Calculating the difference in the free energy of solvation of K^+ and Na^+.

*Philip W. Fowler**
Structural Bioinformatics and Computational Biochemistry Unit
Department of Biochemistry

Summary

This practical walks you through how to calculate a single value for the difference in the free energy of solvation between K^+ and Na^+. The final value includes an estimate of its error and also the graph of $\frac{dU}{d\lambda}$ against λ is plotted. It is intended that this practical is done *after* an introductory seminar. Finally, some extension exercises and references are included at the end.

Instructions

- First, retrieve the tarball of files from http://sbcb.bioch.ox.ac.uk/fowler/files/fe-practical-files.tgz.
- Unpack the tarball.

```
tar zxvf fe-practical-files.tgz
```

- It will produce a folder called `fe-practical-files/`. Within this folder are two subfolders called `common/` and `runs/`. `common/` contains files that are referred to by the simulations but are not changed, such as the GROMACS topology file and the initial gro file. The `runs/` folder contains two mdp files and some shell scripts (more on these later).
- But, I hear you cry, how is this any different to a normal GROMACS simulation? Well, we need to setup a peculiar atom that is sodium when λ = 0 and potassium when λ = 1. If we look in the `common/` folder at the `n2k.top` file we see the following lines

```
[ moleculetype ]
; Name                  nrexcl
Na2K                      3
[ atoms ]
; nr type resnr residue atom cgnr charge mass typeB chargeB massB
  1  opls_407  1     N2K      I1   1   1.000  22.9898 opls_408  1.000 39.0983
```

- These define a new molecule called N2K[†] and could, of course, be parcelled off in their own itp file if we wished. The unusual bit is that the typeB chargeB and massB parameters are set for potassium and the regular type, charge and mass columns are set for sodium. This tells GROMACS which parameters to use at the endpoints (i.e. λ = 0 or 1) and it, for example, linearly combines the van der Waals parameters in between.
- If we now look at the `n2k.gro` file we see that the molecule is of type N2K. There is a neutralising chloride and the remainder of the system is TIP3P water.
- The other differences are in the mdp files. Change to the `runs/` directory and look at the `runs-1.mdp` file. This is just a short 10ps NPT equilibration run and is no different to a regular mdp file. Here GROMACS just treats the N2K residue as if the "typeB" parameters were not set and so the ion is sodium.
- While we are here run this equilibration using the `launch-1.sh` script. This simply calls `grompp` and then `mdrun` to run this brief equilibration. (You need the `./` to force the shell to run something that is not in its `PATH` but is instead in the local folder, otherwise known as `.`)

```
./launch-1.sh
```

*philip.fowler@bioch.ox.ac.uk
†"n2k" stands for "sodium to potassium".

- This will take 3–6 minutes to complete. While you are waiting look at the `run-2-master.mdp` file. This also looks innocuous enough except for the last few settings

```
; Free energy
free_energy             =  yes
init_lambda             =  0.99
delta_lambda            =  0
sc_alpha                =  0.0
DispCorr                =  EnerPres
```

  (I think some of the other settings may be a bit different to what you have seen before use e.g. `pme_order` but these are all trying to make the nonbonded calculations, especially electrostatics, a bit more accurate than normal)

- Setting `free_energy` to yes alerts GROMACS that we are going to attempt a free energy calculation and so it will look for the 'typeB' etc parameters in the top file. The value of `init_lambda` is the value of λ that the simulation will run at. Here I have put 0.99 but I shall come back to this in a moment. The variable `delta_lambda` is the amount by which λ should be advanced each timestep. This parameters is here so that another technique, called slow-growth can be used. This approach is generally now regarded as inaccurate. `Sc_alpha` determines whether the van der Waals potential is softened or not. We don't have to worry about this here so have set it to be zero i.e. a normal 6-12 vdW potential. Finally, `DispCorr` applies a dispersion correction to improve the accuracy of the calculation.

