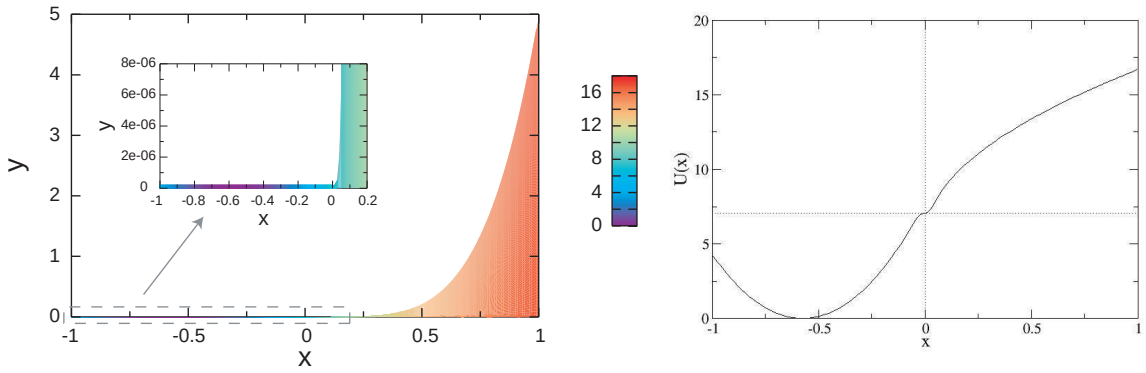
CHEM5302 Fall 2015: Rate Coefficients and Potential of Mean Force for a 2D Potential Model of Protein Folding

Ronald Levy

November 17, 2016

1 Introduction

We are going to obtain rate coefficients in two ways. The goal of this part of the lab is to obtain an Arrhenius plot (Fig. 3.1) of the folding and unfolding rate coefficients on the 2D potential (Fig. 1.1) from first passage time.



(a) (b)

Figure 1.1: (1.1a) A schematic representation of the two-dimensional potential function used here. The colored area corresponds to the accessible region of the (x, y) plane, with the colors representing the magnitude of the potential energy at that (x, y) point (scale bar in kcal/mol). The potential energy is infinite in the non-colored region and for $y < 0$, $x < -1$, and $x > 1$. The inset is an enlarged view of the folded macrostate and transition region. (1.1b) Potential Energy along x coordinate.

Note: see Lecture 13. Conformational Sampling MC and MD (Morkov chain Monte Carlo, trial moves and Metropolis algorithm).

Linux command line and scripting

During this lab, you need to run simulations several times, and those simulations may take minutes to hours. In order to avoid waiting around in order to submit new simulations, you can write a script which includes a set of sequential commands in Linux. There are many ways

of writing such scripts; below we will show one example. Further, because the scripts may take a while to complete, we will be running them on one of our group's server Cb2rr. Each of you has been granted access using your TU AccessNetID and normal password.

First, log on to Cb2rr:

```
ssh -Y username@cb2rr.cst.temple.edu
```

Once you enter your password, your terminal will redirect you to your home directory on Cb2rr. It's important to understand that the home directory on the computer you're using now is **not the same** as the home directory on Cb2rr.

Next, download the source files for the lab.

```
mkdir klab
cd klab
curl -O https://ronlevygroup.cst.temple.edu/courses/2015_fall/\
          chem5302/kinetics_lab.tgz
tar -xzvf kinetics_lab.tgz
```

With whatever text editor you prefer, open the file `simulations/exe.sc`. If you don't have a Linux preferred editor, you can try `gedit`.

```
cd simulations
gedit exe.sc
```

Remember that you need to save your changes before closing `gedit`.

