AmberTools - From the Setup to the Analysis

Vito Federico Palmisano

February 14, 2022

1 System Setup

For whoever is not connected to our machine, this is the architecture of the folders.

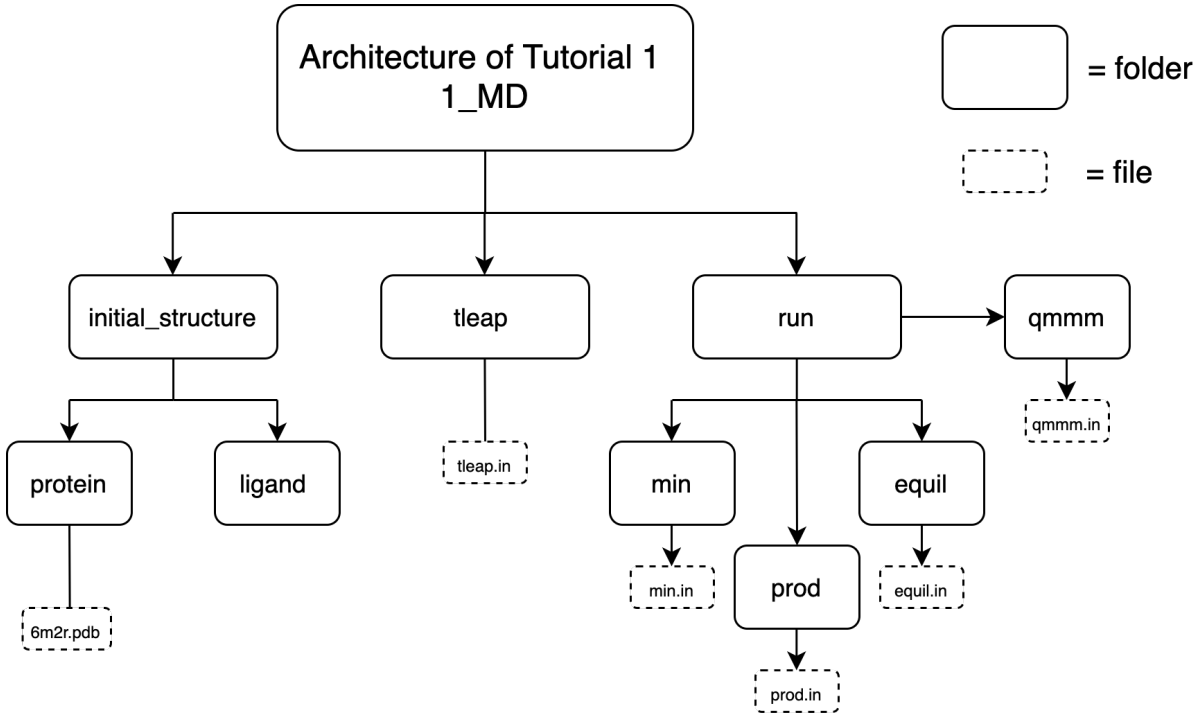

Figure 1: Architecture of the folders for the tutorial

In order to follow the tutorial with the same structure, please create the folders and include the input files which will be explicitly introduced in this pdf.
Before we start, we should check whether AmberTools and Visual Molecular Dynamics (VMD) are functioning in our work station. Please type :

```
sander
```

which should print "Error opening unit 5: File "mdin" is missing or unreadable" since we are not providing any inputs to the program and then type :

```
vmd
```

which is the keyword to open the VMD program. If VMD is not working from your laptop connected to our work station, then please create a folder in your own laptop and send the files from the directory in our work station to your laptop every time you require visualization. For example, let's suppose you are inside a folder in your own computer, then you can send a file from our work station to the current folder as:

```
scp tutorial@150.244.121.202:/data/tutorial/file.dat myfolder/.

scp [source] [destination]
scp -r [source] [destination]  (for folders)
```

Finally, some versions of VMD cannot read binary files (.nc extension), so we will not work with binary files, but usually when you have large simulations the coordinate file is so big that binary files are highly suggested.
If those two programs work, then we can start the tutorial, good luck !