- We need one mdp file for each value of λ that we wish to run at. This tutorial is set to run at 11 values of λ

```
init_lambda = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
```

- All these mdp files will be exactly the same except for the value of `init_lambda` so I have written a shell script (`create-files.sh`) that copies `run-2-master.sh` eleven times and then replaces 0.99 by the correct value of λ. The first step should have finished now so create the mdp files

```
./create-files.sh
```

- You should now have 11 files that have names like `n2k-2-l0.500.mdp`. The `l0.500` means that this is the mdp file with λ = 0.5.

- Now we are ready to run the 11 GROMACS simulations and record the average of $\frac{dU}{d\lambda}$. If you look in `launch-2.sh` you will see that there is an extra flag next to mdrun

```
mdrun -deffnm n2k-2-l0.500 -dgdl n2k-2-l0.500.fep.xvg;
```

- The `-dgdl` flag tells GROMACS what to call the free energy output file. It is, as you can guess, an `Xmgrace` file.

- Running all 11 simulations one after the other will take between 6 and 12 minutes so you might want to get a cup of tea (or hack the script so it runs on more than one processor...).

```
./launch-2.sh
```

- Of course, we could run all 11 at the same time if we had 11 processors....(assuming that we can start from the same structure each time)

- Finished? The last few scripts analyse the data and print your calculated value of the free energy change. The first of these is called `compute-dg.sh`. This extracts the data from the 11 `xvg` files and then runs it through a `Perl` script (`../common/compute-free-energy.pl`) that numerically integrates the data. We only ran for 2 ps at each value of λ and this script discards the first 1 ps of data from each value of λ to remove any large perturbations. It also estimates the error by assuming that the remaining data has a correlation time of `200 fs` i.e. it can be divided into five bins and we can therefore calculate five independent values of the free energy change. These can be found in `n2k-2.dg`. The error can then be estimated in the usual way.

- The `Perl` script also writes out the variation of $\frac{dU}{d\lambda}$ with λ to a file called `n2k-2.dudl`. It is always useful to look at this graphically as errant simulations can easily be seen. To plot this call the last script

  ```
  ./plot-dudl.sh
  ```

- This uses `gnuplot` which you should be in your `PATH`. If it is, you should get a graph similar to mine which is the one I used as an example above.
- When I run this I get a value of 16.8 $\pm$ 0.4 kcal/mol which is in good agreement with the experimental value of 16.7 kcal/mol[1]
- Please write your result on the white board and then we can compare everyone's results.
- Oh, the script called `remove-simulation-files.sh` does just that: it clears away all the output files. So don't run it before you've recorded your result.

Extensions

- increase the number of λ simulations to 21 (i.e. 0, 0.05, 0.1, 0.15, ..., 1) to check that you get a similar answer. Is this approach more accurate or more precise or both?
- try computing the reverse mutation i.e. potassium to sodium. This is an excellent way of checking that you have not underestimated your errors: if you have the forward and reverse results will not agree with one another (this is called hysteresis).
- investigate what effect increasing the length of the λ simulations has on the accuracy or precision and compare that to the effect of changing the number of λ simulations (i.e. what would you do if you only had a fixed amount of computer time?).
- try a different calculation (e.g. calcium to magnesium)
- try using a different forcefield (e.g. GROMOS53a6). This is more difficult and you would need to find out what the ions are called in this forcefield and make the appropriate changes to the `top` and `gro` files.

So why are free energies difficult?

You may now be wondering what all the fuss is about when it comes to protein free energy calculations since these calculations only took 15 minutes to run. We have only changed one atom to another so the perturbation is very small. Also the water relaxes very quickly around the ion and because all the waters are identical it is not very difficult to representatively sample phase space. If we placed our alchemical ion in a protein then we would have to wait at each value of λ for the protein to relax and adapt to the alchemical ion before measuring $\frac{dU}{d\lambda}$. Even if that is fast or insignificant we must wait a long time to allow the protein a chance to sample all its states. Now imagine how much more complex this would be if we replaced our simple atom with amino acid A being mutated into amino acid B. Or even amino acid A being turned into nothing (i.e. all its nonbonded parameters being switched off). There are now many more degrees of freedom and so one would have to run for far, far longer at each value of λ. More values of λ may be needed to characterise the $\frac{dU}{d\lambda}$ curve and the individual simulations themselves are, of course, much larger. But it can still be done. Just not in an hour.

