Tutorial: Introduction to QM/MM simulations using the GROMACS-CP2K Interface

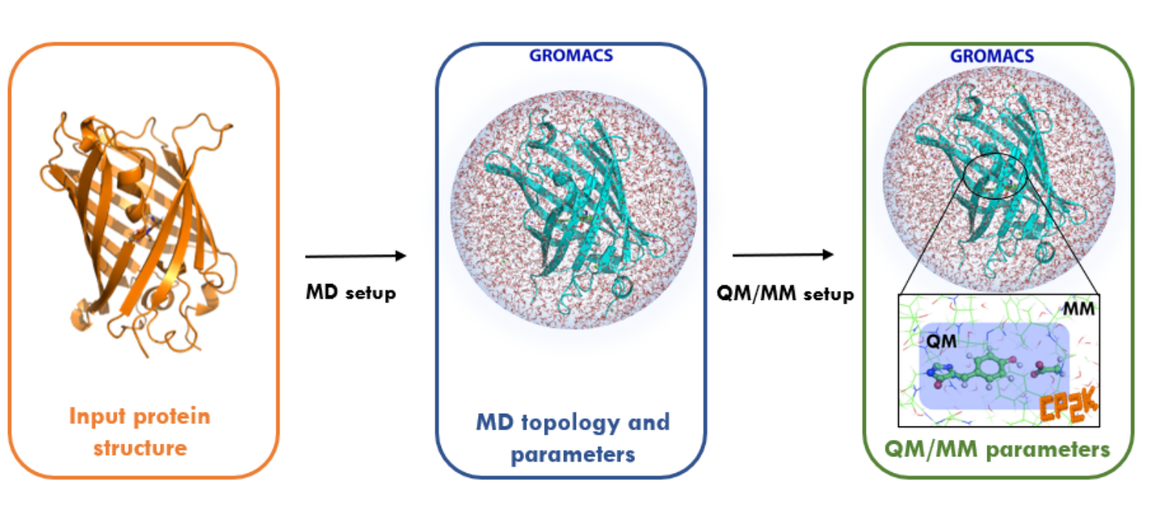
Overview

Questions

- What is GROMACS-CP2K QMMM Interface?
- How it could be used?

Objectives

- Getting started with GROMACS-CP2K Interface
- Learning how to prepare your system for a simple QM calculation
- Umbrella sampling using GROMACS-CP2K Interface
- Make protein QMMM system starting from the PDB structure



Preparing for the tutorial

Everything, which is written inside the `gray box` are a commands, that should be executed in the terminal window, string-by-string, each following with the ENTER button. Please note that `<...>` in the commands means, that everything, including `<>` symbols, must be replaced with your own specific information. Be careful!

Helpful utilities and commands

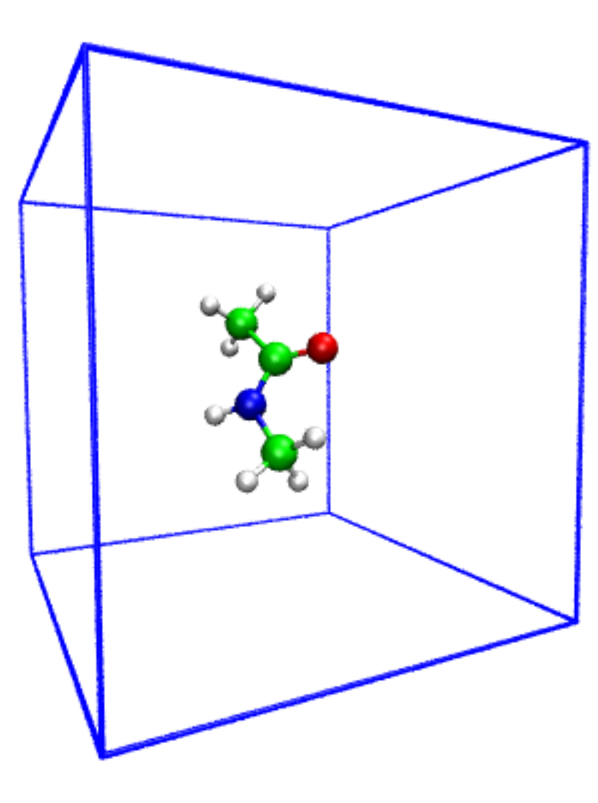
Some exercises will require usage of `less` Linux tool for looking up into the content of files. In case you are not familiar with it, here is a short list of hotkeys, which could be used inside LESS editor:

- q – exit
- / – search for a pattern which will take you to the next occurrence
- n – for next match in forward
- N (SHIFT+n) – for previous match in backward
- g – go to the start of file
- G (SHIFT+g) – go to the end of file

All exercises will require you to submit job for computing using `sbatch run.sh` command. This are typically used for submitting job on HPC clusters you might need to adjust that procedure, depending on available computing resources.

