EFFICIENT METHODS FOR STOCHASTIC SIMULATIONS OF BIOMOLECULE MOTIONS

TINGTING JIANG, JOHN REIF
Department of Computer Science,
Duke University, Box 90129,
Durham, NC27708-0129, USA
{*ruxu, reif* }*@cs.duke.edu*

Abstract

Classic techniques for simulating molecular motions, such as the Monte Carlo and molecular dynamics methods generate individual motion pathways one at a time. In contrast, the goal of stochastic simulation for biomolecule motion is to determine the limit probability distribution of the conformations of a molecule or a set of molecules. We make use of the known biomolecule modeling and also previously developed techniques in robot motion planning, in particular, *probabilistic roadmap methods* (PRM), to construct a roadmap in the conformation space of the molecules. By viewing the roadmap as a Markov chain, we can take advantage of powerful tools from the Markov chain theory and other algebraic methods to explore the kinetics of molecule motions. In this paper, we have developed new techniques which could significantly improve the performance of the simulation including: (a) techniques for partitioning the state space, e.g. Tensor product; (b) techniques for computing eigenvalues of a matrix and their associated eigenvectors iteratively. With these techniques, we propose a hybrid algorithm to compute the limit of the probability of the biomolecule's states more efficiently.

1 Introduction

Many biomolecule motions, such as protein folding are complicated and remain mysterious. So simulations of the biomolecule motions become more and more important. Currently there are two different approaches to simulating molecular motions: single motion pathway and multiple motion pathways. The first approach, including the Monte Carlo [1] or molecular dynamics [2], focuses on only one pathway at a time and has a higher density

of samples along the single particular way. In contrast, a new method called Stochastic Roadmap Simulation (SRS) which simultaneously examines multiple pathways instead of one pathway has appeared[3]. This method is based on PRM[4] and constructs a roadmap consisted of N sample nodes in the configuration space of the molecules (proteins). Based on the roadmap, they explore the kinetics of molecular motion efficiently. In the limit, SRS converges to the same distribution as Monte Carlo simulation. To compute some specific kinetic parameter, SRS method need go through all the N sample nodes in the roadmap and then solve the linear system of size N which is exponential of the number of degrees of freedom (DOFs) of biomolecules. However biomolecules, like proteins have hundreds of DOFs. Therefore, when N is very large, SRS is computationally inefficient due to the limitation of the linear system solvers.

In this paper, we proposed a hybrid algorithm in the SRS frame work to get the limiting probability distribution of the conformations of biomolecules. The main difference between our algorithm and SRS is that our algorithm only visit part of the N sample nodes instead of all the nodes. Hence, it can greatly save computation cost. Particularly, this algorithm focuses on the special case that a molecule could be decomposed of two weakly interacting parts both of which generate *rapidly mixing* Markov chains and then makes use of the Tensor product. The key property of rapidly mixing Markov chain is that it can converge to the limit stationary state quickly. Based on this property, the algorithm uses Monte Carlo method to get the limit probability distributions of the individual parts. Then it combines the two results together to get the limiting probability distribution of the whole molecule iteratively. The strength of this algorithm is that it avoids visiting all the sample nodes. Therefore, it is quite computationally efficient and could be improved by incorporating better iteration methods.

The rest of the paper will be organized as follow. Section 2 introduces the modeling methods for biomolecules. Section 3 gives an overview of rapidly mixing Markov chains. Section 4 shows how to calculate the limiting probability distribution for molecules. Section 5 propose the new hybrid algorithm. Section 6 summarize the main result and point out the future research directions.

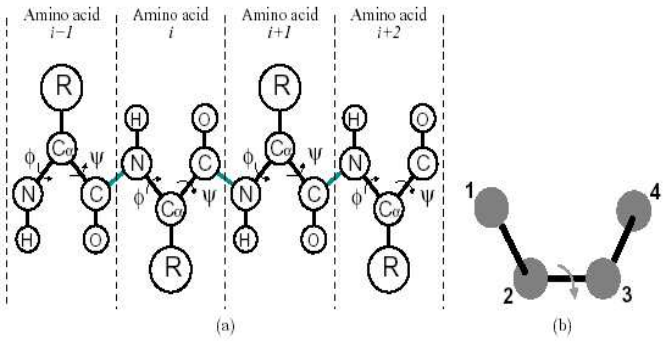

Figure 1: (a) Ramachadran model (R represents any side-chain) (b) A torsional DOF: it is the angle made by the two planes containing the centers of atoms 1, 2, and 3, and 2, 3, and 4, respectively.