You can also try `vi` and/or `vim` to edit text files. `vi` has two basic operation modes: "Normal Mode" which is used to perform macroscopic operations to your document like delete lines, copy/paste, and more; and "Insert Mode" which you use to insert text. "Normal Mode" is the default mode. You enter "Insert Mode" by pressing `i`, and return to "Normal Mode" by pressing `ESC`. The following useful "Normal Mode" commands may be useful and can be entered by first pressing `ESC`

- `:wq` - save and quit
- `:q!` - quit without saving
- `/<search term>` - search for `<search term>` within the document

In the `exe.sc` file, you can enter all the commands you would normally enter into the terminal, and the script will process the second command when the first command finishes. For example, a useful script for this lab might include:

```
#!/bin/bash

tempfs=(296 338 474 592 789)
for i in {0..4}
do
  echo \ Running simulation $((i+1))"
  ../exe_run_MC $((i+1)) 0.05 100 -0.1 0.1 > ${temps[$i]}.dat
  mkdir results_${temps[$i]}
  mv ${temps[$1]}.dat results_${temps[$i]}/
```

```
    mv folding_FPT.dat results_${temps[$i]}/
    mv unfolding_FPT.dat results_${temps[$i]}/
  done
  echo \ Simulations complete!"
```

The first line is necessary to tell the script to act like a terminal. The script will iterate the variable `i` through the values 1 to 5 and create directories and run simulations specific to each value of `i`.

For the future, when you need to execute this script, save and close (using `ESC + :wq` if using `vi`), and enter the following into the terminal one line at a time. Come back to this portion when instructed to later on in the manual.

```
cd ~/klab/simulations
chmod 755 exe.sc
mkdir kinetics
cd kinetics
../exe.sc &> exe.log &
disown
```

`chmod` allows the file you edited to be run as an executable; `./exe.sc` tells you script to run, `&> exe.log` tells your script to write output and errors to `exe.log`, `&` at the end and `disown` tell your job to run in the background, even if you close the terminal.

Using scripts frees you from being bound to the computer. There are some example scripts with the names `exe*.sc`, and you are free to write your own scripts as well or to use any other method you feel most comfortable using to complete to following lab.

2 Launch Simulations and Analysis

To begin, ensure you are in the working directory that you created in the Introduction:

```
pwd
```

The line should return a directory path that ends in `klab/simulations`. If not, navigate to the `simulations` directory.

Test Run

Executable `exe_run_PMF` is used to run a long MC simulation to generate x coordinates sampled at each step. The executable `exe_run_PMF` takes 5 command line arguments which control the program's function:

`tempidx` Temperature index. This index runs from 1 to 5 which correspond to the tempatures 296K, 338K, 474K, 592K, 789K, respectively.

`MCwidth` Width of the MC proposal distribution along the x-axis. We recommend you keep this parameter set at 0.05, though you can play around with it if you have the time and are curious.

`Nsteps` Number of proposed moves in the MC simulations. To obtain enough statistics, we suggest you use no less than 10^9 steps.

`xcoord` x coordinate of the starting point in the MC trajectory.

`ycoord` y coordinate of the starting point in the MC trajectory. Suitable `xcoord` and `ycoord` fall anywhere in the area of potential energy.

The simulation will output a file `x.out` which contains one tenth of the x coordinates sampled by the simulation; without subsampling, the output file will be too bulky.

Run a short simulation at temperature 3 (474K):

```
cd ~/klab/simulations
mkdir test
cd test
../exe_run_PMF 3 0.05 100000000 0 1e-7
```

and plot the trajectory of the MC particle using the x coordinates.

We recommend using `gnuplot` to generate pictures for your lab report. `gnuplot` is a command-line program that can make plots of functions and data. You can invoke `gnuplot` by issuing the command:

```
gnuplot
```

After the welcome message, you can see the shell prompt has been replaced by a gnuplot prompt. Next plot the trajectory recorded in the file `x.out`

```
plot 'x.out' every 1000 with lines title 'Traj'
```

This command line tells gnuplot to plot one data element of every 1000 data recorded in file "x.out", and connect them by line segments. Without mistakes, you should see a similar picture as Fig.2.1, which is a typical trajectory of a two state system. Then we can make this picture better for a lab report by adding labels to the x and y axes.

```
set xlabel 'time (10000 steps)'
set ylabel 'x'
replot
```