Exercise 1: Setting up simple QM system

1) Go to nma directory:
`cd nma`



In the directory located forcefield and **nma.pdb** file with a geometry of simple compuond N-Methylacetamide (NMA)
You can download it and inspect structure with VMD or PyMOL.

2) Make topology for the system using the following command:
`gmx pdb2gmx -f nma.pdb`
choose the following forcefield and water model:

```
Select the Force Field:
From current directory:
1: CHARMM36 all-atom force field (March 2019)
....
....
Select the Water Model:
1: TIP3P       TIP 3-point, recommended, by default uses CHARMM TIP3 with LJ on H
```

Files **topol.top**, **conf.gro** and **posre.itp** should appear in the directory

3) Look into Gromacs input file **em.mdp**:
`less em.mdp`
The following lines contain QMMM MdModule options:

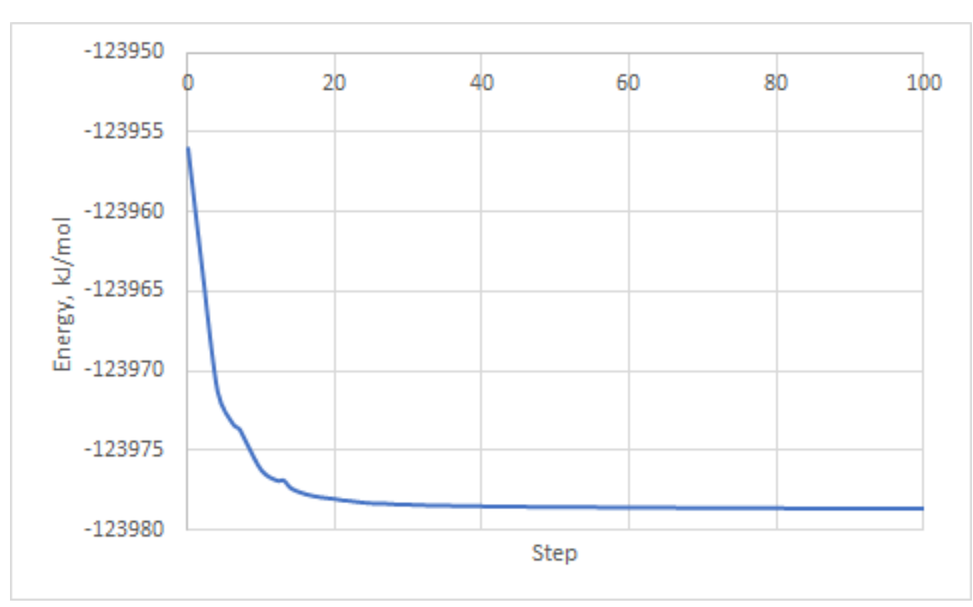
```
; CP2K QMMM parameters
qmmm-cp2k-active              = true   ; Activate QMMM MdModule
qmmm-cp2k-qmgroup             = System ; Index group of QM atoms
qmmm-cp2k-qmmethod            = PBE    ; Method to use
qmmm-cp2k-qmcharge            = 0      ; Charge of QM system
qmmm-cp2k-qmmultiplicity      = 1      ; Multiplicity of QM system
```

4) Lets perform energy minimization first for that molecule using QMMM interface
Generate Gromacs-CP2K simulation file:
`gmx grompp -f em.mdp -p topol.top -c conf.gro -o nma-em.tpr`
file **nma-em.tpr** should appear in the directory

5) Run QMMM simulation:
`sbatch run-em.sh`

6) While job is running you can check the content of **nma-em.inp** that should appear in the directory.
`less nma-em.inp`

7) At the end of the job use the following command to extract potential energy:
`gmx energy`
and choose `5 Potential`
File **energy.xvg** should appear in the directory. It contains data with Potential energy (kJ/mol) against optimization step. You can open it in Grace or copy data into any other software (i.e. Excel).

