Umbrella Sampling - First Case Study

Vito Federico Palmisano

February 17, 2022

1 Introduction

The goal of today is to study the thermal isomerization of azobenzene in water with the help of an enhanced sampling method, umbrella sampling.

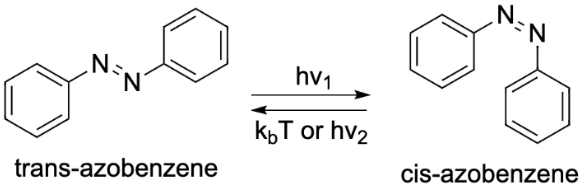

Figure 1: Photo and Thermal isomerization of azobenzene

2 System Setup

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

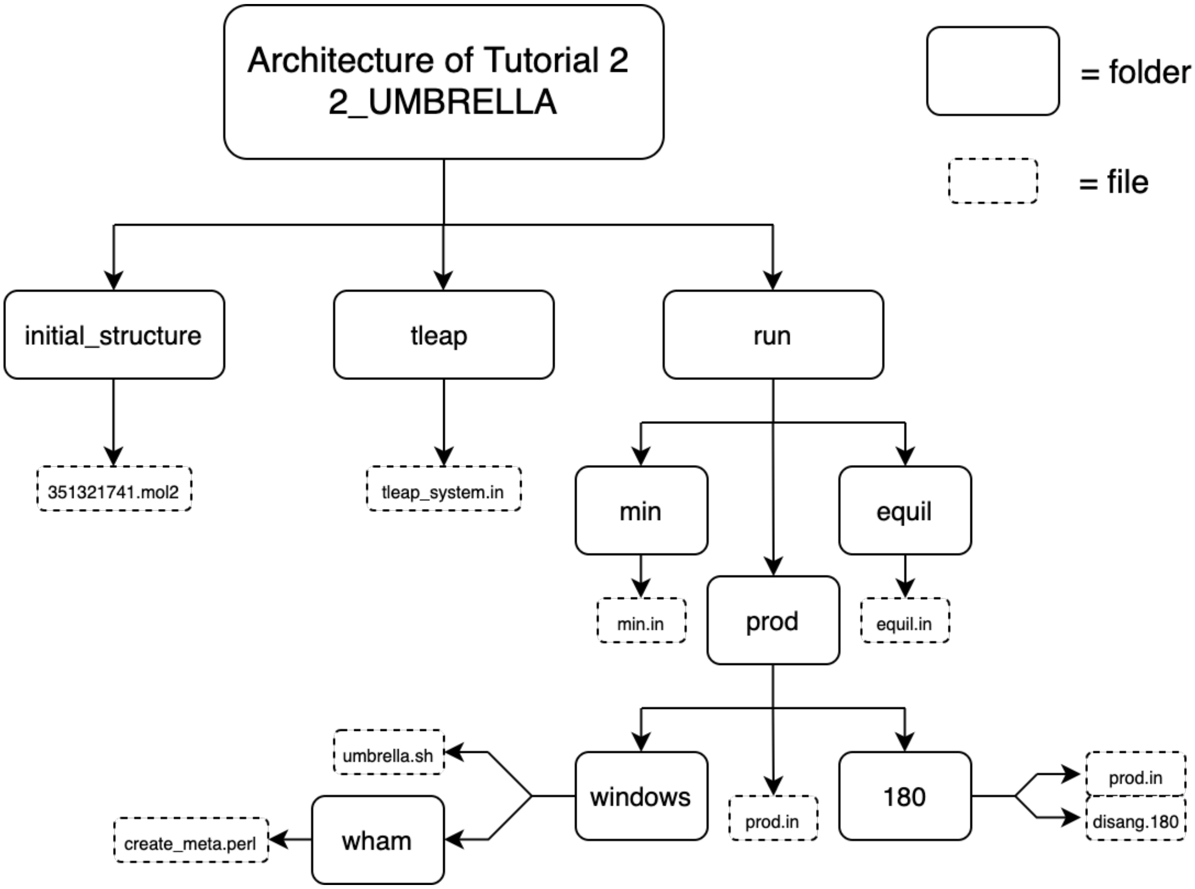

Figure 2: Structure of the folders for the tutorial

Before we start, we should check whether AmberTools and Visual Molecular Dynamics (VMD) are function 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 "scp" the files. Some VMD editions may have some problems with binary files of Amber trajectories, but we will try to solve any problem of any kind in the next few hours.