1.1 Obtain the PDB structure

The https://www.rcsb.org/ is an archive containing all the PDB (Protein Data Base) structure obtained yearly from X-ray crystallography, NMR spectroscopy, cryo electron microscopy experiments performed by scientists all around the world. Each PDB structure has an ID of four letters/numbers, but due to the exponential growth of new structures available, soon it will be changed since there is a limited number of possible combinations available. The website offers the possibility to visualize and download all those 3D structures and also analyze them (sequence alignment, protein quality and symmetry).

Today we will be working with PDB-ID: **6M2R** which is the dopamine transporter bound to norephinephrine (noradrenaline). The structure is already downloaded and placed in each of your folders, in 1-MD/initial-structures/protein
The first step is to open this structure with your favorite text editor to observe it:

```
vi 6m2r.pdb
```

The first part gives a physiological description of the system and different crystallized sequences of the PDB. The actual structure begins in "ATOM 1", and from this part on, it is

fundamental to understand that pdb structure recognition of most of all softwares is **Case Sensitive**, therefore the spacing must be kept the way it is and each element correspond precisely to a column.

ATOM	1	N	ASP	A	25	−23.642	28.571	−36.738	1.00127.32	N
ATOM	2	CA	ASP	A	25	−22.594	29.551	−37.001	1.00127.91	C
ATOM	3	C	ASP	A	25	−21.314	28.874	−37.492	1.00127.30	C
ATOM	4	O	ASP	A	25	−21.324	28.164	−38.501	1.00123.87	O
ATOM	5	CB	ASP	A	25	−22.313	30.383	−35.745	1.00128.50	C
ATOM	6	CG	ASP	A	25	−22.798	31.817	−35.875	1.00132.02	C
ATOM	7	OD1	ASP	A	25	−22.475	32.463	−36.895	1.00127.64	O
ATOM	8	OD2	ASP	A	25	−23.509	32.294	−34.962	1.00131.72	O
ATOM	9	N	GLU	A	26	−20.215	29.093	−36.771	1.00122.26	N
ATOM	10	CA	GLU	A	26	−18.906	28.583	−37.155	1.00112.42	C
ATOM	11	C	GLU	A	26	−18.395	27.614	−36.097	1.00105.13	C
ATOM	12	O	GLU	A	26	−18.354	27.948	−34.907	1.00105.45	O
ATOM	13	CB	GLU	A	26	−17.913	29.731	−37.364	1.00116.92	C
ATOM	14	CG	GLU	A	26	−18.269	30.641	−38.536	1.00120.84	C
ATOM	15	CD	GLU	A	26	−17.232	31.723	−38.787	1.00124.28	C
ATOM	16	OE1	GLU	A	26	−16.614	32.202	−37.808	1.00124.52	O
ATOM	17	OE2	GLU	A	26	−17.036	32.090	−39.969	1.00114.53	O

The 2^{nd} column is the atom index, the 3^{rd} is the atom type, the 4^{th} is the residue type, the 5^{th} is the residue sequence (chain), the 6^{th} is the residue index, the 7^{th}, 8^{th} and 9^{th} are the XYZ coordinates and the last three columns are occupancy, temperature-factor and element symbol. Each residue sequence terminates with the line TER, therefore you could more around the text editor searching for "TER". Now it is time to go from a text editor to a visualization software to observe our system and select only the portion of the PDB that we require.

```
vmd 6m2r.pdb
```

Let's go to **Graphics** → **Representations** and under **Selected Atoms** we can select the atom, residue, chain, etc.. of our interest and use the **Coloring Method**, **Drawing Method** and **Materials** to help us visualize the structure and also appreciate the art which nature gives us at the nanoscale. Select chain A, which is the transporter we are interested in, close the **Graphical Representation** and go to **File** → **Save Coordinates** → **Selected Atoms** and save the chain A, preferably in the same folder and with a different name. Let's open our newly generated file chainA.pdb and observe it again. If you go at the end of it you will notice that all the ligands of chain A have been kept in the structure, therefore remove them all except the LNR which corresponds to the Norephinephrine.
Let's go to the next section and start working with our PDB structure.

1.2 Preparation of Initial Structures

This subsection is divided into Protein and Ligand since we need to treat them as two distict entities for the system preparation.