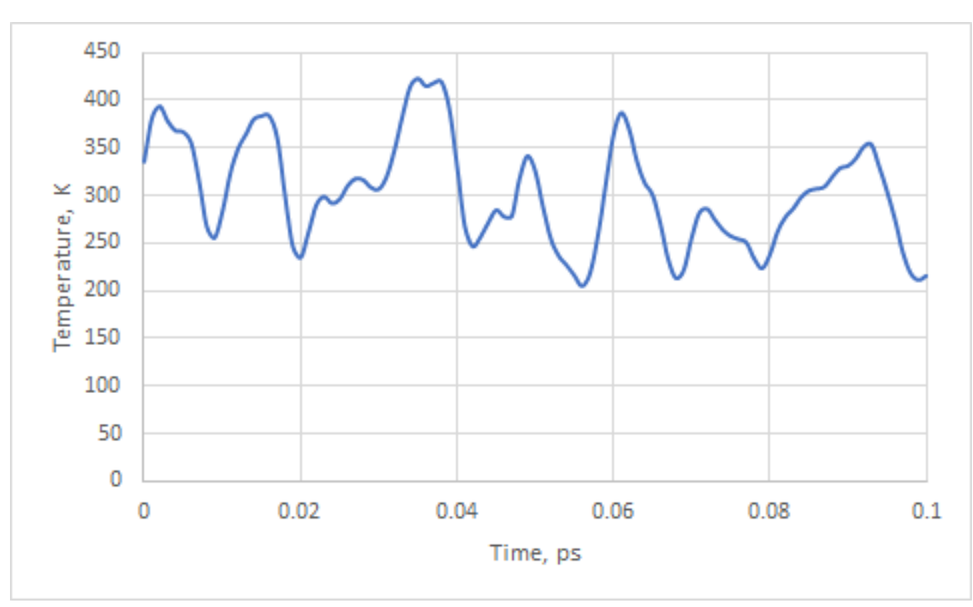


8) Next we will perform short (100 fs) MD simulation with QM. At first look into the **md-nvt.mdp** file:
`less md-nvt.mdp`
It contains parameters for performing dynamics with QM forces in the NVT ensemble at 300K

9) Generate Gromacs-CP2K simulation file:
`gmx grompp -f md-nvt.mdp -p topol.top -c conf.gro -o nma-nvt.tpr`
file **nma-nvt.tpr** should appear in the directory

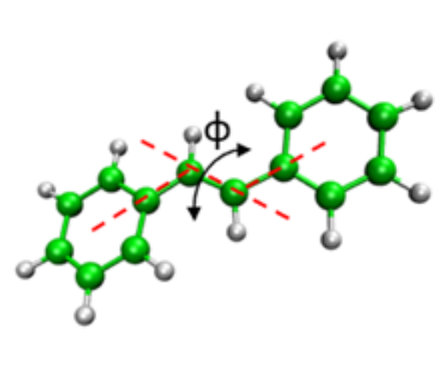
10) Run QMMM simulation:
`sbatch run-nvt.sh`

11) At the end of the simulation you can download trajectory file **traj.trr** and render it using your favorite software (e.g. VMD, PyMOL).
Also you could check temperature as a function of time with the following command:
`gmx energy`
and choose `9 Temperature`
File **energy.xvg** will contain data about Temperature (K) against simulation time (ps).
Notice, how temperature fluctuates around 300K.



Exercise 2: Umbrella sampling simulation with QMMM MdModule

1) Go to stilbene_vacuum directory:
`cd ../stilbene_vacuum`



2) Look up in the table and pick-up any starting structure and dihedral angle value, that are located in the **eq_gro** directory:

Structure	Dihedral angle, φ
md-equilb1.gro	-180
md-equilb2.gro	-173
md-equilb3.gro	-166
md-equilb4.gro	-159
md-equilb5.gro	-152
md-equilb6.gro	-145
md-equilb7.gro	-138
md-equilb8.gro	-131
md-equilb9.gro	-124
md-equilb10.gro	-117
md-equilb11.gro	-110
md-equilb12.gro	-103
md-equilb13.gro	-96
md-equilb14.gro	-89
md-equilb15.gro	-82
md-equilb16.gro	-75
md-equilb17.gro	-68
md-equilb18.gro	-61
md-equilb19.gro	-54
md-equilb20.gro	-47
md-equilb21.gro	-40
md-equilb22.gro	-33
md-equilb23.gro	-26
md-equilb24.gro	-19
md-equilb25.gro	-12
md-equilb26.gro	-5
md-equilb27.gro	2
md-equilb28.gro	9
md-equilb29.gro	16
md-equilb30.gro	23

3) Copy chosen starting structure:
`cp eq_gro/<your starting gro> ./conf.gro`

4) Modify Gromacs input file **qmmm_md_umbrella.mdp** with value of your chosen Dihedral angle:
`sed -i "s/@umbr@/<your dihedral angle>/" qmmm_md_umbrella.mdp`
You can also modify *pull-coord1-init* option in the **qmmm_md_umbrella.mdp** file with vim or any other editor.

