Vol.2011-BIO-26 No.5
2011/9/13

Distance-based Graph Linearization and Sampled Max-sum Algorithm for Efficient 3D Potential Decoding of Macromolecules

Takahiro Shinozaki,[†1,†2] Toshinao Iwaki,[†2] Shiqiao Du,[†2]
Masakazu Sekijima [†2] and Sadaoki Furui [†2]

Three-dimensional structure prediction of a molecule can be modeled as a minimum energy search problem in a potential landscape. Popular ab initio structure prediction approaches based on this formalization are the Monte Carlo methods represented by the Metropolis method. However, their prediction performance degrades for larger molecules such as proteins since the search space is exponential to the number of atoms. In order to search the exponential space more efficiently, we propose a new method modeling the potential landscape as a factor graph. The key ideas are slicing the factor graph based on the maximum distance of bonded atoms to convert it to a linear structured graph, and the utilization of the max-sum search algorithm combined with samplings. It is referred to as Slice Chain Max-Sum and it has an advantage that the search is efficient because the graph is linear. Experiments are performed using polypeptides having 50 to 300 amino acid residues. It has been shown that the proposed method is computationally more efficient than the Metropolis method for large molecules.

1. Introduction

Molecular structure prediction has been a popular topic of research among biologists, physicists, chemists, computer scientists, and researchers from many other fields. Knowing the tertiary structure of macromolecules such as protein is a key in understanding the function of them. There have been many approaches for structure prediction of macromolecules. Among them, molecular dynamics (MD)[1)] and Monte Carlo methods[2)] are two main approaches for ab initio tertiary structure prediction that does not require previously solved structures of similar molecules.

MD is a method of predicting motions of molecules by using Newtonian physics. It tracks down atom movements by calculating the numerical integrations over short time-steps, usually in the order of femtoseconds ($1\,\mathrm{fs} = 10^{-15}\,\mathrm{s}$). A disadvantage of MD is that the amount of calculation is too large for macromolecules that take more than milliseconds to reach its thermodynamic equilibrium.

On the other hand, the Monte Carlo-based methods are based on statistical and thermal physics. In the canonical ensemble condition[3)], where the number of atoms in a system, N, the volume, and the temperature, T, are constant, the probability distribution of the system state given by $3N$-dimensional momenta vector $\mathbf{p}$ and $3N$-dimensional position vector $\mathbf{r}$ can be written as:

$$P(\mathbf{p},\mathbf{r}) = \frac{1}{Z}\exp\left(-\frac{\mathscr{H}(\mathbf{p},\mathbf{r})}{k_B T}\right), \tag{1}$$

$$\mathscr{H}(\mathbf{p},\mathbf{r}) = \sum_{i=1}^{M}\frac{\left|\mathbf{p}_i^2\right|}{2m} + V(\mathbf{r}), \tag{2}$$

where k_B is the Boltzmann constant, m is the mass of an atom, $V(\mathbf{r})$ is the potential function, $\mathscr{H}(\mathbf{p},\mathbf{r})$ is the Hamiltonian that corresponds to the total energy of the system, and Z is a normalization constant. By substituting Equation (2) into Equation (1), we achieve:

$$P(\mathbf{p},\mathbf{r}) = \frac{1}{Z}\exp\left[-\frac{\left|\mathbf{p}^2\right|}{2mk_B T}\right]\exp\left[-\frac{V(\mathbf{r})}{k_B T}\right]. \tag{3}$$

In Equation (3), the exponential term is separated into a product of two exponential terms of momenta and positions. Thus, the probability $P(\mathbf{r})$ that a molecule takes a structure $\mathbf{r}$ is expressed as shown in Equation 4.

$$P(\mathbf{r}) = \frac{1}{Z_{\mathbf{r}}}\exp\left[-\frac{V(\mathbf{r})}{k_B T}\right], \tag{4}$$

where $Z_{\mathbf{r}}$ is a normalization constant. That is, the probability that a molecule takes a particular structure is expressed by its potential energy. The structure that has the highest probability is the one with the lowest potential energy. Therefore, the structure prediction problem is reduced to a search problem of minimum potential energy.

