Unbiased, Scalable Sampling of Constrained Kinematic Loops

Yajia Zhang and Kris Hauser
School of Informatics and Computing
Indiana University Bloomington, USA
{zhangyaj, hauserk}@indiana.edu

***Abstract*—We propose a Monte Carlo technique for generating conformations of a kinematic chain under loop closure and other severely restrictive constraints. The problem is cast as one of inference on a probabilistic graphical model that links configuration variables, spatial variables (e.g., atom positions), and constraints in a unified framework. We employ two technical contributions: first, a sparse model allows scalable inference to be performed on large kinematic chains, and second, the mathematical foundations for unbiased sampling of a probability density restricted to an implicit submanifold specified by loop closure constraints. The method is demonstrated to improve the speed of protein loop sampling when integrating prior from steric clashes, Ramachandran plots, and B-factors.**

***Keywords*-protein loops; conformation sampling; Markov chain Monte Carlo**

I. Introduction

The movement of kinematic chains is an area of interest in the geometry of folding linkages, robot motion planning, and protein structure prediction. One of the most valuable tools in this area is the ability to generate (i.e., sample) conformations of a chain that satisfies certain hard feasibility constraints, such as loop closure and collision avoidance, and soft preference constraints, such as low energy and high likelihood. These constraints impose a major computational challenge because the subset of feasible and favorable conformations is a minuscule (sometimes zero) volume subset of the space, particularly around folded protein structures. Due to the large number of degrees of freedom in interesting biological macromolecules (ranging up to hundreds or thousands), new techniques are needed to sample severely constrained conformations in an efficient manner.

Monte Carlo (MC) techniques have a long history of use in computational biology because they can quickly explore multiple energy minima and transition pathways whereas molecular dynamics and optimization techniques often get stuck in single local minima [1], [8], [11], [14]. They are also able to generate conformation *ensembles*, which are useful for a number of applications such as understanding equilibrium fluctuations [5], [15]. However, there is a dilemma in setting the perturbation size; small perturbations increase the rate of successful moves but slow the rate at which conformation space is explored. Moreover, standard MC techniques are not directly applicable to protein loops, which are extremely relevant to understanding protein function [7]. Existing methods for loop computation include discrete search [5], optimization [7], and inverse kinematics (IK) sampling methods [2], [3], [13], [17]. But, we still lack methods that are simultaneously scalable to large loops (e.g., >10 residues) and that generate unbiased ensembles of conformations.

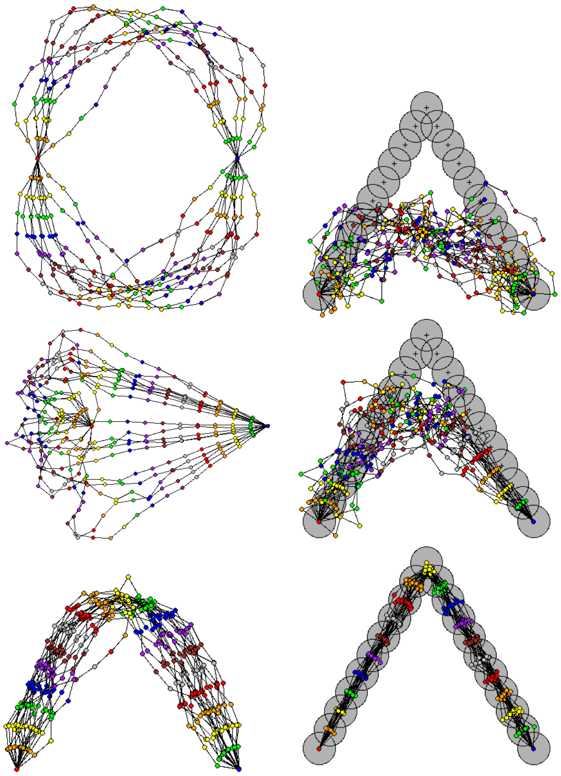

Fig. 1. Three sampling methods for a 20-link closed-loop chain. At left, the prior gives preference to joint angles with small magnitude. At right, the prior gives preference to joint positions in a triangle shaped distribution (crosses: means, shaded circles: 3σ spreads). Top: Sampling joint angles followed by numerical loop closure, best 20/20,000 samples. Middle: Sampling with RLG [3], best 20/20,000 samples. Bottom: Our method, displayed every 40'th sample. These sample sets are generated by our method approximately as fast as RLG and an order of magnitude faster than numerical loop closure.