5) Add group of atoms which will be treated with QM to the index file (in that case all atoms are QM):
`gmx make_ndx -f conf.gro -n index.ndx`

```
> 0
> name 7 QMatoms
> q
```

6) Generate Gromacs-CP2K simulation file:
`gmx grompp -f qmmm_md_umbrella.mdp -p topol.top -c conf.gro -n index.ndx -o stilbene.tpr`
file **stilbene.tpr** should appear in the directory

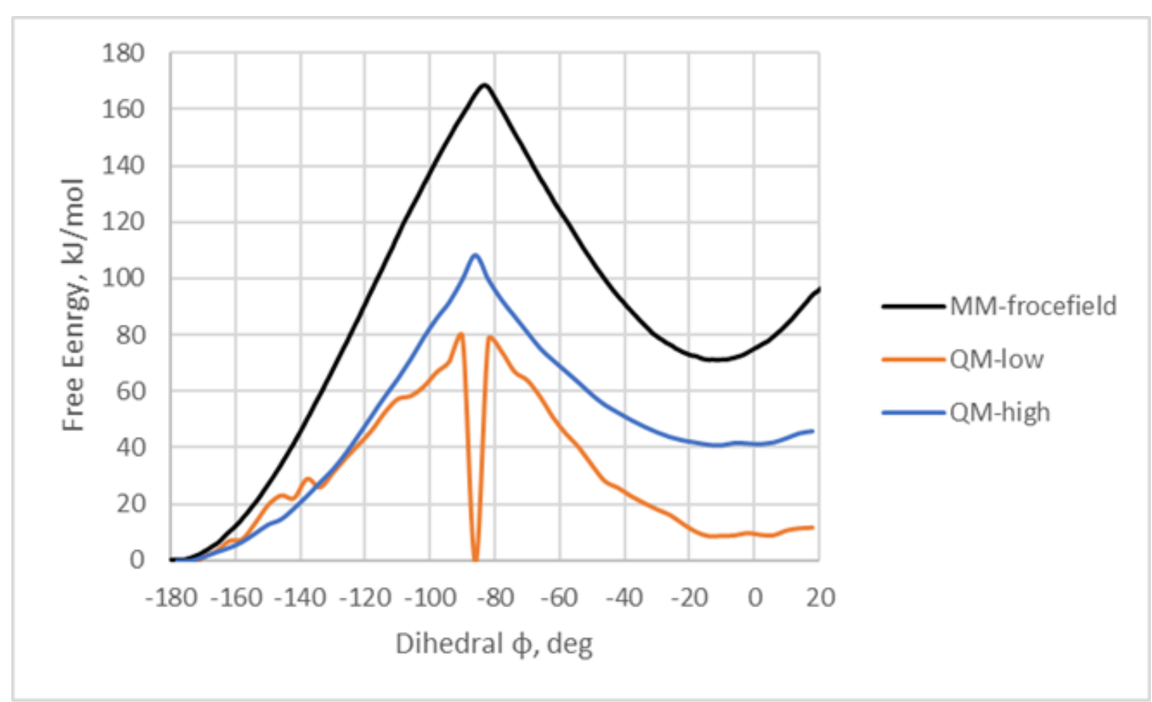
7) Run QMMM simulation:
`sbatch run.sh`

8) While job is running you can check the content of **stilbene.inp** that should appear in the directory.
`less stilbene.inp`
and of **qmmm_md_umbrella.mdp**
`less qmmm_md_umbrella.mdp`

9) At the end of the job you could check **pullx.xvg** file.
`less pullx.xvg`
It contains information about chosen coordinate dynamics over the simulation trajectory. By performing that sampling over the many points along reaction coordinate and gathering all *.tpr and pullx.xvg files you could produce free-energy profile of the reaction with `gmx wham` tool.

