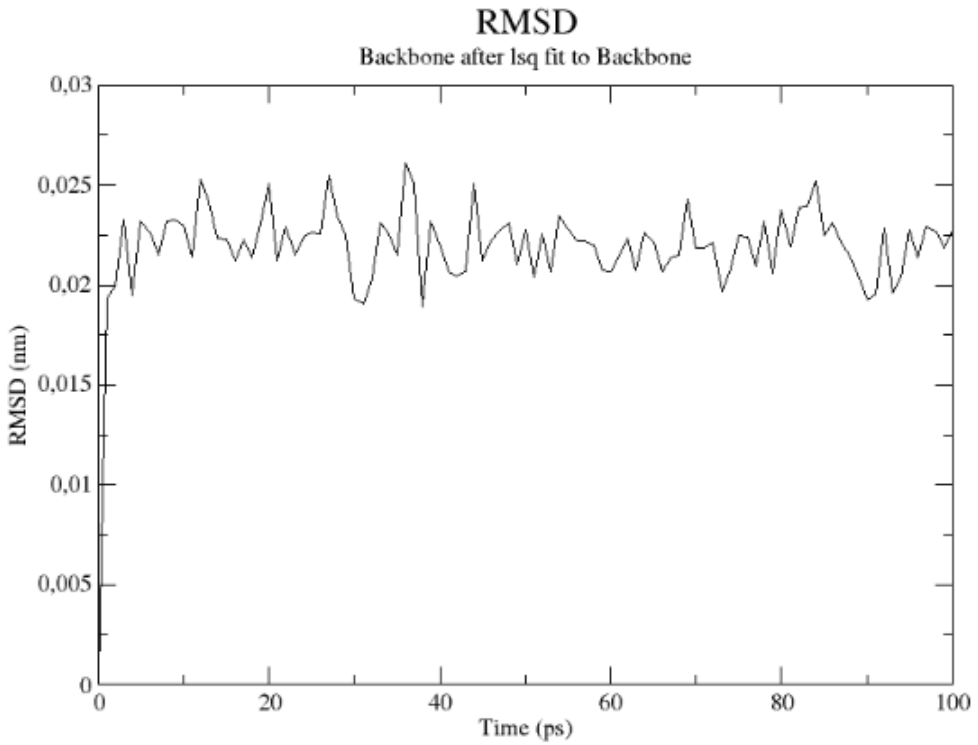
```
gmx rms -f npt.trr -s npt.tpr -o npt_rmsd.xvg
```

At the prompt, choose:

4 4

This will generate the file `npt_rmsd.xvg`

The RMSD values are very stable over time, indicating that the system is well-equilibrated at the desired pressure.



Simulation workflow

This is the real simulation to observes how system evolves over time.
Still uses NPT ensemble, but with no position restraints.

10. Production run

Generates the *.tpr* file:

```
gmx grompp -f md.mdp -c npt.gro -t npt.cpt -p topol.top -o md.tpr
```

Run the simulation:

```
gmx mdrun -v -deffnm md
```

The simulation time is defined in the md.mdp file.
We will run for **1 ns**, using the Parrinello-Rahman barostat and V-rescale thermostat.

The relevant information can be found in the usual files.
To visualize the trajectories, you need to load: md.gro + md.xtc in pymol, VMD…

Simulation workflow: Analysis of the production run

Energy:

```
gmx energy -f md.edr -o md.xvg;  (choose 18
```

RMSD:

```
Printf "4 4"| gmx rms -f md.trr –s md.tpr -o md.xvg;
```

Hydrogen bonds

```
gmx make_ndx -f md.gro -o index.ndx
gmx hbond -f md.xtc –s md.tpr –n index.ndx –num hbnum.xvg
```

RMSF (Root square fluctuation) for specific time frames (-b and –e)

```
gmx rmsf -f md.xtc –s md.tpr –o rmsf.xvg –res (-b 1000 –e 10000)
```

Clustering frames of the trajectory:

```
gmx cluster -f md.xtc –s md.tpr -method gromos -cutoff 0.15 -b 50000 -g -cl (-e -dt 10 -n index.ndx)
```

Visualization: jmol

You can visualize the file.gro directly or convert it to a file.pdb using GROMACS:

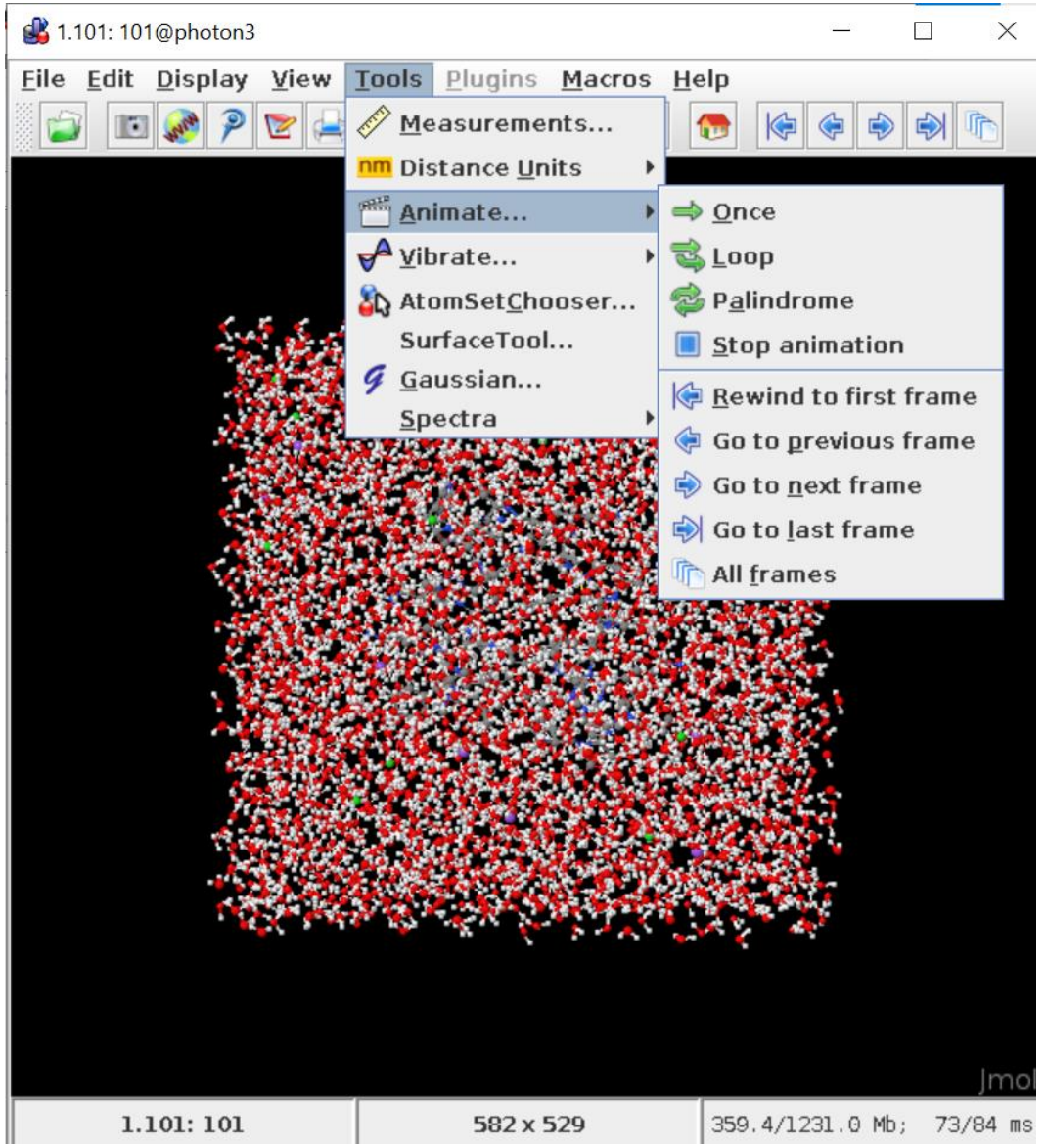
```
gmx editconf -f npt.gro -o npt.pdb
```