1.2.1 Protein

Let's copy the ligand to its corresponding folder and forget about it for this subsection.

```
grep LNR chainA.pdb > ../ligand/lig.pdb
```

Open again the chainA.pdb and remove the ligand, that is all the lines containing the LNR residue name.
Now we begin to use the Amber packages, and the first one allows us to clean up the PDB file from any atom or residue indices which may not be recognized later on by other Amber packages.

```
reduce -trim chainA.pdb > chainA_noh.pdb
pdb4amber --reduce chainA_noh.pdb > amber.pdb
```

The "reduce -trim" command removes all the hydrogens from the pdb structure, since many times the atom types of the H atoms are defined differently between different softwares. The "pdb4amber –reduce" instead controls all the other atom types, changes them accordingly to the software and adds all the removed hydrogens with the correct atom type. For a more precise explanation for those commands please go to the amber manual (Amber20) at p.203.

1.2.2 Ligand

Enter the previously created folder called "ligand". To treat the ligand, we will be using another Amber package called Antechamber which is used for recognizing the atom type; (2) recognizing bond type; (2) judging the atomic equivalence; (3) generating residue topology file; (4) finding missing force field parameters and supplying reasonable and similar substitutes. Please type :

```
antechamber -h
antechamber -L
```

to visualize on your screen the commands available (-h) and the formats available (-L). We will be using this package to generate from our pdb file a mol2 file which contains most of the information required for force field recognition. For small organic molecules, Amber uses the General Amber Force Field (GAFF) which contains 33 basic atom types and 22 special atom types, where the the equilibrium bond lengths and bond angles came from ab initio calculations at the MP2/6-31G*. The force constants for bonds and angles were estimated using empirical models, and the parameters in these models were trained using the force field

parameters in the traditional AMBER force fields. I've noticed that sometimes antechamber does not change the atom type generated by the PDB database, therefore I suggest to first convert the pdb structure into an xyz file and then convert it to a mol2 file, to avoid possible mistakes in the mol2 recognition (p.287). Please type :

```
antechamber -i lig.pdb -fi pdb -o lig.com -fo gcrt
antechamber -i lig.com -fi gcrt -o lig.mol2 -fo mol2 -c bcc -at gaff
```

At this point we have a mol2 file which will be later on treated by the force field when we build the full system. The final step is to generate an extra file (frcmod = force field modification) which takes care of the modifications that are needed for the molecule but not given by the force field. Please type :

```
parmchk2 -h
parmchk2 -i lig.mol2 -f mol2 -o lig.frcmod
```

The parmchk2 package reads a mol2 file or a force field file,

($AMBERHOME/dat/leap/parm/gaff.dat)

The parkmchk2 package writes the modifications, adds the missing force field parameters and prints the problematic parameters with a scoring penalty number.
If everything works fine, we can move to the next section and build the system.

1.3 Build the Solvated System

We finally have all the required files with a correct nomenclature and it is time to build the system. The system generation is taken care of by the "tleap" package which allow us to generate an Amber coordinate file and a parameter file which contains all the information given by the force field used. Let's move out of the initial-structure folder, enter the tleap folder and copy both the protein and ligand files generated (amber.pdb - in the protein folder) and (lig.mol2, lig.frcmod - in the ligand folder). In the current directory you will find an input file, tleap.in so please open it :

```
source leaprc.gaff2
source leaprc.protein.ff19SB
source leaprc.water.tip3p

#Load structures and parameters

MOL = loadmol2 lig.mol2
loadamberparams lig.frcmod
saveoff MOL lig.lib

PRO = loadpdb amber.pdb
```

```
#Combine ligand and protein

COM = combine {PRO MOL}

#Solvate the combined system

solvateoct COM WAT 12

#Save pdb, parameter, coordinate files

saveamberparm COM system.parm7 system.rst7
savepdb COM system.pdb

quit
```

and execute it as :

```
tleap -f tleap.in
```

In this input file we've source the required force fields, loaded the previously created ligand and protein file, used the function "combine" to make a unique file for both, solvate the combined file in water and saved the topology (system.parm7) and the coordinate (system.rst7) files (p.213). If everthing proceeded correctly, then we can start our first molecular dynamics simulation.

