Structural Bioinformatics Practical Manual Calculating Protein-Protein interaction Potential of Mean Force using Molecular Dynamics Simulations and Umbrella Sampling

K. Anton Feenstra – k.a.feenstra@vu.nl
Sanne Abeln, Juami van Gils and Maurits Dijkstra

March 2, 2021

1 Introduction

In this practical you will be introduced to free energy methods for calculating free energies of a physical process that is relevant to biology. Using the molecular dynamics technique, combined with umbrella sampling, you will calculate the free energy or, equivalently the potential of mean force (PMF), associated with breaking the interaction between two proteins in a complex.

We calculated interaction energies for the same protein complex as the one in this practical using both atomistic and coarse-grained forcefields (May *et al.*, 2014); in the same study we also included another, larger, complex. Earlier, this complex has also been studied computationally by others (Cui *et al.*, 2008), but please note that they used very different computational methods to calculate the free energies. Please refer to these papers for more background detail, if you are interested.

This practical will help you learn:

1. how to perform MD simulation in practice.
2. where PMF of protein-protein interaction comes from and the approach to calculate the PMFs
3. to understand umbrella sampling and comparision between two different methods calculating PMFs (umbrella sampling in this practical and constraint force integration in the May et al. (2014) paper). **Make sure you have read the May et al. (2014) paper before starting the practical.**

To remind yourself some basic knowledge, please refer to Chapters 12, 13 and 14 on protein dynamics, thermodynamics and MD simulations. There may be some additional specific information in the PPI lecture slides, that is not presented in the chapters. More detailed background reading:

- For an excellent, detailed introduction to molecular dynamics simulations (physical principles, force fields, algorithms), please refer to the GROMACS Manual.

- For an introduction to the umbrella sampling technique we will be using here to calculate the PMF, please refer to the short but good Wikipedia page on Umbrella Sampling.

2 Deliverables

Throughout this practical manual you will find questions. Please make a document in which you answer each of these questions one by one (there are 10 in total). Please indicate which question you are answering. You only have to hand in the questions in the question blocks. There are some additional questions in the text to help guide you through the assignment. Please think about these and ask questions about them during the practical sessions, but you do not have to hand them in. You can (and should) discuss them with the practical assistants for feedback during the practical sessions. Please add a short caption for each figure, do not answer the question in the caption. Captions do count towards the word limit.

Submit your report via `canvas.vu.nl`.

To be efficient, you do not need to answer each question before you continue with the rest of the assignment. Some steps involve a bit of computation time (or a lot if you want to improve your results later on), so you can work on some of the other questions while you are waiting.

Question 1: [10 points, max 150 words] The technique used in the May *et al.* (2014) paper for calculating the PMF is constraint force integration.

→ Describe the essential difference between constraint force integration and umbrella sampling.

→ Name two main assumptions that need to hold for these methods to work correctly (only one is explicitly mentioned in the paper).

---

3 Starting Up

For the calculations we will use the GROMACS molecular simulation package. Visualization of trajectories will be done with Chimera.

Download the pmf_prak_dist_2020.tar.gz file in canvas that contains the necessary scripts and various input files. You can unpack it using the following command:

```
tar -xf pmf_prak_dist_2020.tar.gz
```

We will use the `PlotXVG.py` script for displaying graphs of the simulation outcomes for analysis. You can find this script in the `src` directory in the supplied archive file; you will need to give its full path when you run it (or make sure your shell can find it). (Alternatively, you can cut-and-paste and/or import the numerical columns into your graphing program of choice, or write a script in `R` if you like. However, we can offer very limited support for such solutions.)

You should install these programs on your own computer. At the end of this manual, you will find a summary of all the software used, and pointers to where to install them from. If you are using the VU server you will only need to install chimera.

3.1 Getting the Necessary Files

- Download the tarball archive from Canvas and unpack it in your working directory.
- Get the crystal structure `1VET` from the `PDB` (choose the `PDB` file format for download)
- go to the `PPI` subdirectory (which you just unpacked)
- Open the `1VET` structure in your favourite PDB viewer.
  - **(?)** How many chains are present in the structure?
  - **(?)** What secondary structure types (strand, helix, coil) are involved in the interaction?

3.2 Generating the Topology

We will let GROMACS process the input coordinate file, and set up some of the force field parameters for us. But before that, we need to check if the crystal structure `1VET.pdb` is complete, i.e. it has all atoms present in all amino acids.

