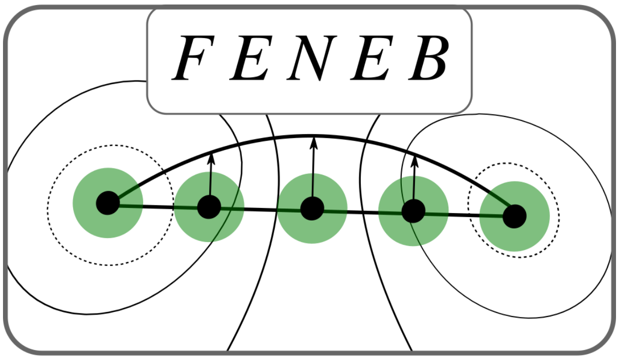


Minimum free energy pathways with nudged elastic bands

FENEB TUTORIAL

Jonathan A. Semelak
jsemelak@qi.fcen.uba.ar

1. What is FENEB

A tool for performing nudged elastic bands simulations on the free energy surface using Amber .nc files.

2. Requirements

This code requires NETCDF Libraries.

3. Compilation

Simply execute “make”:

```
>>$  make
```

Where the NETCDF_DIR is assumed to be /usr, and can be replaced with :

```
>>$  make NETCDF_DIR=/your/location
```

If everything goes well, you will see the following message:

```
FENEB installation is complete! 🎉
Don't forget to export the bin path: export PATH=$PATH:/home/usuario/Programas/feneb/bin
```

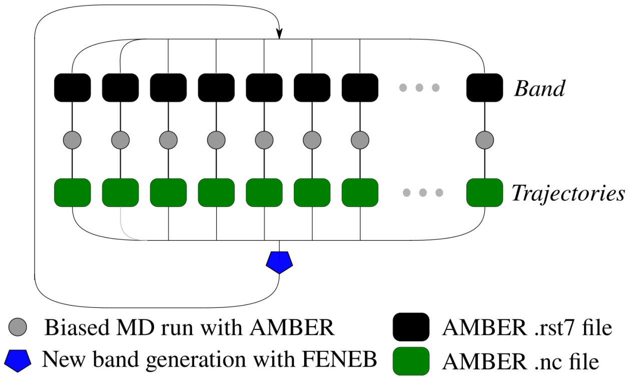
After installation, export the bin path (as indicated in the message):

```
>>$  export PATH=$PATH:/your/path/to/feneb/bin
```

If you happen to use this code within a script, you should at this line (or you can just add it to your .bashrc file).

4. Usage

Run molecular dynamics (MD) with AMBER, feed this code with .rst7 and .nc files, and you will get new .rst7 files to continue the optimization. Additionally, every time this code is executed, an output file will be generated with thermodynamic information.



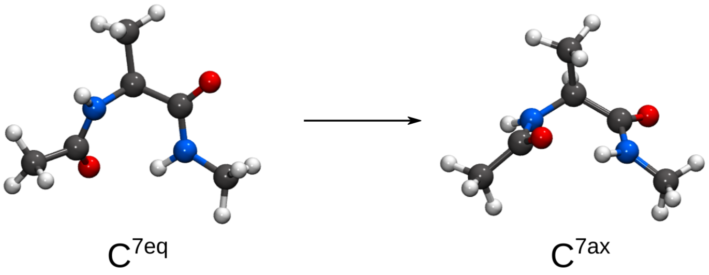
Knowledge of the AMBER package is assumed, and we recommend checking the following two references for theory and notation:

- Semelak, et. al. (2023). Minimum Free Energy Pathways of Reactive Processes with Nudged Elastic Bands. Journal of Chemical Theory and Computation.
- Bohner, et. al. (2014). Nudged-elastic band used to find reaction coordinates based on the free energy. The Journal of Chemical Physics, 140(7), 074109.

Using a bash script is recommended for automatization. For this reason, a feneb_wizard is provided, which helps you to run in parallel the required MD simulations, executing feneb, and iterating until the desired convergence criteria is reached.

5. Tutorial

Here we will show how to optimize a minimum free energy pathway (MFEP) for the alanine dipeptide conformational change from the local minima C^{7eq} to C^{7ax}, using the FENEB method:



We will use minimal inputs, following the simplest protocol to optimize a MFEP. We will use the **feopt_wizard**, **feneb_wizard** and **bandbuilder** tools.