The contribution of this paper is a unified and scalable probabilistic graphical modeling framework for Monte Carlo sampling of kinematic chains. It is particularly well-suited for closed loops (Figure 1) but can also be advantageous for chains with free endpoints as well (Figure 2). The technique uses a blocked Gibbs sampler that proposes movements of small subchains of conformation angles at once, along with a Metropolis-Hastings technique that guarantees an unbiased sampling of the loop-closure submanifold for that block. Due to the small block size, each energy function is local and adjustments are extremely fast. Our method is mathematically proven to be unbiased and numerical experiments demonstrate

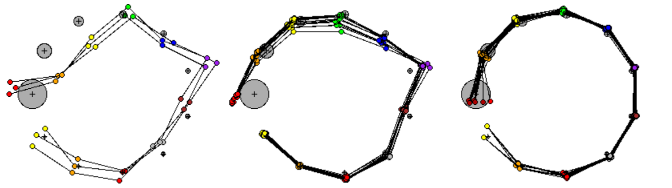

Fig. 2. Comparing MC sampling methods for a free-endpoint chain with heterogeneous prior distribution over joint positions (crosses: means, shaded circles: 3σ spreads). For each method, 400 iterations are run and every 20'th sample is retained, taking approximately 10 s time on a standard PC. Left: standard MC takes steps that are too large and only generates 3 unique samples. Middle: MC with step size reduced by 10 has a higher success rate but slower convergence. Note the lack of variance in the leftmost point. Right: Our method.

that our method generates higher quality samples with lower computational cost than existing approaches.

II. Related Work

Protein loop closure techniques are used in generating variability in equilibrium conformations and missing fragment reconstruction. Fiser et al's [7] optimization approach uses an energy function which encodes spatial restraints and preferences on dihedral angles, and then runs a computationally expensive minimizer. Discrete search methods are able to explore a wider space of conformations by incrementally building a tree of clash-free subchain conformations [6], [16]. But, these methods face a problem of combinatorial explosion and are rarely applied to residues of length greater than 10. They also suffer from discretization artifacts and are not able to close the gap with a terminal atom exactly. A third class of methods are inverse kinematics (IK) techniques from the robotics field for sampling closed-loop conformations [2]–[4], [13]. However, these methods do not take energies into account during sampling, so some authors employ a secondary energy optimization step to generate more plausible conformations [13], [15], [17]. For each of these methods it is difficult to discuss properties of the sampling *distribution*, which is throughly entangled with the sampling *procedure*, and empirical testing is employed to argue that a method samples well. In contrast, our method provides a unified probabilistic framework for both modeling a desired distribution and sampling from it in a mathematically rigorous fashion.

Probabilistic graphical models have been shown to be powerful tools for inference in large domains, and have been applied in a limited sense to protein structure prediction. They have been applied to side-chain prediction [18] and prediction of macromolecular assemblies from electron density maps [10]. Their application is reasonably well understood in the discrete case, but continuous variables often prove challenging. We derive the mathematical foundations needed to apply this approach to continuous distributions constrained to nonlinear implicit manifolds.

III. Problem Statement and Background

This section will first review the fundamentals of Markov Chain Monte Carlo (MCMC) techniques, and then describe the implementation challenges for constrained kinematic chains.