- open a terminal and go to the `PPI` directory
- make (`mkdir`) a directory `AddAtoms`, and `cd` there
- put the `1VET.pdb` crystal structure file here
- run the Modeller script `ModellerTools.py` on `1VET.pdb`:

  ```
  python3 ../../src/ModellerTools.py 1VET.pdb
  ```

This will produce a `1VET_add.pdb` which has missing sidechain atoms added.

Now we're ready to use the GROMACS protein pre-processor and topology generator `pdb2gmx`.

- go back to the `PPI` directory (`cd ..`) and then to a new directory `Pdb2Gmx`
- copy the output file of the previous step:
  `cp ../AddAtoms/1VET_add.pdb .`
- generate the GROMACS topology for `1VET.pdb`:

  ```
  pdb2gmx -f 1VET_add.pdb -o 1VET.gro
  ```

  then select the `GROMOS96 53a6` forcefield.
  if asked, select the following:

  - select `SPC` for Water Model
  - select `1: NH2` for N-terminus
  - select `1: COOH` for C-terminus

  - select `1: SPC` for Water Model

  (Depending on your GROMACS version, options will be different or not presented; choose intelligently!)

- we now have the GROMACS molecular configuration (`1VET.gro`), position restraints files (`posre_Protein_chain_*.itp`), and the topology (`topol_Protein_chain_*.itp` and `topol.top`).

If you're curious, you can try to run `pdb2gmx` on the original crystal structure file `1VET.pdb`; but be sure to re-run with the correct input later as some output files will get overwritten.

3.3 Setting Up the System

We will set up the size of our simulated system.

- create a new `EditConf` directory in the `PPI` directory
- copy the molecular configuration of the previous step:
  `cp ../Pdb2Gmx/1VET.gro .`
- use the GROMACS tool `editconf` to set the box size so there is 1.2 nm space around the molecule and it is centered in the box:

  ```
  editconf -c -princ -d 1.2 -f 1VET.gro -o conf.gro
  ```

  when you are asked for group(s) to work on, select `Protein`.
- the output configuration (`conf.gro`) contains our molecule centered in a rectangular box.

3.4 Energy Minimizing the System

We are starting a new simulation, and to do that properly we must first relax any strain in our starting conformation by energy minimization. This strain mainly arises from inaccuracies in the X-ray crystallography structure determination experiment.

- change to the `Em` directory in the `PPI` directory (this one is already there)
- copy the molecular configuration of the previous step:
  `cp ../EditConf/conf.gro .`
- copy the topology generated by `Pdb2Gmx`:
  ```
  cp ../Pdb2Gmx/*.top .
  cp ../Pdb2Gmx/*.itp .
  ```
- A parameter file for the GROMACS pre-processor is already present in this subdirectory, you should have a look by opening `minim.mdp` in a text editor. Have a look at these parameters to get some feeling of what is needed to set up an MD simulation. The GROMACS website has a page explaining all of them in some detail: `manual.gromacs.org/archive/4.6.7/online/mdp.html`.

- preprocess the input for the energy minimization:

  ```
  grompp -f minim.mdp -c conf.gro -p topol.top
  ```

- perform the energy minimization:

  ```
  mdrun -v -nt 1
  ```

  If your computer has hyper threading (many do), it makes sense to tell `mdrun` with the `-nt` option to only use as many threads as there are physical cores – if you don't it may actually be slower! (Depending on the GROMACS version and install, this option may not be available - in which case it will only use a single thread.) When you run GROMACS on the `compute` servers during the practical session, please also use `nt -1`, since other students are using it as well.

- The output configuration is written to `confout.gro`

- Use `g_confrms` to fit the starting and minimized structures on top of each other for easy comparison;
  `g_confrms -f1 conf.gro -f2 confout.gro`
  when you are asked for group(s) to work on, select `Protein`.

  (?) The output file `fit.pdb` will contain both structures, open it in your protein viewer. Can you now see any differences? (You may need to look very closely.)

  (?) The tool also calculates the RMSD, look it up in the output. How large is it?

  (?) Explain if your observation makes sense, knowing that the energy minimization optimizes the structure to a local minimum?