Given the information about bonds and necessary coefficients, as well as coordinates of each atom, potential energy $V(\mathbf{r})$ of a molecule can be calculated using the following

†1 Chiba University
†2 Tokyo Institute of Technology

equation[4]:

$$V(\mathbf{r}) = \sum_{b\in\mathcal{B}} k_b \left(d_b^{eq} - d_b\right)^2 + \sum_{a\in\mathcal{A}} k_a \left(\theta_a^{eq} - \theta_a\right)^2 + \sum_{d\in\mathcal{D}} k_d \left(1 + \cos[n\phi_d - \delta_d]\right) + \sum_{\mathcal{F}} \left\{ \left(\frac{A_{ij}}{r_{ij}^{12}}\right) + \left(\frac{B_{ij}}{r_{ij}^{6}}\right) + \left(\frac{q_i q_j}{\epsilon r_{ij}}\right) \right\}, \quad (5)$$

where $\mathcal{B}$, $\mathcal{A}$, $\mathcal{D}$, and $\mathcal{F}$ are sets of bonds, bond angles, dihedral angles, and non-bonded atom pairs. The terms regarding $\mathcal{B}$, $\mathcal{A}$, and $\mathcal{D}$ are functions of bond length d_b, bound angle θ_a, and dihedral angle ϕ_d, respectively. The constants k_b, d_b^{eq}, k_a, θ_a^{eq}, k_d, δ_d are their parameters. For the non-bonded terms, $r_{i,j}$ is a distance between an atom pair, q_i is a constant that depends on an atom, and $A_{i,j}$ and $B_{i,j}$ are constants that depend on an atom pair, and ϵ is a dielectric constant.

The challenge of the search is to find a global minimum in the potential landscape that has many local optima. The Monte Carlo approach searches the minimum by randomly generating candidate structures or samples $\mathbf{r_t}$ and evaluating their energy $V(\mathbf{r_t})$. If a uniform distribution is used to generate the samples, then the number of samples required to find a good solution increases quickly for the molecular sizes since the search space is exponential to the number of atoms, and the search will fail even for a moderate-size molecule.

The Metropolis method[5] generates a sequence of samples according to the probability distribution that is tied to the potential energy by Equation (4). In the method, a candidate sample is generated based on a current state. The candidate is always accepted as the next state if it decreases the energy. In addition, it is accepted with a certain chance even if it increases the energy. With this procedure, the search is efficient since it puts priority on low energy regions while maintaining the ability to escape from local minima. However, it still suffers from the exponential increase of the search space and does not work well for macromolecules consisting of thousands of atoms[6].

In this paper, we propose a new method named Slice Chain Max-Sum (SCMS) that is based on modeling the potential landscape by a factor graph. A factor graph that represents a potential of a molecule has many cycles. With the cycles, there is no efficient search algorithm that is guaranteed to converge. Therefore, we convert it to a linear structured graph by aggregating the factors. In general, such conversion is computationally not tractable for a large graph[7]. For this problem, we propose an efficient conversion algorithm that is based on the maximum distance between bonded atoms of the underlying molecule. Given the linear structured graph, a dynamic programming based efficient max-sum search algorithm[8] is applied in combination with samplings of candidate atom positions at each node.

The organization of this paper is as follows. In Section 2, some basics of factor graphs are reviewed. In Section 3, the proposed method is described. Experimental conditions are described in Section 4 and the results are shown in Section 5. Finally, conclusions and future works are given in Section 6.

2. Factor Graph and Related Algorithms

2.1 Factor Graph

A factor graph[9] G is a bipartite graph to represent a decomposed structure of a function that can be expressed by a product of component functions or factors as shown in Equation 6.

$$g(X_1, X_2, \cdots, X_N) = \prod_{i=1}^{M} f_i(C_i), \quad (6)$$

where X_n is a variable and $C_i \in \{X_1, X_2, \cdots, X_N\}$ is a set of the variables. The factor graph corresponding to Equation 6 consists of variable nodes $X = \{X_1, X_2, \cdots, X_N\}$, factor nodes $F = \{f_1, f_2, \cdots, f_M\}$, and arcs E. A value of a factor node f_i is determined by the nodes that correspond to the elements of C_i. To represent this relationship, a factor node f_i and a variable node X_n are connected by an undirected arc if $X_n \in C_i$. The value of a factor graph is the product of the values of all factors. For example, a function shown in Equation (7) can be represented by a factor graph shown in Figure 1.

$$g(X_1, X_2, X_3, X_4, X_5) = f_1(X_1, X_2) + f_2(X_2, X_3, X_5) + f_3(X_2, X_3) + f_4(X_3, X_4) \quad (7)$$

By taking a logarithm, Equation (6) has a form shown in Equation (8).

$$g'(X_1, X_2, \cdots, X_N) = \sum_{i=1}^{M} f_i'(C_i). \quad (8)$$

A factor graph can also be used to represent this summation based decomposed structure. The logarithmic operation does not affect the structure of the factor graph.

