

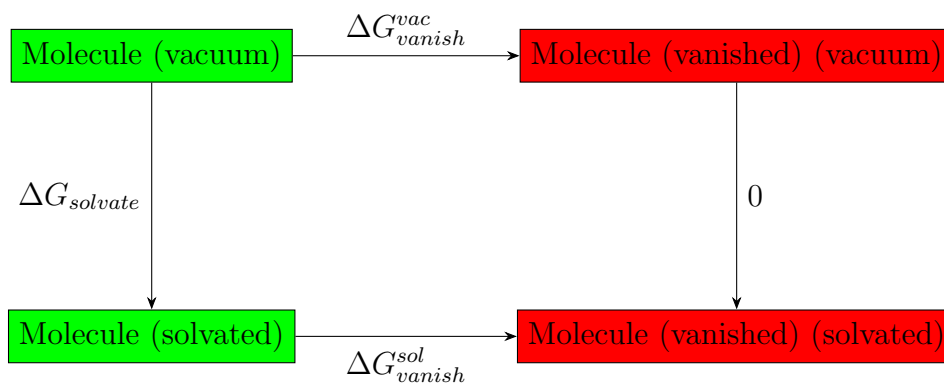
Molecular Dynamics Exercise VI: Calculating the Solvation Free Energy of Methane using the BAR Method

Tristan Alexander Mauck

December 23, 2024

This report deals with the calculation of the solvation free energy of a methane molecule in water using alchemical transformations. To this end, the BAR method was used with an initial window separation of $\Delta\lambda = 0.05$, leading to a result within 5% of the reported literature value. Subsequent runs were conducted with more sampling and smaller window separation, both improving the result. All simulations were performed using the MD software GROMACS.

1 Introduction and Procedure



As the modelling of the actual solvation process is computationally difficult, one usually takes a different route to calculate solvation free energies [1]. This starts by creating a thermodynamic cycle such as the one presented above. First, the molecule sits in vacuum. From here two processes are considered: Solvating the molecule, which is the path of interest, and vanishing the molecule. Vanishing the molecule is done by turning off all molecular potentials, meaning that the molecule is effectively decoupled from its environment. After vanishing the molecule, one considers the solvation of the vanished molecule. This process does not have to be considered later in the calculations as due to the deactivated potentials there is no interaction between the molecule and the solvent and hence also no change in free energy. The cycle is closed by a path from the solvated molecule to the vanished molecule in the solvent. This process is essentially a decoupling of the molecule from the surrounding solvent. Making use of the fact that for a state function, such as the free energy G , the sum of all energy changes in a cycle is zero it is now possible to avoid the actual solvating process. Its free energy difference can be found from the other processes. This can be expressed in equations as follows [1]:

$$\Delta G_{solvate} + \Delta G_{vanish}^{sol} - \Delta G_{vanish}^{vac} = 0$$

$$\Delta G_{solvate} = \Delta G_{vanish}^{vac} - \Delta G_{vanish}^{sol}$$

So the desired free energy of solvation can be expressed as the difference of the free energy changes induced by decoupling the molecule from its environment in vacuum and in solvent. Additionally, the free energy difference of the vanishing process of the molecule in vacuum can also be left out of the calculation as it cancels with the contributions of the intramolecular forces during the process of the vanishing of the molecule in the solvent. All in all, the solvation energy is given by the negative value of the free energy difference of vanishing the molecule in the solvent. To simplify the simulation setup in this exercise, only the deactivation of the Van der Waals Forces (LJ potential) is considered. The Coulomb forces are considered to already have been deactivated.

While the explanation above has shown how costly simulation of the solvation process can be avoided, it remains to clarify how the 'vanishing' process can be implemented. Technically, the 'vanishing' process belongs to a larger family of processes known as alchemical transformations. Here, unphysical processes such as the removal of atoms, addition of atoms and changing of atom types are modelled. For this a linear coupling parameter λ is added linking the initial and final Hamiltonian. Varying the parameter from 0 to 1.0 turns the initial structure into the final structure. In the 'vanishing' process, better called decoupling process, the final Hamiltonian of our molecule H_B would be the initial Hamiltonian without the Lennart-Jones potential terms (Remember: The Coulomb term has already been removed in both.).

$$H(\mathbf{r}^N, \mathbf{p}^N, \lambda) = (1 - \lambda)H_A + \lambda H_B \quad \text{with } \lambda \in [0, 1] \quad (1)$$

Formally, one requires now a method that allows the computation of the free energy difference between $\lambda = 0$ and $\lambda = 1$. As these states behave naturally fairly different one tends to split this transformation into many smaller. This means one find the free energy differences between states λ and state $\lambda + \Delta\lambda$ with $\Delta\lambda$ given by how many steps one needs or can afford. In this report, the Bennett Acceptance Ratio (BAR) method was used to calculate the free energy difference between subsequent states. This method iteratively solves the following relation for the unknown values C and ΔG . Here ΔG is the free energy difference between the initial state and the final state (denoted i and f in the equation) [3].

$$\beta\Delta G = \ln\left(\frac{\langle f(U_i - U_f + C) \rangle_f}{\langle f(U_f - U_i - C) \rangle_i}\right) + C \quad (2)$$

Here f is the Fermi function and the angle brackets denote ensemble averages taken from the respective simulation (f or i).