- The energy minimization log can be viewed in `md.log`

- The next step expects a PDB file as input, so we need to convert our minimized output conformation to PDB:
  `editconf -label A -f confout.gro -o confout.pdb`

- Fix chain identifiers that are needed to tell one protein from the other, and the names of the C-terminal oxygen atoms. From residue `MET 123` onwards, you need to change the chain identifier (`A`) to `B`; easiest is to do a find-replace of "` A `" into "` B `". You also need to change the names of two oxygen atoms: "`O1`" into "`O`" and "`O2`" into "`OXT`" to conform with the PDB naming conventions that modeller expects. Please make sure you keep the ***columns aligned*** in the PDB file, as the PDB format is strictly a fixed-width column format. Save the file as `1VET-em.pdb`

4 Umbrella Sampling

4.1 Coarse-graining

Up to now we have been working with the detailed 'atomistic' protein structures. To speed things up, we will use a so-called 'coarse-grained' force field for the remainder of the calculations. In this case, we will be using the MARTINI forcefield, developed by the group of Siewert-Jan Marrink in Groningen (Monticelli *et al.*, 2008).

In addition, there are a lot of separate things to take care of; we need to put both protein chains at a range of different distances for the umbrella sampling. And at each distance, we need to carefully equilibrate the system by energy minimization, adding water, more energy minimization, and some short MD simulations with the proteins (partially) constrained so the water can adjust to interacting with the proteins.

Ideally, for maximum learning effect, you should be doing each of those steps by yourself, but that would need a lot more time than we currently have. So instead, we have a script that takes as input a pdb file, a set of chains to 'pull' using the umbrella potential and a range of distances to calculate. Just be sure to remember that the script hides a lot of complexity! We will go into some of the intricacies later.

4.2 Distance

To do the umbrella sampling, we need to put the two protein chains at a number of different distances with respect to each other. But we first need to know what this distance is in the crystal structure. Using `chimera` we can calculate the center of mass (COM) distance between the chains of our protein.

- open Tools → General controls → Command line. Then execute:

```
define centroid :.A
define centroid :.B
dist c1 c2
```

(?) Which distance did you measure? What is the unit of this measure?

(?) Which distances should you use for the umbrella sampling simulations (minimum, maximum)?

4.3 Equilibrations

Now, we proceed to do the umbrella sampling setup.

- go to the the `PPI` directory
- copy the output conformation of the previous (EM) step:
  `cp Em/1VET-em.pdb ./1VET-em.pdb`
- run the script, and be sure to choose an appropriate distance range (`<start>` and `<end>` values) based on the distance you measured in section 4.2:
  `python3 ../src/Pull.py -f 1VET-em.pdb -c A:B -d <start>:<end>:0.1`

  If the script throws errors about not finding modules like `Bio.PDB` that means BioPython was not (properly) installed. Refer to the end of this manual for some instructions how to fix this for your account. Also make sure that all the necessary modules (see start of the manual) are loaded.

The script will run for a while, because it will do the equilibration minimizations and MD runs for each distance (it takes about one minute for each distance). It will also produce a lot of output on the terminal, so you can see it is still working. If the process stalls for more than a minute or so that means something is starting to go wrong in the simulation and you may cancel the calculations for that distance by pressing `<ctrl-c>` in the terminal where the script runs. The script will then proceed to the next distance. (If you don't do this, the simulation will ultimately terminate by itself, but that just takes longer.)

When the script finishes, you don't actually have any real output yet; it just readied things for the production simulations. You will find a directory called `1VET-em`. If you look in it, you will see a list of directories, one for each distance (`d1.8`, `d1.9`, etc.). Inside each of them, you will find separate subdirectories for each of the equilibration steps that have already been performed by the script:

`em_vac`
: Energy minimization (EM) in vacuum (so, just the protein). This allows the protein to 'adjust' to the forcefield (which, in a way, is a different 'environment' than the crystal structure).

`em_vac_posre`
: EM in vacuum with position restraints so the protein cannot change conformation too much.

`solvate`
: No minimization or simulation here, but just defining the size of the simulation box and filling it with water molecules (for different distances, here you will get different box sizes and subsequently different amounts of water added – some of it in between the two protein chains). The interface between protein and water will be 'rough'; this will be solved in the next three steps.

`em_sol_posre`
: EM with position restraints on the protein, so the newly added water molecules can adopt an optimal orientation with respect to the protein surface, i.e. forming local hydrogen bonds.

`md_sol_posre`
: MD (dynamics) simulation, but still with position restraints on the protein. Here the protein is mainly fixed, but all the water molecules are free to move around and find a nice arrangement around the protein molecules; this completes the 'solvation' process. This allows water molecules to fill up holes that were left between the protein surface and the water molecules.

`md_sol_eq`
: More MD, but now without restraining the protein molecules. The whole system is now free to adapt. The Umbrella potential is already switched

on, so the protein molecules are not completely free to change the COM distance.