If those two programs work, then we can start the tutorial, good luck !

2.1 Obtain or Build Azobenzene

The 3D structure of a ligand can either be build with your favorite molecular builder (Avogadro, IQmol) or, most of the times, can directly be found in the ZINC database (*https* : *//zinc.docking.org/substances/home/*) which is a database of all the commercially available chemical compounds, prepared for docking. The file was already downloaded from the database and placed in the correct directory. Please open the mol2 file of azobenzene, which has a long digit number when dowloaded :

```
cd /data/tutorial/yourname/2_UMBRELLA/initial_structures/
vi 351321741.mol2
```

To avoid any nomenclature troubles with Amber, please convert the structure to XYZ file and then convert it again to mol2 including the charges and the gaff parameters :

```
antechamber -i 351321741.mol2 -fi mol2 -o lig.com -fo gcrt
antechamber -i lig.com -fi gcrt -o lig.mol2 -fo mol2 -c bcc -at gaff
```

And let's immediately generate the frcmod file which takes care of the modification that are needed for the molecule given by the force field. Please type:

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

If both mol2 and frcmod files were generated without errors, we can move to the next step and build the solvated system.

2.2 Build the Solvated System

Please enter the tleap folder and copy the ligand files previously generated (lig.mol2, lig.frcmod). In the current directory you will find one input file called tleap-system.in, please open it :

```
source leaprc.gaff2
source leaprc.water.tip3p

MOL = loadmol2 lig.mol2
loadamberparams lig.frcmod

saveoff MOL lig.lib

solvateoct MOL WAT 10

saveamberparm MOL system.parm7 system.rst7
```

```
savepdb MOL system.pdb

quit
```

We have just solvated the azobenzene molecule with an octahedral box of water and generated the topology file (system.parm7) and the coordinate file (system.rst7) which will be the files required to run a molecular dynamics simulation in the next section.

3 Running Simulation

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

```
vmd system.parm7 system.rst7

or

vmd system.pdb
```

and once you are done, please enter the "run folder. You will see three subfolders : "min", "heat","prod" which are the basic ingredients for the simulation. Please enter the "min" folder, and open the min.in file.

```
 Minimization
&cntrl

imin.   = 1       ! Turn on minimization
maxcyc  = 2000.   ! Maximum number of minimization cycles
ncyc    = 1000    ! 1000 steepest-descent steps

ntpr    = 2       ! Print energies every 2 steps
```

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 (2000) of which the last 1000 the minimization algorithm is switched from steepest descent to conjugate gradient.
Please execute it as :

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

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=10000,     ! 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=8000, value1=0.0, value2=300.0 /
&wt type='TEMP0', istep1=8000, istep2=10000, value1=300.0, 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     = 40000    ! Number of timesteps
  dt         = 0.002    ! Time step in ps

  ntpr       = 200      ! Energy is printed every N steps
  ntwx       = 200      ! Trajectory is printed every N steps
  ntwr       = 200      ! 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 !!!!!

4 Restrained Simulation and Umbrella Sampling

Up until now, we've build a system and sampled the probability distribution of the end state, but this method is not optimal if one wish to estimate the difference in free energy between states. The difference in free energy between two states A and B can be identified as the energy difference at the point where the two energy difference distributions δ_A and δ_B intersect. Therefore two or more states have to be connected as part of the same overall system to allow sufficient sampling in the intersection region. Assume we want to calculate the energy difference between A and B, we then want to define a combined Hamiltonian H$_{comb}$ such that the important configurations of this combined Hamiltonian are primarily composed of configurations important to states A and B. The H$_{comb}$ will be some function of H$_A$ and H$_B$, which is dependent on a coupling parameter λ where :