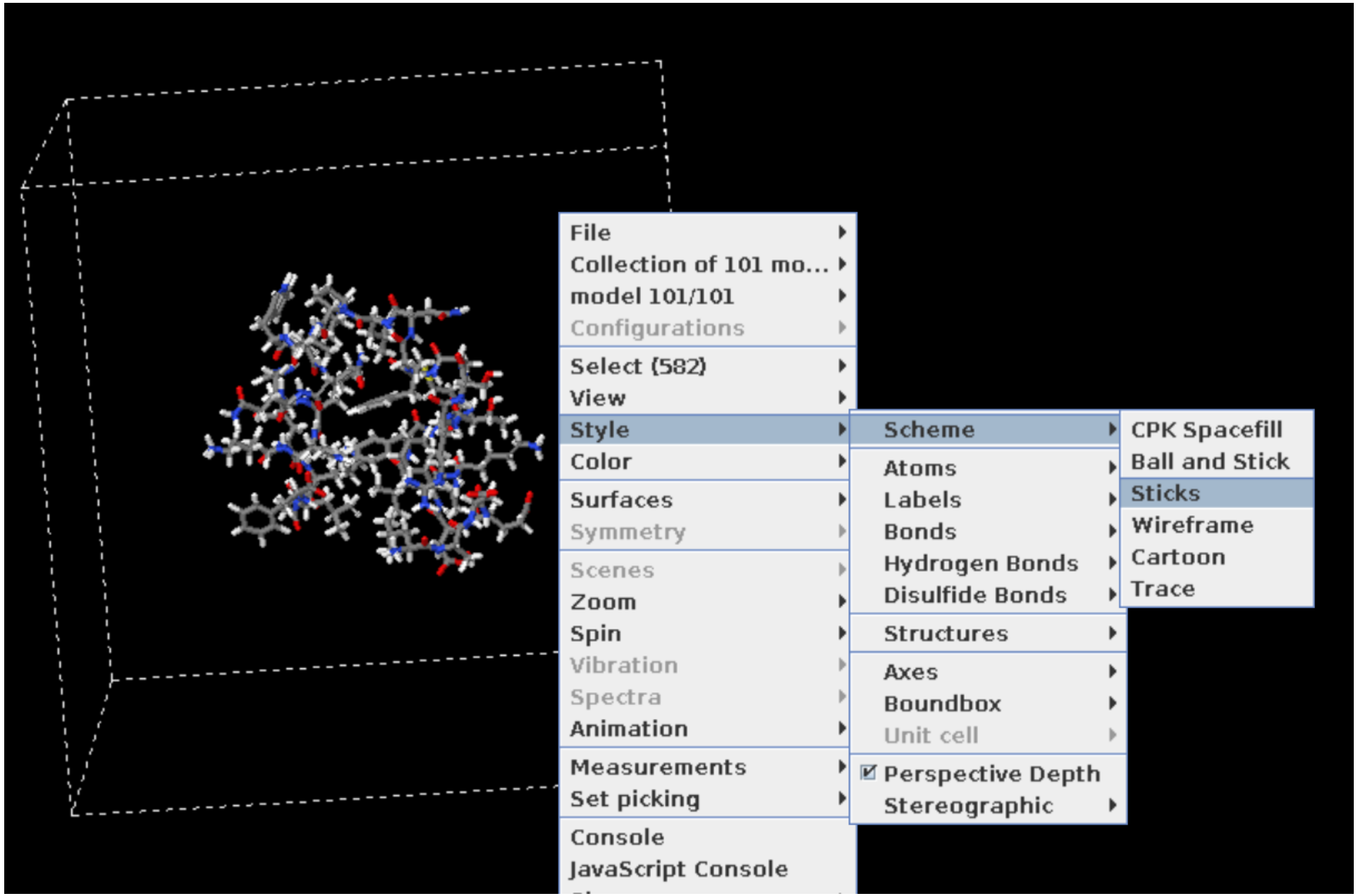
To visualize the trajectory, you can convert the file.trr or file.xtc from gromacs to a pdb file using trjconv tool in GROMACS:

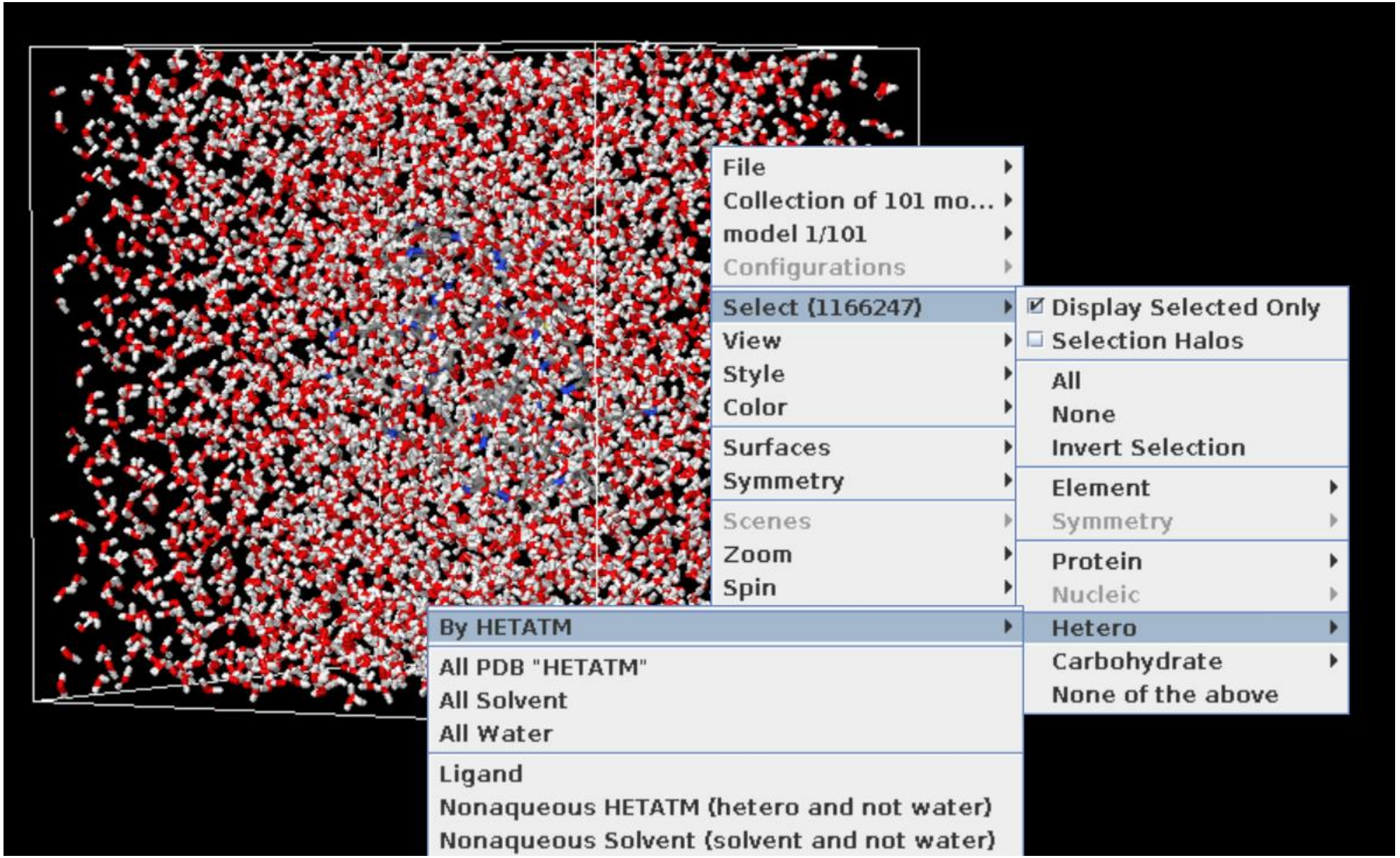
```
gmx trjconv -s npt.tpr -f npt.trr -o
npt.pdb -pbc whole -conect
```

Display -> bounding box (show the box)

In the console:
select all; wireframe; select solvent; wireframe -0.1;
select protein; cartoons
display selected




File
Collection of 101 mo...
model 101/101
Configurations
Select (582)
View
Style
Color
Surfaces
Symmetry
Scenes
Zoom
Spin
Vibration
Spectra
Animation
Measurements
Set picking
Console
JavaScript Console
Scheme
Atoms
Labels
Bonds
Hydrogen Bonds
Disulfide Bonds
Structures
Axes
Boundbox
Unit cell
Perspective Depth
Stereographic
CPK Spacefill
Ball and Stick
Sticks
Wireframe
Cartoon
Trace

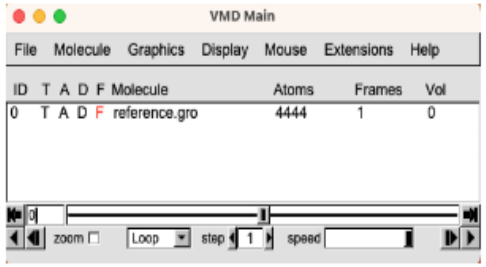

File
Collection of 101 mo...
model 1/101
Configurations
Select (1166247)
View
Style
Color
Surfaces
Symmetry
Scenes
Zoom
Spin
Display Selected Only
Selection Halos
All
None
Invert Selection
Element
Symmetry
Protein
Nucleic
Hetero
Carbohydrate
None of the above
By HETATM
All PDB "HETATM"
All Solvent
All Water
Ligand
Nonaqueous HETATM (hetero and not water)
Nonaqueous Solvent (solvent and not water)

Trajectories visualization in VMD

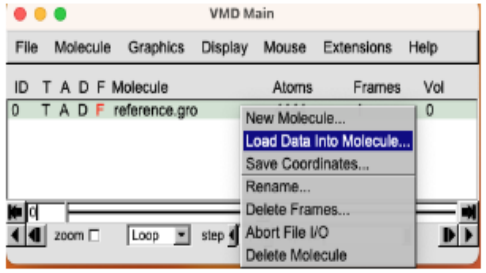
load md.gro -> load data into molecule; md.xtc

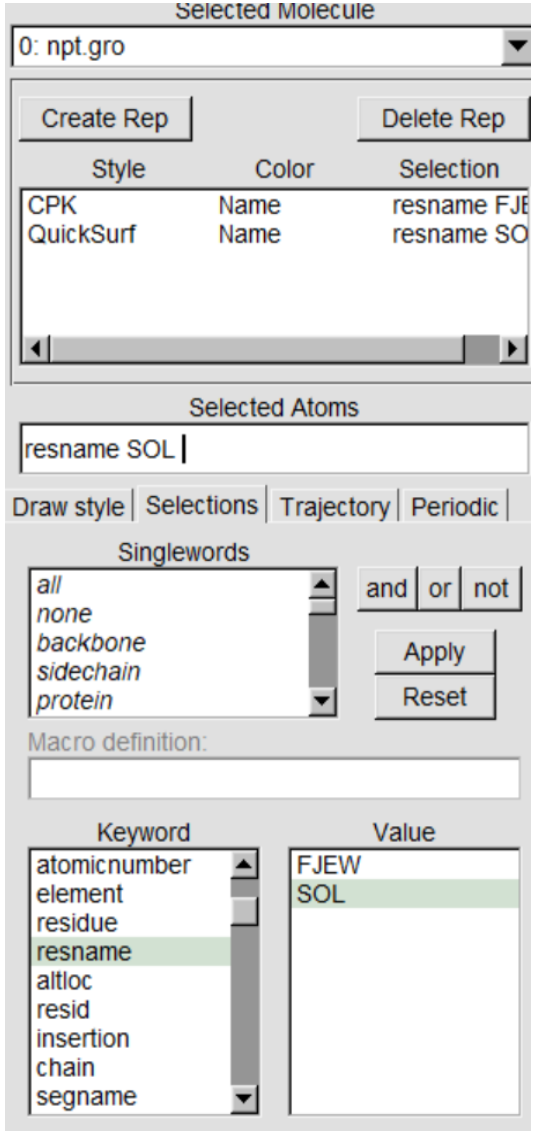
This approach involves a two-step process. Firstly, you load the structure file, and then you can load the trajectory itself, integrating it with the reference structure.