A. Sampling Framework

Let the state variables of a system be denoted $\mathbf{x} = (x_1, \ldots, x_n)$. Hard constraints and soft preferences are encoded into a nonnegative scoring function $\Phi(x_1, \ldots, x_n)$. The score must have finite integral, is zero at states that violate hard constraints, and higher values indicate higher desirability. Φ can be considered as an unnormalized probability density, and our goal is to generate samples with probability proportional to Φ. In other words, we wish to sample from the normalized density P defined as:

$$P(x_1, \ldots, x_n) = \frac{1}{Z}\Phi(x_1, \ldots, x_n) \tag{1}$$

where Z is a proportionality constant that ensures that P integrates to 1.

B. Sparse Factored Models

It is convenient to represent Φ in a *factored* form:

$$\Phi(x_1, \ldots, x_n) = \prod_i \phi_i(S_i) \tag{2}$$

where each ϕ_i is known as a *factor* and each S_i is a subset of $\{x_1, \ldots, x_n\}$ known as the *domain* of the factor ϕ_i. For example, in protein structure prediction we may have factors for Ramachandran plots for each pair of dihedral angles (φ, ψ), steric clashes, energy functions defined over atom positions, and prior knowledge from B-factors or electron density maps.

Probabilistic graphical models, such as Bayesian networks and Markov random fields, are inherently factored. A graphical model is *sparse* if each variable x_i is involved in only a handful of factors (i.e., bounded by a constant unrelated to n), and hence only interacts directly with a few other variables. An important consequence is that this additional structure makes probabilistic inference in sparse models computationally tractable (polynomial in n), whereas inference in dense models is intractable (in general, exponential in n). The conversion of a kinematic chain from dense to sparse form, as described below, is a key step in our method. Our implementation currently supports:

- Ramachandran plots $\phi_{RP(r)}(\varphi, \psi)$ which vary by residue r.
- Steric clashes $\phi_{SC(j,k)}(p_j, p_k)$ which are 0 if atom j collides with atom k and 1 otherwise.
- B-factors defined as a Gaussian $\phi_{BF(j)}(p_j) = \frac{1}{c\sqrt{2\pi B_j}} \exp - \frac{||p_j - \mu_j||^2}{2B_j c^2}$ where μ_j is the predicted atom position and B_j is the B-factor value in the protein's PDB file. A constant of proportionality c can be set by the user according to his/her confidence in the quality of the B-factor estimates.

Although each factor can be evaluated quickly, over thousands or millions of evaluations they accumulate significant computational cost. Sparseness is important to exploit; for example, when a few variables are changed, the change in

Φ can be calculated quickly by only evaluating those factors involved. This leads to significant savings compared to recomputing Φ from scratch.

C. Markov Chain Monte Carlo Methods

Sampling from complex distributions of the form (1) is almost always handled using Markov Chain Monte Carlo (MCMC) methods. Starting from an initial state $\mathbf{x}^{(0)}$, these methods generate a sequence of states $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ that forms a random walk whose probability density asymptotically approaches P.

There are two main classes of MCMC algorithms. First, Gibbs sampling is commonly used for obtaining a list of random samples from multivariate probability distribution. This algorithm applies when it is easier to sample from the conditional density rather than the entire joint density, particularly for sparse factored models. Given the sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})$ at time k, this algorithm generates the next state $\mathbf{x}^{(k+1)}$ by sampling a single variable $x_i^{(k+1)}$ from the conditional density

$$P(x_i^{(k+1)} | x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_n^{(k)}) \tag{3}$$

and keeping the remaining variables fixed. The variable will be updated as soon as its value has been sampled. The index i is incremented in looping fashion. Blocked Gibbs sampling is a variation of Gibbs sampling by grouping multiple variables as a block and sampling the block from the joint distribution conditioned on all other variables.