Question 2: [10 points, max 150 words] We like you to think a bit on what each of these steps mean; please note these are new things to you, which have not been explained in detail in the lectures, nor in the papers presented. There is a section in the online GROMACS manual (How-to "Steps to perform a simulation") that gives some hints and pointers, but also includes a lot of further detail that is not so relevant here, so try not to get confused there.

→ Consider why we need (short) simulations before your production run using umbrella sampling and just after energy minimization. What is the difference between this short simulation and energy minimization?

4.4 Production simulations

Finally, there is another directory (`md_sol_prod`), which contains the input file for the production MD simulation, one for each distance, if all equilibrations went right.

(?) Which distances failed? Can you offer an explanation why?

For each of the `md_sol_prod` directories we now need to start the production simulations using:

```
mdrun -v
```

(Here, like before, it makes sens to use `-nt` to limit the number of threads to the number of physical cores in your computer.)

In stead of starting all simulations manually, you can also use a small script like:

```
for i in d[0-9].[0-9]*/md_sol_prod; do
    echo $i
    cd $i
    mdrun -nt 1 -v
    cd ../..
done
```

Each simulation will take about a minute minutes (up to about 3, larger distances have more water and thus run slower). So you should be able to finish this within an hour (on a single computer in the practical room). You can also consider copying the `topol.tpr` file to some other computer to distribute the workload, and then copy the output back. Make sure to keep track of which distance you were simulating in which place (it is probably best to recreate the `d?.?` directory with just the single `md_sol_prod` directory and `topol.tpr` file inside).

5 Analyzing the Results

Once all simulations have finished, there are two ways to process the results. One is using the GROMACS tool that implements the weighted histogram analysis

(WHAM) method, and is easy but gives you no insight into what is happening. The other is doing some manual binning and averaging of distances and energies, which is what we will do first.

5.1 Analysis according to umbrella sampling

The wikipedia page on Umbrella Sampling gives a short but concise introduction into the principle of umbrella sampling – which is what we have done using harmonic potentials. You can think of this umbrella potential as similar to how a bond interaction is modelled in the forcefield: there is a reference distance, an actual distance (which changes throughout the simulation) and a force constant.

5.1.1 Extracting Energies and Distances

We now first need to extract from each of our simulation the distances (requires some small processing using the `pullx.xvg` output file that has been produced by `mdrun`) as well as the total potential energies and the umbrella sampling energies (using the `g_energy` tool). We can automate this using:

```
for dir in d?.?*/md_sol_prod; do
    cd $dir
    ( echo COM-Pull-En; echo Potential ) | g_energy -f ener.edr -s topol.tpr
    awk '$1+0==$1{print $1, sqrt($5^2+$6^2+$7^2);next}{print}' pullx.xvg > dist.xvg
    cd ../..
done
```

This creates an `energy.xvg` containing the umbrella potential energy and the total potential energy, and a `dist.xvg` file containing center of mass distances, both as a function of time.

It is instructive to have a look at them graphically. For the distances, you can look at all of them together using:

```
python3 PlotXVG.py -xvg_f d?.?*/md_sol_prod/dist.xvg
```

(Ignore the legend, this is not correct now.) You will see a number of traces overlaying each other, each corresponding to one of your simulations.

For the energies, just have a look at one of them:

```
python3 PlotXVG.py -xvg_f d2.0/md_sol_prod/energy.xvg
```

You will see two traces: one around zero (the Umbrella energy, called 'COM Pull En.' which stands for Center of Mass Pulling energy), and another at a large negative value (the total Potential energy). If you are curious, you may want to zoom in on both to see how they behave.

Question 3: [10 points, max 200 words] Plot the COM Pull energy vs. distance for each of your distances. Note, you cannot simply use `python3 PlotXVG.py -xvg_f d?.?*/md_sol_prod/energy.xvg` like you did for the distances, since there are multiple sets of energies in each file. You can manually extract the right columns from the xvg files, or you can modify the PlotXVG.py script to select the columns you need.

→ Show the plot for one of your starting distances. What figure would you expect for an infinitely long sampling?

→ Show the COM Pull energies for your simulations that started at (approximately) 2.0 nm and 2.7 nm. What is the difference between the two?

→ Discuss how you can explain this difference.

Question 4: [10 points, max 100 words]

→ Show all the COM Pull energies in the same plot (COM Pull energy vs. distance). Based on this plot, where do you expect your free energy minimum to lie and why?