5.1. Defining the free energy surface

The first step is to define free energy surface (FES) by selecting which atoms will integrate the *reaction coordinate space* (or RCS). The thumb rule for this is to choose all atoms involved in large geometrical modifications during the process. In the case of chemical reactions, atoms participating in the forming and breaking of bonds are recommended to be included on the RCS. In the case we address here, we will choose those atoms forming the Ramachandran angles of the alanine dipeptide.

According to the files from the **tutorial** folder, they correspond to indexes 5,7,9,15 and 17.

5.2 Optimizing reactants and products

This is not a chemical reaction, but we will refer to the band extreme as reactants and products for simplicity. Inside the **1-REACTANTS** folder, you will find a bash script **run_feopt.sh**, and an **inputs** folder find the following files:

ALAD.prmtop	This is the topology file
prod.mdin	This is an AMBER input for NVT biased MD
feneb.in	This is the input for this code

Additionally, the folder **STEP-0** contains the initial structure for the reactants optimization, and the folder **STEP-1** contains the outputs (we will get to that point later).

Lets inspect the content of the prod.mdin and feneb.in files:

```
>>$  cat run_feopt.sh
```

```
#!/bin/bash

SANDERPATH=sander #Should be replaced by the corresponding path to sander
FENEBPATH=feneb      #Should be replaced by the corresponding path to feneb
STARTSTEP=1            #We are starting a new optimization
MAXSTEPS=5             #Maximum optimization steps to be performed
DELETENC=T             #Delete .nc files after analyzing them, to save space
MDIN=prod.mdin         #Amber input file for each MD

feopt_wizard -s $SANDERPATH \
          -f $FENEBPATH \
          -x $STARTSTEP \
          -d $MAXSTEPS \
          -m $MDIN \
          -g $DELETENC
```

```
>>$  cat prod.min
```

```
 &cntrl
  imin=0,
  ntx=1,
  ntwr=100,
  ntpr=10,
  ntwx=10,
  ioutfm=1,
  ntxo=1,
  nstlim=10000,
  dt=0.001,
  ntt=3,
  tempi=300.0,
  temp0=300.0,
  ig=2022, !This is the seed. It should be set to random (-1), but here is set  to 2022 for reproducibility
  gamma_ln=5.0,
  ntp=0,
  ntb=0,
  cut=999.,
  ntr=1, !Turn on restraints on cartesian coordinates
  restraint_wt=50.0,  !Specify restrain force constant
  restraintmask='@5,7,9,15,17',  !Specify which atoms will be affected by the restraint
 &end
```

The prod.mdin file corresponds to a biased MD run in the NVT ensemble, with a restraint in the cartesian coordinates of the atoms specified in restraintmask. The restraintmask selection must be the same as the desired RCS.

```
>>$  cat feneb.in
```

```
prefix ALAD           ! prefix for files
per F                 ! This system is not periodic
nrep 1                ! Only 1 replica/image -> This a free energy optimization (not a band!)
nrestr 5              ! 5 atoms integrate the RCS
mask 5 7 9 15 17      ! Indexes of atoms in the RCS
kref 100              ! Restraint force constant. Must be twice that of the .mdin file (because of AMBER things)
skip 500              ! Number of initial frames to be omitted
ftol 2.25             ! Convergence criterion (kcal/mol A)
stopifconverged T  ! If convergence is reached, the simulation will stop
```

Where **prefix** corresponds to the prefix for the initial coordinates, the reference coordinates, and the output (optimized) coordinates. In dis case prefix is set to “ALAD”, so files will be named by the FENEB code as:

ALAD_f_numberofimage.rst7	Last frame of MD from current optimization step.
ALAD_fprev_numberofimage.rst7	Last frame of previous MD; only for 2 step onwards.
ALAD_r_numberofimage.rst7	Restraint reference coordinates for current optimization step.
ALAD_o_numberofimage.rst7	Restraint reference coordinates for next optimization step.
ALAD_o_av_numberofimage.rst7	Average coordinates for current optimization step.

The **nrep** variable corresponds to the number of images the FENEB band contains.

IMPORTANT: When **nrep** is set to 1, the code automatically understands that we are only performing a free energy optimization, and not a FENEB optimization.