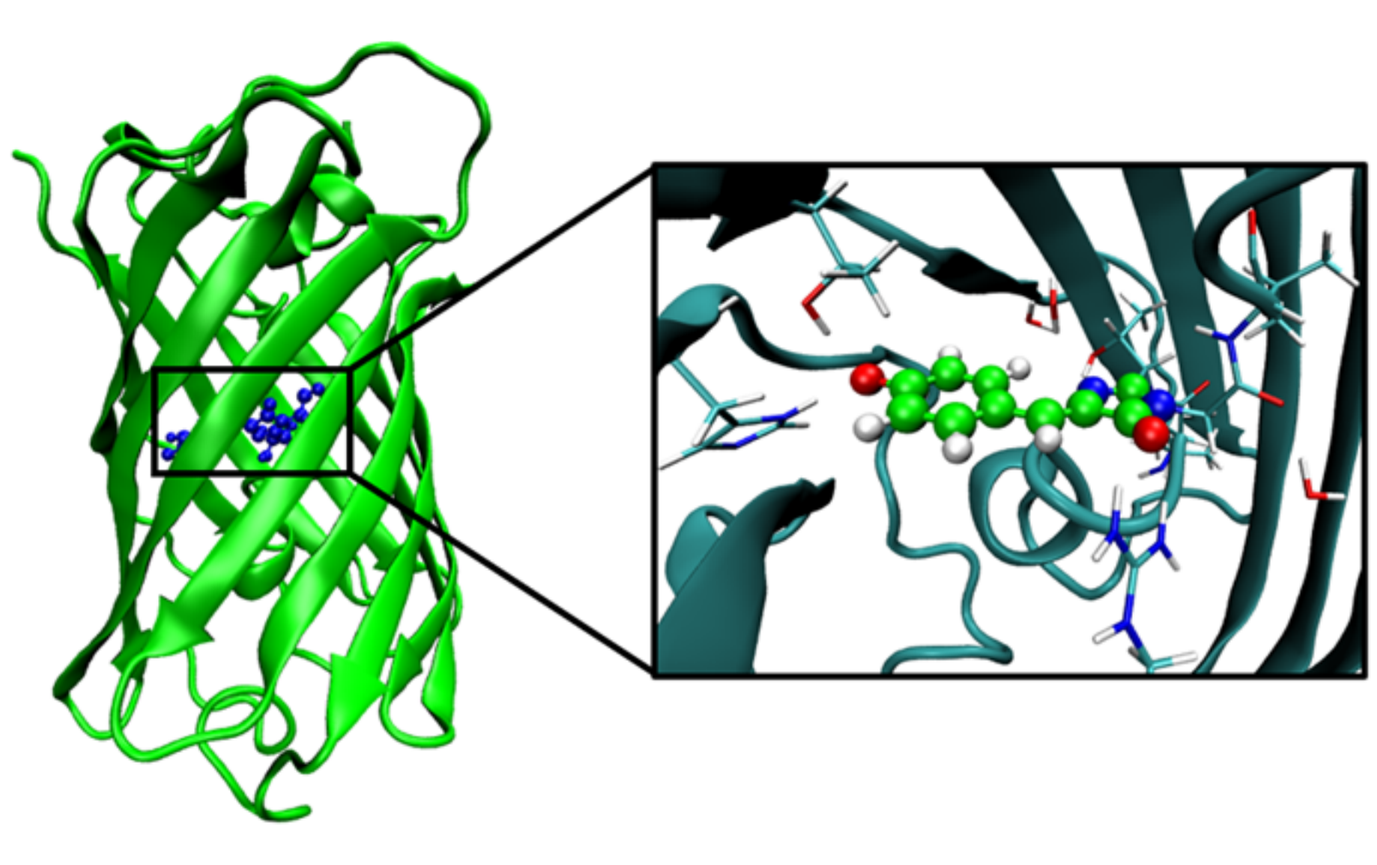
10) Sample output files for all umbrella windows are located in **profile-100fs** directory.
Go to that directory and generate profile using information gathered over 100 steps (100 fs) of the simulation:
`cd profile-100fs`
`gmx wham -it tpr-files.dat -ix pullx-files.dat -o -hist -unit kJ -min -180 -max 20 -b 0 -bins 50`
Files **profile.xvg** and **histo.xvg** should appear in the directory.
profile.xvg contains data about Free energy (kJ/mol) against Dihedral angle (deg).
histo.xvg contains distribution of the dihedral angle in each particular window.
You can download and open them in Grace or copy data into any other software (i.e. Excel).

11) Check and compare the free energy profiles generated from 100 steps (100 fs) you have just generated and from 10000 steps (10 ps) of QMMM MD simulation (**profile-10ps.xvg** file located in the same directory).



Exercise 3: Setting up simple protein system starting from the PDB file

1) Go to egfp directory:
`cd ../../egfp`



In the directory located forcefield and **4eul.pdb** file with a strucutre of EGFP protein dowloaded from PDB databank.
Note that file was modified, missing atoms have been added, first and last residues has been marked as N- and C- terminus, periodic box has been added 10 x 10 x 10 nm.
In addition forcefield for non-standard Chromophore residue CRO66 has been generated with Antechamber.
You can download it and inspect structure with VMD or PyMOL.

2) Make topology for the system using the following command:
`gmx pdb2gmx -f 4eul.pdb`
choose the following forcefield and water model:

```
Select the Force Field:
From current directory:
1: AMBER03 : Neutral GFP
....
....
Select the Water Model:
 1: TIP3P     TIP 3-point, recommended
```

Files **topol.top**, **conf.gro** and **posre.itp** should appear in the directory

3) Solvate te system in the **conf.gro**
`gmx solvate -cp conf.gro -o conf.gro -p topol.top -shell 10`

