

Luis Vollmers | Dr. Maria Reif | Prof. Martin Zacharias

Molecular Dynamics (SS2025)
Exercise 6

Calculation of the Solvation Free Energy of Methane Using the BAR Method

Report Tasks

These simulations will take a considerable amount of time to complete. You should first test how long a single production run (for one lambda window) takes, either on your own computer or in the CIP-Pool. Based on this, you can reduce the number of simulation steps to achieve a shorter runtime. For reference, we provide output files from full-length runs. In your analysis, please include both your own simulation results and the reference data (if your simulations used fewer steps).

1. Complete the missing parts in the two bash scripts `run_lambda_simulations.sh` and `generate_mdp_files.sh` and include your code suggestions in the report (as a screenshot or similar).
2. Use `gmx bar` to generate your `bar.xvg` and `barint.xvg` and include the plots in your report with a brief discussion.
3. Discuss your result for ΔG_{solv} and compare it to the one in the literature.

Acknowledgement

This exercise was inspired by a tutorial offered by Justin Lemkul¹.

Introduction

In contrast to prior tutorials that focused on exploring different algorithms, this tutorial centres on a specific application: the calculation of free energies. Free energy calculations are critically important in computational biophysics because they quantify various biologically relevant processes, such as stability alterations from point mutations, small ligand binding, and solvation free energy.² Setting up and analysing these simulations involves numerous choices experienced users must carefully select.³

The system of interest in this tutorial is an aqueous system with a single methane molecule. The goal is to calculate the solvation free energy, ΔG_{solv} , for the methane molecule. This exercise will also convey an understanding of the physical and mathematical aspects involved in such calculations and the computational framework provided by GROMACS. While an in-depth exploration of free energy calculations is beyond the scope of this tutorial, additional sources

¹http://www.mdtutorials.com/gmx/free_energy/03_workflow.html

²<https://doi.org/10.1063/1.2221683>

³<http://dx.doi.org/10.1002/jcc.21450>

can be consulted for further information.⁴⁵⁶

The property to be calculated, ΔG_{solv} , was reported in 2003 by Shirts et al. They found a value of $2.42 \text{ kcal mol}^{-1}$ for the OPLS-AA force field.⁷ By following the steps described below, it should be possible to replicate their results with reasonably good agreement.

Theoretical Considerations

Solvation free energy, ΔG_{solv} , also known as the Gibbs free energy of solvation, is a fundamental concept in physical chemistry. It quantifies the free energy change when a solute dissolves in a solvent, usually under standard conditions (pressure of 1 bar, temperature of 298.15 K, and concentration of 1 mol L^{-1}). The process of solvation entails changing the whole chemical environment of the solute. Thus, the solute might rearrange because its intra-molecular interactions adjust to the solvent contacts that suddenly present themselves. Exposed hydrophobic entities would aggregate within each other, while buried hydrophilic entities might become solvent-accessible. Especially for complex molecules and complex solvents (it does not have to be water), there is no way to estimate the solvation free energy intuitively given the multitude of (competing) enthalpic and entropic driving factors. Since the Gibbs free energy quantifies the spontaneity of chemical processes, finding a way to calculate this property from first principles would render such an estimate possible. Then, a negative ΔG_{solv} would indicate a readily dissolving solute. The general thermodynamic equation to calculate ΔG_{solv} for a binary solution of the solute and the solvent entails three states and their free energy. The solute alone (G_{solute}), the solvent alone (G_{solvent}) and the two components mixed (G_{solution}). The process of solvation associated with ΔG_{solv} is bringing them both together. Thus, it is the difference between the mixed and the unmixed systems.

$$\Delta G_{\text{solv}} = G_{\text{solution}} - G_{\text{solute}} - G_{\text{solvent}} \quad (1)$$