2.2 Factor Graph Representation of a Molecular Structure

The potential $V(\mathbf{r})$ of a molecule expressed by Equation (5) can be represented by a factor graph since it has the decomposed form. The factor graph abstracts the dependency structure between the atom positions and the potentials omitting the detailed functional forms.

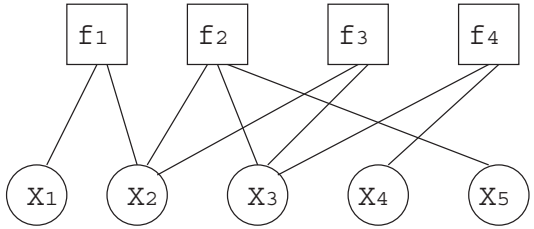


Fig. 1 An example of a factor graph.

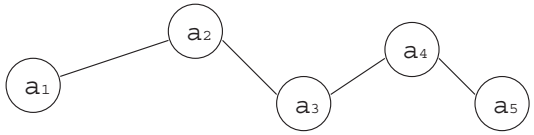


Fig. 2 A molecule consisting of five atoms.

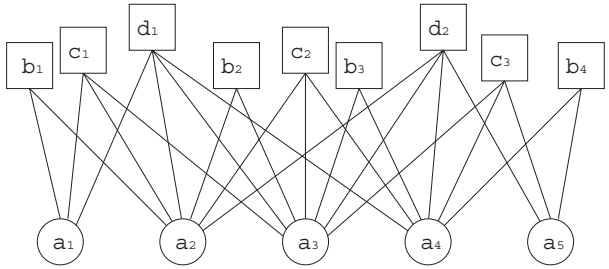


Fig. 3 A factor graph that represent the potential energy of the molecule in Figure 2. The variable a_i denotes 3-dimensional Cartesian coordinate of an atom.

For example, if a potential of a molecule shown in Figure 2 consisting of five atoms a_1 to a_5 is modeled by a set of bond length potentials $d_1(a_1, a_2)$, $d_2(a_2, a_3)$, $d_3(a_2, a_3)$, $d_4(a_2, a_3)$, bond angle potentials $c_1(a_1, a_2, a_3)$, $c_2(a_2, a_3, a_4)$, $c_3(a_4, a_4, a_5)$, and dihedral potentials $d_1(a_1, a_2, a_3, a_4)$, $d_2(a_2, a_3, a_4, a_5)$, then its factor graph is represented as shown in Figure 3.

2.3 Max-sum algorithm

When the variables X_j of a factor graph G are all discrete, a configuration of the variables that gives the global maximum (or minimum) of the factor graph is found by

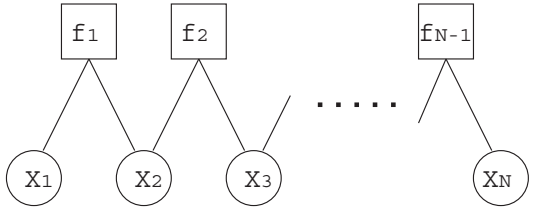


Fig. 4 A factor graph that has a linear structure.

enumerating and evaluating all the combinations of their values. However, the number of combinations of the values are exponential to the number of variables, and this straightforward approach does not work for a large graph.

Fortunately, there exist efficient algorithms to find the global maximum when the factor graph does not have cycles. Max-product and max-sum are such algorithms. The max-product is used for the product decomposition as in Equation 6, and the max-sum is used for the summation decomposition as in Equation 8. Their procedures are basically the same and only the difference is that product operations in max-product are replaced with summations in max-sum. A special case of a factor graph that does not have cycles is a linear structured graph as shown in Figure 4. Here, as the simplest case, the max-sum algorithm is explained when the graph is linear.