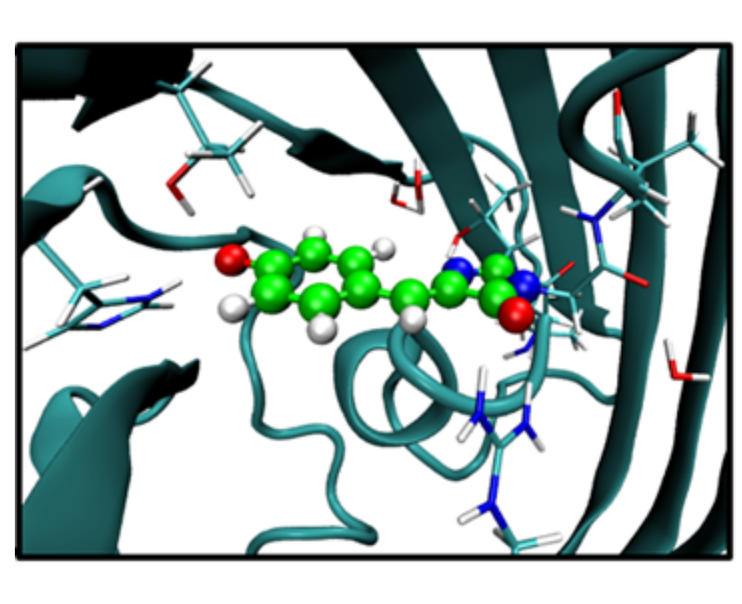
4) Now we need to make our system neutral by adding 6 Na+ ions
To do that first generate tpr file:
`gmx grompp -f em.mdp -p topol.top -c conf -o egfp-genions.tpr -maxwarn 10`
then use the following command to replace 6 random water molecules with Na+ ions
`gmx genion -s egfp-genions.tpr -p topol.top -o conf.gro -neutral`
select group 13 of SOL molecules `Select a group: 13`
after that manipulations your **conf.gro** and **topol.top** files will contain solvated and neutralized protein system.

5) The next step would be minimization and short classical equilibration NVT trajectory.
First generate and run energy minimization:
`gmx grompp -f em.mdp -p topol.top -c conf -o egfp-em.tpr`
`sbatch run-em.sh`
Wait until simulation will be completed.
Then perform 100 ps NVT simulation starting from the optimized structure to equilibrate our system:
`gmx grompp -f md-mm-nvt.mdp -p topol.top -c conf.gro -t egfp-em.trr -o egfp-mm-nvt.tpr`
`sbatch run-mm-nvt.sh`
while simulation is running you could check **em.mdp** and **mm-nvt.mdp** files for the details of classical MD simulations

6) Next step would be changing simulation from classical forcefiled to QMMM.
First generate index.ndx file that would contain QMatoms group, marking QM atoms in our protein: `gmx make_ndx -f conf.gro`
and do the following input

```
> a 938-956
> name 18 QMatoms
> q
```

Look into the **conf.gro** with VMD or PyMOL and make sure that atoms from 938 to 956 are the same as shown in spheres on the following figure:

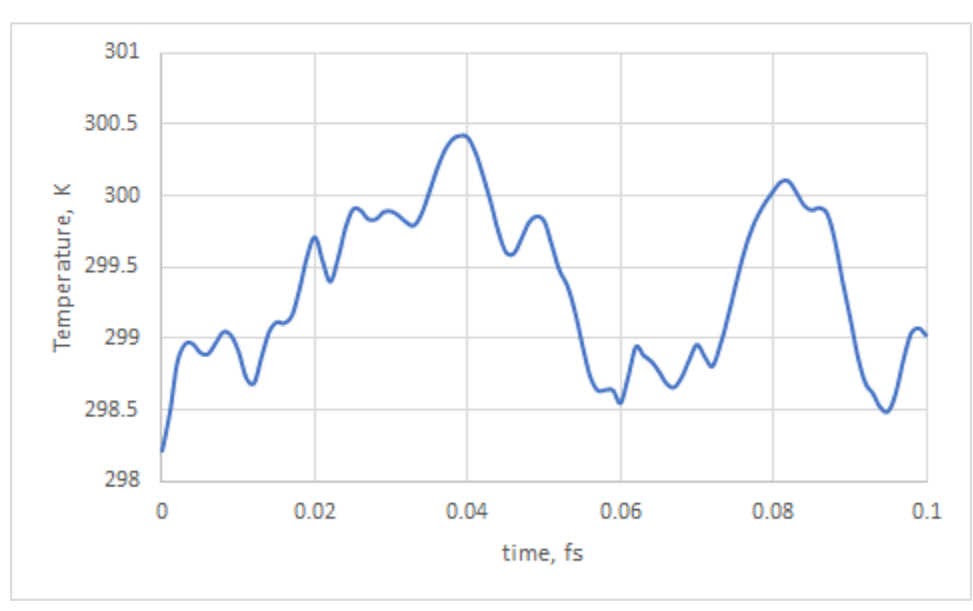


7) Now we are ready to generate QMMM simulation file:
`gmx grompp -f md-qmmm-nvt.mdp -p topol.top -c conf.gro -t egfp-mm-nvt.trr -n index.ndx -o egfp-qmmm-nvt.tpr`
Here we are using classically equilibrated trajectory **egfp-mm-nvt.trr** as a starting point for QMMM simulation.