The control granted by computational methods enables the user to design a meticulous strategy to conduct this calculation. Technically, a simulation could be conducted, where the reverse process is probed, and the solute is magically pulled out of the solvent into the gas phase (vacuum). The force necessary to separate the two components would then be related to ΔG_{solv} . However, a more elegant approach for the system of interest is the so-called *alchemical route*. Alchemical means that computational operations are conducted that are impossible in real-world physics, e.g., making a particle disappear into nothingness. Calculating the solvation free energy via the alchemical route means decoupling the van-der-Waals and Coulombic interactions between the solute and the solvent, effectively transforming the solute into a *ghost particle* that is formally part of the simulation but does not affect the system anymore, i.e. does not interact with its surroundings anymore. The fully interacting state is usually denoted as state *A* and the fully decoupled (ghost-like) state *B*. Despite this seemingly unphysical procedure, the alchemical route has a rigorous mathematical framework and yields fairly accurate results.⁸⁹ With both, the alchemical route and equation (1) in mind a thermodynamic cycle can be constructed to showcase the required simulation setup.

Starting from the complete solution, hence G_{solution} in equation (1), the solute and the solvent are fully interacting (see figure 1). Since decoupling the solvent is computationally expensive

⁴<http://dx.doi.org/10.1021/jp102971x>

⁵<http://dx.doi.org/10.1002/jcc.10052>

⁶[http://dx.doi.org/10.1016/0021-9991\(76\)90078-4](http://dx.doi.org/10.1016/0021-9991(76)90078-4)

⁷<https://doi.org/10.1063/1.1587119>

⁸Kirkwood (1935) <https://doi.org/10.1063/1.1749657>

⁹Zwanzig (1954) <https://doi.org/10.1063/1.1740409>

because of the sheer number of molecules, decoupling the non-bonded parameters of the solute is the sensible choice. The next step would be to transfer the non-interacting solute out of the solvent into a vacuum. This process does not require work and thus does not contribute to ΔG_{solv} . As a last step, the intermolecular interactions are turned on again (the decoupling is reversed), and the solute adjusts its conformation according to its vacuum environment. This conformational change is non-existent for a methane molecule since directly connected atoms have bonded interactions only. Thus, the vacuum contribution $\Delta G_{A \rightarrow B}^{\text{vac}}$ results in zero as well.

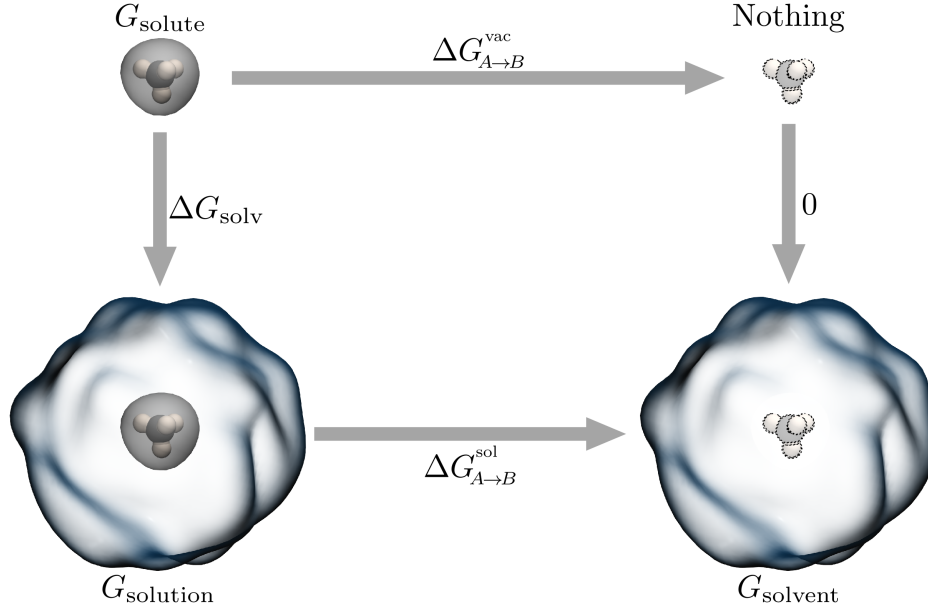


Figure 1: The solute, here a methane molecule, is transferred from a vacuum into water (left arrow). There are two intermediate systems shown on the right side. The upper one only comprises the decoupled solute, visualised by transparency and dotted outlines, and the lower system shows the decoupled solute in the solvent. Each system is labelled with its respective term from equation (1). In a thermodynamic cycle, only the end states matter; therefore, the alchemical vanishing of the solute within the aqueous solution (three right arrows) can be chosen to indirectly obtain ΔG_{solv} . The contribution from each arrow needs to be summed up, yielding the solvation free energy.