Second, the Metropolis-Hastings (M-H) algorithm addresses the problem that it is hard to sample directly from P and compute the normalization factor Z explicitly. Instead, the algorithm samples a candidate move from $\mathbf{x}^{(k)}$ to $\mathbf{x}'$ from a *proposal distribution* $Q(\mathbf{x}'; \mathbf{x}^{(k)})$, and *accepts* the move $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}'$ with probability

$$\alpha = \min\left(1, \frac{P(\mathbf{x}')Q(\mathbf{x}^{(k)}; \mathbf{x}')}{P(\mathbf{x}^{(k)})Q(\mathbf{x}'; \mathbf{x}^{(k)})}\right). \tag{4}$$

The Z term in the numerator and denominator cancel out, so we can use Φ directly instead of P. If the move is rejected, then the current state is maintained: $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)}$.

The remaining question is how to choose a proposal distribution that we can sample from and evaluate. The acceptance strategy must evaluate (4) exactly so that the M-H algorithm respects the *detailed balance* condition that guarantees that the Markov chain converges to the proper stationary distribution. One of our key contributions is a technique for evaluating Q exactly when sampling from closed chain submanifolds, which enables our method to generate an unbiased sampling sequence.

There also exist many variants and hybridizations of these two basic methods. Our technique is one such a hybrid. M-H is used to avoid calculation of normalization factors, and blocked Gibbs sampling is used to scale better to large chains.

D. Kinematic Chains

Consider a jointed kinematic chain with reference frames $T_0, T_1, \ldots, T_N$, connected with relative rotational angles $q_1, \ldots, q_N$. For a protein backbone, there is a one to one correspondence between frames and atom positions along the backbone $p_1, \ldots, p_M$, and the rotational variables are simply the backbone dihedral angles $\varphi_1, \psi_1, \ldots, \varphi_{N/2}, \psi_{N/2}$. Although our technique generalizes in a straightforward manner to branched structures (e.g., protein side-chains or multi-limbed robots), we currently consider only linear structures, with or without fixed endpoints.

It may be tempting to define the system state with a minimal set of coordinates, e.g., $\mathbf{x} = (T_0, q_1, \ldots, q_N)$, because each subsequent frame $T_1, \ldots, T_N$ can be determined from $\mathbf{x}$ through straightforward forward kinematics. However, this approach eliminates sparsity in the probabilistic model because a factor defined on T_N will depend on all variables, a factor defined over T_{N-1} will depend on all variables except q_n, and so on. Moreover, if a sampler is asked to generate certain variables from a density defined over $T_1, \ldots, T_N$ (for example, atom positions), may be biased unless it computes the determinant of an $N \times N$ metric tensor for each evaluation of Φ. This is a consequence of nonlinear transformations of distributions (see Appendix). On the other hand, computing determinants takes $O(N^3)$ time, which scales poorly with large N.

The key step of our method is to consider an expanded state that incorporates all spatial variables along with the conformation variables: $\mathbf{x} = (q_1, \ldots, q_N, T_0, \ldots, T_N)$. The joint probability density is then defined over angles and reference frames of all links along the chain:

$$\Phi(\mathbf{x}) = \prod_i \phi_i(S_i) \prod_{j=1}^{N} \phi_{kinematic}(T_j | T_{j-1}, q_j) \tag{5}$$

where each S_i is now a subset of $\{q_1, \ldots, q_n, T_0, \ldots, T_n\}$, and where $\phi_{kinematic}$ is the forward kinematic transform that defines the frame T_j in terms of the prior frame T_{j-1} and the relative angle q_j. Because the transform is deterministic, $\phi_{kinematic}$ should be thought of as an indicator function. Taking the convention that each frame's origin lies on its joint's axis:

$$\phi_{kinematic}(T_j | T_{j-1}, q_j) = \begin{cases} 1 & \text{if } T_j = T_{j-1} T_j^{rel} R(a_j, q_j) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where T_j^{rel} is the relative transformation of frame j relative to frame $j-1$ and $R(a, q)$ is the rotation of angle q about axis a. Fixed-endpoint constraints can also be encoded with indicator factors $\phi_{closure}(T_0)$ and $\phi_{closure}(T_n)$ that are zero everywhere except at the fixed frames.