Question 5: [10 points, 50 words for the short caption] As you increase the COM distance, the box size increases. Consequently, there are more water molecules in the box and thus more favorable interactions between the water molecules in the box. Therefore, the average potential energy of the system decreases as the COM distance increases.

→ Using your simulation results, create a figure that supports this claim.

5.1.2 Unbiasing the umbrella potential

The umbrella potentials that we applied at different distances, will have biased the simulation to remain close to that distance. The statistics (distances, energies) are therefore also biased in the same way. Since we know *exactly* how the biasing umbrella potential works, we can re-calculate it from the output, and apply the reverse biasing. In this way we can exactly recover the statistics *as if* we had not applied the umbrella potential in the first place.

The most important formula needed, used to calculate an arbitrary property A from your simulation, is as follows:

$$\langle A \rangle = \frac{\langle A/w \rangle_w}{\langle 1/w \rangle_w} \tag{1}$$

where w is the weight according to the umbrella potential, $\langle \cdots \rangle_w$ denotes the average over the simulation including the umbrella (*w*eighted) potential, and $\langle A \rangle$ the *proper* (unweighted) average of property A that we are actually interested in. For our setup, w is a function of the umbrella energy $E_{Umbrella}$ for a given conformation, according to Boltzmann's formula:

$$w = e^{-E_{Umbrella}/k_B T} \tag{2}$$

To properly go from observed probabilities in umbrella sampling, to free energies, we need to solve a complicated set of equations. The main problem is that we are combining results from simulations at different distances, pretending that these simulations are all directly comparable. But some simulations may be at a distance that is inherently less likely than others. This lower likelihood should lead to a higher free energy (smaller negative numbers), but this method cannot do that.

5.2 Automated analysis using WHAM

To solve the problem of the relative likelihood of each of the individual simulations, the weighted histogram analysis was developed. There are several resources available about this topic (Kumar *et al.* (1992); Bartels (2000); WHAM), but they are quite technical and involve a lot of sophisticated mathematical analysis.

Here, we will just be using the `g_wham` method as available within GROMACS, as follows:

- make a list of `tpr` files:
  `ls -1 d?.?*/md_sol_prod/topol.tpr > tpr-files.dat`
- make a list of output 'pull force' (`pullf.xvg`) files:
  `ls -1 d?.?*/md_sol_prod/pullf.xvg > pullf-files.dat`
- Use the GROMACS tool `g_wham` to perform a '**w**eighted-**h**istogram **a**nalysis **m**ethod': `g_wham -it tpr-files.dat -if pullf-files.dat -o -hist`

`g_wham` will iteratively combine the separately sampled energies into one combined free energy profile, or potential of mean force (PMF). The PMF can be plotted using `PlotXVG.py` using the following command:
`python3 PlotXVG.py -xvg_f profile.xvg`
The associated histograms that were used by the weighted histogram method can be displayed by typing:
`python3 PlotXVG.py -xvg_f histo.xvg`

An important success factor for umbrella sampling is sufficient overlap between neighbouring umbrella simulations (this overlap, among other things, is what allows the WHAM method to estimate the relative likelihoods of the simulations). The `histo.xvg` file will show you a histogram for each of the umbrella simulations; you should see that the histograms are all centered around different distances. But you will probably also see that overlap in several cases is quite limited.

Question 6: [10 points, max 150 words] Compare the histograms with the PMF plot (add both figures in your report).

→ Explain the effect of the sampling, as observed in the histogram on the PMF plot. Consider some examples.

To increase the overlap there are several things you can do:

- add an additional simulation at intermediate distance to 'close the gap' (run the `Pull.py` script again, now setting the range so it will only do this additional distance(s); then re-do the analyses above as appropriate)
- combine your output with that of one of your colleagues. Each simulation starts with randomly generated starting velocities, so these results are not identical. Just add the appropriate files to the `tpr-files.dat` and `pullf-files.dat` you used for `g_wham` above.

Question 7: [10 points, max 150 words] Choose **one** of the above solutions.

→ Explain which one you chose and what changed/improved in your plots.

→ Compare your results to those of a fellow student who chose a different solution.

We will first compare your results to the plots in the May et al. (2014) paper, which we will consider as a reliable reference here.

Question 8: [15 points, max 200 words] Compare the PMFs you obtained from `g_wham` with the ones presented in the paper.

→ In case of ideal sampling, do you expect to see any differences between the PMF plot in May et al. and your PMF plot? What do you observe? Refer back to Question 1 if needed.

Now, finally, we can start to interpret some of our results.