1. Open VMD. Go to *File* → *New Molecule*. This will open the *Molecule File Browser* menu. Click on *Browse* and look for the gro or pdb file, select it, and click *Load*. The loaded molecule will display on the *VMD Main* menu.



2. Now click on the loaded file to select it and then right-click on it. Select *Load Data Into Molecule*. Once again, click on *Browse*, look for the trajectory file, and *Load* it. You will see that the initial structure will start moving as the trajectory slowly loads.





Trajectories visualization in PyMOL

load md.gro

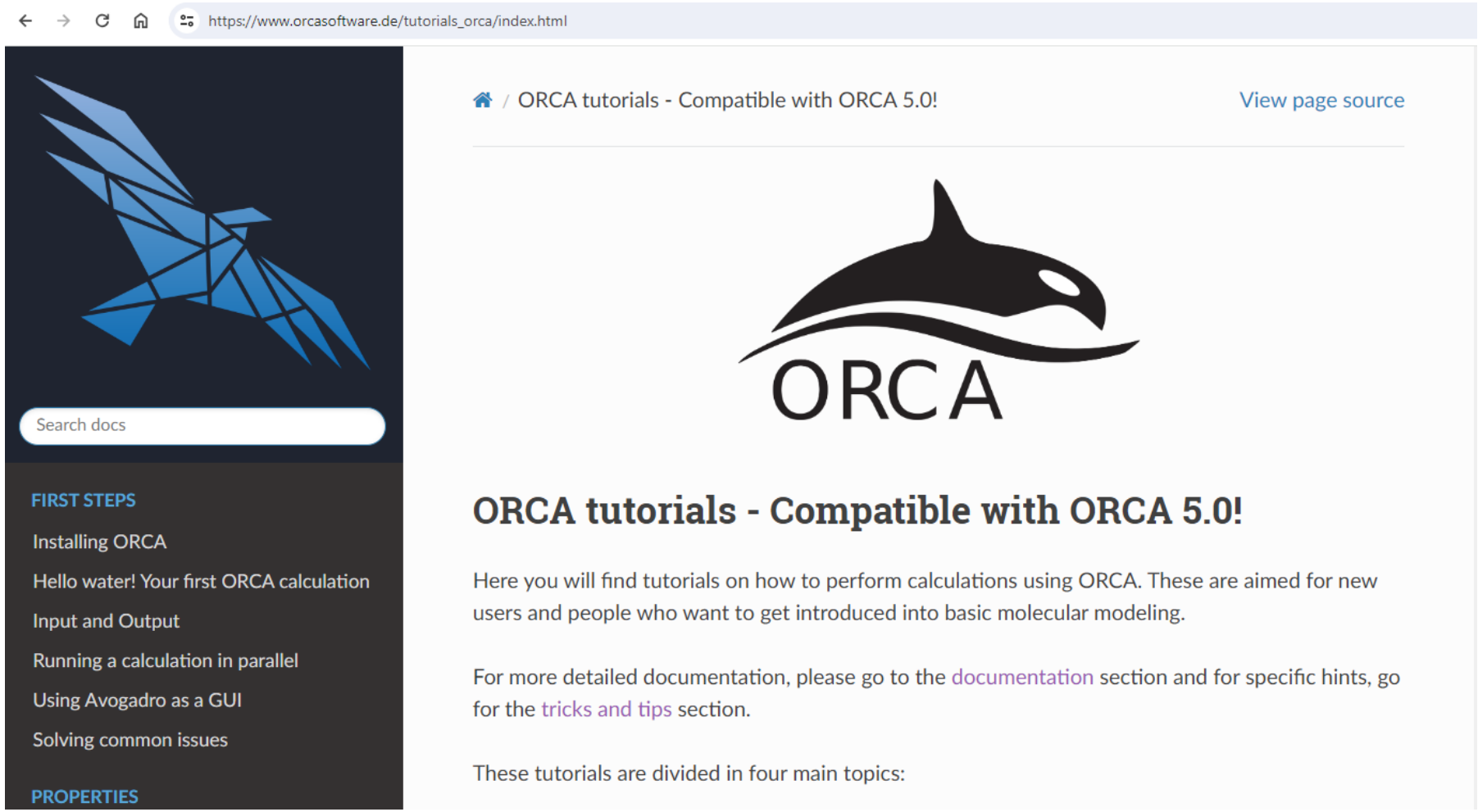
Then go to File-> Open-> select npt.trr or md.xtc