Therefore, the only contribution that is necessary to calculate the solvation free energy is $\Delta G_{A \rightarrow B}^{\text{sol}}$, which denotes the alchemical vanishing of the solute within the aqueous solution. However, the vanishing cannot be conducted within a single simulation. For two states to be eligible for free energy calculation, a considerable energetic and conformational overlap must exist.

Therefore, the solute does not vanish in one step but via a multistep approach. The coupling parameter λ is introduced, which ranges between zero and one and specifies the decoupling progress from state A ($\lambda = 0$) to B ($\lambda = 1$). Theoretically, it can be applied to the full Hamiltonian of the system containing the kinetic and potential energy or only to individual parts, e.g. the Lennard-Jones potential V_{LJ} (part of the potential energy U).

$$V_{\text{LJ}}(r_{ij}, \lambda) = \frac{(1 - \lambda)C_{12}^A + \lambda C_{12}^B}{r_{ij}^{12}} - \frac{(1 - \lambda)C_6^A + \lambda C_6^B}{r_{ij}^6} \quad (2)$$

$$V_{\text{LJ}}(r_{ij}, \lambda) = \frac{(1 - \lambda)C_{12}^A}{r_{ij}^{12}} - \frac{(1 - \lambda)C_6^A}{r_{ij}^6} \quad \Bigg| \quad C_{12}^B = C_6^B = 0 \quad (3)$$

The coupling strategy can be tuned to arbitrarily interpolate between both end states by selecting the appropriate λ spacing. For every λ , an MD simulation has to be performed to sample

the system in each λ -state. This procedure can be understood as yet another thermodynamic cycle, applied to the only step that remained from figure 1.

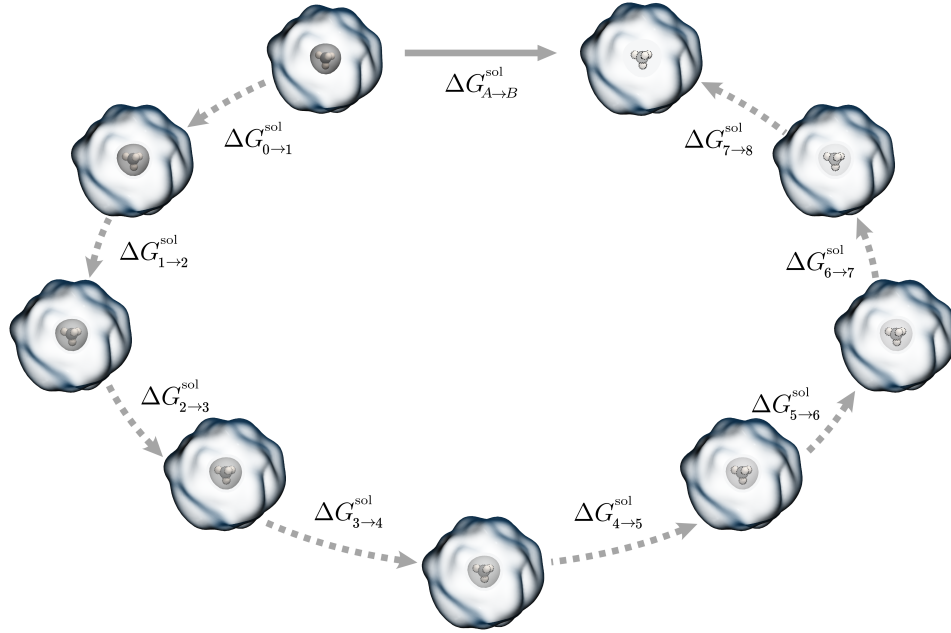


Figure 2: Visualisation of the coupling/decoupling strategy. The direct but error-prone calculation of $\Delta G_{A \rightarrow B}^{sol}$ is circumvented by a thermodynamic cycle that introduces unphysical intermediates (here: 7, 9 states in total) utilising the coupling parameter λ_n that assumes values between zero and one. Each λ -state is simulated, and the free energy contributions are then calculated with an analysis method, e.g. the Bennett acceptance ratio (BAR) method.