The principle of the max-sum algorithm is to utilize the independence structure of a graph. When the graph is linear, this is simply done by pushing the max operation to the right in the corresponding equation as shown in Equation (9).

$$\begin{aligned}
&\max_{X_N, X_{N-1} \cdots, X_2, X_1} g(X_1, X_2, \cdots, X_{N-1}, X_N) \\
&= \max_{X_N, X_{N-1} \cdots, X_2, X_1} \{ f_{N-1}(X_N, X_{N-1}) + \cdots + f_2(X_3, X_2) + f_1(X_2, X_1) \} \\
&= \max_{X_N, X_{N-1}} \left\{ f_{N-1}(X_N, X_{N-1}) + \cdots + \max_{X_2} \left\{ f_2(X_2, X_3) + \max_{X_1} \{ f_1(X_1, X_2) \} \right\} \right\}. \qquad (9)
\end{aligned}$$

To visualize the search process, a graph called lattice shown in Figure 5 is used. The horizontal axis of the graph is the variables and the vertical axis is their values. In the figure, all the variables are assumed to have K values for simplicity. Let $L_{n,k}$ be a lattice node corresponding to k-th value of X_n, and let $x_{n,k}$ be its value. First, for each lattice node $L_{2,k}$ corresponding to X_2, $\max_j \left\{ f_1\left(x_{1,j}, x_{2,k}\right) \right\}$ is calculated. Let $acc_{2,k}$ be the maximum score and let $marc_{2,k}$ be the arc that corresponds to it. They are recorded at

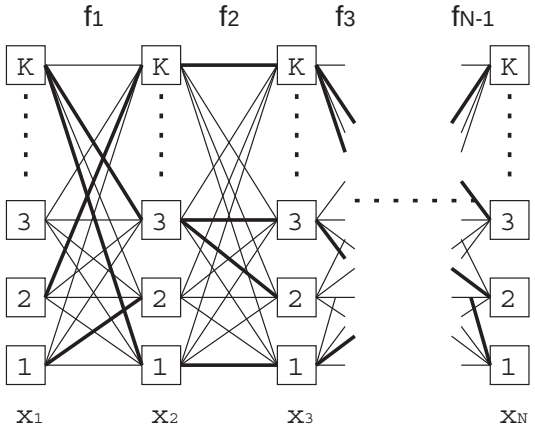


Fig. 5 Lattice is used to find the maximum path in a linear structured graph.

the lattice node $L_{2,k}$. Next, after $acc_{n,k}$ and $marc_{n,k}$ are calculated for all $k \in \{1, 2, \cdots, K\}$ at $(n-1)$-th step where $n = 2, 3, \cdots, N-1$, $\max_j \left\{acc_{n,j} + f_n\left(x_{n,j}, x_{n+1,k}\right)\right\}$ is calculated for each k, and the maximum score and the corresponding arc are recorded at $L_{n+1,k}$. The search proceeds from the left hand side to the right of the lattice.

After all the lattice nodes $L_{n,k}$ are evaluated, the maximum score of the factor graph is obtained by $max_k\,(acc_{N,k})$. The variable configuration that gives the maximum is obtained by backtracking the lattice nodes from the maximum node at the right hand side to the left following the arcs stored at each lattice node. The computational cost of the former left-to-right procedure is $O\left(K^2N\right)$ and the cost for the backtracking is $O\,(N)$. Therefore, the cost of max-sum is linear to N, which is much lower than the exponential cost $O\left(K^N\right)$ when all the combinations of the values are enumerated without using the graph structure. The minimum of the graph can be obtained by simply negating all the factors and applying the max-sum algorithm.

2.4 Metropolis sampling

The Metropolis sampling is one of the representatives of the Markov Chain Monte Carlo (MCMC) method that can generate samples following an arbitrary probability distribution $G\,(X)$[5]. In the algorithm, first a candidate of a sample X^* is generated from a proposal distribution $q\,(X|X\langle t-1\rangle)$ given an initial state $X\langle t-1\rangle = \{X_1\langle t-1\rangle, X_2\langle t-1\rangle, \cdots, X_N\langle t-1\rangle\}$. At the very beginning, arbitrary chosen seed value $X\langle 0\rangle$ is used as the initial state. Any distribution that satisfies the symmetric constraint $q\,(Y|X) = q\,(X|Y)$ can be used as the proposal distribution, as far as deriving samples from the distribution is easy. Then an acceptance ratio shown in Equation (10) is calculated at each step $t \in \{1, 2, \cdots\}$, and the candidate is accepted with that ratio.

$$Arate = \min\left\{1, \frac{G\,(X^*)}{G\,(X\langle t\rangle)}\right\}. \tag{10}$$

This is done by getting a value *rnd* from a uniform distribution ranging from 0.0 to 1.0, and accepting the candidate if $rnd \leq Arate$. When the candidate is accepted as the t-th sample $X\langle t\rangle = X^*$, it is used as an initial state of the next step. When it is rejected, the value of the previous sample is copied $X\langle t\rangle = X\langle t-1\rangle$. When the sequence $X\langle 0\rangle, X\langle 1\rangle, \cdots X\langle T\rangle$ is long enough, the distribution of $X\langle t\rangle$ approaches $G\,(X)$. Since the correlations between adjacent samples are high, only every I-th sample is retained when independent samples are required, where I is a sufficiently large number.

