

Institute of Theoretical Computer Science

Lecturer: Prof. Jack Snoeyink

Assistant: Yves Brise

April 25, 2007

Geometric Computations in Molecular Biology SS07

Exercise Set 5

Course Webpage: <http://www.ti.inf.ethz.ch/ew/courses/GCMB07/>

Due date: -

Classroom Exercise Finding H-bonds

The first exercise that we are going to look at today is a self explanatory exercise that can be found in the teaching section of the Richardson lab webpage, where we accessed MolProbit last week. It consists of five different molecules in which you have to add the hydrogen bonds. When you think that you found them all, you can check if your solution is correct.

Here are the instructions:

- Go to <http://kinemage.biochem.duke.edu/software/mage.php> (there is a link from the course homepage), and download the Mage viewer for your specific platform. Just save the executable to your desktop.
- Go to the course homepage and download the file HbondPractice.kin from today's lecture material. Alternatively, the HbondPractice.kin file can be found on the page <http://kinemage.biochem.duke.edu/teaching/bch222/>.
- Drag the .kin file and drop it over the executable. Two windows should open. One contains the visualization of a molecule and the other one contains further explanations.
- Start adding hydrogen bonds by clicking on the hydrogen that is donating the electron and then on the acceptor of the electron. The connection will be made visible as a green line.
- Follow the explanations in the secondary window.

Classroom Exercise Optimizing RMSD computation

In this exercise we are going to discuss what can be done to improve the running time of your RMSD computations, especially in Matlab. We look at two approaches. The first one is purely theoretical and connected to this particular problem. The second one is a general tool available in Matlab, the so called “profiler”, which let’s you exactly see where the most time is spent in your code.

Only compute RMSD value:

We have seen that we can compute the RMSD value by doing the actual rotation of one of the point sets, and then computing the squared norms of the differences of the left and the right system.

$$RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\dot{r}'_{r,i} - \dot{q} \dot{r}'_{l,i} \dot{q}^*\|^2}$$

In fact, if we are only interested in the value, but not in the coordinates of the rotated point set, then there is a much easier way to achieve this. Let’s concentrate on the sum under the square root.

$$\begin{aligned} \sum_{i=1}^n \|\dot{r}'_{r,i} - \dot{q} \dot{r}'_{l,i} \dot{q}^*\|^2 &= \sum_{i=1}^n (\dot{r}'_{r,i} - \dot{q} \dot{r}'_{l,i} \dot{q}^*) \cdot (\dot{r}'_{r,i} - \dot{q} \dot{r}'_{l,i} \dot{q}^*) \\ &= \sum_{i=1}^n \|\dot{r}'_{r,i}\|^2 + \sum_{i=1}^n \|\dot{q} \dot{r}'_{l,i} \dot{q}^*\|^2 - 2 \sum_{i=1}^n (\dot{r}'_{r,i}) \cdot (\dot{q} \dot{r}'_{l,i} \dot{q}^*) \end{aligned}$$

The first term on the last line is easy. In the second term, we can drop the multiplication by \dot{q} and \dot{q}^* , because rotation does not change the norm of a vector. Finally, the last term is what we wanted to maximize, in order to minimize the difference between the two vectors. The term can be developed to $\dot{q}^T N \dot{q}$, and we found that this is maximized for \dot{q} being the unit eigenvector corresponding to the largest eigenvalue, λ_{max} , of N . Therefore,

$$\sum_{i=1}^n (\dot{r}'_{r,i}) \cdot (\dot{q} \dot{r}'_{l,i} \dot{q}^*) = \dot{q}^T N \dot{q} = \dot{q}^T \lambda_{max} \dot{q} = \lambda_{max},$$

and

$$RMSD = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n \|\dot{r}'_{r,i}\|^2 + \sum_{i=1}^n \|\dot{r}'_{l,i}\|^2 - 2\lambda_{max} \right)}$$

This latter expression can be computed considerably faster than the former one, because there is no actual rotation involved, but it gives you nothing more than the RMSD value.

Run your command in the Matlab profiler:

In the Matlab window, choose **Desktop -> Profiler** to open the profiler window. Then, in the input line at the top you can enter any Matlab command. This command will then be executed normally, except that the profiler keeps track of the running time. Once the command exits, the profiler will show you a overview of your code and indicate costly operations.