The Bennett acceptance ratio (BAR) method¹⁰ is the method of choice for calculating the contributions showcased in figure 2, i.e. $\Delta G_{i \rightarrow j}^{sol}$ for the transition between to λ -states. The general idea is to sample configurations X_i with the Hamiltonian that corresponds to a certain λ_i , $H(\lambda_i)$. The Hamiltonian concerning a certain λ can be rewritten as U_i the potential energy since it is the only contribution affected by the decoupling strategy and for a consistent notation. The lengthy mathematical derivation can be looked up elsewhere¹¹, nonetheless, the final equation utilises $X_i = \{x_1, \dots, x_i, \dots, x_M\}$ and $X_j = \{x_1, \dots, x_j, \dots, x_N\}$ the configurations sampled from two adjacent Hamiltonians (i.e. simulations) with sample sizes N and M , and evaluates these configurations with their corresponding and each others potential energy functions U_i and U_j .

$$\Delta G_{i \rightarrow j} = \ln \left[\frac{\sum_{x_j} f(U_i(x_j) - U_j(x_j) + C)}{\sum_{x_i} f(U_j(x_i) - U_i(x_i) - C)} \right] + C - \ln(N/M) \quad (4)$$

$$C = \Delta G_{i \rightarrow j} + \ln(N/M) \quad (5)$$

Here, the function $f(x)$ is the Fermi function $f(x) = \frac{1}{1 + \exp(x)}$, and the constant C is to be determined by iteratively applying the above equations until the self-consistent results do not change anymore.

¹⁰Bennett (1976) doi.org/10.1016/0021-9991(76)90078-4

¹¹Bennett (1976) doi.org/10.1016/0021-9991(76)90078-4, eq. 12 mostly

Setup and Simulations

Each part of this theoretical consideration has its computational counterpart. Due to the aforementioned relevancy of free energy calculations, the GROMACS software implements the decoupling strategy via the MDP file. Also, the BAR analysis can be conducted using the `gmx bar` command. The force field and the system coordinates are given in the TOP and GRO files, as per usual. It contains the methane molecule (called ALAB) and 750 TIP3P water molecules within the OPLS-AA force field approach. In the downloadable TOP file, the charges of the methane molecule are set to zero which is appropriate since the coulombic term is negligibly small according to Shirts et al.¹². Thus, incorporating a second complicated decoupling strategy does not add didactic value and can be assumed properly conducted prior to the van der Waals decoupling at hand.

The simulation phases comprise the usual energy minimisation, NVT and NPT equilibrations and the production simulations. With the difference that every λ value requires its own set of simulations. Herein, the decoupling strategy consists of 21 equidistant λ values from 0 to 1, thus in steps of 0.05. Two additional bash scripts can be downloaded, which ease housekeeping of the variety of different output files, `generate_mdp_files.sh` and `(run_lambda_simulations.sh)`. The first one generates all necessary MDP files for running all simulation phases for each λ from the downloadable source MDP files, and the second one executes all the runs in an automated fashion. However, these scripts are incomplete and must be augmented by the user where indicated. Once both scripts are fully functional, they can be executed in the following way:

```
./generate_mdp_files.sh
```

This bash script will prompt the user to decide if a test run should be conducted first. A test run simply means that all MDP files will only run for 500 frames, drastically reducing the simulation time to a couple of minutes, even on older hardware. The dummy output files produce unreliable free energy estimates but still result in an error-free BAR analysis (see below). Running the script in test mode is highly recommended and can be followed by denying the test run to generate MDP files with the appropriate number of frames as input for `run_lambda_simulations.sh`. Files generated with `generate_mdp_files.sh` are named e.g. `em_0.mdp`, `nvt_1.mdp` or `prod_19.mdp`, and define not only the integrator, thermostat, barostat, and cutoff scheme but also any other fancy restraints and coupling/decoupling schemes that require algorithmic implementation. The corresponding free energy options provided by GROMACS are quite flexible and, therefore, a bit complex. The relevant ones are elaborated in table 1. Although working parameters have been predetermined and adjusted in the source MDP, it is important to understand the free energy-related options before proceeding. For even further details, refer to the manual.¹³

Table 1: Tabular description of parameters related to free energy calculations in GROMACS. On the left is the MDP-parameter with its corresponding value, and on the right is a brief description.