By introducing a normalization term, the function g shown in Equation (6) can be regarded as a joint probability distribution $G\,(X_1, X_2, \cdots, X_{N-1}, X_N)$ as shown in Equation 11. Therefore, the Metropolis sampling can be applied to derive samples from the factor graph. Although, the Metropolis sampling does not utilize the decomposed structure of the factor graph.

$$G\,(X_1, X_2, \cdots, X_n) = \frac{1}{Z}\prod_{i=1}^{m} f_i\left(C_j\right). \tag{11}$$

In the sampling process, the normalization term does not affect the result and can be omitted since it is a constant. In the following of this paper, the Metropolis method is simply referred to as MCMC.

3. Proposed Method

3.1 Basic ideas and assumptions

The basic idea of the proposed slice chain max-sum (SCMS) method is to apply the max-sum algorithm to find the minimum potential energy structure of a molecule modeled as a factor graph. However, the inconvenient characteristics about molecular structure prediction are that the factor graph contains cycles and the positions of atoms are continuous values. In order to deal with these problems, SCMS first convert the factor graph to a linear graph and utilize sampling to discretize the positions. For the simplicity of our experiments, it is assumed that the potential energy V of a molecule is based

Vol.2011-BIO-26 No.5
2011/9/13

Step 1: Represent potential of a molecule as a factor graph having cycles. Initialize atom coordinates.
Step 2: Slice the molecule in 3D space by parallel planes with an interval equal to three times of maximum bond length.
Step 3: For each slice, aggregate variable nodes of the factor graph that correspond to the atoms in the slice into a single composite node S_m.
Step 4: Aggregate factor nodes that only depend on S_m and S_{m+1} into a single composite factor node F_m. If an original factor only depends on S_m, it is merged to either F_{m-1} or F_m. This makes a linear structured factor graph.
Step 5: In each composite node S_m, sample candidate positions of atoms according to the potential fixing positions of atoms in other slices. The samples are regarded as possible values of the composite node.
Step 6: Apply max-sum on the linear factor graph to find a minimum energy atom configuration.
Step 7: Output the configuration after enough iterations or go to step 2.

Fig. 6 Procedure of proposed SCMS.

solely on factors representing bond length, bond angle, and dihedral angle potentials as follows.

$$V(\mathbf{r}) = \sum_{b\in\mathcal{B}} k_b \left(d_b^{eq} - d_b\right)^2 + \sum_{a\in\mathcal{A}} k_a \left(\theta_a^{eq} - \theta_a\right)^2 + \sum_{d\in\mathcal{D}} k_d \left(1 + \cos[n\phi_d - \delta_d]\right). \quad (12)$$

3.2 Procedure of SCMS

Figure 6 describes the procedure of the proposed SCMS. First, a factor graph that represent a molecule is constructed based on a topology of the molecule, and a configuration of all atom's coordinates are initialized. The factor graph consists of variable nodes that represent atom positions and factor nodes regarding bond lengths, bond angles, and dihedrals, and it contains many cycles. In the factor graph, factors regarding a bond length span two atoms. Similarly, factors regarding a bond angle span three atoms, and factors regarding dihedral span four atoms. Therefor, factors of the graph span at most four atoms.