Hybrid methods: QNIOM calculations

Adapted from:
https://www.orcasoftware.de/tutorials_orca/multi/basics.html

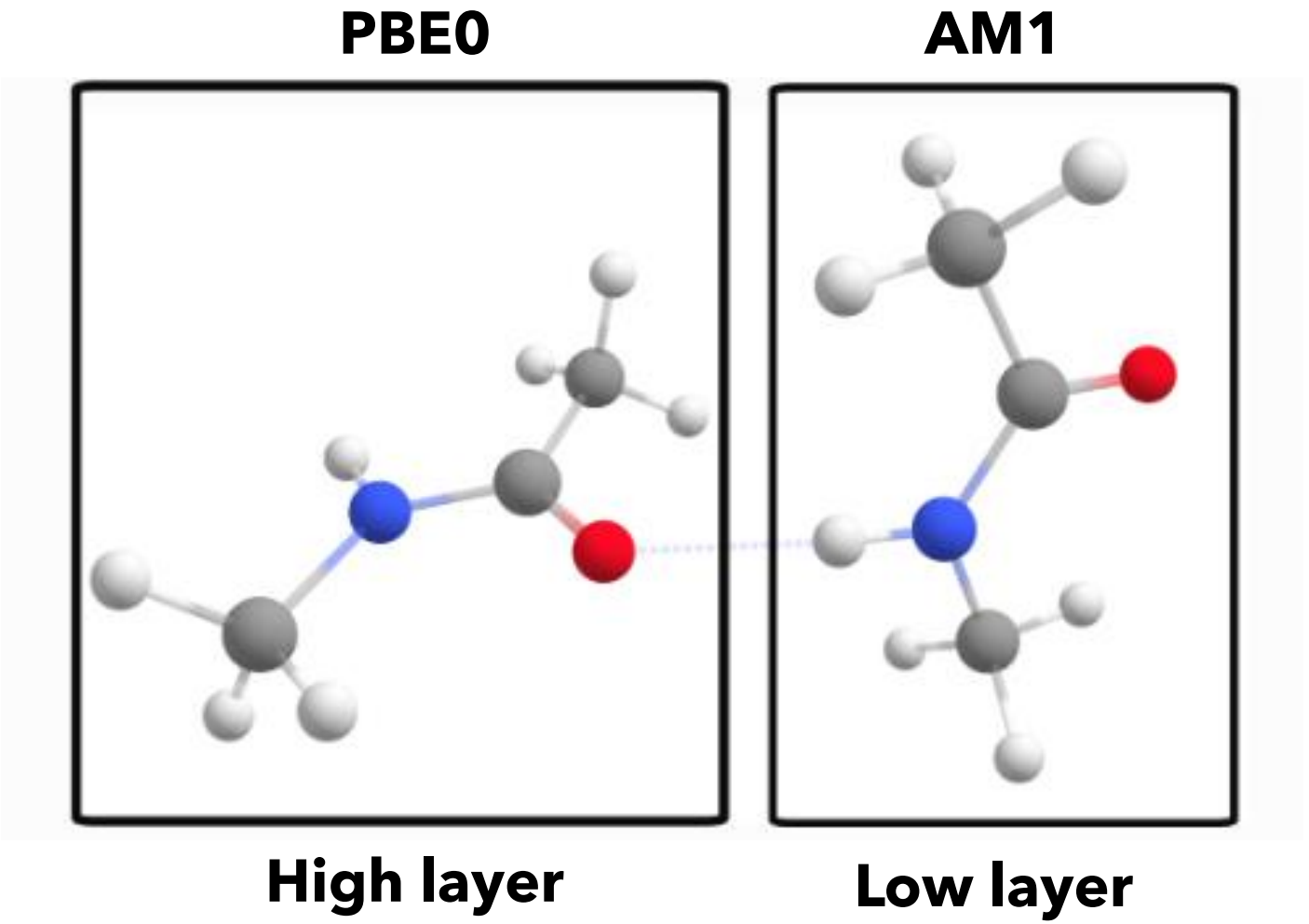
QM Program: ORCA

https://www.orcasoftware.de/tutorials_orca/index.html



ONIOM QM/AM1

We are splitting the system in two regions:



Considerations about the calculation

1. The *Multiscale model* is *QM1/QM2*, using *AM1* as *QM2*

2. The default coupling scheme is *subtractive*. This is related to how the final energy is calculated.

3. The default embedding scheme is *electrostatic*. This is related to how both QM systems interact electrostatically.

4. The full size of the system is 24 atoms, and the size of each subsystem is 12.

Orca QM/AM1 input

```
# ONIOM Single point calculation
%pal nprocs 1
end
!QM/AM1 PBE0 D4 DEF2-TZVP
%QMMM QMATOMS {0:11} END END

* xyz 0 1
C   -0.701502936  -0.290627698   2.406884396
H   -1.183295956   0.395647770   3.098874220
H    0.349561571  -0.030321572   2.307833035
H   -0.794056854  -1.291605451   2.824039291
.
.
.
*
```

Atoms **0** to **11** will be treated at the QM level.
(Orca atom start counting from zero!)

Running the calculations

```
module load orca/5.0.4

qorca -i oniom.inp
```

Output

```
                          ****************************
                          *    2-layered ONIOM       *
                          ****************************


------------------------------------------------------------------------------




Multiscale model                              ... QM1/QM2
QM2 method                                    ... AM1
Coupling Scheme                               ... subtractive
Embedding Scheme                              ... electrostatic
PrintLevel                                    ... 1
Method for determining QM2 charges            ... Loewdin
Charge of total system                        ... 0
Charge alteration scheme                      ... Charge shifting
Scaling factor for CS scheme                  ... 0.06
Gradient projection scheme                    ... Lever rule
Scaling factor for QM2 charges                ... 1.00
AutoFF method                                 ... XTB2
AutoFF boundary                               ... no covalent bonds
Point charges in QM calc. from MM atoms ... 12
                from charge shift scheme  ... 0
Printing trajectories, xyz and allxyz files:
        QMMMRegion                            ... YES
        activeRegion                          ... YES
        activeRegionExt                       ... NO
        qmRegion                              ... YES
        qm2Region                             ... NO
```

Calculations necessary for the subtractive scheme: full system using QM2, high level region (small system) using QM1.

```
                  ************************************
                  *       QM/    AM1 SP Energy       *
                  ************************************


---------------------------------------------------------------------------
                         AM1 - FULL SYSTEM
---------------------------------------------------------------------------

Running /software/softs/orca_5_0_4_linux_x86-64_shared_openmpi411/orca oniom_L_QM2.inp >oniom_L_QM2.lastout

--------------------------------   --------------------
FINAL SINGLE POINT ENERGY (L-QM2)      -74.098710681396
--------------------------------   --------------------
---------------------------------------------------------------------------
                                  AM1 - SMALL SYSTEM
---------------------------------------------------------------------------

Running /software/softs/orca_5_0_4_linux_x86-64_shared_openmpi411/orca oniom_S_QM2.inp >oniom_S_QM2.lastout

--------------------------------   --------------------
FINAL SINGLE POINT ENERGY (S-QM2)      -37.053921173457
--------------------------------   --------------------
---------------------------------------------------------------------------
                                QM - SMALL SYSTEM
---------------------------------------------------------------------------
--------------------------------
CARTESIAN COORDINATES (ANGSTROEM)
--------------------------------
  C     -0.701503   -0.290628    2.406884
```

The ONIOM energy is given as FINAL SINGLE POINT ENERGY (QM/QM2):

```
Starting D4
-------------------------   ----------------
Dispersion correction           -0.006076459
-------------------------   ----------------
```

High level region energy:

```
-------------------------   --------------------
FINAL SINGLE POINT ENERGY      -248.351653539665
-------------------------   --------------------
```