With (5) encoded properly, few factors include any given variable in their domain, and the model is sparse. However, we have added the complication of maintaining a valid kinematic structure, because the set of $\mathbf{x}$ for which Φ is nonzero lies on a lower-dimensional manifold. Technically speaking, the probability density must be considered with respect to a base

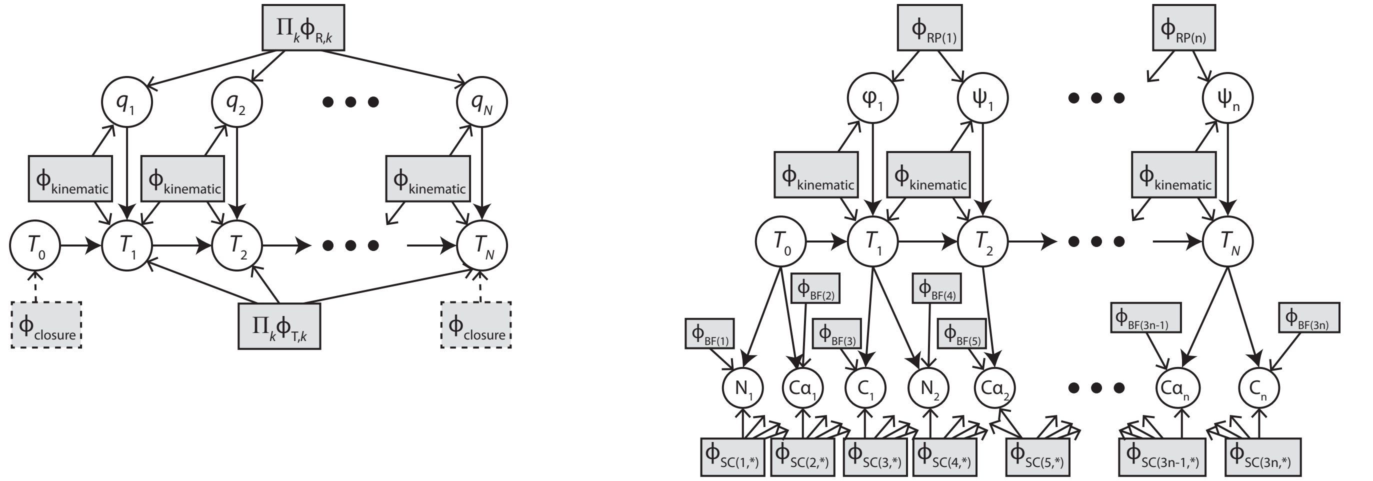

Fig. 3. Left: sparse graphical model relating N joint angles and link transformations via local factors. Right: instantiation of the model for an n-residue protein backbone, with an additional layer accounting for atom positions.

measure that assigns finite, nonzero density to the manifold. For 3D chains, the state space has dimensionality $7N$ but the manifold has dimensionality $6+N$ for free-endpoint chains or $N-6$ for fixed-endpoint chains. (For 2D chains, these constants become 4, 3, and 3, respectively.) The next section will describe how we handle these submanifolds in detail.

IV. Method

A. Summary

Our sampler uses a blocked Gibbs sampling method that simultaneously samples subsets of variables that are sufficiently large to give at least one continuous degree of freedom of movement. It is unrealistic to sample exactly from the block conditional density, so we use the Metropolis-Hastings criteria to accept or reject a move.