Then, in step 2, the molecule in 3D Cartesian space is sliced by parallel planes with an interval w as shown in Figure 7. The interval is chosen to three times of the maximum bond length d_{max} of the molecule as shown in Equation 13.

$$w = 3d_{max} + \epsilon, \quad (13)$$

where ϵ is a small positive value. Ideally, the direction of the slicing is chosen so as

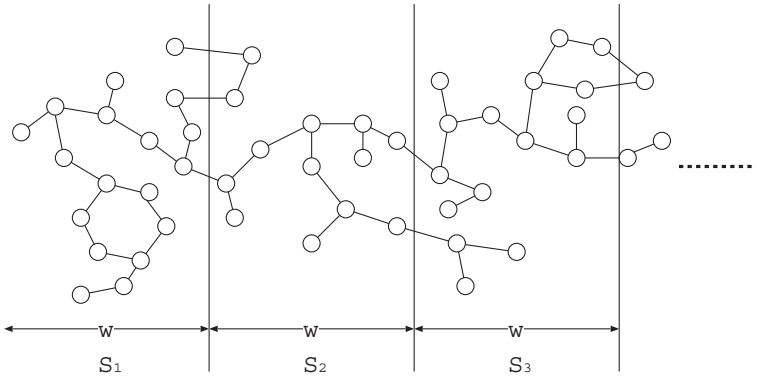


Fig. 7 Slicing a molecule by parallel planes with an interval w.

to maximizes the number of sliced segments. This is done by finding the longest axis of the molecule and arranging the planes to form a right angle to it. In our experiment, however, we simply slice the molecule with respect to the x, y, or z-axis, depending on which direction the molecule is the longest.

The slicing divides the molecule into multiple segments and the variable nodes of the factor graph are grouped according to the segments. In step 3, the nodes in the same group are aggregated to form a single composite variable node S_m for $m = 1, 2, \cdots, M$, where M is the number of the sliced segments.

In step 4, the factor nodes that only depend on S_m and S_{m+1} are aggregated into a single composite factor node F_m. If a factor only depends on S_m, it is merged to either F_{m-1} or F_m. The choice between F_{m-1} and F_m is arbitrary. With the slicing, it is guaranteed that all the original factors span at most two adjacent slices. This is because the original factors span at most $3d_{max}$ length in the Cartesian space since they span at most four atoms. Therefore, this process makes a linear structured factor graph as shown in Figure 8. In other words, the factor graph having cycles is efficiently converted to a linear graph using the information of atom distances of the underlining molecule in 3D space.

In step 5, sampling is applied at each composite variable node S_m according to potentials represented by the composite factors F_{m-1} and F_m fixing positions of atoms in other slices. With the sampling, a finite set of positions of atoms are generated and they

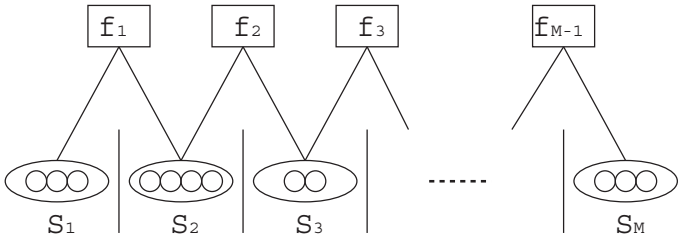


Fig. 8 A linear structured factor graph having composite variable and factor nodes representing a potential of a molecule.

are regarded as possible values of the composite node. Therefore, the factor graph with continuous variables is approximated by the one with discrete variables. In this study, MCMC is used for the sampling. Given the discrete factor graph, max-sum is applied in step 6 to find a global minimum among all the combinations of the samples over the slices, where the number of combinations is exponential to M. During the max-sum process, potential energies are re-calculated and the energies estimated during the MCMC sampling are not used.

After the max-sum procedure, a new atom position configuration is obtained. In step 7, if the drop of the energy from the previous configuration is small and the process is converged, the configuration is output and the process is terminated. Otherwise, the process is repeated from step 2 using the new configuration as an initial value. The steps from 2 to 7 is referred to as an epoch.

3.3 Properties of SCMS

In the max-sum step of SCMS, it finds the global minimum among K^M combinations of candidate positions of the atoms when K samples are used at each composite node. The computational cost to explore the exponential space is only $O\left(K^2M\right)$. Therefore, it is desirable to choose M as large as possible so that larger possibilities are investigated. This is why the slicing is performed along the longest axis in SCMS. As a special case, if there is only a single slice and only a single epoch is applied, SCMS reduces to the MCMC method. Since larger molecules have larger M for a given slice interval, it is expected that SCMS is more advantageous than MCMC when it is applied to larger molecules.

When MCMC is used for the sampling at each composite node, adjacent samples in the generated sequence have strong correlation as it has been explained in Section 2.4. Therefore, K consecutive samples will cover only a small region unless K is large enough. On the other hand, K affects the computational cost with the squared order K^2. In this study, to cover wider region while limiting the number of samples used at each composite node, every I-th sample in a KI-length sequence is used.