The **nrest** and **mask** variables indicate how many and which atoms form the RCS. The kref variable is the force constant used in the biased MD (and must be twice the one from the prod.mdin file, because of how AMBER defines the restraint potential). The units are kcal/(mol Å^2)

The **skip** variable indicates how many steps from the MD trajectory will be omitted for the analysis (the first skip **steps** are omitted).

The **ftol** variable indicates the convergence criterion. The optimization is assumed to be converged when the maximum (root mean square) atomic free energy gradient is lower than **ftol**.

Now we are ready to execute the **feopt_wizard** through the **run_feopt.sh** script, that basically will run MD, execute the FENEB code, run MD, execute the FENEB code, until convergence.

Note the script requires the user SANDER and FENEB paths.

Then, we execute it:

```
>>$  ./run_feopt.sh
```

In this case the code we will only execute a single step (notice the variable MAXSTEPS is set to 1 in the feopt_script.sh file), because that is enough to optimize the structure. The feopt_wizard will print the following in the screen:

```
------------------------------------------------------------------------------------------------------------------
                                        FEOPT WIZARD
------------------------------------------------------------------------------------------------------------------

Welcome to Feneb Wizard, a tool designed to automate the running of a Free Energy Optimizaion.

USAGE:
feopt_wizard [OPTIONS]

OPTIONS:
-s  Specify the path to sander [default: sander]
-f  Specify the path to feneb [default: feneb]
-m  Specify the sander input [default: prod.mdin]
-x  Specify the starting optimization step [default: 1]
-d  Specify the maximum optimization steps to be performed [default: 10]
-g  Delete .nc files after processing (enter T for True) [default: T]
-a  Append maxgrad evolution (enter T for True) [default: T]
------------------------------------------------------------------------------------------------------------------
NOTES:
1. The following files must be provided in the 'inputs' folder:
    - feneb.in file and any other necessary feneb.* file (like feneb.extrema1 and feneb.extrema2)
    - A sander input for the MD simulations (prod.mdin by default, as specified by -m)
    - The topology file named PREFIX.prmtop, where PREFIX is read from the feneb.in file (prefix)

2. The 'STEP-(K-1)' folder:
   In each optimization step (K), the simulation is run in a 'STEP-K' folder, with the restraint coordinates
   taken from the previous one. This folder is automatically generated by this script, but for the first step (-x 1),
   a 'STEP-0' folder must be manually generated.

------------------------------------------------------------------------------------------------------------------
                                  STARTING THE FEOPT SIMULATION
```

```
-----------------------------------------------------------------------------------------------------------------

The restraint reference will be updated after each MD simulation.

---------------
 STEP: 1
---------------
Running MD
Running Free Energy optimization
Convergence achieved: Stopping simulation...
-----------------------------------------------------------------------------------------------------------------
                                        HAPPY LANDING
                                   FEOPT OPTIMIZATION CONVERGED
-----------------------------------------------------------------------------------------------------------------
```

Several files will be generated in a folder called “STEP-1”, being the most important ones the **ALAD_o_av_1.rst7** and **feneb.out** files.

ALAD_o_av_1.rst7	Mean coordinates obtained from the MD (only for RCS, the rest correspond to the initial coordinates). If the system is converged, here is where you want to look at.
feneb.out	This the output of FENEB.
feneb.gradients	Mean coordinates obtained from the MD (only for RCS) and free energy gradients.

The most important thing to look at the feneb.out file is the “Max force:” line, where the maximum atomic free energy gradient is printed. In this case, it is equal to 0.67 kcal/(mol Å), that satisfies the 2.25 kcal/(mol Å) convergence criterion. The wizard grep this line to the **maxgradevolution.dat** file.

If more optimization steps were needed, the script would generate a **ALAD_o_1.rst7** file**,** and copy this file along with the **ALAD_f_1.rst7** file to a **STEP-2** folder, and so on. You can play with this if you set the stopifconverged variable to False.