2 Running Simulation

Let's first enjoy our solvated system by opening it in vmd:

```
vmd system.parm7 system.rst7
```

or

```
vmd system.pdb
```

Please remember that you cannot visualize the topology and the coordinate files alone because the former is missing the XYZ coordinates and the latter is missing the atom type therefore don't panic if you try to open the topology alone in vmd and you don't see anything. Play with the visualization software as much as you want and when you are done, exit the tleap folder and enter the "run" folder. You will see four subfolders : "min", "heat", "prod" which are the basic ingredients for the simulation and "qmmm" which will be used later on. Please enter the "min" folder, and open the min.in file.

```
Minimization
&cntrl

imin=1        ! Turn on minimization
maxcyc = 200 ! Maximum number of minimization cycles
ncyc = 100    ! 100 steepest-descent steps

ntpr = 2      ! Print energies every 2 steps
ntxo = 2      ! Print binary file
/
```

where the purpouse of the minimization is to find a local energy minimum of the starting structure so that the MD simulation does not "blow up" (i.e. the forces on any atom are not so large that the atoms move an unreasonable distance in a single timestep). In this situation the maxcyc provides the total number of steps (200) of which the last 100 the minimization algorithm is switched from steepest descent to conjugate gradient.

To run the simulation we need to use the sander (Simulated Annealing with NMR-Derived Energy Restraints) package of Amber which carries out energy minimization and molecular dynamics. Please execute it as :

```
sander -O -i min.in -p system.parm7 -c system.rst7 -o min.out
-r min.rst7 -inf min.info
```

The "-O" allows to overwrite files. When the calculation is done, we can open the min.out file and look if the total energy has converged. If so, we can continue and heat up the system. This stage is called heating but could also be confused with equilibriation; the name does not really matter if you understand the purpose. Ultimately, we want to run a simulation in a particular thermodynamic ensemble (NVE,NPT) at a particular state point (target energy, temperature, and pressure) and collect data when those conditions are reached. Usually, even though velocities are assigned according to the correct distribution, a thermostat will still need to add or remove heat from the system as it approaches the correct partitioning of kinetic and potential energies. For this reason, it is advised that a thermostatted simulation is performed prior to a desired production simulation, even if the production simulation will ultimately be done in the NPT ensemble. So the heating corresponds to a portion of the simulation where we want to relax to the temperature of interest. Please enter the heat folder and open the heat.in file:

```
Heat
 &cntrl
   imin=0,           ! Turn off minimization
   ntx=1,            ! Our starting file has no input velocities
```

```
  irest=0,            ! This is NOT a restart of an old MD simulation
  nstlim=3000,        ! Number of timesteps
  dt=0.002,           ! 2fs timestep


  ntf=2,              !Setting to not calculate force for SHAKE
  ntc=2,              !Enable SHAKE to constrain all bonds with H


  tempi=0.0,          !Initial thermostat temperature in K
  temp0=300.0,        !Final thermostat temperature in K



  ntpr=1,             ! Print energies every 1 step
  ntwx=10,            ! Print coordinates every 10 steps to the trajectory
  ioutfm=0,           ! Print coordinates in ASCII
  iwrap=1,            ! Wrap coordinates into primary box
  cut=8.0,            ! Nonbonded cutoff, in Angstroms


  ntb=1,              !Periodic boundaries for constant volume
  ntp=0,              !No pressure control
  ntt=3,              !Temperature control with Langevin thermostat
  gamma_ln=2.0,       !Langevin thermostat collision frequency


  nmropt=1,           !NMR restraints ON
  ig=-1,              !The seed for the pseudo-random number generator
 /


&wt type='TEMP0', istep1=0, istep2=3000, value1=0.0, value2=300.0 /
&wt type='END' /
```

Now the file became a bit more complicated, because more variables are added since we want to get closer to a real system. To run the heating simulation, we require the topology (system.parm7), but also the coordiantes of the minimized structure (min.rst7), therefore please copy those two files in the "heat" folder. To run the simulation, please execute it as :

```
sander -O -i heat.in -p system.parm7 -c min.rst7 -o heat.out
-x heat.crd -r heat.rst7 -inf heat.info
```