2 Modeling

2.1 Conformation Space

For proteins, we use the off-lattice model where the backbone torsional angles are used [5]. The only DOFs in our model of the protein backbone are the Ramachadran torsional angels ϕ and φ [6](Fig. 1). For the sake of simplicity, all the side chains are modeled as spheres and have zero DOF. So the model for a k residue sequence will consist of 2k links and 2(k-1) revolute joints. (The two rotations at the ends don't contribute). Its DOFs is 2(k-1) corresponding to $2(k-1)$ ϕ and φ angles. In other words, the configuration space of a protein with $n+1$ amino acids can be formulated as [7]:

$$C = \{q | q \in \mathbf{S}^{2n}\}$$

We can make similar assumptions and simplifications for modeling other biomolecules, such as DNA and RNA. Besides geometric conformation restriction, the energy constraints are also required because molecular conformations usually have low energies. So this property must be included in the collision checker of PRM method. Now by modeling proteins as articulated robots and including the energy function in the collision checker, the problem of finding protein folding pathways has been transformed into a problem of robot path planning. So the same motion planning

algorithms including PRMs developed for robots can be obviously applied to molecules.

2.2 Node Generation

Based on the above off-lattice model, a given configuration $q \in C$ can be generated by assigning each torsion angle a value in its allowable range and determining the position and orientation of the root atom. Once these parameters are known, the coordinates of each atom in the protein can be calculated and then used to determine the potential energy of the conformation. In the following discussion, we will assume the sampling is uniform for simplicity. Once a node q is randomly generated by some sampling strategy, the probability of accepting it or not is based on its potential energy. If the potential energy $E(q)$ is high, then the probability of accepting this node is low. Otherwise, if its potential energy is small, it will be accepted with high probability. One way to define the acceptance criterion is as follows [5,7] :

$$\mathbf{P}(\text{p is accepted}) = \begin{cases} 1 \text{ if } E(q) < E_{min} \\ \frac{E_{max} - E(q)}{E_{max} - E_{min}} \text{ if } E_{min} \leq E(q) \leq E_{max} \\ 0 \text{ if } E(q) > E_{max} \end{cases}$$

The two thresholds E_{max} and E_{min} can be empirically set.

2.3 Roadmap Construction

Assume we have N sample conformations now. For every node v_i, we find v_i's nearest neighbors according to a reasonable metric such as Euclidean or RMS distance in the conformation space C and connect them to v_i. For every pair of neighboring nodes v_i and v_j, the energy difference is denoted by $\triangle E_{ij} = E(v_j) - E(v_i)$ and the transition probability M_{ij} from v_i to v_j is defined as

$$\mathbf{M_{ij}} = \begin{cases} \frac{1}{d_j} exp(-\Delta E_{ij}/k_B T) & \text{if } \frac{\epsilon_j/d_j}{\epsilon_i/d_i} < 1; \\ \frac{1}{d_i} & \text{otherwise;} \end{cases}$$

where ϵ_i and ϵ_j are the Boltzmann factors at v_i and v_j, and d_i and d_j are the number of neighbors of v_i and v_j. The Boltzmann factor at v_i can be expressed as $\epsilon_i = exp(-E_i/k_B T)$. To make

the sum of the transition probability for each node equal to 1, we need define the self-transition probability:

$$M_{ii} = 1 - \Sigma_{j \neq i} M_{ij}.$$

Now for each node v_i in the roadmap, it has d_i adjacent edges. For each of these edges, there is an associated probability. Actually, we have determined the transition probabilities from the node to its k neighbors and the self-transition probability. Assume there are N nodes in the roadmap in total, we can get an $N \times N$ probability transition matrix.

2.4 Energy Calculation

To simplify the energy computation, we can make some assumptions. First, we could only consider the van der Waals energy and neglect the electrostatic energy and torsional energy which are comparatively small. Second, for those pairs of atoms that are far apart, the contribution to the total energy is trivial. So we just consider the atoms that are close together because they contribute largely to the total energy. By these simplifications, we can define the potential energy for a specific protein configuration:

$$E_{total} = \sum_{\text{close atom pairs}} (A/r_{ij}^{12} - B/r_{ij}^{6}).$$

Here the meaning of "close" can be determined by a cut-off distance. A and B are constants. E_{total} defined above is only a rough approximation about the real potential energy. Actually we can adopt more precise approximation if we need.