Question 9: [10 points, max 100 words] Providing sampling has been sufficient, your PMF should contain four major features: a downward slope, a global minimum, an upward slope and a plateau. These can all be observed nicely in the figures in May *et al.* (2014).

→ For each of these features, name the distance range it occurs in and explain what physical phenomenon it describes. Include the following terms in your explanation: steric clashes, physical contact, hydrophobic effect, water-mediated interactions.

Question 10: [5 points, max 100 words] The MARTINI force-field does not explicitly describe backbone hydrogen bonds.

→ Where may backbone hydrogen bonds be particularly important for this complex? Refer back to the structural properties of the interface you looked at in the beginning of the practical (section 3.1).

---

6 Software:

Software used in this exercise. Those marked with asterisk (‘*’) are strictly essential; also at least one protein viewer (like `rasmol` or `chimera`) and a graphing tool (the supplied `PlotXVG.py` script, or an external tool like `xmgrace`, `matplotlib`, `gnuplot`, or a spreadsheet) is essential.

On Ubuntu/Debian systems you can easily install most of this as follows:

```
sudo apt-get install gromacs grace rasmol modeller
sudo apt-get install python python-matplotlib python-biopython
```

Dependencies and paths to tools for `Pull.py` are defined in `src/Env.py`.

- `gromacs` * www.gromacs.org
  the `Pull.py` script assumes GROMACS version 4.0.5, and also works with the installed GROMACS 4.5.5, but may not work equally well with other versions.
- `dssp` for defining secondary structure; a pre-compiled linux version is supplied.
- `modeller` * www.salilab.org/modeller
- `xmgrace`
  for plotting.
- `rasmol`
  simple protein structure viewer. If you need a simple Protein structure viewer, `rasmol` is a good alternative to the others mentioned below. For this practical, you only need one of them, so you can choose one that you’re already familiar with.
- `pymol`
  advanced protein structure viewer
- `chimera` www.cgl.ucsf.edu/chimera/
  advanced protein structure modelling tool
- `VMD` www.ks.uiuc.edu/Research/vmd
  advanced protein structure modelling tool
  - `pbctools`
    If you have `VMD` on your own laptop, you can install the `pbctools` package to visualize the boundaries of the simulation box. In the ‘extension’ menu, choose the ‘`Tk` console’. There you can enable by `package require pbctools`.
- `python` *
  - `biophyton` * biopython.org/wiki/
  - `matplotlib`

– If you need to install BioPython on your own account (without root access), please execute the following:

```
wget http://biopython.org/DIST/biopython-1.63.tar.gz
tar zxvf biopython-1.63.tar.gz
cd biopython-1.63/
python3 setup.py build
python3 setup.py install --user
```

  Afterwards, you may clean up the `tar.gz` file as well as the whole `biopython-1.63` directory (clearing some 100MB disk space). Further info on `biopython.org/wiki/Download`.

References

Bartels, C. (2000). Analyzing biased monte carlo and molecular dynamics simulations. *Chemical Physics Letters*, **331**(5–6), 446 – 454.

Cui, Q., Sulea, T., Schrag, J. D., Munger, C., Hung, M. N., Nam, M., Cygler, M., and Purisima, E. O. (2008). Molecular dynamics-solvated interaction energy studies of protein- protein interactions: the MP1-p14 scaffolding complex. *J Mol Biol*, **379**(4), 787–802.

GROMACS Manual. GROMACS Website. `www.gromacs.org/Documentation/Manual`.

How-to. Steps to perform a simulation. GROMACS Website. `www.gromacs.org/Documentation/How-tos/Steps_to_Perform_a_Simulation`.

Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry*, **13**(8), 1011–1021.

May, A., Pool, R., van Dijk, E., Bijlard, J., Abeln, S., Heringa, J., and Feenstra, K. A. (2014). Coarse-grained versus atomistic simulations: realistic interaction free energies for real proteins. *Bioinformatics*, **30**(3), 326–334.

Monticelli, L., Kandasamy, S. K., Periole, X., Larson, R. G., Tieleman, D. P., and Marrink, S.-J. (2008). The martini coarse-grained force field: Extension to proteins. *J Chem Theory Comput*, **4**(5), 819–834.

Umbrella Sampling. Wikipedia. `en.wikipedia.org/wiki/Umbrella_sampling`.

WHAM. Weighted histogram analysis method. AlchemistryWiki. `www.alchemistry.org/wiki/Weighted_Histogram_Analysis_Method`.