When the calculation is done we can open the output "heat.out". Fortunately Amber provides also a perl script to automatically extract the summary of all the state functions. Please execute:

```
process_mdout.perl heat.out
```

and you will see many files appearing in your folder. Let's plot for example the temperature and then the total energy:

```
xmgrace summary.TEMP
```

and

```
xmgrace summary.ETOT
```

If everything went fine, we can proceed to the next and final step of MD simulation, the production run. Now we want to move from an NVT ensemble to an NPT ensemble and start collecting data from the point where the system is equilibriated. Please enter the prod folder and open the prod.in file:

```
Production
 &cntrl
  imin       = 0         ! No minimization but molecular dynamics
  irest      = 1         ! This is NOT a restart of an old MD simulation
  ntx        = 7         ! Our starting file has input velocities

  ntb        = 2         ! Periodic Boundary Conditions in the NPT

  ntp        = 1         ! Isotropic scaling of volume
  barostat   = 1         ! Berendsen barostat
  pres0      = 1.0       ! Pressure in bars
  taup       = 2.0       ! Relaxation time is ps

  cut        = 8.0       ! Cutoff Lennard-Jones real-space Ewald int
  ntc        = 2         ! Enable SHAKE to constrain all bonds with H
  ntf        = 2         ! Setting to not calculate force for SHAKE

  tempi      = 300.0     ! Initial temperature
  ntt        = 3         ! Langevin thermostat
  gamma_ln   = 1.0       ! Collision frequency in ps-1

  nstlim     = 1000      ! Number of timesteps
```

```
  dt          = 0.002     ! Time step in ps

  ntpr        = 1         ! Energy is printed every N steps
  ntwx        = 10        ! Trajectory is printed every N steps
  ntwr        = 20        ! Restart file is printed every N steps
  ntxo        = 1         ! ASCII format for final coord, vel, box size
  ioutfm      = 0         ! ASCII format for trajectory (xyz) file
  iwrap       = 1         ! Wrap coordinates into primary box
 /
```

Again, copy the topology (system.parm7) and the coordinate file of the last frame of the heating run (heat.rst7) to the "prod" folder. Please execute it as :

```
sander -O -i prod.in -p system.parm7 -c heat.rst7 -o prod.out
-r prod.rst7 -x prod.crd -inf heat.info
```

Please, for more information on this section check (p.335).
It is time for a break !!!!!

3 Post Processing Analysis

We finally performed a MD simulation, and now the science begins. Until now we created a setup and simulated a solvated protein-ligand structure but the most important thing is to extract information out of it. The final Amber package that I will show you today is called CPPTRAJ, which is the post processing trajectory analyzer. The program works both interactively and as a script which can be executed as :

```
(script)
cpptraj -i file.in
```

or

```
(interactively)
cpptraj
```

where the file.in should include the same lines of code that you would write interactively. But to better understand it I will guide you through the interactive approach. First, we need to enter cpptraj and load the topology file :

```
cpptraj
```

```
parm system.parm7
```

already at this stage, without loading the actual trajectory we could extract many information from this complicated file. But before we continue, let's discuss the **Atom Mask Selection** and **Distance Mask Selection** which will allows us to select only a portion of our system for analysis.

3.1 Atom Mask Selection

Tha main characters are " @ and : ". The @ corresponds to atom selection, the : corresponds to residue selection from the pdb columns that we observed at the beginnig, for example :

```
@1-100          ! Select atoms from 1 to 100

:1-100          ! Select residues from 1 to 100

@CA             ! Select all the alpha carbons

:TRP            ! Select all the tryptophan residues

:TRP@CA         ! Select all the alpha carbon of all tryptophan
```

And also the logical operands & (and), | (or) , ! (not) work, for example :

```
!@CA,C,O,N,H            ! Exclude all the backbone atoms

:1-500@O&!(LYS,ARG)     ! Oxygens of res 1 to 500 excluding LYS and ARG
```

We could go on with many more examples but please check Ch.3 (3.2) of the cpptraj manual ($https://amber-md.github.io/cpptraj/CPPTRAJ.xhtml$.

3.2 Distance Mask Selection

There is the possibility also to select based on distance, and therefore include or exclude based on atom or residue distance. For example :

```
@1-100<@3.0     ! Select everything within 3$\AA$ from atom 1 to 100

:33>:20.0       ! Select everything out of 20.0 from residue 33
```

It is time to combine this atom selections with all the other functionalities that this program offers.