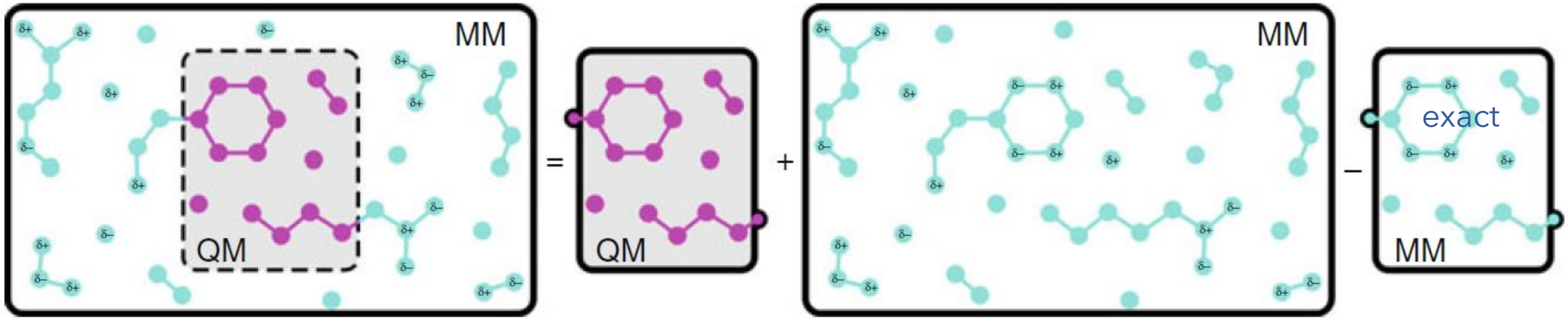
binding_property.txt

Onion:

```
-----------------------------------   --------------------
FINAL SINGLE POINT ENERGY (QM/QM2)     -285.396443047604
-----------------------------------   --------------------
```

Subtractive QM/MM (ONIOM)

$$E_{QM/MM} = V_{MM}\left(R+NR\right)+V_{QM}\left(R\right)-V_{MM}\left(R\right)$$



☺ No communication between QM and MM routines
☹ Require FF for R region
☹ No polarization of R due to NR

Computing the binding energies between two amides

Repeat the calculation with the molecules far from each other.
(input file: far.inp)

And compute the binding energy: $$E_{binding}^{ONIOM} = E_{far}^{ONIOM} - E_{close}^{ONIOM}$$

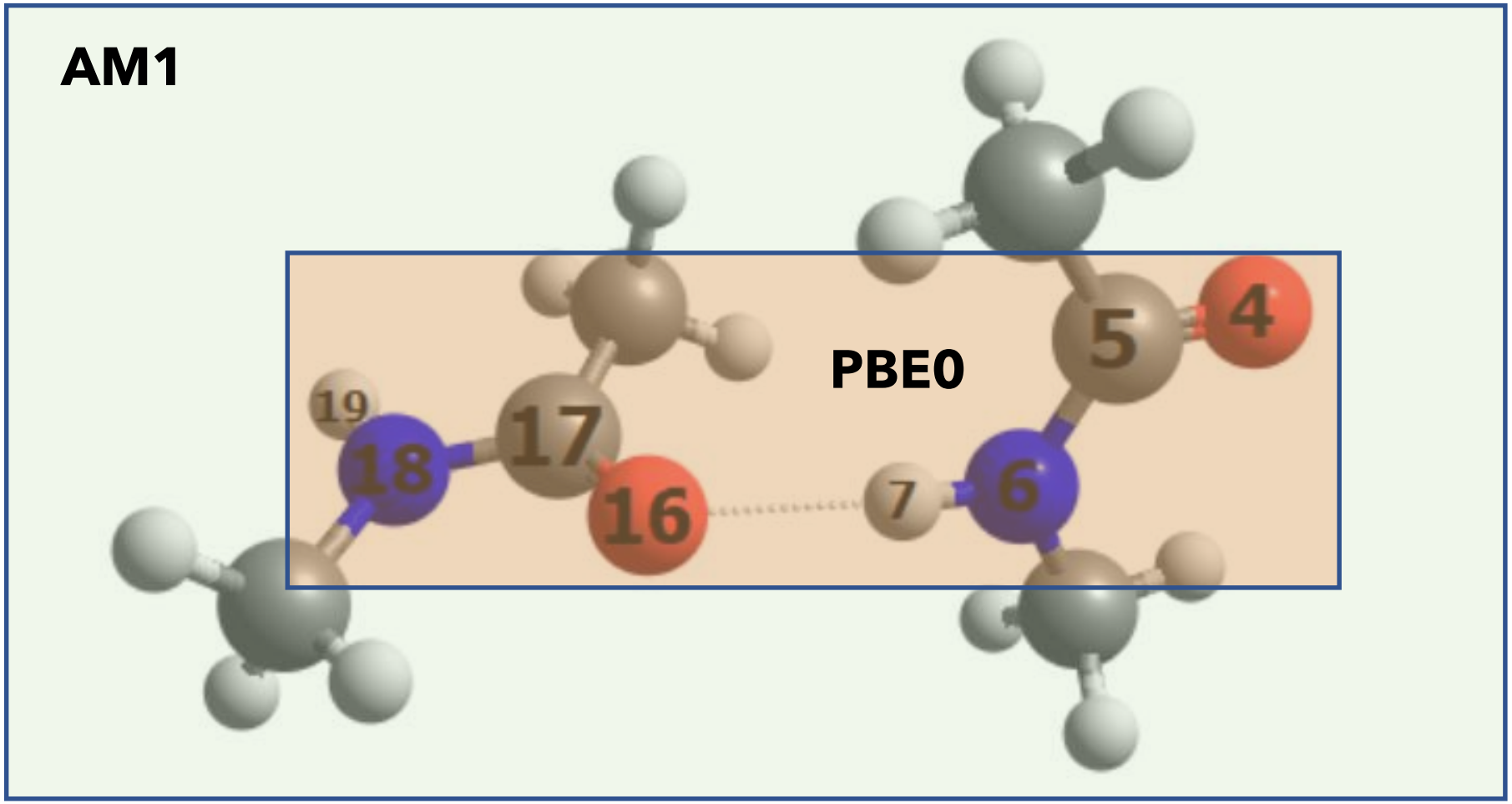
The reference value is 8.7 kcal/mol [Hobza 2011]

Repeat these two calculations using only qm method (input qm.inp)

Question: Why do you think the error is so large?

