Programming Project

May 2014, Bioinformatik

1 Aim 1

2 Introduction 1

3 Monte Carlo Simulation 2

3.1 Monte Carlo technical details 5

4 Enveloping distribution sampling 3

5 Energies 4

6 Implementation 4

6.1 Monte Carlo machinery 5

6.2 Monte Carlo moves 5

6.3 Calculations 5

6.4 Division of labour 6

7 Analysis 6

8 Tell Prof Schwabe und Prof Torda 6

9 Escape 6

10 References 7

1 Aim

We want to estimate free energy differences between different states of a molecule using energies calculated from a quantum chemical method. We will simulate a molecule using Monte Carlo, but in an energy function which combines two states.

2 Introduction

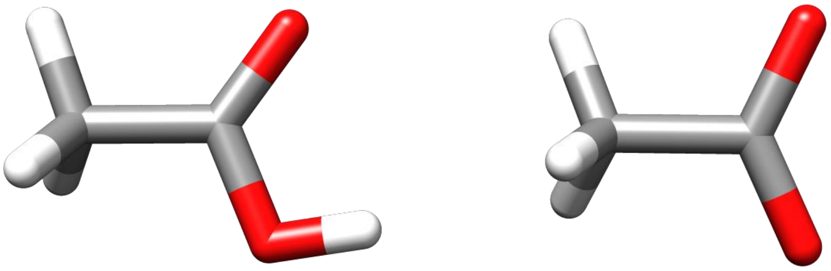
In much of physics and chemistry we worry about the free energy difference ΔG between two states. We have models for potential energy like the formulae you know for electrostatics or gravity. We do not have good models for free energy since there is an entropic (S) contribution to $G = H - TS$ where H is potential energy (enthalpy). Entropy is

$S = -k\sum_i p_i \ln p_i$ where the summation runs over all of the i states. The only reasonable way to estimate entropy is from sampling. We will use the Metropolis Monte Carlo technique.

Chemists worry about protonation. This is usually a reaction like

$$\mathrm{HA} \rightleftharpoons \mathrm{H}^+ + \mathrm{A}^- \qquad (1)$$

where A^- is an anion. Because the project takes place in the Zentrum für Bioinformatik, we must work on a biological example such as essig/acetic acid (CH_3COOH) or glycine (NH_3CH_2COOH). For acetic acid, the system only has one rotatable bond:



The A^- in this system is the CH_3COO^- in the right of the diagram.

3 Monte Carlo Simulation

Monte Carlo simulation means sampling from a probability distribution. In physics, this is usually the Boltzmann distribution where the probability p_i of state i is given by $p_i = \frac{e^{\frac{-G_i}{kT}}}{Z}$ where G is the free energy, k is Boltzmann's constant, T is the temperature and Z is known as the partition function. Your first guess would be to pick a state randomly, calculate its energy and accept the state with a probability p_i. This will not work. You cannot calculate Z.

Metropolis Monte Carlo avoids this problem by considering two states i and j. The ratio of probabilities is given by

$$\frac{p_i}{p_j} = \frac{e^{\frac{-E_i}{kT}}}{\frac{e^{\frac{-E_j}{kT}}}{Z}} = \frac{e^{\frac{-E_i}{kT}}}{Z} \frac{Z}{e^{\frac{-E_j}{kT}}} \qquad (2)$$

$$= e^{\frac{-\Delta E}{kT}}$$

where we use E for energy in a general sense. In our case, it will be a potential energy using a quantum mechanical recipe. We use this nomenclature since our system will be *in vacuo* where concepts of pressure and temperature are rather ill defined. This leads to a method one can directly implement

```
generate starting configuration $\mathbf{r}_o$ and calculate $E_0$
while (not happy)
        generate $\mathbf{r}_{new}$
        calculate $E_{new}$ and $\Delta E$
        if $\Delta E < 0$
                set $\mathbf{r}_o$ to $\mathbf{r}_{new}$
        else
                x = rand [0:1]
                if $x \leq \exp\left(\frac{-\Delta E}{kT}\right)$
                        set $\mathbf{r}_o$ to $\mathbf{r}_{new}$
```

4 Enveloping distribution sampling

If you want to calculate the free energy difference for the reaction A ⇌ B you could run a simulation for a very long time and measure the number of samples, n_A, n_B in the A and B states and say $\Delta G = RT \ln \frac{n_A}{n_B}$. The problem is that if there is an energy barrier between the states, you might need nearly infinite computer time. You would also have to define when the system is in state A and state B. For problems like this, people developed methods where you simulate the states separately and use a recipe to mix the energies. This is the idea of the enveloping distribution method.[1]

Our energy function that we use in the Monte Carlo is a combined energy,

$$E_R(\boldsymbol{r}) = \frac{1}{kT}\ln\left(\exp\frac{E_A(\boldsymbol{r})}{kT} + \exp\frac{E_B(\boldsymbol{r})}{kT}\right) \tag{3}$$

$E_R(\boldsymbol{r})$ is a reference energy as a function of the set of coordinates, $\boldsymbol{r}$. This might look similar to the Hamiltonian $\mathcal{H}(\boldsymbol{r}, \boldsymbol{p})$ you have met in other courses, but in Monte Carlo, we do not have velocities or momenta. Equation 3 has a physical meaning. We have one simulation and one set of coordinates, but the energy is calculated twice with two different energy functions. In the case of the reaction in (1), one energy function has the proton, H bound to the anion. In the second energy function, the proton has coordinates, but it disappears from the energy calculation and the system will have a net negative charge (A^-).