3 Rapidly Mixing Markov Chain Techniques

Once we construct the roadmap G and define the transition probabilities for any two adjacent nodes in the roadmap, a Markov chain is generated from the roadmap. Moreover, if the roadmap G contains only one strongly-connected component, then the Markov chain represented by G is *ergodic* [8]i.e. the final stationary distribution of the Markov chain exists. However, we are also interested in the mixing time from the starting distribution to the final stationary distribution. If a Markov chain can quickly converge to its stationary distribution regardless of the starting state,

we refer such chains as *rapidly mixing*. Rapidly mixing is a high non-trivial requirement because the number of states in the chain is exponentially large and we hope it converges after visiting a tiny fraction of the state space. Rapidly mixing Markov chains is a well-established method to compute the limit probability of ergodic Markov chain. It has been used for approximately solving some very hard $\sharp P$-complete problems such as counting problems for objects of size d, estimating the volume of convex polytope in d and random walk in a d-dimensional hypercubes. In those cases, the number of conformation states N is exponential in the parameter of d, yet some algorithms have been designed with which the time costs are polynomial functions of d . So rapidly mixing Markov chain has become a very important tool in approximation algorithms.

4 Calculating Limiting Probability Distribution for Molecules with NonInteracting or Weakly Interacting Parts

If a molecule is very large, it is possible that there are several parts whose motions are independent or almost independent if they are far away enough. We are interested in this special property and try to exploit it in our calculation of the limit probability of the molecule. Tensor product is a very useful technique to deal with independent sets and partition the state space. First we consider a very simple case that all parts are independent each other. In this case, we get a nice property that the larger Markov chain for the whole molecule is just the Tensor product of the smaller Markov chain for each independent part. Then we generalize the result further to more complicated cases by divide-conquer method.

4.1 Tensor Product

Let $R = (\rho_{ij}), 1 \leq i, j \leq r$ and $S = (\sigma_{ij}),\ 1 \leq i, j \leq s$, be $r \times r$ and $s \times s$ matrices, respectively. The *Tensor Product* of R and S is the $rs \times rs$ matrix

$$\mathbf{R} \bigotimes \mathbf{S} = \begin{pmatrix} \rho_{11}S & \rho_{12}S & \dots & \rho_{1,r}S \\ \rho_{21}S & \rho_{22}S & \dots & \rho_{2,r}S \\ \vdots & \vdots & \dots & \vdots \\ \rho_{r,1}S & \rho_{r,2}S & \dots & \rho_{r,r}S \end{pmatrix}$$

Several key properties of Tensor products are listed below [9]:

Lemma 1:

- $A \bigotimes B \bigotimes C = A \bigotimes (B \bigotimes C) = (A \bigotimes B) \bigotimes C$
- $(A \bigotimes B)(C \bigotimes D) = (AC) \bigotimes (BD)$; assume that the ordinary multiplications AC and BD are defined.

4.2 Molecules with NonInteracting Parts

Suppose there are two atoms A and B which are connected by a bond. For each atom, there are two possible states. A could be at state A_1 or A_2. B could be at state B_1 or B_2. Furthermore, we assume the two atoms' state transitions are independent, i.e. the probability for atom A to transform from A_1 to A_2 is independent from the probability for atom B to transform from B_1 to B_2, etc. Therefore we have two separate transition matrices for the two atoms. The 2×2 transition matrix M_A denotes the transition probabilities for atom A. And M_B denotes the transition probabilities for atom B. Specifically a_{ij} represents the probability for atom A from state i to j. The same for b_{ij}.

$$\mathbf{M_A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \mathbf{M_B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

Now if we consider the two atoms together as a molecule, it has four possible states, say 11, 12, 21, 22. The first number denotes the state of A and the second one denotes the state of B. Since A and B transform independently, the probability from $i_1 j_1$ to $i_2 j_2$ is the product of $a_{i_1 i_2}$ and $b_{b_1 b_2}$. So we can get a 4×4 transition matrix M for the molecule as follows:

$$\mathbf{M} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix}$$

Actually, $M = M_A \bigotimes M_B$.