4. Experimental Conditions

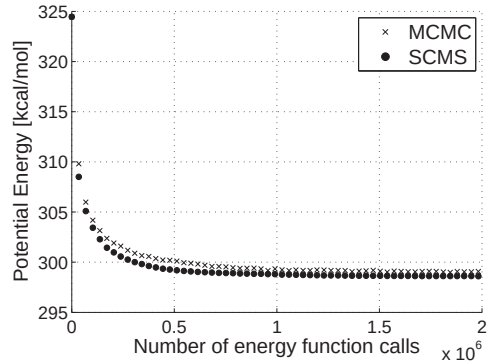
4.1 Data Preparation

We prepared four different length poly-peptides for testing the proposed method. They are 50-mer, 100-mer, 200-mer and 300-mer poly-alanaines. The structures of them were built with LEap program included in the AMBER 11 package [10]. The parameters required for calculating potential energies such as bond stretching constants and equilibrium bond lengths, were taken from the amber99 force field.

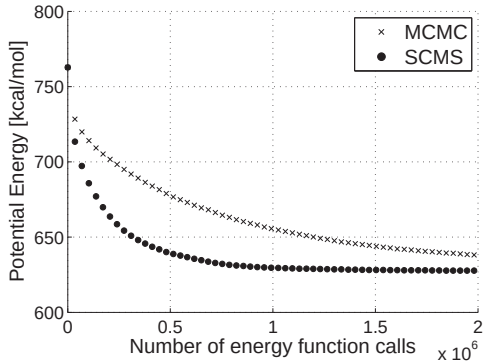
4.2 Comparison Method

Our primal aim is to compare conventional MCMC and the proposed SCMS in terms of computational cost required to find a lower energy structure. We measured their performance by looking at how potential energy decreases as the increase of the number of calls of a potential energy function. For the MCMC method, a call of an energy function corresponds to an evaluation of potential energy of a candidate molecular structure. However, a molecule is sliced in SCMS and potential energies are evaluated for the slices both for the sampling and for the max-sum processes. In order to make a direct comparison possible between SCMS and MCMC results, a normalized number of energy function call is defined as a total number of evaluations of bond, angle, and dihedral energies divided by the total number of bonds, angles, and dihedrals in a molecule. For the MCMC method, the normalized number of the energy function calls is equivalent to the number of the evaluation of a candidate molecular structure.

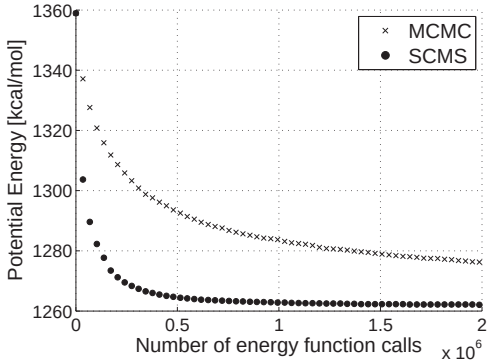
As for the proposal distribution for the MCMC sampling used in the MCMC and SCMS-based methods, a Gaussian distribution was used. The standard deviation of the Gaussian distribution was set to 0.0001Å based on a preliminary experiment so that the performance of the baseline MCMC method is maximized. For the SCMS method, the number of samples K and the interval I were set to 50 and 400, respectively. The same random seed was used for both MCMC and SCMS.



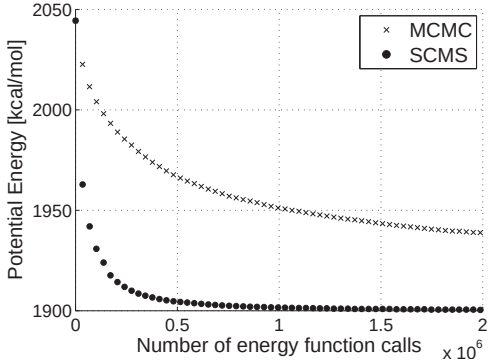
(a) 50-mer poly-alanine. Actual CPU time at 2 million calls was 56 minutes for MCMC and 57 minutes for SCMS.



(b) 100-mer poly-alanine. Actual CPU time at 2 million calls was 113 minutes for MCMC and 115 minutes for SCMS.



(c) 200-mer poly-alanine. Actual CPU time at 2 million calls was 233 minutes for MCMC and 239 minutes for SCMS.