The consequences may not be intuitive. The atoms move and either one of the states (energy functions) will dominate and the system behaves as if it has a proton. If the atoms move somewhere else, the other energy function will dominate in equation 3 and the system behaves as if it has less of a proton. What we want is for the system to move between the two states and sample both evenly. This is not so easy.

Imagine the system has the best energy when it is protonated. If the atoms move so as to favour the deprotonated states, this will make the total energy less negative and any move in this direction is less likely to be accepted. We can artificially fix this. The problem is that the average energies $\langle E_A \rangle$ and $\langle E_B \rangle$ might be very different. We can correct for this by adding an offset to the energies,

$$E_R(\boldsymbol{r}) = \frac{1}{kT}\ln\left(\exp\frac{E_A(\boldsymbol{r}) - E_A^R}{kT} + \exp\frac{E_B(\boldsymbol{r})}{kT}\right) \tag{4}$$

Where E_A^R is some constant that makes the energies in states A and B roughly equal.

5 Energies

Energies will be calculated using orca.[2]

6 Implementation

The first implementation will be slow. If it appears to work, one can discuss faster methods.

6.1 Monte Carlo machinery

The Monte Carlo machinery will be primitive. Your code will move atoms to generate trial coordinates. It will then call orca to calculate the potential energy and decide whether or not to accept the move, according to the scheme on page 3.

6.2 Monte Carlo moves

In Monte Carlo schemes, you have to move atoms randomly. Almost anything is allowed, as long as it does not introduce any bias in the search space. If you move atoms too far, the energy becomes very high and the move is rejected. If you move atoms too little, the simulation does not explore much space. We will start with two kinds of move

1. Pick an atom at random and move it by δr in each of the x, y and z directions where $-r_{max} < \delta r < r_{max}$ and r_{max} is a value like 0.005 nm (about $\frac{1}{20}$Å).
2. Pick a bond at random and apply a rotation about the dihedral angle so that $-\theta_{max} < \delta_\theta < \theta_{max}$

6.3 Monte Carlo technical details

- You have to save the configurations that you visit, or at least save the properties you are interested in (structure, energy).
- At the start of a simulation, the system is not at equilibrium. You normally start simulating and throw the first 1000 or 10 000 steps away.
- The energy function
- Set the Boltzmann constant to $k = 1.38 \times 10^{-23}$JK^{-1}. Be careful with J and kJ. Some programs print energies in kcal mol^{-1}.

6.4 Calculations

6.4.1 The systems

We will start with CH_3COOH. If this works, we would like to try the same calculation on the neutral and zwitterionic forms of glycine (NH_2CH_2COOH). If you write nice code, it should be able to cope with different molecules.

6.4.2 Simulations

1. Simulate CH_3COOH.
2. Simulate CH_3COO^-

We will use these two simulations to

- get values for r_{max} and θ_{max}
- get a value for E_A^R in equation 4
- see if we can introduce rotations around dihedrals of $\pi/2$ (or any other big value) to make the system move faster

3. Simulate in mixed energy form from equation 4.

6.5 Division of labour

- Programming rotations, moves and resetting the molecules centre of mass to some known position
- The interface between your code and orca.

7 Analysis

At the end, we want to be able to extract ΔG values from the simulation with the mixed energy form. This will be discussed later, but there will be two issues

- A correction for the E_A^R term
- Trying to define the two states and answer the question, when is the system more protonated and when is it not protonated ?

8 Tell Prof Schwabe und Prof Torda

Could you please tell us

- Have you met the Boltzmann distribution in other lectures ?
- Have you met Monte Carlo / importance sampling / the partition function ?
- Have you met free energies and Gibbs entropy ?
- What programming languages do you know ?

9 Escape

This project has more chemistry and less programming than the one offered by Prof Kurz. If you already know more chemistry than you think is healthy and do not want to meet a Hamiltonian, you can ask if you can join Prof Kurz's group.

10 References

1. Christ CD, van Gunsteren WF: Enveloping distribution sampling: A method to calculate free energy differences from a single simulation. *J Chem Phys* 2007, **126:**184110.
2. [http://cec.mpg.de/forum/]