<code>free-energy = yes</code>	Interpolate between topology A to topology B via the λ -states. Any λ -states would be ignored if this option would be set to no.
--------------------------------	---

¹²<https://doi.org/10.1063/1.1587119>

¹³<https://manual.gromacs.org/documentation/2023-beta/user-guide/mdp-options.html>

<code>init-lambda-state = 0</code>	Specifies which column of the lambda vector should be used for the current simulation run. As shown below, λ -values are compiled into lists, and the corresponding column specifies the transition degree between states A and B. Zero means starting from the first column.
<code>delta-lambda = 0</code>	Increment per time step for λ . Instead of instantiating each λ -value and thus the alchemical transformation, the whole process can happen incrementally by specifying choosing non-zero for this option. However, this can result in serious errors and is therefore omitted.
<code>calc-lambda-neighbors = 1</code>	Controls the number of lambda values for which Delta H values will be calculated. E.g. for a value of 1 and a <code>init-lambda-state</code> of 10, energy differences concerning λ -states 9 and 11 are computed.
<code>vdw-lambdas = ...</code>	A list of λ -values for the transformation of van der Waals interactions.
<code>couple-moltype = Methane</code>	Name given in the TOP file [<code>moleculetype</code>] section that is to be alchemically transformed from state A to state B.
<code>couple-lambda0 = vdw</code>	Tweaks the kind of non-bonded interactions that are present at $\lambda = 0$. Here, charges are set to zero, and van der Waals interactions are considered.
<code>couple-lambda1 = none</code>	Tweaks the kind of non-bonded interactions that are present at $\lambda = 1$. Here, all non-bonded interactions will be turned off. This will require soft-core potentials to avoid singularities.
<code>couple-intramol = no</code>	All intra-molecular non-bonded interactions for [<code>moleculetype</code>] <code>couple-moltype</code> are replaced by exclusions and explicit pair interactions (as they would without <code>free-energy</code> set to yes).
<code>nstdhdl = 10</code>	The frequency for writing $dH/d\lambda$ and possibly Delta H to <code>dhdl.xvg</code> .

The λ spacing might be the most important part of the alchemical route and is implemented within the λ vectors in each MDP file. The λ vectors for any `*.mdp` contains 20 columns and six rows.

```
; init_lambda_state 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20
vdw-lambdas = 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
0.75 0.80 0.85 0.90 0.95 1.00
coul-lambdas = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00
bonded-lambdas = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00
restraint-lambdas = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00
mass-lambdas = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00
```

```
temperature-lambdas = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00
```

Although `vdw-lambdas` is the only non-zero row, the other ones are still listed to showcase what else might be specified. The option `init-lambda-state = 0` specifies the Hamiltonian with the 0-th element in the `vdw-lambdas` list, here `0.00`. Setting `initial-lambda-state = 1` would correspond to the next element, hence `0.05`, while `initial-lambda-state = 20` would specify the final column, resulting in a λ -value of `1.00` for van der Waals interactions meaning they would be turned off completely according to equation (3).

The other bash script is called `run_lambda_simulations.sh`. It also needs to be completed, and one of the tasks is to fill in the gaps and append code suggestions to the report. After successful completion, `run_lambda_simulations.sh` should first create the following directory hierarchy:

```
lambda_0/
lambda_1/
lambda_2/
lambda_3/
lambda_4/
```

This way, all steps in the workflow are executed within a single directory for each value of `init-lambda-state`, which is a convenient way to organise the simulations and their output. Then, the aqueous methane system is simulated, and if no error occurs, the terminal output should look something like the below:

```
Starting minimisation for lambda = 0 ...
```

```
#####
# ENERGY MINIMIZATION STEEP #
#####

#####
#   NVT EQUILIBRATION       #
#####

#####
#   NPT EQUILIBRATION       #
#####

#####
#   PRODUCTION MD          #
#####
.
.
.
```

However, should there be a mistake, the bash script should report an error to the user, as seen below.

```
Starting minimisation for lambda = 0 ...
```

```
#####
# ENERGY MINIMIZATION STEEP #
#####
```

```
something went wrong, check /long/absolute/path/to/lambda_0/em/mdrun.em
```


Notice how the bash script specifies a file containing more information on what might have caused the error. Given that the missing parts were filled in correctly, all the simulations should be executed automatically. The whole workflow can take quite a long time (**about 8 hours on 4 CPUs**); therefore, the simulations should be started as early as possible. Once all of the production simulations are complete, the resulting data can then be analysed.