In the GROMACS implementation an energy minimization, an NVT equilibration, an NPT equilibration and a production run was conducted for each value of the parameter λ [2]. The number of steps and time steps for the equilibration and the production run were set as follows: NVT: 5 000 steps at $\Delta t = 0.002$ ps, NPT: 5 000 steps at $\Delta t = 0.002$ ps and for the Production Run: 250 000 steps at $\Delta t = 0.002$ ps. The temperature coupling was implicitly included in the stochastic integrator producing Langevin dynamics and the barostatting was done using the Parinello-Rahman Method.

To efficiently create the various mdp files and run the simulations two bash scripts were used. These are shown below:

```
#!/bin/bash

mkdir -p MDP
echo "Should we just test the simulation workflow? [y/n]"
read TEST

# for loops a very useful tools in bash and also in almost
# every other programming language as well
# here, a variable is set iteratively to the values listed after 'in'
# These values, are to be set by you. They must match the names already in use
# in this directory as can be seen below: ${mdp}.mdp corresponds to the mdp
# files in your exercise directory. If you are unsure, ask the tutor or the internet.
for mdp in 'em' 'npt' 'nvt' 'prod'
do
    for i in {0..20}
    do
        # What means the '-E' option of 'sed'? Add some comments
        sed -E "s/(init_lambda_state\s+=\s+)/O/\1$i/" ${mdp}.mdp > ./MDP/${mdp}_${i}.mdp

        if [[ $TEST == y ]];
        then
            # what does this command do? Add some comments
            sed -i 's/nsteps.*\/nsteps = 500/g' ./MDP/${mdp}_${i}.mdp
        fi
    done
done
```

Figure 1: Bash script for creating energy minimization, thermal equilibration, pressure equilibration and production run files (.mdp).

```
# Set some environment variables
FREE_ENERGY=$(pwd)
MDP=$FREE_ENERGY/MDP

# make the user aware of his/her choices
echo "Free energy home directory set to $FREE_ENERGY"
echo ".mdp files are stored in $MDP"

# loop over all values of lambda
# $(()) is called a command substitution, where the output of that command
# can be transferred into a new command or, in this case into
# a variable used in a for loop for more information google 'command substitution bash'
# and for more information about seq, google it or type 'man seq' in your terminal
for l in $(seq 0 20)
do
    mkdir -p lambda_$l
    echo "Starting minimization for lambda = $l ..."

    # Define Variables
    # the following two variables are very important for the md runs. To set them correctly, of course one
    # would have to understand what they are used for. in vim you can search the file by typing e.g. '/wd_prev' + <enter>
    # you can find the next/previous instance by hitting <N> / <P> keys.
    # What are they used for? Can you set them freely or are they constrained to certain values?
    wd_prev=""
    sim_prev="methane_water"
    lambda_dir=$FREE_ENERGY/lambda_$l          # current lambda value

    # loop over all different simulation steps
    for sim in 'em' 'nvt' 'npt' 'prod'
    do
        # preliminaries before the simulations are executed
        cwd="$lambda_dir/$sim"                  # set the current working directory
        cat $sim.banner                          # Concatenate the according banner to give information about the current run (just for fun)
        mkdir -p $cwd                            # create the current working directory

        # start the run of the corresponding lambda and simulation step
        gmx grompp -f $MDP/$sim $l.mdp -c $wd_prev/$sim_prev.gro -p $FREE_ENERGY/topol.top -o $cwd/$sim.tpr -po $cwd/mdout.mdp &>$cwd/grompp.$sim
    || { echo "something went wrong, check $cwd/grompp.$sim"; exit; }
        gmx mdrun -deffnm $cwd/$sim &>$cwd/mdrun.$sim || { echo "something went wrong, check $cwd/mdrun.$sim"; exit; }

        # save the name of the current simulation step to access it in the next iteration
        sim_prev=$sim                          # name of next simulations input file
        wd_prev="$lambda_dir/$sim"
    done
done
```

Figure 2: Bash script for running all subsequent simulations and providing the correct files for each simulation.

Requested Comments:

1. 'sed' is a stream editor and allows for efficient editing of text in files. With the '-E' option the extended regular expressions are enabled for the editor. This is used to extract the i-th lambda state from the mother file and setting it to the newly created file.
2. Being in the if-loop which is entered upon activating the test mode, this line reduces the number of steps of all simulations to 500. In this way, they will run quickly and the result can be double-checked. This is done by finding the pattern 'nsteps.*' and replacing it with the pattern 'nsteps = 500'.

