Macromolecular Docking - project seminar

Peter Mühlbacher - a1253030

April 14, 2015

Contents

Abstract

Weekly reports on advances, encountered difficulties and implementations for the seminar on artificial intelligence will be gathered in this document.

Goals

In this seminar I aim to examine the problem of (flexible) protein-protein docking.

1 Week 1

1.1 The Original Goal

Originally I wanted to approach the problem of protein folding by investigating the asymptotic ($t \to \infty$) behaviour of the probability density function of a given protein. As the changes over time of a given protein[1] can be modelled by a Langevin equation $\dot{x} = -\nabla U(x) + \beta\dot{w}$, the corresponding probability density function $p(t, x)$ is described by the forward Fokker-Planck equation $\frac{\partial p}{\partial t} = \tilde{\beta}\Delta p + \operatorname{div}(p\nabla U)$ whose asymptotic behaviour has been investigated in Nadler et al. (2008).

However, there are (at least) two reasons why this does not work:

- In order to get a meaningful low dimensional representation of the system, a considerable margin between two adjacent eigenvalues $\lambda_{k+1} \gg \lambda_k$ of the Fokker-Planck operator is needed and it can be shown that this roughly corresponds to having a potential function U with k local minima[2]. Taking these properties into consideration it seems futile to employ this method of dimensionality reduction as the potential function in the area of protein folding has thousands of local minima.

- Even if solutions could be found explicitly in a reasonable amount of time the problem of finding and exact potential function U still remains unsolved[3]. As we are working with approximations it does not seem to make any sense trying to study its asymptotic long-term behaviour.

As a result of above considerations my topic of this seminar was restricted to docking. In the special case of rigid docking both molecules are assumed to be in a metastable state which drastically cuts the the complexity of the state space.

[1] understood as a single point in high dimensional space

[2] c.f. (Nadler et al., 2008, pp.9), p.9

[3] c.f. Neumaier (2006), p.14ff

2 Week 2

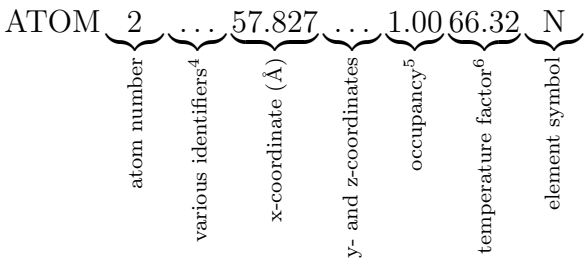
2.1 Benchmarks

Visit `http://zlab.umassmed.edu/benchmark/` for a benchmark on various docking problems that are ordered by difficulty. There are test cases for rigid docking as well as flexible docking, again ordered by amount of typically changing parameters like torsion angles.

2.2 PDB

The *Protein Data Bank Format* is a plain text data file, storing (amongst other information) the coordinates of every single atom in a protein.

A typical *PDB*-file's row might look like this:



For a list of PDB files for various proteins visit `http://www.rcsb.org/`.

why aren't footnotes showing?

2.3 Modelling the Potential

Given two metastable molecules $M_i = (\mathbf{x_i}, \alpha_i, \beta_i, \gamma_i, \{\mathbf{a_{ij}}\}_{j=1}^{n_i})$, $i \in \{1, 2\}$. In the following $\mathbf{x_i} \in \mathbb{R}^3$ will be called the molecule's (or the system's) point of reference, $\alpha_i, \beta_i, \gamma_i \in [0, 2\pi)$ the system's rotation (where α_i corresponds to the rotation around the x-axis relative to its point of reference; analogously for β_i, γ_i) and $\mathbf{a_{ij}} \in \mathbb{R}^3$ the system's atoms with (absolute) coordinates $\mathbf{x_i}+\mathbf{a_{ij}}$, where n_i is the number of atoms.

The Lennard-Jones potential of a system consisting of two points $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ with distance $r^2 = ||a - b||$ is defined as

$$V_{LJ}(r) = \varepsilon \left(\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^{6} \right), \tag{1}$$

where ε is the size of the potential well and σ the distance at which the potential reaches its minimum. Both are usually given by experiments.

In order to investigate the energy landscape for a docking problem it is natural to consider the Lennard-Jones potential for a system consisting of two proteins M_1, M_2 given by

$$\tilde{V}_{LJ}(M_1, M_2) = \sum_{i=1}^{n_1+n_2} \sum_{j>i}^{n_1+n_2} V_{LJ}(r_{ij}) \tag{2}$$

if we set r_{ij} to be the distance between the ith and jth atom[7].

If we want to explicitly calculate its gradient with respect to the free parameters $\mathbf{x_1}, \alpha_1, \beta_1, \gamma_1$ it is instructive to write down the entire formula as a function of those variables. First of all, to faciliate computation, we split up the summation into three parts $\tilde{V}_{LJ}(M_1, M_2) = \tilde{V}^1_{LJ}(M_1) + \tilde{V}^2_{LJ}(M_2) + \tilde{V}^3_{LJ}(M_1, M_2)$, where $\tilde{V}^1_{LJ}$ refers to all combinations of atoms of M_1, $\tilde{V}^2_{LJ}$ to those of M_2 and $\tilde{V}^3_{LJ}$ to the summation over all pairs $(a_{1i}, a_{2j})_{i\in\{1,\dots,n_1\}, j\in\{1,\dots,n_2\}}$. It is apparent that $\tilde{V}^1_{LJ}$ and $\tilde{V}^2_{LJ}$ do not change as a function of the arguments chosen above (however, they would in case we also introduced variable torsion angles and such). To calculate the gradient it thus suffices to inspect $\tilde{V}^3_{LJ}$ as a function of the free parameters $x, y, z, \alpha, \beta, \gamma$ describing the position of M_1:

$$\tilde{V}^3_{LJ}(\mathbf{x}_1, \alpha_1, \beta_1, \gamma_1) = \varepsilon \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left(\frac{\sigma}{r_{ij}}\right)^{12} - 2\left(\frac{\sigma}{r_{ij}}\right)^{6} \tag{3}$$

where $r^2_{ij} = \|\tilde{a}_{1i} - a_{2j}\|$ and $\tilde{a}_{1i}$ are the first system's ith particle's absolute coordinates given by

$$R_x(\alpha_1) R_y(\beta_1) R_z(\gamma_1) \mathbf{a}_{1i} + \mathbf{x}_1,$$

where

$$R_x(\alpha_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_1 & -\sin\alpha_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 \end{pmatrix},$$

$$R_y(\beta_1) = \begin{pmatrix} \cos\beta_1 & 0 & -\sin\beta_1 \\ 0 & 1 & 0 \\ \sin\beta_1 & 0 & \cos\beta_1 \end{pmatrix}$$

[7] $i, j \in \{1, \dots, n_1 + n_2\}$

and

$$R_z(\gamma_1) = \begin{pmatrix} \cos\gamma_1 & -\sin\gamma_1 & 0 \\ \sin\gamma_1 & \cos\gamma_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Now one can explicitly compute the partial derivatives $\frac{\partial \tilde{V}_{LJ}}{\partial p} = \frac{\partial \tilde{V}^3_{LJ}}{\partial p}$, for p being one of the parameters.

As of now we can compute the potential $\tilde{V}_{LJ}(\mathbf{X})$ for a given point $\mathbf{X} = (\mathbf{x}_1, \alpha_1, \beta_1, \gamma_1)$ in the state space and its gradient $\nabla\tilde{V}_{LJ}(\mathbf{X})$ and that is all we need for algorithms like steepest descent (which would still require rescaling of the parameters as to avoid extensive zig-zagging) or diffusion maps.

Furthermore, note that the computation of the potential and the gradient can be combined into one double loop, making[8] it $O\left((n_1+n_2)^2\right)$.

3 Week 3

3.1 Ordering Dihedral Angles by Variance

To be able to approximate the real docking process as closely as possible while only letting some chosen dihedral angles be variable, one first needs to order them accordingly. One such order might be given by regarding each such angle Θ_i as a discrete random variable and ordering them by their variances.

More precisely, a set of a single molecule's uniformly distributed[9] conformations $\{M_k\}_{k=0}^n$ induces a discrete probability space $(\{M_k\}_{k=0}^n, \mathscr{A}, P)$ where $P(M_k) := \frac{V_{LJ}(M_k)}{Z}$, where $Z := \sum_{k=1}^n P(M_k)$ is a normalising factor and for all A in $\mathscr{A}$: $P(A) = \sum_{M\in A} P(M)$.

For a fixed Θ_i we obtain the expected value $E[\Theta_i] = \sum_{k=1}^n P(M_k)\Theta_{ik}$, were Θ_{ik} is the k-th conformation's i-th dihedral angle.

The variance $\mathrm{Var}[\Theta_i]$ is given by $E[(\Theta_i - E[\Theta_i])^2]$.

Other ideas included:

- Looking if such a list already existed for some molecules, but I couldn't find any.

would it be more efficient to compute sample trajectories and just use them (regardless of their dis-

[8] although the computation of the gradient is likely to result in a big constant factor of operations

[9] This requirement is necessary because we do not want to count "very similar" conformations twice

- Taking different conformations from `rcsb.com` and using PCA on the phase space consisting of the dihedral angles, but given the number of dimensions compared to the amount of samples available[10] this is not a promising approach either.

- Take Ramachandran plots, interpret them as conditional probabilities and combine all of them. The data for this approach should be available, but combining all those single results may be hard.

3.2 Rotation About an Arbitrary Axis in 3D

To simulate flexible docking it is essential to be able to determine the single atoms' Cartesian coordinates, given one point of reference, the distance between two adjacent atoms, the angles of three consecutive atoms and the dihedral angles.

Given four atoms $\{a_i\}_{i=0}^3$, s.t. a_j and a_{j+1} are covalently bonded their dihedral angle is the smallest angle between the two planes π_0 and π_1 which are uniquely determined by $\{a_i\}_{i=0}^2$ and $\{a_i\}_{i=1}^3$, respectively.

Given $\{a_i\}_{i=0}^2$ one sets a_3 in $_0$, s.t. it has the prescribed distance as well as the angle between a_1, a_2 and a_3. Now we want to rotate a_3 about the line $a_2 + \overrightarrow{a_1 - a_2}$.

Clearly this is not a linear mapping unless $a_2 = 0$, but by introducing a forth dimension we can turn three-dimensional translations into linear mappings.

See `http://inside.mines.edu/fs_home/gmurray/ArbitraryAxisRotation/` for a derivation of the according formula.

References

Nadler, Boaz, Ronald R. Coifman, Ioannis G. Kevrekidis, Stéphane Lafon, and Mauro Maggioni (2008), "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems." *Society for Industrial and Applied Mathematics*, URL `http://www.wisdom.weizmann.ac.il/~nadler/Publications/diffusion_map_MMS.pdf`. Multiscale Model. Simul.

[10] For albumin there are 104 entries as of 14.04.2015.

Neumaier, Arnold (2006), "Molecular modeling of proteins and mathematical prediction of protein structure." URL http://www.mat.univie.ac.at/~neum/ms/protein.pdf.