Utilising bash scripts can be a double-edged sword. Firstly, writing a functioning bash script takes time; sometimes, it could take longer than setting things up manually. Secondly, if there is an error somewhere in the bash script, simulations will not finish or end up the way they were intended, and the whole bash script must be executed again.

However, both these drawbacks can be mitigated. Firstly, looking at the workload of this tutorial, there is a total of 84 simulations to run. Seamlessly executing all `gmx grompp` and `gmx mdrun` commands by hand would take a long time, so a bash script is worth it. Secondly, bash scripts have numerous benefits, like reproduction, documentation, and shareability, to name just a few. Last but not least, brainlessly copying and pasting or re-executing commands is not engaging for most operators, and the fun and challenge of coming up with a working bash script is not to be underestimated as a motivational factor.

Analysis

The BAR module of GROMACS makes the calculation of ΔG_{solv} very simple. Therefore, all `prod.xvg` files produced by `run_lambda_simulations.sh` must be included in the `gmx bar` command. This can be achieved by copying all the files in a shared directory or (easier) by specifying the correct GLOB. If the aforementioned directory structure is unchanged, the correct command can be seen below.

```
gmx bar -f lambda_*/prod.xvg -o -oi
```

In addition to producing `bar.xvg`, `barint.xvg`, and `histogram.xvg`, the program will print lots of useful information to the terminal, which should look something like this:

Detailed results in kT (see help for explanation):

lam_A	lam_B	DG	+/-	s_A	+/-	s_B	+/-	stdev	+/-
0	1	0.08	0.00	0.03	0.00	0.03	0.00	0.25	0.00
.
.
19	20	-0.05	0.00	0.00	0.00	0.00	0.00	0.10	0.00

Final results in kJ/mol:

point	0 -	1,	DG	0.19 +/-	0.00
.....
.....
.....
point	19 -	20,	DG	-0.13 +/-	0.00
total	0 -	20,	DG	-9.13 +/-	0.09

Examining the content of the output files is essential, as they provide a comprehensive understanding of the decoupling process and the effectiveness of the sampling. From the DG values (corresponding to $\Delta G_{i \rightarrow j}^{sol}$ in figure 2), it becomes evident that some transitions lead to an increase in the free energy of solvation, while others improve the solvation of methane. The values printed for `s.A` and `s.B` represent the configurational overlap between states *A* and *B*. This information is crucial for assessing the adequacy of sampling and the reliability of the

calculated free energy measures.¹⁴

The `bar.xvg` file contains the relative free energy differences for each λ interval (i.e., between neighbouring Hamiltonians) and should look similar to figure 3 when plotted with `xmgrace`.