Rezac, Riley, Hozba, JCTC **2011**, *7*, 2427.

Selecting the proper QM region




```
%QMMM QMATOMS {4:7} {16:19} END END
```

References and suggestions

Protein in water:
https://www.compchems.com/gromacs_protein_water/md.mdp

https://www.compchems.com/how-to-run-a-molecular-dynamics-simulation-using-gromacs/#what-is-gromacs

https://gromacstutorials.github.io/doc-sphinx/build/html/tutorials/bulksolution.html#

Graphene:
https://www.erastova.xyz/teaching/practical-simulations-for-molecules-and-materials/material-simulations/graphene-simulation-set-up/

Diverse GROMACS tutorials:
http://www.mdtutorials.com/gmx/index.html

ONIOM:
https://www.orcasoftware.de/tutorials_orca/multi/basics.html#our-first-oniom-single-point

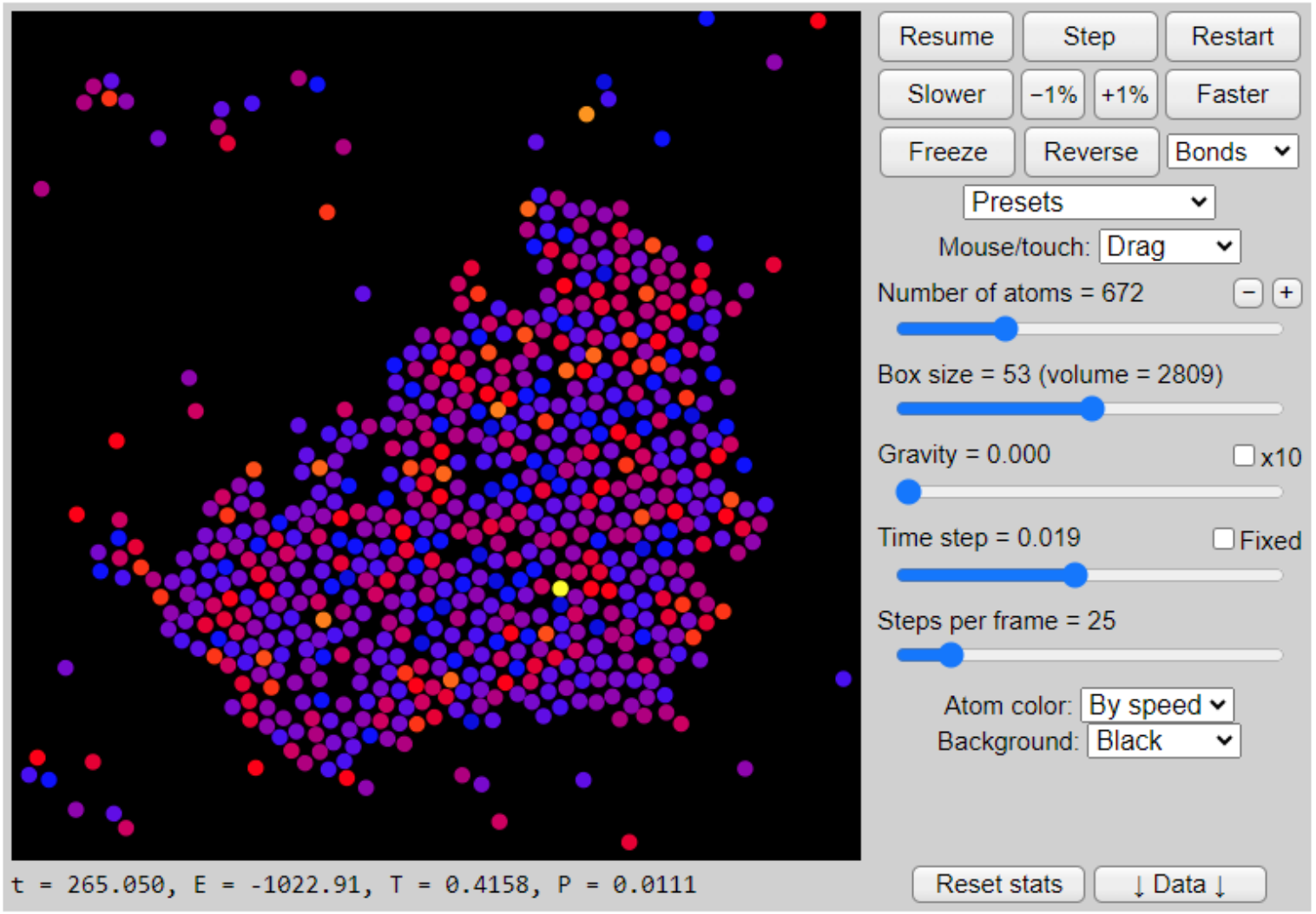
References and suggestions

Visualization:
https://manual.gromacs.org/current/how-to/visualize.html

JMOL:
https://chemapps.stolaf.edu/jmol/docs/examples-11/new.htm?topic=138

https://earth.callutheran.edu/Academic_Programs/Departments/BioDev/omm/jsmolnew/scripting/molmast.htm#nb

Interactive Molecular Dynamics



This web app simulates the dynamics of simple atoms and molecules in a two-dimensional universe. The force between the atoms is weakly attractive at short distances, but strongly repulsive when they touch. Use the simulation to explore phases of matter, emergent behavior, irreversibility, and thermal effects at the nanoscale.

https://physics.weber.edu/schroeder/md/

MDWEB

Molecular Dynamics on Web

INB

- Home
- General Help
- Getting started
- Structure checking
- Ligand checking
- Workspace
- MDMoby ontology
- Operations
- Workflows
- Icons
- Software
- References
- Setup tutorial
- **Analysis tutorial**
- Run tutorial

MDWeb Analysis Tutorial

MDWeb provides a friendly environment to analyse your own generated molecular dynamics trajectories. With this short tutorial, you will be able to upload a trajectory and run a set of analysis, checking for example the stability of your system or information about flexibility.

1. **Registration**
2. **Starting Project**
3. **Uploading a trajectory**
4. **Running Analysis**

Tutorial Steps

1. **Registration**

   The first thing to do is to choose between working as an ***anonymous*** user or alternatively as a ***registered*** user. We **strongly** recommend working as a registered user, as it has some important advantages.

   Anonymous user's projects are completely removed once the user is disconnected and also when session expires (after some minutes of inactivity), and therefore working as anonymous user is only suited for a first impression of the web server.

   Registration process will just take a minute --> **Registration**.

   Once logged in, the user **workspace** appears. In this **workspace**, all projects of the user will be shown.