3.3 Topology File Commands

Let's go back to our topology and apply now the knowledge of the atom/distance mask selection. Those commands control the reading and writing of topologies and cpptraj supports Amber topology, PDB, Mol2, CIF, Charmm PSF, Gromacs topology, SDF, Tinker. For example we can use the commands : angleinfo, atominfo, bondinfo, charge, dihedralinfo, mass, parmbox, parmstrip and many more. Please try to execute those commands with the atom/distance mask selection. Let's load our topology file from inside cpptraj :

```
parm system.parm7
```

or

```
cpptraj                    ! In case you exited the program
parm system.parm7
```

And for example:

```
charge :1-100          ! print charge of residues 1 to 100
dihedralinfo :MOL      ! print dihedrals of our ligand
parmbox                ! print the box information
parmstrip :WAT         ! remove the water from the topology
```

Of course, every time you want to then save your action, you need to write "out" followed by the name of the file where you would like to save your information and then "go" to start the actions you provided, for example :

```
dihedralinfo :MOL out dihedral-lig.dat
go
```

Apart from the new topology that we create by removing the water molecules, where we require an additional keyword to save :

```
parmwrite out system-nowat.parm7
```

Let's remember this action because it will be important for the next section. For more information please go to Ch.9 of the cpptraj manual.

3.4 Trajectory File Commands

Up until now we have only worked with the topology, therefore it is time to also add the trajectory to cpptraj and apply again our previous knowledge.

```
cpptraj
parm system.parm7
trajin prod.crd
```

We've just loaded the trajectory on top of the topology file. As for VMD, you cannot only load the trajectory but you always require a topology first, otherwise you would be loading random coordinates in space with no physical meaning. The "trajin" command controls the reading of the trajectory and accepts Amber trajectory (.crd), Amber NetCDF (.nc), Amber restart (.rst7), Charmm DCD trajectory (.dcd), Charmm Coordinates (.cor), Charmm restart (.res), PDB (.pdb), Mol2 (.mol2) and also Gromacs, Tinker and Desmond files.
It can combine many different trajectories (only if the topology of this trajectories is the same, otherwise you won't be able to load it) and you can also extract different frames with a "start,stop,step" style, for example :

```
cpptraj
parm system.parm7
trajin *.crd                ! Load all the .crd trajectories

or

trajin prod.crd 1 50 2      ! Load from frame 1 to 50 every 2 frames
```

And to save your trajectory :

```
trajout newtraj.crd
```

Most of the times cpptraj recognizes the output from the extension but you can also remind him by rewriting the extension next to the output file, also in the case where you want to give it a different extension but keep the specified format,

```
trajout newtraj.gustavo  nc
```

Trajout also works for .crd, .rst7, .pdb, .mol2 and many other extensions. Let's suppose for some reason you have lost the pdb of the full system, then you could obtain in in the following way.

```
cpptraj
parm system.parm7
trajin prod.crd 1 1 1    ! Load from frame 1 to 1 every 1 frame
trajout system-f1.pdb pdb
go
```

```
exit
```

and you will find a system-f1.pdb corresponding to the pdb file of the first frame of the production run. For more options on the trajectory file commands please go to Ch.10 of the cpptraj manual.

3.5 Action Commands

The action commands operated on the trajectory files loaded into cpptraj and help us extracting all the information we want from our system in time. There is almost 100 actions available therefore I will list few of them just to provide example on the input and apply again the knowledge we acquired until now. Let's start from the beginning :

```
cpptraj
parm system.parm7
trajin prod.crd 1 40 2
distance @8487 @8488 out distance-co.dat
angle @8493 @8494 @8495 out angle-ccc.dat
go
exit
```

I really hope you understood the basic ingredients of post processing, because I believe that it is a powerfull tool to use. Ambertools is free, therefore all packages that we encountered in our path can be used and explored. Now we can move to the QM/MM approach with more confidence.