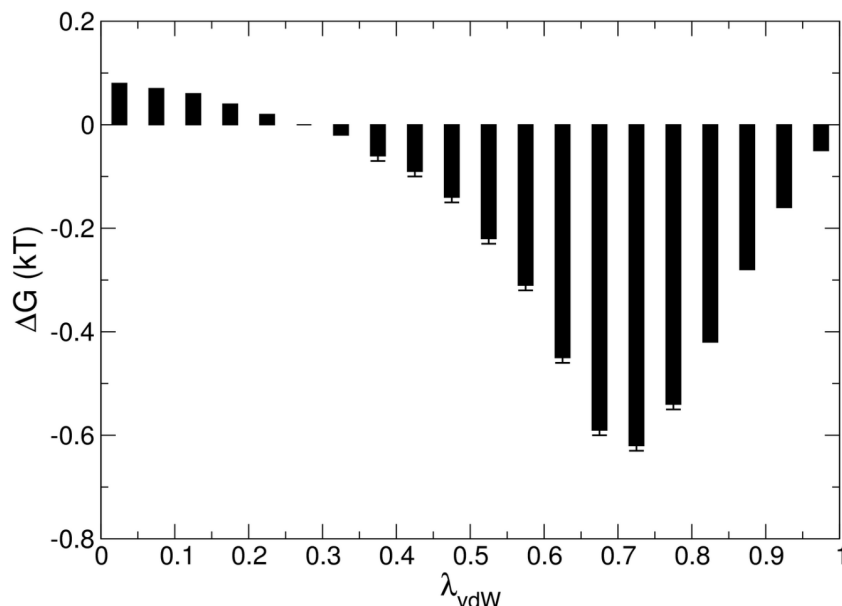


Figure 3: The x-axis contains the discrete λ_i values utilised in the decoupling process conducted here. The y-axis contains the relative free energy difference between two neighbouring λ_i values shown as black bars.¹⁵

The free energy differences between simulation i and simulation $i + 1$ are calculated via equation (4). GROMACS does not use the configurations but instead directly writes out the potential energies $U_i(x_j)$, $U_i(x_i)$ and so on, every `nstdhdl = 10` steps. The total solvation free energy can be obtained by summing up all these intermediate free energy changes.

The `barint.xvg` file plots the cumulative ΔG as a function of λ . Thus, λ_{20} corresponds to the sum of ΔG from λ_0 to λ_{20} . In figure 4 `barint.xvg` is shown together with the data of `bar.xvg` illustrating the cumulative ΔG .

¹⁴<https://aip.scitation.org/doi/pdf/10.1063/1.2011391>

¹⁵http://www.mdtutorials.com/gmx/free_energy/07_analysis.html

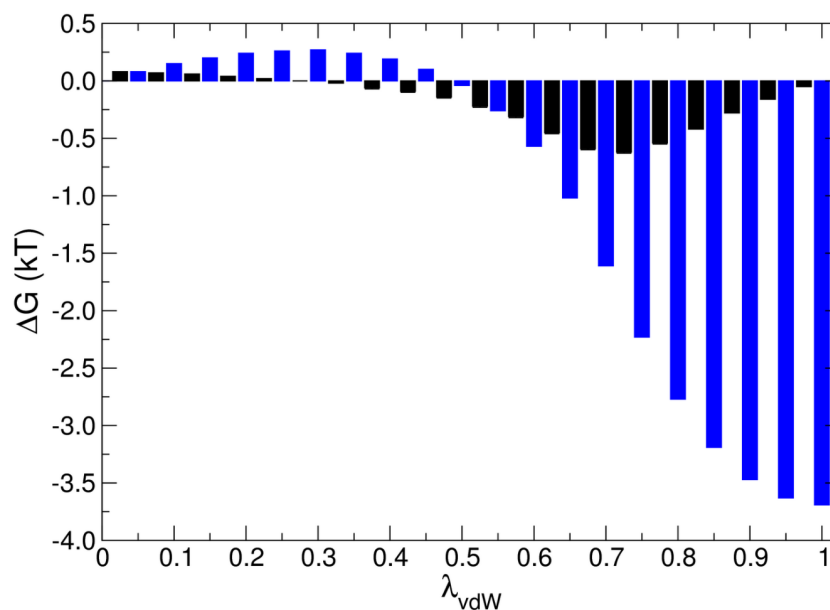


Figure 4: The x-axis contains the λ -increments, and the y-axis shows the free energy difference in units of kT . Additionally to the black bars, the cumulative free energy differences are shown as blue bars.¹⁶

By taking the value of ΔG at λ_{20} , we can recover the value of ΔG_{solv} by converting from kT to kcal mol^{-1} . The literature value for the solvation free energy of methane is $(2.42 \pm 0.01) \text{ kcal mol}^{-1}$ (see Shirts et al. ¹⁷ table II). Report the output from `gmx bar` and how well your simulation setup reproduces the literature value. You may try increasing your simulation time or decreasing your lambda spacing to further compete with the publication, but this would be voluntary. Apart from these two possibilities, discuss in your report what other options may increase the accuracy of your result.

¹⁶http://www.mdtutorials.com/gmx/free_energy/07_analysis.html

¹⁷<https://doi.org/10.1063/1.1587119>