At last, the picture can be saved to an "eps" file by

```
call '../gnuplot_4eps' 'traj.eps'
```

Note the "gnuplot_4eps" is our pre-made format file for gnuplot. Now enter an "exit" or "quit" command or type "Control-D" to end `gnuplot` and come back to the shell prompt. If you prefer using "png" file for you lab report instead of "eps" file, run the following command

```
convert -density 500 traj.eps traj.png
```

where number 500 is the option to control the resolution of the "png" file. You can check your "eps" or "png" (or any) file by command `xdg-open`

```
xdg-open traj.eps
xdg-open traj.png
```

or

```
evince traj.eps
eog traj.eps
```

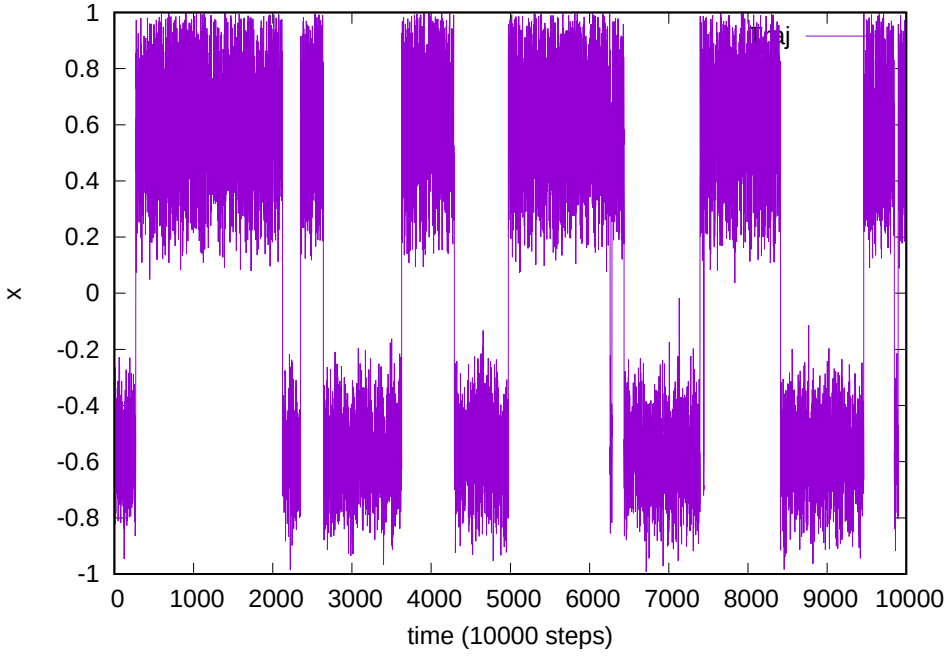


Figure 2.1: Trajectory of the MC particle in the 2D potential.

If you prefer using Excel to plot pictures, the data files need to be copied to your local desktop because Cb2rr server is running Linux without Excel. The following command line copies (synchronizes) the "klab" directory to the home directory of your local desktop

```
rsync -urlv username@cb2rr.cst.temple.edu:~/klab ~/
```

Potential of Mean Force (PMF) for the 2D potential

The goal of this part of the lab is to obtain rate coefficients from transition state theory. You will observe how the PMF varies with temperature and examine the activation free energy for both folding and unfolding reactions.