Since the initial structure actually possessed coordinates from the RCS that corresponded to a local minimum in our FES, we can take the **ALAD_o_av_1.rst7** file, change its name to **ALAD_r.rst7**, and continue our life.

```
>>$   mv ALAD_o_av_1.rst7 ALAD_r.rst7
```

Additionally, in this case we will avoid recalculating the free energy gradients for the extremes of the band in each iteration. So, we will save them with a specific name, as required by the feneb code:

```
>>$   mv feneb.gradients feneb.reactants
```

Now we repeat the same with the products, working in the **2-PRODUCTS** folder. We will change the name of the new **ALAD_o_av_1.rst7** file, but this time to **ALAD_p.rst7**:

```
>>$  mv ALAD_o_av_1.rst7 ALAD_r.rst7
 >>$  mv feneb.gradients feneb.products
```

NOTE: FENEB always numbers as “1” the image of a free energy optimization. It does not know whether it corresponds to reactants or products.

5.3 Generating the initial band

The FENEB code includes a tool called **bandbuilder** which takes reactant and products coordinates, and generates the initial band. Reactants and products coordinates should be aligned (with VMD or CPPTRAJ, for instance), prior to executing bandbuilder (and, if you are saving the gradients, prior to their optimization). The folder **3-INITIALBAND** contains:

ALAD.prmtop	This is the topology file
ALAD_r.rst7	Reactants optimized coordinates
ALAD_p.rst7	Products optimized coordinates
bandbuilder.in	Input for BANDBUILDER
run_bandbuilder.sh	This is a bash script to run BANDBUILDER and generate a trajectory file for easy visualization
ALAD_BAND_0.nc	Trajectory file with the images generated.

The **run_bandbuilder.sh** script calls the **bandbuilder** tool to generate a set of structures (images), and move them to a new folder named STEP-0. An **ALAD_BAND_0.nc** file with all structures will be also generated, using CPPTRAJ.

```
>>$  cat run_feopt.sh
```

```
#!/bin/bash

BANDBUILDERPATH=bandbuilder   #Should be replaced by the corresponding path to bandbuilder
CPPTRAJ=cpptraj               #Should be replaced by the corresponding path to cpptraj
IMAGES=15                     #Number of images to be generated (according to bandbuilder.in)
TOPOLOGY=ALAD.prmtop          #Topology file
NAME=ALAD                     #Prefix for all coordinates files.

#Run bandbuilder

$BANDBUILDER

#Generate .nc file for further visualization
for ((i=1; i<=IMAGES; i++));
```

```
do
echo "trajin ${NAME}_r_$i.rst7" >> input.cpptraj
done
echo "trajout ${NAME}_BAND_0.nc netcdf" >> input.cpptraj
$CPPTRAJ $TOPOLOGY input.cpptraj
rm input.cpptraj
#Move images to the corresponding file
mkdir -p STEP-0
mv *_r_*.rst7 STEP-0/.
```

In this case the script has already been executed, and the folder **STEP-0** contains the coordinates of the images generated: **ALAD_r_1.rst7**, **ALAD_r_2.rst7**, **…**, **ALAD_r_15.rst7**.

If you want to do it yourself, just type:

```
>>$  ./run_bandbuilder.sh
```

The **bandbuilder.in** file syntax is similar to that of the feneb.in method:

```
>>$   cat bandbuilder.in
```

```
prefix ALAD_r          ! Coordinate files will be called ALAD_r_1.rst7, ALAD_r_2.rst7, …, ALAD_r_15.rst7.
rcfile ALAD_r.rst7     ! Reactant coordinates
pcfile ALAD_p.rst7     ! Products coordinates
nrep 15                ! Number of replicas/images
nrestr 5               ! 5 atoms integrate the RCS
mask 5 7 9 15 17       ! Indexes of atoms in the RCS
per F                  ! The system is not periodic
```

Here the only new variables are **rcfile** and **pcfile**, that correspond to the reactants and coordinates files. With this input file, **bandbuilder** will generate coordinate files for the band interpolating reactants and products RCS coordinates, and the rest of the coordinates will be those of the reactants files. The band will contain 15 images, according to the **nrep** variable.