$$\lambda = 0 \qquad H_{comb} = H_A$$

$$\lambda = 1 \qquad H_{comb} = H_B$$

The λ can be called collective variable or reaction coordinate and can be thought as a constrained variable which is set to a certain value, similarly to when we perform a potential energy surface scan in quantum chemistry. Once the reaction coordinate is chosen, which implies that you are aware of both the reactant and the product, we can apply the staging/windowing of the simulation, which means that we will perform many independent simulations at different values of λ. The strategy of adding a biased potential energy term to each windowed simulation was named Umbrella Sampling, probably because in each window, an harmonic (umbrella) potential is applied to avoid the sampling of configurations outside the λ parameter. In our case, for the azobenzene, the λ parameter will be the dihedral of C-N-N-C, which correspond to the central dihedral.
Therefore we will first continue by learning how to run a constrained MD simulation. Please open the "180" folder and open the disang.180 file:

```
Harmonic restraints for 180 deg
 &rst.
   iat=4,5,6,7,                                      ! atom index of C–N–N–C
   r1= 60.0, r2=180.0, r3=180.0, r4=300.0,           ! reaction coordinates
   rk2=200.0, rk3=200.0,                             ! biased potential
 /
```

Here, the iat specifies the atom index, the r1 to r4 define the share of the potential and an harmonic potential rk2 is applied between r1 and r2 while rk3 is applied between r3 and r4. In this case to have give a perfect harmonic potential centered at 180, rk2 must be set equal to rk3 and r2 equal to r3. Now instead, let's 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        = 10         ! 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
  nmropt      = 1
/
&wt
  type='DUMPFREQ', istep1=50,
&end
&wt
  type='END',
&end
DISANG=disang.180
DUMPAVE=dihedral_180.dat
```

where the final part of the file allows sander to read the applied potential and print the value of the dihedral every 50 steps.
Please execute it as :

```
sander -O -i prod_180.in -p system.parm7 -c prod.rst7 -r prod_180.rst7
```

```
-x prod_180.crd -inf prod_180.info
```

Before we start windowing the simulation, let's check whether the dihedral constraint was applied. To do so we could compare the dihedral-180.dat file just generated with the dihedral of the previous simulation without the contraint applied. So let's go back to the "prod" folder and apply the knowledge we previously acquired with cpptraj.

```
cpptraj
parm system.parm7
dihedral @4 @5 @6 @7 range360 out dihedral.dat
go
exit
```

If we now compare this file with the dihedral file in the "180" folder we can see that the potential applied actually blocked that dihedral to apprimately 180 degrees.
Let's now enter the "windows" folder, and open the umbrella.sh file :

```
#!/bin/bash -f

export AMBERHOME=/usr/license/amber/amber20
export PATH=${PATH}:${AMBERHOME}/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$AMBERHOME/lib
export CUDA_VISIBLE_DEVICES=0


delta=$(echo "-3" | bc )
rini=$(echo "180" | bc )
limit=61

n=1

while [ $n -le $limit ]
do


r1=$(echo "$rini + $delta*($n - 1)"  | bc ) #center of window r2=r3
r1min=$(echo "$r1 - 5.0" | bc )  # r1
r1max=$(echo "$r1 + 5.0" | bc )  # r4

[ -e md$r1.in ] && rm md$r1.in
[ -e dihedral$r1.dat ] && rm dihedral$r1.dat
```

```
# Print MD file.
echo umbrellatest >> md$r1.in
echo  \&cntrl >> md$r1.in
echo  imin       = 0         ! No minimization  >> md$r1.in
echo  irest      = 1         ! Simulation is restarted  >> md$r1.in
echo  ntx        = 7         ! Coord, vel and box  >> md$r1.in
echo  ntb        = 2         ! Periodic Boundary Conditions in the NPT
>> md$r1.in
echo  ntp        = 1         ! Isotropic scaling of volumen >> md$r1.in
echo  barostat   = 1         ! Berendsen barostat >> md$r1.in
echo  pres0      = 1.0       ! Pressure in bars >> md$r1.in
echo  taup       = 2.0       ! Relaxation time is ps >> md$r1.in
echo  cut        = 8.0       ! Cutoff for LJ and Ewald  >> md$r1.in
echo  ntc        = 2         ! Bonds H have SHAKE >> md$r1.in
echo  ntf        = 2         ! Bond interactions H  >> md$r1.in
echo  tempi      = 300.0     ! Initial temperature >> md$r1.in
echo  temp0      = 300.0     ! Final temperature >> md$r1.in
echo  ntt        = 3         ! Langevin thermostat >> md$r1.in
echo  gamma_ln   = 1.0       ! Collision frequency in ps-1 >> md$r1.in
echo  nstlim     = 1000      ! Number of time steps >> md$r1.in
echo  dt         = 0.002     ! Time step in ps >> md$r1.in
echo  ntpr       = 10        ! E is printed every N steps >> md$r1.in
echo  ntwx       = 5         ! Traj is printed every N steps >> md$r1.in
echo  ntwr       = 5         ! Rst file is printed N steps >> md$r1.in
echo  ntxo       = 1         ! ASCII format final c,v,box >> md$r1.in
echo  ioutfm     = 0         ! ASCII format for trajectory file >> md$r1.in
echo  nmropt     = 1         ! Restraints, e.g., Umb Samp >> md$r1.in
echo / >> md$r1.in