Run the simulation at temperature 3 (474K):

```
cd ~/klab/simulations
mkdir pmf
cd pmf
../exe_run_PMF 3 0.05 1000000000 0 1e-7
```

and **make a histogram** of the x coordinates. Because we will run this script for 5 temperatures, it is better to change the name of the output file now to avoid being overwritten later.

```
mv x.out T3x.out
```

The histogram can be constructed by running

```
../Histogram.py -f T3x.out -i 1 -n -1.0 -x 1.0 -b 100 > T3.hist
```

The program "Histogram.py" makes a histogram using the first column of data file "T3x.out". There are 100 bins in the histogram, the left edge of the first bin is -1.0 and the coordinate right edge of the last bin is 1.0. Run "Histogram.py -h" to see all the options and how to use this script. The histogram made by "Histogram.py" has been normalized. In other words

$$\sum_{i=1}^{N} w_i h_i = 1\,, \tag{1}$$

where w_i and h_i are the width and height of the i^{th} bin respectively. Unlike the "T3x.out" file, the size of "T3.hist" is small, you can open "T3.hist" file using `gedit`. As you can see, there are two columns in the file "T3.hist": the first column is the coordinate of each bin x_i and the second column is the height of each bin h_i. The width of each bin is $w_i = x_{i+1} - x_i = 0.02$.

Given this histogram, the PMF, denoted by W_i, is given by

$$W_i = -k_B T \ln P_i = -\frac{1}{\beta} \ln P_i \tag{2}$$

where $P_i = w_i h_i$ is the fractional population in each bin of the histogram when each bin is normalized by the total number of MC steps. This calculation can be easily carried out by the following one-line Perl script (type it in one line) or Excel

```
perl -wnal -e ' printf("%12g %12g %12g\n",

  $F[0], $F[1], (-log($F[1]*0.02))) ' T3.hist > tmp.hist
```

This Perl script copies the first (x_i) and the second (h_i) column of file "T3.hist", and adds the third column $W_i = -log(h_i * 0.02)$. Note the width of each bin w_i is 0.02.

There is one careful point to consider; we have to set up a common point for all temperatures so that we can compare PMF at different temperatures and calculate the pre-exponential factor of Arrhenius equation. Usually, this common point is chosen as the top of the transition (as you can see in Fig 2.2). Now open the file "tmp.hist" using `gedit` to find out the maximum of the third column W_i and record it as $Wb3$. The maximum is around position $x = 0$. Then run this one-line Perl script with the shift $Wb3$ (type it in one line):

```
perl -wnal -e ' printf("%12g %12g %12g\n",

  $F[0], $F[1], (-log($F[1]*0.02)-Wb3)) ' T3.hist > T3result.dat
```

Fig 2.2 should give you a general idea of what a reasonable PMF looks like (based on the potential in Fig 1.1b). **Plot βW vs. x.** You can check your result "T3result.dat" by using `gnuplot`. After invoking `gnuplot`, issue command:

```
plot 'T3result.dat' using 1:3 with linespoints title '474 K'
```

"using 1:3" in the command line means plotting the third column data (y axis) vs the first cloumn data (x axis). Does your picture make sense compared with Fig.2.2? It is not necessary to save this picture now.

Run the PMF script for all 5 temperatures. Remember to use the corresponding shift value in the one-line Perl script (or Excel) for each temperature. At the end, you can plot all these PMF (W_i vs x) curves together by using `gnuplot` and save the picture as "pmd.eps":

```
plot 'T1result.dat' using 1:3 with linespoints title '296 K'
replot 'T2result.dat' using 1:3 with linespoints title '338 K'
replot 'T3result.dat' using 1:3 with linespoints title '474 K'
replot 'T4result.dat' using 1:3 with linespoints title '592 K'
replot 'T5result.dat' using 1:3 with linespoints title '789 K'
set xlabel 'x'
set ylabel 'W (k_BT)'
replot
call '../gnuplot_4eps' 'pmf.eps'
```

If the sampling in $x > 0$ region is not good for low temperatures, you can extend (double) the length of simulations at those temperatures.

> **Question 2.1**: How does the plot (βW vs x) compare across the different temperatures? Can you explain the vertical order of the curves at different temperatures?