If both atoms have stationary probability distributions, say $\pi_A = (p_1, p_2)$ and $\pi_B = (q_1, q_2)$. p_i represents the final probability for A staying at A_i and q_i represents the final probability for B staying at B_i. Since the stationary probability distribution of A is π_A, there is a smallest k_1 such that $M_A^k = (\pi_A, \pi_A)^T$ for $k \geq k_1$.

Similarly, there is a smallest k_2 such that $M_B^k = (\pi_B, \pi_B)^T$ for $k \geq k_2$. Assume $k_0 = max\{k_1, k_2\}$. Then $M_A^k = (\pi_A, \pi_A)^T$ and $M_B^k = (\pi_B, \pi_B)^T$ for $k \geq k_0$. According to Lemma 1, we have

$$M^k = M_A^k \bigotimes M_B^k = (\pi_A, \pi_A)^T \bigotimes (\pi_B, \pi_B)^T \text{ for } k \geq k_0$$

So the molecule does have a stationary probability distribution π and the final stationary transition matrix is just the Tensor product of the final stationary transition matrices of A and B. It is easy to show that $\pi = (p_1q_1, p_1q_2, p_2q_1, p_2q_2)$. More generally, assume the size of A is $r \times r$ and the size of B is $s \times s$. And the π_A is a r-dimensional vector, π_B is an s-dimensional vector. Then the limit probability distribution π for M is a $(r \times s)$-dimensional vector. We can prove that $\pi^{(k)} = {\pi_A}^{(i)} \times {\pi_B}^{(j)}$ where $k = s(i-1)+j$, $(1 \leq i \leq r, 1 \leq j \leq s)$ and $\pi^{(k)}$ denotes the kth component of the vector π.

The above result can be easily generalized to the case of more than two independent parts. Assume there are $n(n \geq 2)$ independent parts in a bio-molecule. Furthermore, it could be generalized to the cases of the molecules with weakly interacting parts.

4.3 Molecules with Weakly Interacting Parts

Suppose there is a long chain of molecules, it can be divided into several parts that weakly interact each other. In this case, we can calculate the limit probability by divide and conquer method.

First we assume there are only two parts A and B which are not independent in a molecule C. Both M_A and M_B are rapidly mixing and the interaction between A and B is weak. Let $M_C = M_A \bigotimes M_B$ and $\tilde{M_C}$ be the transition matrix for C. Since the interaction between A and B is weak, we could imagine the $\tilde{M_C}$ is just a perturbation of M_C. Let $\tilde{M_C} = M_C + E$.

Since M_A and M_B are rapidly mixing, by Monte Carlo method, we can get the limit probability distribution of A and B in polynomial time of their number of DOFs. The time cost is logarithmic of N. Then, based on the limits of subsystems A and B, we can easily calculate π_C, the limit probability distribution of C. Since $\tilde{M_C} = M_C + E$ and E is very small, we can use iterative method to get the limit of $\tilde{M_C}$. This idea implies a hybrid algorithm to solve such kind of systems in the next section.

5 Hybrid Algorithm

Based on the above arguments, now we propose a hybrid algorithm to calculate the limit probability distribution for the kind of biomolecules with two weakly interacting parts both of which are rapidly mixing. The algorithm is hybrid because it solves the subsystems by Monte Carlo method and then based on the results of subsystems it iteratively calculates the limit probability distribution of the whole system by some numerical methods, such as Davidson's method [10].

Algorithm LimitProbability (C)
INPUT: C is a biomolecule composed of two weakly interacting parts A and B both of which are rapidly mixing;
OUTPUT: The limit probability distribution of C.

BEGIN

1: **for** A and B **do**
2: (i) Node Generation
3: (ii) Roadmap Construction
4: (iii) Calculate the limit probability distribution by Monte Carlo method
5: Based on the composition of the limit probability distributions of A and B, select prospective conformation states of C and construct a transition matrix for C which is much smaller than NxN;
6: Start from the composition of the limit probability distributions of A and B, iteratively calculate the limit probability distribution of C until the difference between the two continuous results is small enough.