2 Simulation Results and Discussion

The simulation results are presented in figure 3. In the histogram the change in free energy between each subsequent states is shown as well as the cumulative free energy difference. While the computed free energy differences are positive for small λ , the values turn negative for $\lambda > 0.5$. Consequently, the cumulative free energy difference is first positive, but turns then negative. The final result for the solvation energy is $\Delta G_{\text{solvation}} = -2.13 \frac{\text{kcal}}{\text{mol}}$. This is a 4.5% deviation from the literature value of $\Delta G_{\text{solvation}} = -2.24 \frac{\text{kcal}}{\text{mol}}$. There could be several reasons for this: The spacing of the parameter λ might not have been sufficient to capture all details of the changes of the free energy. This is supported by the fact that a simulation run with $i_\lambda \in (0, 1, \dots, 40)$, hence a reduction of $\Delta\lambda$ by half, yielded an improved result with $\Delta G_{\text{solvation}} = -2.15 \frac{\text{kcal}}{\text{mol}}$ (4.0% deviation). An additional run with $\Delta\lambda = 0.01$ improved the result again to $\Delta G_{\text{solvation}} \approx -2.18 \frac{\text{kcal}}{\text{mol}}$ (2.7% deviation). On another note, the number of steps in the production run might not have been enough to represent an accurate ensemble average which is needed in the BAR method. Here a run with the number of step for the production run doubled yielded again $\Delta G_{\text{solvation}} = -2.18 \frac{\text{kcal}}{\text{mol}}$ (2.7% deviation).

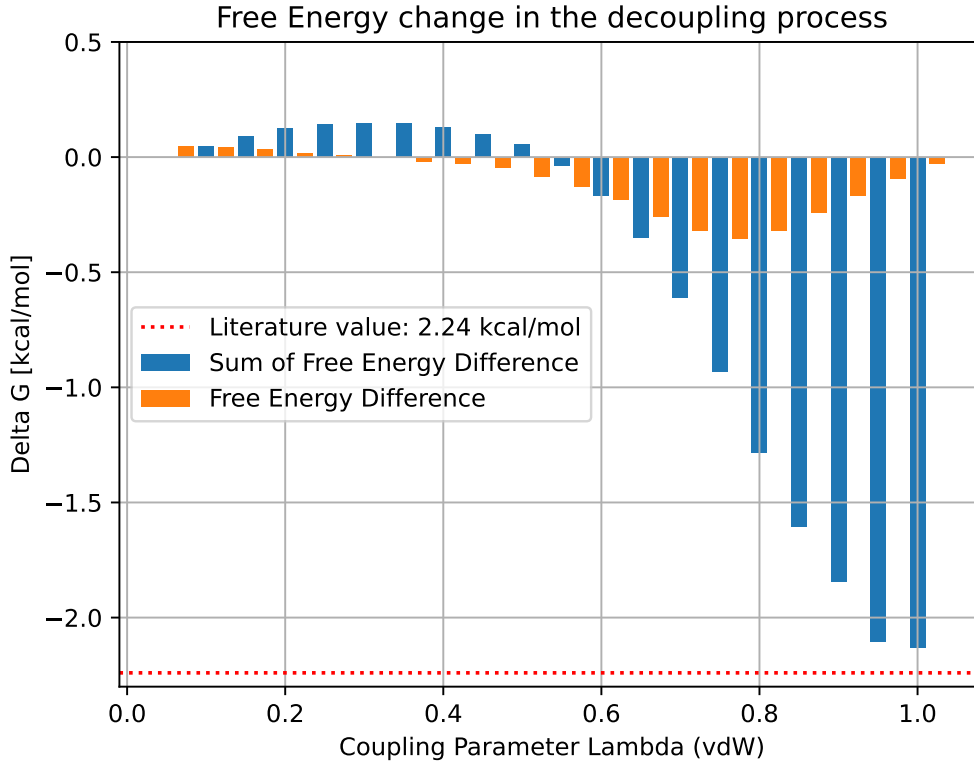


Figure 3: The free energy change (and the accumulated free energy) for the decoupling of the van der Waals forces of Methane in water.

3 Bibliography

- [1] Zacharias, Martin et Al. Molecular Dynamics: From Basics to Applications Lecture Notes; 2023
- [2] Vollmers, Luis; Zacharias, Martin; Reif, Maria; Molecular Dynamics: Exercise 6 - Calculation of the Solvation Free Energy of Methane Using the MBAR Method; 2023

[3] Bennett, Charles; Efficient Estimation of Free Energy Differences from Monte Carlo Data; Journal of Computational Physics; 1976