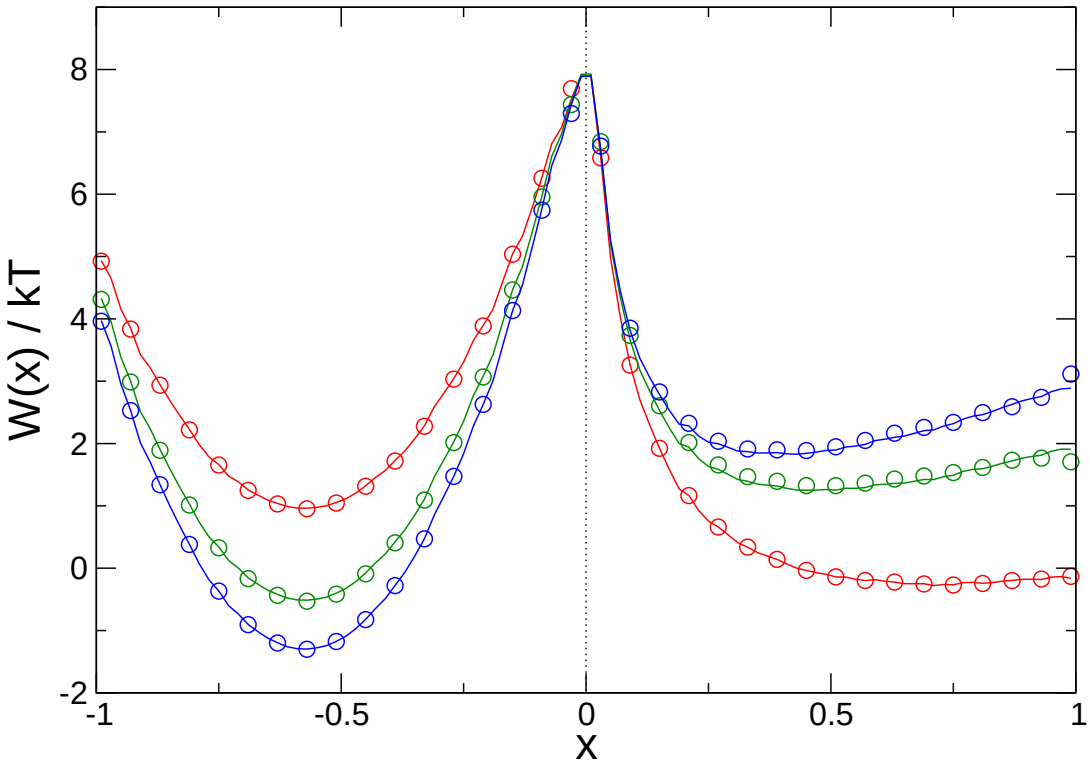


Figure 2.2: The PMF along the x coordinate at three temperatures, 395K, 431K, and 526K (blue, green and red respectively). Solid lines are the exact PMFs calculated by numerical integration of the potential, while the circles are derived from kinetic network simulations at each temperature.

MC kinetics on the 2D potential

The executable `exe_run_MC` is used to generate folding and unfolding first passage times (FPT), of which you will calcluate the mean (MFPT) to obtain rate coefficients $k = MFPT^{-1}$.

Next, we will need to run the executable `exe_run_MC`. This program takes 5 command line arguments which control the program's function:

`tempidx` Temperature index. This index runs from 1 to 5 which correspond to the tempatures 296K, 338K, 474K, 592K, 789K, respectively.

`MCwidth` Width of the MC proposal distribution along the x-axis. We recommend you keep this parameter set at 0.05, though you can play around with it if you have the time and are curious.

`Nevents` Number of folding and unfolding events. The program will stop running once it completes `Nevents` folding/unfolding events.

`lxbound` Left boundary of the buffer region along the x-axis.

`rxbound` Right boundary of the buffer region along the x-axis. `lxbound` and `rxbound` define the buffer region within $x \in [-1, 1]$ which partitions the region which defines the folded state from that which defines the unfolded state. For example, a buffer

region of $[-0.05, 0.05]$ would define the folded state (F) to be $x \in [-1, -0.05]$ and the unfolded state (U) to be $x \in [0.05, 1]$.