8) Run QMMM simulation:
`sbatch run-qmmm-nvt.sh`
While simulation is running you could inspect **md-qmmm-nvt.mdp** and check that QM part in that case has charge -1.

```
; CP2K QMMM parameters
qmmm-cp2k-active              = true   ; Activate QMMM MdModule
qmmm-cp2k-qmgroup             = QMatoms; Index group of QM atoms
qmmm-cp2k-qmmethod            = PBE    ; Method to use
qmmm-cp2k-qmcharge            = -1     ; Charge of QM system
qmmm-cp2k-qmmultiplicity      = 1      ; Multiplicity of QM system
```

9) At the end of the simulation you can download trajectory file **egfp-qmmm-nvt.trr** and render it using your favorite software (e.g. VMD, PyMOL).
Also you could check temperature as a function of time with the following command:
`gmx energy -f egfp-qmmm-nvt.edr`
and choose `16 Temperature`
File **energy.xvg** will contain data about Temperature (K) against simulation time (ps).



Exercise 4: Using non-standard parameters in CP2K input

1) Stay in the same egfp directory

2) Copy **egfp-qmmm-nvt.inp** file:
`cp egfp-qmmm-nvt.inp egfp-qmmm-spec.inp`

3) Modify **egfp-qmmm-spec.inp** file you have just copied with `vim egfp-qmmm-spec.inp` or any other editor.
Insert between `&END DFT` and `&QMMM` lines an additional `&PROPERTIES` section.
Final result should look like that:

```
  &END DFT
  &PROPERTIES
    &TDDFPT
       NSTATES      5
       MAX_ITER    10
       CONVERGENCE [eV] 1.0e-3
    &END TDDFPT
  &END PROPERTIES
  &QMMM
```

This will order CP2K to also calculate 5 excited states at each MD step with TDDFT.

4) Generate Gromacs-CP2K simulation file:
`gmx grompp -f md-qmmm-spec.mdp -qmi egfp-qmmm-spec.inp -p topol.top -c conf.gro -t egfp-qmmm-nvt.trr -n index.ndx -o egfp-qmmm-spec.tpr`

5) Run simulation:
`sbatch run-qmmm-spec.sh`

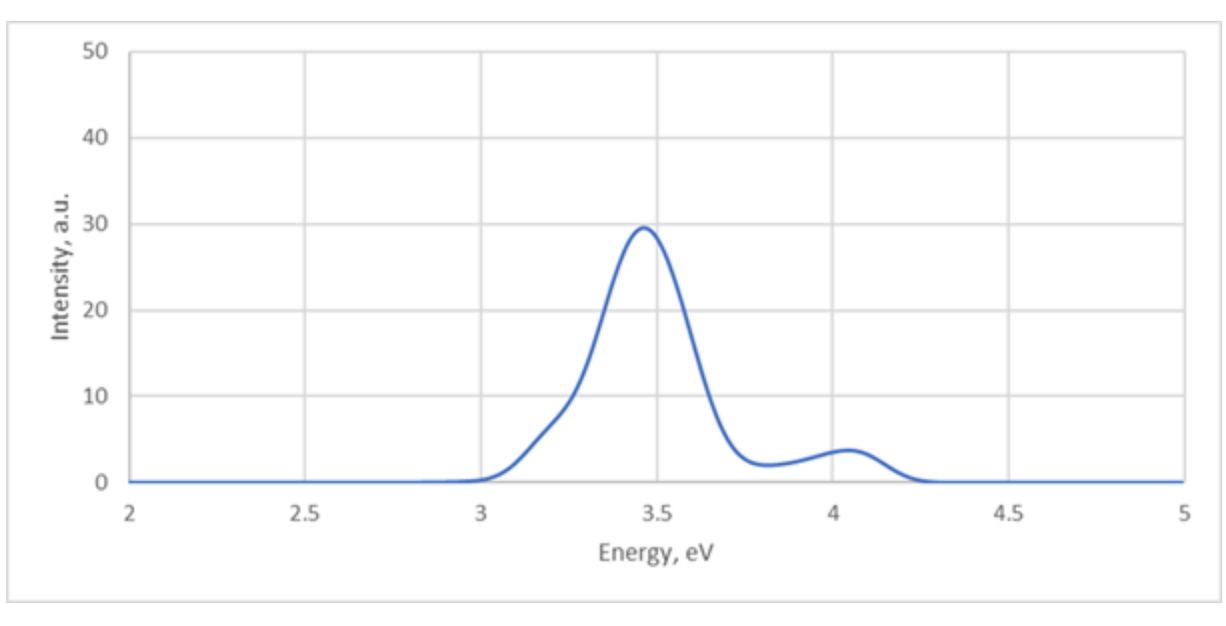
6) While it is running inspect content of **md-qmmm-spec.mdp** file:

```
; CP2K QMMM parameters
qmmm-cp2k-active              = true   ; Activate QMMM MdModule
qmmm-cp2k-qmgroup             = QMatoms; Index group of QM atoms
qmmm-cp2k-qmmethod            = INPUT  ; Method to use
```