Another option would be to use an intermediate structure (usually called a “transition state candidate”) to split the interpolation in two. In that case (not explored in this tutorial), the bandbuilde.in file would contain the following two extra lines:

```
usets T                ! Use a transition state candidate
tsfile ALAD_r.rst7     ! Transition state candidate coordinates
```

5.4 FENEB Optimization

Now we are ready to optimize a MFEP for the process C^{7eq} → C^{7ax}. We have an initial band consisting of 2 previously optimized extremes, and 13 structures ready to be optimized until they lay in a MFEP**.**

In the **4-REACTION** folder, we have both the inputs and the outputs of the FENEB optimization. Inputs are the **STEP-0** folder obtained in the section 5.3, which contains the initial band, a bash script **run_feneb.sh**, and an **inputs** folder with the following files:

ALAD.prmtop	This is the topology file
prod.mdin	This is an AMBER input for NVT biased MD
feneb.in	This is the input for this code
feneb.reactants	Free energy gradients for reactants
feneb.products	Free energy gradients for products

The prod.mdin file is identical to that of section 2, but with a force constant ten times higher. The **run_feneb.sh** executes the **feneb_wizard**. This wizard coordinates the calls to AMBER and FENEB, and creates folders to save the outputs of each optimization step.

```
>>$  cat run_feneb.sh
```

```
#!/bin/bash

SANDERPATH=sander #Should be replaced by the corresponding path to sander
FENEBPATH=feneb      #Should be replaced by the corresponding path to feneb
STARTSTEP=1            #We are starting a new optimization
MAXSTEPS=10            #Maximum optimization steps to be performed
DELETENC=T             #Delete .nc files after analyzing them, to save space
MDIN=prod.mdin         #Amber input file for each MD
NCPU=2                 #Number of cpu-threads available

feneb_wizard -s $SANDERPATH \
        -f $FENEBPATH \
        -x $STARTSTEP \
        -d $MAXSTEPS \
        -c $NCPU \
        -m $MDIN
```

The feneb.in file is similar to that of the free energy optimizations:

```
>>$  cat  feneb.in
```

```
prefix ALAD          ! prefix for files
per F                ! This system is not periodic
nrep 15               ! The band contains 15 replicas/images
nrestr 5             ! 5 atoms integrate the RCS
```

```
mask 5 7 9 15 17    ! Indexes of atoms in the RCS
kref 1000           ! Restraint force constant. Must be twice that of the .mdin file (because of AMBER things)
kref 500            ! Spring force constant
skip 500            ! Number of initial frames to be omitted
ftol 2.25           ! Convergence criterion (kcal/mol A)
stopifconverged T   ! If convergence is reached, the simulation will stop
```

Where the only new variable is **kspring**, which specifies the value for this constant. Additionally, we note again that the force constant is ten times higher than in the free energy optimizations of reactants and products. By default, the “uncoupled” version of FENEB is performed.

In this case, we perform 10 optimization steps. It is not enough to reach the established convergence criterion, but the results are accurate enough. We are ready to execute the run_feneb.sh script, which calls the feneb_wizard to automatize the run. Note that it is assumed that 2 cpu-threads are available (for larger parallelization, use a larger value for the -c option).