While running, the executable will output the rate coefficients to standard output, and when complete, it will generate two output files, `folding_FPT.dat` and `unfolding_FPT.dat` which contain the first passage times for each folding and unfolding event, respectively.

Buffer region size

Before we run at different temperatures, we must first determine an appropriate buffer region size. Run `exe_run_MC` at temperature 3 (474K) for 100 (un)folding events with buffer regions $[-0.01, 0.01]$, $[-0.05, 0.05]$, $[-0.1, 0.1]$, and $[-0.2, 0.2]$.

```
cd ~/klab/simulations
mkdir buffer
cd buffer
```

Note that these output files will be overwritten if you run `exe_run_MC` again, so it's best to copy them to a new name. Therefore, it may be beneficial to run the executable as follows:

```
../exe_run_MC tempidx MCwidth Nevents lxbound rxbound > buffer_idx.dat
mkdir results_buffer_idx
mv buffer_idx.dat results_buffer_idx/
mv folding_FPT.dat results_buffer_idx/
mv unfolding_FPT.dat results_buffer_idx/
```

where idx is the index of the buffer region size. This way, the results for a single definition of buffer region are stored in an appropriately labeled directory.

Question 2.2: How do the apparent first passage times change with the size of the buffer?

Question 2.3: By looking at the first passage times, which buffer region is most problematic? State your reasoning and explain why the observed behavior makes sense.

MC kinetics

Now choose an appropriate buffer region and run `exe_run_MC` for all 5 temperatures with that buffer region. (We already did it at the beginning of this lab.)

Question 2.4: Once you have the rate coefficients, plot them as an Arrhenius plot. Does it agree with your expectations?

3 Final Analysis

Recall the Arrhenius equation,

$$k = A \cdot e^{-\frac{\Delta G^\dagger}{k_B T}}. \tag{3}$$

We can find out $\beta \Delta G^\dagger$ from the PMF plot (obtained from last step), which is the PMF barrier between either folded state or unfolded state and the transition region. Determine all k_f and k_u (in terms of A) at the five temperatures based on the PMF plot. Then draw a graph of $\ln \frac{k}{A}$ vs $\frac{1}{T}$ and give the best linear fit to them.

> **Question 3.1**: Does this graph agree with the graph your previous $\ln k$ vs. $\frac{1}{T}$ plot? How can you test the consistency between the $\ln k$ vs. $\frac{1}{T}$ and $\ln \frac{k}{A}$ vs. $\frac{1}{T}$ plots?

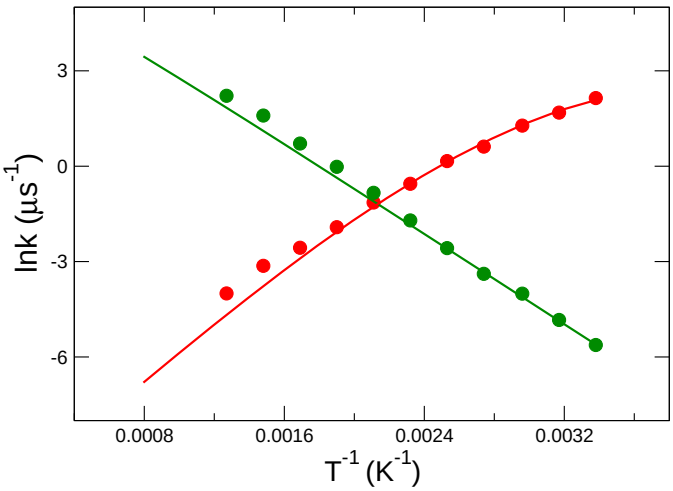


Figure 3.1: Arrhenius plot of the folding rates of the model system. Folding and unfolding rates are indicated by red and green, respectively. The line represents the folding rate from MC simulation in units of 10^{-6} per MC step, which is calculated by using the Arrhenius equation based on activation energies derived from the PMF along x axis. The circles represent the rates from MC simulation.