echo \&wt type = \'DUMPFREQ\', istep1 = 1 / >> md$r1.in
echo \&wt type = \'END\' / >> md$r1.in
echo DISANG = dihedral$r1.dat >> md$r1.in
echo DUMPAVE = dihedral$r1.out >> md$r1.in

echo Harmonic restraints for $r1 dihedral >> dihedral$r1.dat
echo \&rst iat = 4,5,6,7 >> dihedral$r1.dat
echo r1=$r1min, r2=$r1, r3=$r1, r4=$r1max, >> dihedral$r1.dat
echo rk2=200, rk3=200, >> dihedral$r1.dat
echo /  >> dihedral$r1.dat

rnext=$(echo "$r1 + $delta" | bc )

sander -O -i md$r1.in -o md$r1.out -p system.parm7 -c
ini$r1.rst -r md$r1.rst -ref ini$r1.rst -x md$n.mdcrd
```

```
cp md$r1.rst ini$rnext.rst

(( n++ ))
done
```

This is a script that takes the restart file of the constrained 180 degrees simulation and sequentially run a simulation decreasing the dihedral angle by 3 degrees. This means that we will have a total of 60 windows, each corresponding to a simulation where the starting file corresponds to the last frame of the previous simulation. The ini180.rst file correspond to the last frame of the 180 degrees constrained simulation, with a different name to adapt it for the script. So copy the file to the "window" folder and execute the script:

```
cp ../180/prod_180.rst7 ini180.rst
cp ../180/system.parm7 .
./umbrella.sh
```

which have generated a restart file (.rst) , a coordinate file (.mdcrd) and a output file (.out) where the dihedrals are printed for each window.
Let's first visualize the full trajectory, and in order to do so we need to combine the trajectories of each simulation, which of course can be done with cpptraj. Please let's rename the trajectories from 1 to 9 to make them recognizable for the subsequent action of cpptraj. If you don't rename them, then cpptraj will not give the order from 1 to 60.

```
for i in {1..9}
do
mv md$i.mdcrd md0$i.mdcrd
done
```

and now let's combine them with cpptraj:

```
cpptraj
parm system.parm7
for i in md*.mdcrd
trajin $i 1 last 60
autoimage :MOL
done
trajout final.crd
go
exit
```

and please open the it with vmd:

```
vmd system.parm7 final.crd
```

And we should also check whether there is overlap between the different windows. Please plot all the dihedral files :

```
xmgrace dihedral{1..180}.out
```

which gives us an indication of the dihedral visited in each simulation. Probably it will not be enough to give a good estimate of the relative free energy, but some overlap is observed. The final stage of the umbrella sampling is to use the Weighted Histogram Analysis Method (WHAM) code to construct the potential of mean force (PMF) from the dihedral data collected. The exact explanation of this method goes beyond the scope of this tutorial, but we will still perform the analysis in order to obtain our first PMF plot. Please copy all the dihedral files into the "wham" folder, enter the folder and open the create-meta.perl file :

```
#!/usr/bin/perl -w
use Cwd;
$wd=cwd;

print "Preparing meta file\n";

$name="meta.dat";
$incr=-3;
$force=0.12184;

&prepare_input();

exit(0);

sub prepare_input() {

    $dihed=180;
    while ($dihed >= 0) {
      print "Processing dihedral: $dihed\n";
      &write_meta();
      $dihed += $incr;
    }
}

sub write_meta {
    open METAFILE,'>>', "$name";
    print METAFILE <<EOF;
dihedral$dihed.out $dihed.0 $force
EOF
    close MDINFILE;
}
```

which was already adjusted for our system. Please execute it as :