Further reading

There are a number of books on molecular modelling that cover this subject[2,3]. Leach[2] is the easiest to read but does not go into any great detail. Frenkel and Smit[3] is much more physical and includes more on the theory. Another excellent source of information is review papers[4–8]. Rodinger and Pomès[4] gives a good high-level overview of how the recent theoretical developments are blurring the distinctions between what were separate approaches. Jorgensen[6] examines how free energy calculations are useful in drug discovery and Chipot and Pearlman[7] give a good historical account. Review papers rarely go into the theory and the books are usually not up to date; one of the best sources I have found is the Encyclopedia of Computational Chemistry[9–12]. If you can find it it is worth

photocopying the relevant chapters. Scientific papers where these techniques are used are another good place to look[13,14]. Finally, both the GROMACS[†] and NAMD[‡] websites have tutorials on how to run free energy calculations.

References

[1] Marcus, Y., 1994. A simple empirical model describing the thermodynamics of hydration of ions of widely varying charges, sizes, and shapes. *Biophys. Chem.* 51(2-3):111–127. doi:10.1016/0301-4622(94)00051-4.

[2] Leach, A. R., 2001. *Molecular Modelling. Principles and Applications*. Pearson Education Ltd., Edinburgh, second edition.

[3] Frenkel, D. and B. Smit, 2002. *Understanding Molecular Simulation*. Academic Press, London, second edition.

[4] Rodinger, T. and R. Pomès, 2005. Enhancing the accuracy, the efficiency and the scope of free energy simulations. *Curr. Opin. Struct. Biol.* 15:164–170. doi:10.1016/j.sbi.2005.03.001.

[5] Kofke, D. A., 2005. Free energy methods in molecular simulation. *Fluid Phase Equilib.* 228-229:41–48. doi:10.1016/j.fluid.2004.09.017.

[6] Jorgensen, W. L., 2004. The Many Roles of Computation in Drug Discovery. *Science* 303:1813–1818. doi:10.1126/science.1096361.

[7] Chipot, C. and D. A. Pearlman, 2002. Free energy calculations: The long and winding gilded road. *Mol. Sim.* 28:1–12. doi:10.1080/08927020211974.

[8] Simonson, T., G. Archontis and M. Karplus, 2002. Free energy simulations come of age: Protein-ligand recognition. *Acc. Chem. Res.* 35:430–437. doi:10.1021/ar010030m.

[9] Jorgensen, W. L., 1999. Free energy changes in solution. In P. von Ragué Schleyer, (editor) *Encyclopedia of Computational Chemistry*, volume 2, pages 1061–1070. Wiley, Chichester.

[10] Mark, A. E., 1999. Free energy perturbation calculations. In P. von Ragué Schleyer, (editor) *Encyclopedia of Computational Chemistry*, volume 2, pages 1070–1083. Wiley, Chichester.

[11] Straatsma, T. P., 1999. Free energy simulations. In P. von Ragué Schleyer, (editor) *Encyclopedia of Computational Chemistry*, volume 2, pages 1083–1089. Wiley, Chichester.

[12] Pearlman, D. A. and B. G. Rao, 1999. Free energy calculations: Methods and applications. In P. von Ragué Schleyer, (editor) *Encyclopedia of Computational Chemistry*, volume 2, pages 1036–1061. Wiley, Chichester.

[13] Fowler, P. W., S. Geroult, S. Jha, G. Waksman and P. V. Coveney, 2007. Rapid, accurate and precise calculation of relative binding affinities for the SH2 domain using a computational grid. *J. Chem. Theo. Comp.* 3(3):1193–1202. doi:10.1021/ct6003017.

[14] Michielin, O. and M. Karplus, 2002. Binding free energy differences in a TCR-peptide-MHC complex induced by a peptide mutation: A simulation analysis. *J. Mol. Biol.* 324:547–569. doi:10.1016/S0022-2836(02)00880-X.

[†] `http://www.gromacs.org`
[‡] `http://www.ks.uiuc.edu/Research/namd`