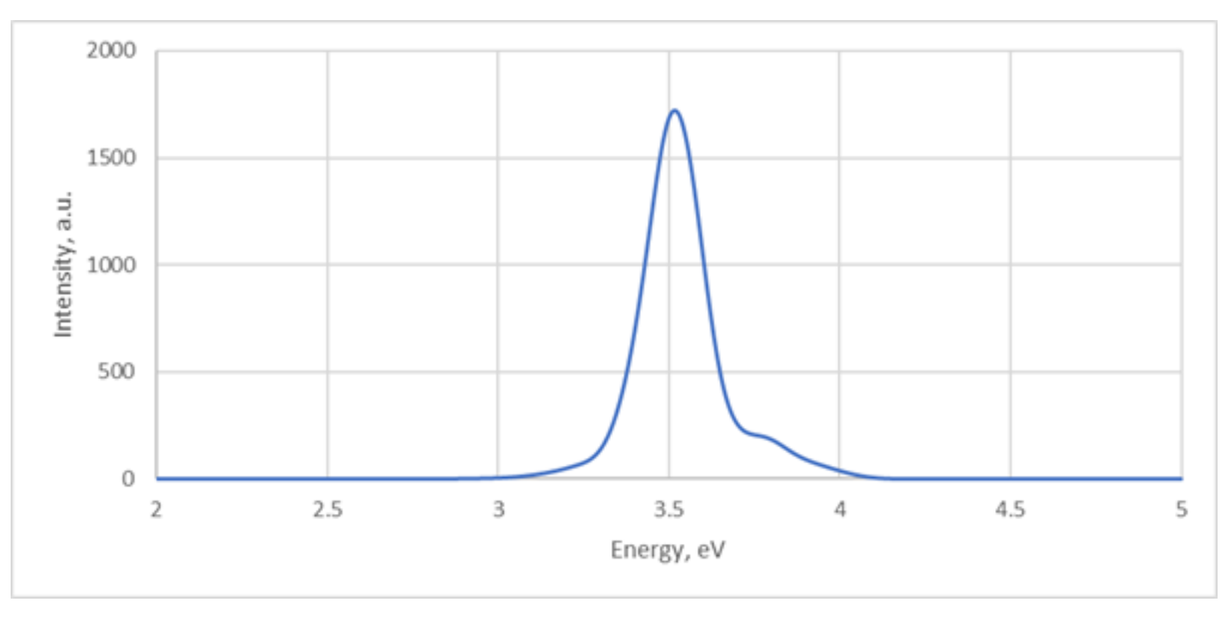
Combination of **qmmm-cp2k-method = INPUT** and **-qmi egfp-qmmm-spec.inp** instructs GROMACS to use external input file or CP2K. Note, that file will be saved into tpr and no longer needed after **grompp**

7) After job is finished, we need to gather information about excitation energies over the calculated trajectory:
`grep " TDDFPT|" egfp-qmmm-spec.out | awk '{ print $3 " " $7 }' > excitations`
The **excitations** file should appear in the directory, it will consist out of two columns. First column is an excitation energy (in eV) and second is an oscillator strength (in a.u.) for each excitation computed by CP2K. Final absorption spectra could be convolved by representing each excitation with gaussian function and sum up over all of them.

8) Convolve the spectra using provided Python script:
`./conv.py excitations 0.1 2 5`
File **spec.xvg** should appear in the directory. You can open it in Grace or copy data into any other software (i.e. Excel).
As an example, convolved spectra with 0.1 eV half-width gaussians over 100fs (100 steps) trajectory:



9) Spectra collected over 3 ps (3000 MD steps) will look like that:



Key Points

- QM simulation could be activated by adding several parameters into the mdp file
- Most of the simulation techniques from GROMACS are available also within QMMM
- When doing advanced sampling with QMMM one should be aware of the distribution and final profile quality
- *qmmm-qmmethod = INPUT* should be used for providing your own CP2K input file
- Advanced properties, like absorption spectra could be calculated using external input files