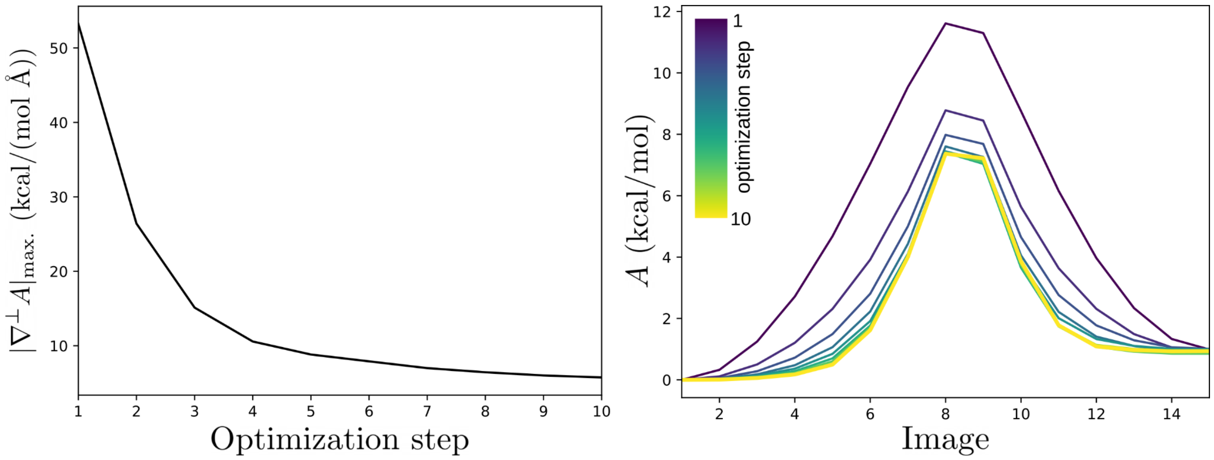
```
>>$  ./run_feneb.sh
```

The feneb_wizard will print the following in the screen:

```
---------------------------------------------------------------------------------------------------------------
                                        FENEB WIZARD
---------------------------------------------------------------------------------------------------------------

Welcome to Feneb Wizard, a tool designed to automate the running of Feneb calculations.

USAGE:
feneb_wizard [OPTIONS]

OPTIONS:
-s  Specify the path to sander [default: sander]
-f  Specify the path to feneb [default: feneb]
-m  Specify the sander input [default: prod.mdin]
-c  Specify the number of available CPU threads [default: 1]
-x  Specify the starting optimization step [default: 1]
-d  Specify the maximum optimization steps to be performed [default: 10]
-g  Delete .nc files after processing (enter T for True) [default: T]
-a  Append band and maxgrad evolution (enter T for True) [default: T]
-u  Set sleep duration of the script [default: 10]
-v  Specify the units for sleep duration (s for seconds, m for minutes, d for days) [default: s]
-r  Resume a previous simulation (enter T for True) [default: F]
---------------------------------------------------------------------------------------------------------------
NOTES:
1. The following files must be provided in the 'inputs' folder:
   - feneb.in file and any other necessary feneb.* file (like feneb.reactants and feneb.products)
   - A sander input for the MD simulations (prod.mdin by default, as specified by -m)
   - The topology file named PREFIX.prmtop, where PREFIX is read from the feneb.in file (prefix)

2. The 'STEP-(K-1)' folder:
  In each optimization step (K), the simulation is run in a 'STEP-K' folder, with the restraint coordinates
  taken from the previous one. This folder is automatically generated by this script, but for the first step (-x 1),
  a 'STEP-0' folder must be manually generated.

3. This script checks if all the MD runs have finished before executing the feneb code.
  If they haven't, the script sleeps for a predefined time (as set by -u and -v) and checks again.
  Typically, only a few minutes (or even seconds) are necessary.

-------------------------------------------------------------------------------------------------------------------
                                    STARTING THE FENEB SIMULATION
-------------------------------------------------------------------------------------------------------------------

The simulation will be parallelized according to the required number of
MD simulations to be performed and the number of cpu threads available.

The 13  MD simulations will be run in 6 packages of 2 each
and an extra one with 1 simulations
---------------
 STEP: 1
---------------
Running MD
Running package 1 (range from 2 to 3)
all Images of package 1 have finished
Running package 2 (range from 4 to 5)
all Images of package 2 have finished
Running package 3 (range from 6 to 7)
all Images of package 3 have finished
Running package 4 (range from 8 to 9)
all Images of package 4 have finished
Running package 5 (range from 10 to 11)
image number 10 has not finished
taking a 10 s  long nap...
all Images of package 5 have finished
Running package 6 (range from 12 to 13)
all Images of package 6 have finished
all remaining Images have finished
Running feneb optimization
---------------
 STEP: 2
---------------

*** more lines ***
-------------------------------------------------------------------------------------------------------------------
                                      HAPPY LANDING
                                MAXIMUM NUMBER OF STEPS REACHED
-------------------------------------------------------------------------------------------------------------------
```

After a few seconds, 10 folders will be generated, each containing mean coordinates (“**_o_av_**” files), optimized coordinates (“**_o_**” files) and others. The feneb.out file contains information of the FENEB optimization step. The profile.dat file contains the free energy evolution along the corresponding band.

The **feneb_wizard** will also grep useful information to the files **bandevolution.dat** and **maxgradevolution.dat.** The content of these files allows us to monitor the FENEB optimization process, and can be easily accessed with xmgrace, or a python script:



The obtained barrier is of 7.3 kcal/mol, which shows an excellent agreement with the 7.9 kcal/mol value that is obtained with longer simulation times and more optimization steps.