```
./create_meta.perl
```

This script have generated a file called meta.dat which is an input file for the wham program telling the filename of each dihedral file, the minimum of the harmonic potential and the force constant used. Please remember that Amber uses $k(x - x_0)^2$ with k in kcal/mol/rad^2 whereas the WHAM program uses 0.5k(x-x$_0$)^2 with k in kcal/mol/deg^2 therefore we need to multiply our force constant by $2(\pi/180)^2$. In order to execute the WHAM program please type :

```
wham P 0 180 60 0.01 300 meta.dat result.dat
```

where :

```
P = reaction coordinate is periodic
0 180 60 001  = generates a PMF from 0 to 180 using 60 bins
(every 3 degrees)
0.01 = tolerance for reconstructing the PMF
300 = 300K (temperature)
0 = do not pad data points
meta.dat = meta file
result.dat = output file
```

The program makes the the histograms, modifies them depending on the potential used, coverts each histogram to a free energy curve, iteratively aligns all curves using a least squares fit and writes the complete PMF curve to the output file. To plot the PMF, please extract the first two columns, execute it as:

```
cat result.dat | awk '{print$1,$2}' > pmf.dat
```

If everything proceeded correctly, this is the PMF plot that you should obtain. The trans isomer corresponds to the minimum at 180°, with a transition state and another minimum approaching around 0°.

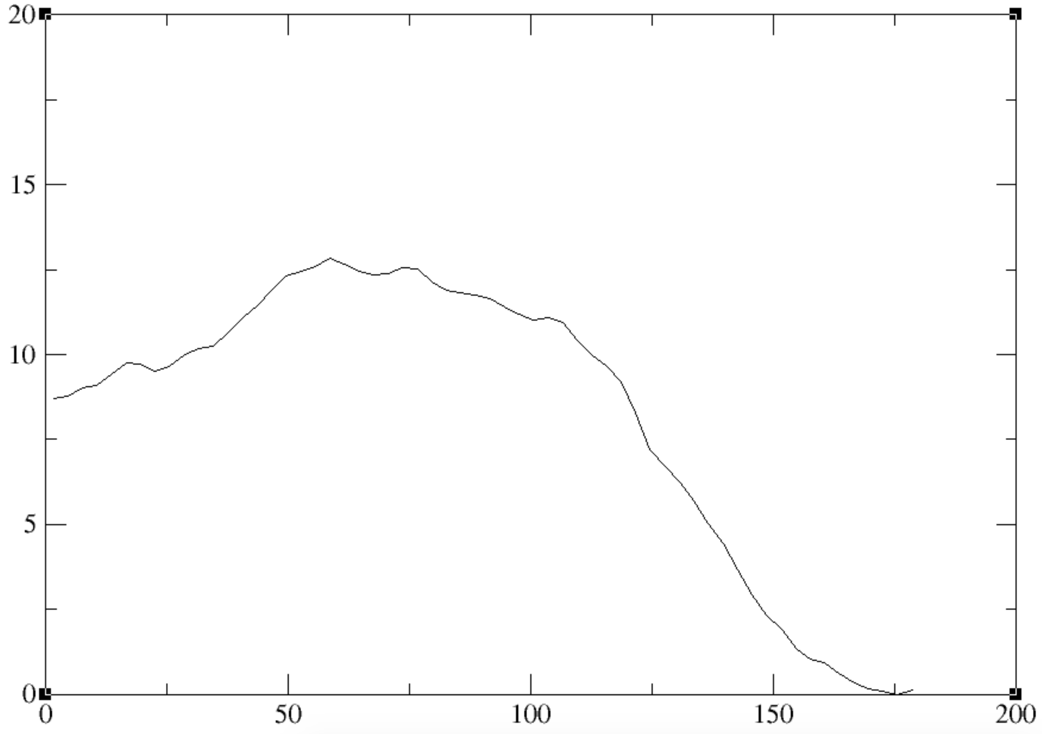

Figure 3: Potential of mean force along the C-N-N-C reaction coordinate

This is the end of the tutorial, I really hope you enjoyed it and see you next time !