END

In the first step of the above algorithm, it avoids constructing transition matrix M_A and M_B explicitly which could take exponential time in the dimension d. In contrast, the first step can be finished in polynomial time because both M_A and M_B are rapidly mixing. As for the second step, based on the results from the first step, we have the prefigure for the stationary distribution for C and therefore can dramatically reduce the size of the transition matrix. The last step makes use of iterative method to approx-

imate the eigenvector corresponding to the largest eigenvalue of M_C which is just 1. It is easy to see this eigenvector is just the limit probability distribution of C. In a word, this algorithm can obtain the limit probability distribution of C much faster than the previous methods.

6 Conclusion and Future Work

In this paper, we have presented a hybrid algorithm in the SRS frame work to get the limiting probability distribution of the conformations of biomolecules. The key idea of this algorithm is divide and conquer. Specifically, we considered the special case that a molecule could be decomposed of two weakly interacting parts both of which generate rapidly mixing Markov chains and therefore made use of the Tensor product. Based on the decomposition, we adopted Monte Carlo method to get the limit probability distribution of the individual parts and then combined them together to get the limiting probability distribution of the whole molecule iteratively. The strength of this algorithm is that it could greatly reduce the computation cost because it makes good use of special properties of rapidly mixing and weakly interaction. Therefore, this method is quite efficient and could be improved by incorporating better iteration methods. We would like to apply this algorithm to some specific examples. However, several interesting issues require further exploration.

The first step of the implementation is to find a suitable data set. Then we could borrow and modify some available software, for instance, the software which has been released by Stanford group recently to do the node generation and roadmap construction. Third, we will implement a good numerical method to iteratively get the limit probability distribution of the biomolecule.

We would also like to further develop the sampling strategy. Currently we use the uniform sampling method. It is not always a good choice because the number of degrees of freedom in molecule problem is high. Simple uniform sampling would take too long to provide sufficiently dense coverage of the conformation space and the number of sample states N would be very large. We could try some of biased sampling strategies which have been applied successfully in robotics applications [11,12] to make N smaller. If we use non-uniform sampling strategy, how to adjust the definition

of transition probabilities appropriately? Another motivation to use nonuniform sampling strategy is that we could possibly lose the conformation states in the free space if we use uniform sampling strategy. Because in that case, N is very large and in the stationary distribution π, the value of components corresponding to the conformation states in the free space might be very small. Thus we might lose those information because of the trivial value. So we might need some sampling techniques to get around of this dilemma.

Acknowledgments.

This work is partially supported by DARPA/AFSOR F30602-01-2-0561, NSF ITR EIA-0086015, NSF EIA-0218376, and NSF EIA-0218359. We thank Dr. Xiaobai Sun, M. Serkan Apaydin, Dr. David Richardson, Dr. Jane Richardson and all the other people who have given kindly suggestions and help to this paper.

References

1. M. Kalos and P. Whitlock. *Monte Carlo Methods.* John Wiley and Sons, New York, 1986.
2. J. Haile. *Molecular Dynamics Simulation: Elementary Methods.* John Wiley and Sons, New York, 1992.
3. Apaydin, Brutlag, Guestrin, Hsu, and Latombe. In *Annual International Conference on (Research in) Computational (Molecular) Biology*, volume 6, 2002.
4. L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Technical Report CS-TR-94-1519, Stanford University, Department of Computer Science, Aug. 1994.
5. G. Song and N. M. Amato. In T. Lengauer, D. Sankoff, S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the Fith International Conference on Computational Biology (RECOMB-01)*, pages 287–296, New York, Apr. 22–25 2001. ACMPress.
6. G. Ramachandran and V. Sasisekharan. *Adv. Protein Chem*, 23:283–437, 1968.
7. N. M. Amato, K. A. Dill, and G. Song. In *Proceedings the 6th International Conference on Computational Molecular Biology (RECOMB)*, Apr. 2002.

8. H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, New York, 1994. 3rd edition.
9. J. Johnson, R. Johnson, D. Rodriguez, and R. Tolimieri. 1990.
10. E. R. Davidson. *Journal of Computational Physics*, 17:87–94, 1975.
11. N. Amato, O. Bayazit, L. Dale, C. Jones, and D. Vallejo. 1998.
12. V. Boor, M. Overmars, and A. van der Stappen. In *In Proc. of IEEE Int. Conf. Robotics and Automation*, Detroit, MI, 1999.