(d) 300-mer poly-alanine. Actual CPU time at 2 million calls was 345 minutes for MCMC and 364 minutes for SCMS.

Fig. 9 Normalized number of energy function calls and potential energy for poly-alanines with varied lengths. The proposed SCMS method finds the lower energy structure faster than MCMC for larger molecules.

5. Experimental Results

We performed our experiments as described above on TSUBAME 2.0 [11]. Figure 9 shows the result for every four poly-alanine, (a) 50-mer, (b) 100-mer, (c) 200-mer and (d) 300-mer, respectively. In the case of (a) 50-mer, the performance of conventional MCMC and the proposed SCMS are almost the same. However as the poly-peptides grow longer, SCMS becomes much more efficient than MCMC. This is because SCMS is more advantageous than MCMC for larger molecules as explained in Section 3.3. Since the computational cost of the MCMC and SCMS methods are dominated by the evaluation of the energy functions, their CPU time is mostly linear to the number of the normal-

Vol.2011-BIO-26 No.5
2011/9/13

ized energy function calls. For (d) 300-mer, the actual CPU time required for MCMC and SCMS were respectively 346 and 364 minutes for 2 millions of the normalized number of energy function calls. The minimum energy found by MCMC after 2 millions of the energy function calls was 1938.8 (kcal/mol). By using SCMS, lower value than that was achieved with only 0.10 million calls or 19 minutes, which was 1930.8 (kcal/mol). The minimum energy found by SCMS after 2 million calls was 1900.4 (kcal/mol), which was significantly lower than the energy value 1938.8 (kcal/mol) obtained by the MCMC method.

6. Conclusion

In this paper we proposed SCMS method for predicting the structure of macromolecules based on modeling the potential landscape by a factor graph. The factor graph is converted to a linear structured graph by aggregating the factors based on a maximum distance between bonded atoms. Then, the continuous variables of the factor graph are approximated by a finite set of samples and the efficient max-sum search algorithm is applied. This process is iterated until it converges. The experimental results show that while SCMS gives similar performance as MCMC when it is applied to a relatively small polypeptide consisting of 50 alanines, it significantly outperforms MCMC for larger polypeptides.

There are several important improvements that need to be made in the future. Firstly, the search cost by SCMS could be further reduced by, for example, introducing the beam pruning in the max-sum process. Secondly, the potential function used in the experiments is simple, and it does not take intermolecular forces into account. The intermolecular forces are important to predict tertiary structure and they must be incorporated. Thirdly, As we only compared SCMS with MCMC, it must be compared to other prediction methods to further analyze the performance. While the experiments were performed using a single macromolecule as an input, it is not a requirement of SCMS. As far as a linear structured factor graph is constructed, multiple or many molecules can be treated. Therefore, it would be also interesting to apply SCMS to quaternary structure prediction to analyze assembly of several proteins.

Acknowledgments This work was conducted as part of KAKENHI (23650068). Part of this research was also supported by JST, Research Seeds Program.

References

1) McCammon, J. and Harvey, S.: *Dynamics of proteins and nucleic acids*, Cambridge University Press (1987).
2) Mitsutake, A., Sugita, Y. and Okamoto, Y.: Generalized-Ensemble Algorithms for Molecular Simulations of Biopolymers, *Biopolymers*, Vol.60, pp.96–123 (2001).
3) Landau, L.D. and Lifshitz, E.M.: *Statistical*, Butterworth-Heinemann, Oxford, 3rd edition part 1 edition (1980).
4) Leach, A.R.: *Molecular Modelling: Principles and Applications*, Prentice Hall (2001).
5) Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E.: Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics*, Vol.21, pp. 1087–1092 (1953).
6) Brunette, T. J. and Brock, O.: Improving protein structure prediction with model-based search, *Bioinformatics*, Vol.21, pp.66–74 (2005).
7) Lauritzen, S.L. and Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems, *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol.50, No.2, pp.pp. 157–224 (1988).
8) Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc. (2006).
9) Frey, B.J.: *Graphical models for machine learning and digital communication*, MIT Press, Cambridge, MA, USA (1998).
10) University of California: *AMBER 11*, San Francisco (2010).
11) Tokyo Institute of Technology: *Global Scientific Information and Computing Center 2011* (2011). `http://www.gsic.titech.ac.jp/sites/default/files/gsic2011E.pdf`.