   Now, we are ready to start our first **MDWeb** analysis project.

https://mmb.irbbarcelona.org/MDWeb/help.php?id=tutorialAnalysis

GRAPHENE SHEET IN WATER

PROTOCOL FOR SET UP

This example was adapted from:
https://www.erastova.xyz/teaching/practical-simulations-for-molecules-and-materials/material-simulations/graphene-simulation-set-up/

1. Create a new working directory GRAPHENE. Copy the folder 'charmm36.ff' forcefield file into it.
   The charmm36 force field can be obtained from:
   https://mackerell.umaryland.edu/download.php?filename=CHARMM_ff_params_files/charmm36-jul2022.ff.tgz

2. Write a unit cell file, named GRA.gro, and copy the following content into it:

GRA.gro:

```
GRA: 1 1 Rcc=1.420 Rhole=0.000 Center: Ring
4
    1GRA   C1      1   0.061   0.071   0.000
    1GRA   C2      2   0.184   0.142   0.000
    1GRA   C3      3   0.184   0.284   0.000
    1GRA   C4      4   0.061   0.355   0.000
    0.245951    0.426000    0.284000
```

3. Create a sheet of graphene in *xy*–plane:

```
gmx genconf -f GRA.gro -o GRA_sheet.gro -nbox 15 10 1
```

4. Build the topology file:

a) First, create a new file in charm36.ff, called graphene.n2t. This file contains specific parameters for your system. Copy information into it:

graphene.n2t :

```
C    CG2R61   0.00      12.011  1    C 0.142
C    CG2R61   0.00      12.011  2    C 0.142   C 0.142
C    CG2R61   0.00      12.011  3    C 0.142   C 0.142   C 0.142
```

This file should be place inside the folder ‘charmm36.ff’, which is in you work directory

b) Then, generate a topology for your system:

```
gmx x2top -f GRA_sheet.gro -o gra.top -name GRA -nexcl 3 -ff charmm36 -kb
255224 -kt 334.72 -kp 12.9704 –alldih
```

c) After that, create the file GRA.itp by copyng the file gra.top and removing a header until [moleculetype] and the description of [system] and [molecules], as shown by selection below:

```
cp gra.top GRA.itp
```

GRA.itp:

```
;
;       File 'gra.top' was generated
;       By user: josene (2003)
;       On host: data
;       At date: Sat Nov  4 17:26:44 2023
;
;       This is a include topology file
;
;       Created by:
;                           :-) GROMACS - gmx x2top, 2022.3 (-:
;
;       Executable:   /home/josene/softs/gromacs/bin/gmx_mpi
;       Data prefix:  /home/josene/softs/gromacs
;       Working dir:  /home/josene/tutorials/tutorial-GROMACS/erastova/graphene
;       Command line:
;         gmx_mpi x2top -f GRA_sheet.gro -o gra.top -name GRA -nexcl 3 -ff charmm36 -kb 255224 -kt 334.72 -kp 12.9704 -alldih
;       Force field was read from current directory or a relative path - path added.
;

; Include forcefield parameters
#include "./charmm36.ff/forcefield.itp"

[ moleculetype ]
; Name            nrexcl
GRA                 3

[ atoms ]
;   nr       type  resnr residue  atom   cgnr     charge       mass  typeB    chargeB      massB
     1     CG2R61      1    GRA     C1      1          0     12.011
     2     CG2R61      1    GRA     C2      2          0     12.011
     3     CG2R61      1    GRA     C3      3          0     12.011
     4     CG2R61      1    GRA     C4      4          0     12.011   ; qtot 0
     5     CG2R61      2    GRA     C1      5          0     12.011
     6     CG2R61      2    GRA     C2      6          0     12.011
     7     CG2R61      2    GRA     C3      7          0     12.011
     8     CG2R61      2    GRA     C4      8          0     12.011   ; qtot 0
    37   598   599   600     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
   597   598   599    40     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
   597   598   599   600     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
    40   599   600   559     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
    40   599   600   561     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
   598   599   600   559     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00
   598   599   600   561     1  6.000000e+01  1.297040e+01  3.000000e+00  6.000000e+01  1.297040e+01  3.000000e+00

[ system ]
; Name
GRA

[ molecules ]
; Compound        #mols
GRA                 1
```

d) Change the size of the box in *z*-direction, this will create a vacuum, that you can then fill up with water. Open GRA_sheet.gro and modify the last line to:

```
3.68927   4.26000   3.28400
```

e) Center the molecule in the box:

```
gmx editconf -f GRA_sheet.gro -o GRA_sheet-centered.gro -c
```

f) create a new .top file, including links to all the forcefield and .itp files needed for your work. In this example, tip4p water has also been included as following. Call this file gra-w.top.

g) Solvate the system. The number of waters will automatically be updated in .top file

```
gmx solvate -cp GRA_sheet-centered -o GRA_w.gro -p gra-w.top
```

gra-w.top:

```
; Include forcefield parameters
#include "./charmm36.ff/forcefield.itp"

; Include topology for GRA
#include "./GRA.itp"
#include "./charmm36.ff/tip3p.itp"

[ system ]
; Name
GRA in water

[ molecules ]
; Compound        #mols
GRA                 1
SOL              1475
```

Pay attention to the number of SOL molecules. It should be the same as in the .gro file.

Now the system is ready to start the simulation.

5) Run an energy minimization. Make sure to include **periodic_molecules = yes** in your .mdp file

min.mdp

```
; Parameters describing what to do, when to stop and what to save
integrator  = steep         ; Algorithm (steep = steepest descent minimization)
emtol       = 1000.0        ; Stop minimization when the maximum force<1000.0kJ/mol/nm
emstep      = 0.01          ; Minimization step size
nsteps      = 50000         ; Maximum number of (minimization) steps to perform

; Parameters describing how to find the neighbors of each atom and how to calculate
the interactions
nstlist         = 1         ; Frequency to update the neighbor list and long range
forces
cutoff-scheme   = Verlet    ; Buffered neighbor searching
ns_type         = grid      ; Method to determine neighbor list (simple, grid)
coulombtype     = PME       ; Treatment of long range electrostatic interactions
rcoulomb        = 1.0       ; Short-range electrostatic cut-off
rvdw            = 1.0       ; Short-range Van der Waals cut-off
pbc             = xyz       ; Periodic Boundary Conditions in all 3 dimensions
periodic_molecules = yes
```

Then run:

```
gmx grompp -f min.mdp -c GRA_w.gro -p gra-w.top -o min1.tpr
gmx mdrun -v -deffnm min1
```