4 Running QM/MM Simulations

When we run a QM/MM simulation in **sander**, the system is partitioned into two regions, a QM region consisting of a portion of the system defined as a "qmmask" and all the rest of the system that is not part of the QM region. There is no universal choice for the selection of the QM region, but consider that having more than 80-100 atoms in the QM region will lead to simulations that are very expensive. Also, in choosing the QM/MM boundary it is always better to cut non-polar bonds than to cut unsaturated or polar bonds. Let's move out of the prod folder and enter the "qmmm" folder. Please open the qmmm.in file :

```
Production
 &cntrl
  imin       = 0         ! No minimization but molecular dynamics
  irest      = 0         ! This is NOT a restart of an old MD simulation
  ntx        = 1         ! Our starting file has input velocities

  ntb        = 2         ! Periodic Boundary Conditions in the NPT
```

```
  ntp         = 1          ! Isotropic scaling of volumen
  barostat    = 1          ! Berendsen barostat
  pres0       = 1.0        ! Pressure in bars
  taup        = 2.0        ! Relaxation time is ps

  cut         = 8.0        ! Cutoff Lennard-Jones real-space Ewald int
  ntc         = 2          ! Enable SHAKE to constrain all bonds with H
  ntf         = 2          ! Setting to not calculate force for SHAKE

  tempi       = 300.0      ! Initial temperature
  ntt         = 3          ! Langevin thermostat
  gamma_ln    = 1.0        ! Collision frequency in ps-1

  nstlim      = 20         ! Number of timesteps
  dt          = 0.002      ! Time step in ps

  ntpr        = 1          ! Energy is printed every N steps
  ntwx        = 1          ! Trajectory is printed every N steps
  ntwr        = 2          ! Restart file is printed every N steps
  ntxo        = 1          ! ASCII format for final coord, vel, box size
  ioutfm      = 0          ! ASCII format for trajectory (xyz) file
  ifqnt       = 1          ! Turn on QM/MM option
 /
 &qmmm
  qmmask     = ':MOL',   !  QM region mask
  qmcharge   = 0,        !  QM region charge
  qmcut      = 8.0,      !  QM region LJ cutoff
  qm_theory  = 'PM6',    !  Call External QM software
  qm_ewald   = 0,        !  Use a real-space cutoff for QM-QM and QM-MM
 /
```

Please execute it as:

```
sander -O -i qmmm.in -p system.parm7 -c prod.rst7 -o qmmm.out -r
qmmm.rst7 -x qmmm.crd -inf qmmm.info
```

In this situation we have performed a simulation using the built-in semiempirical PM6 of Amber. The software offers the possibility to interface with most of ab initio and DFT methods of of many different softwares as Orca, Gaussian, ADF, NWChem, QChem, TeraChem, MRCC, Fireball (p.156).
Recently a stunning software came out, called **MoBioTools**, which is able to generate inputs for QM/MM calculations in the simplest way possible, but will be my colleague and also developer of the software to guide you through it the next tutorial.

Therefore as a final example, let's use the Amber interface with orca to look at a simple input for this QM/MM simulation, please enter the "orca" folder and open orca.in

```
Production
 &cntrl
  imin       = 0         ! No minimization but molecular dynamics
  irest      = 0         ! This is NOT a restart of an old MD simulation
  ntx        = 1         ! Our starting file has input velocities

  ntb        = 2         ! Periodic Boundary Conditions in the NPT

  ntp        = 1         ! Isotropic scaling of volumen
  barostat   = 1         ! Berendsen barostat
  pres0      = 1.0       ! Pressure in bars
  taup       = 2.0       ! Relaxation time is ps

  cut        = 8.0       ! Cutoff Lennard-Jones real-space Ewald int
  ntc        = 2         ! Enable SHAKE to constrain all bonds with H
  ntf        = 2         ! Setting to not calculate force for SHAKE

  tempi      = 300.0     ! Initial temperature
  ntt        = 3         ! Langevin thermostat
  gamma_ln   = 1.0       ! Collision frequency in ps-1

  nstlim     = 20        ! Number of timesteps
  dt         = 0.002     ! Time step in ps

  ntpr       = 1         ! Energy is printed every N steps
  ntwx       = 1         ! Trajectory is printed every N steps
  ntwr       = 1         ! Restart file is printed every N steps
  ntxo       = 1         ! ASCII format for final coord, vel, box size
  ioutfm     = 0         ! ASCII format for trajectory (xyz) file
 /
 &qmmm
  qmmask     = ':MOL',   !  QM region mask
  qmcharge   = 0,        !  QM region charge
  qmcut      = 8.0,      !  QM region LJ and Ewald cutoff
  qm_theory = 'EXTERN',!   Call External QM software
 /
 &orc
 method = 'xtb'          !  Method for QM calculation
```

Please execute it as :

```
sander -O -i orca.in -p system.parm7 -c prod.rst7
-r qmmm.rst7 -x qmmm.crd -inf qmmm.info
```

If the calculation ended correctly, we are done for this tutorial. The next time we will apply all this acquired knowledge to a smaller system and actually extract something chemically meaningful out of it.

I really hope you enjoyed it !