The key subroutine, Sample-Block-MH, takes as input the previous sample $\mathbf{x}^{(k)}$ and a block B of consecutive joint angles and their intervening frames. It then samples a candidate move, and accepts it according to the M-H criterion. Pseudocode is as follows:

```
**Sample-Block-MH**($\mathbf{x}^{(k)}$, $B$):
  1) Sample a candidate conformation $\mathbf{x}'_B$ of $B$ at random,
     keeping the rest of the chain $\mathbf{x}_C^{(k)}$ fixed.
  2) Compute the M-H acceptance probability
     $\alpha = \min\left(1, \frac{\Phi_B(\mathbf{x}'_B) Q_B(\mathbf{x}_B^{(k)} | \mathbf{x}_C^{(k)})}{\Phi_B(\mathbf{x}_B^{(k)}) Q_B(\mathbf{x}'_B | \mathbf{x}_C^{(k)})}\right).$
  3) Accept the move $\mathbf{x}_B^{(k+1)} \leftarrow \mathbf{x}'_B$ with probability $\alpha$.
```

Here the subscript B denotes the subset of variables in the block, while the subscript C denotes the complement of the block. The score Φ_B calculates the product of factors ϕ_i whose domains S_i overlap with B, which is more efficient than recomputing Φ from scratch. The remaining details of the method — the block size, the block sampling procedure, and calculating the sampling probability Q_B — are described in detail in the remainder of this section.

To generate a new conformation $\mathbf{x}^{(k+1)}$ of the entire chain, Sample-Block-MH is called several times with overlapping blocks incremented sequentially down the chain. Thanks to sparsity, each pass is performed in $O(N)$ time, which takes a fraction of a second for chains with hundreds of variables. For each block we call Sample-Block-MH up to n_B times or until a move is accepted ($n_B = 10$ in our implementation). With larger values of n_B the method is more likely to generate successful moves, but may spend too long in particularly constrained regions of the chain.

B. Blocked Gibbs Sampling Using Inverse Kinematics

Standard Gibbs sampling in our problem does not work because loop closure constraints constrain the conditional density of any variable given the rest (3) to a Dirac. Hence, the state would never change. Assuming a 3D chain, no mixing occurs for 5 or fewer angles (except possibly at singular conformations, which occupy a set of measure zero in conformation space). For 6 angles, analytical inverse kinematics (IK) techniques are available to compute solutions for a pair of fixed end frames [4]. In fact, any number between 0 and 16 solutions may exist for a given 6-angle problem. Nevertheless, 6-angle blocked Gibbs sampling is not suitable because a random walk can only access a finite set of states.

With 7 angles, we have sufficient freedom to sample from a 1-dimensional manifold of solutions. In general, a block of b angles admits a $b-6$ dimensional solution manifold. Denote the block $B = \{q_i, \ldots, q_{i+b-1}, T_i, \ldots, T_{i+b-2}\}$, and let us call the first $b-6$ angles of the block $q_i, \ldots, q_{i+b-7}$ the *independent* subchain. Call the remaining 6 angles the *dependent* subchain. This is illustrated for a planar chain in Figure 4. A sampling procedure is as follows:

```
**Sample-Block**
  1) Sample values for the independent subchain at random.
  2) Attempt to close the chain by calculating an analytical
     IK solution for the dependent subchain. We use the
     method of [4].
```

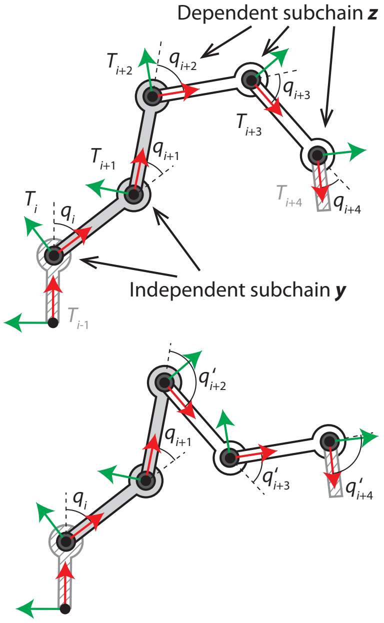

Fig. 4. Top: a 5-angle block for a planar chain with fixed end frames T_{i-1} and T_{i+4}. Bottom: a second IK solution for the dependent subchain.

3) If more than one IK solution exists, one is picked at random, and if no solution exists, the process terminates with failure.

It is recommended that $b \geq 7$ be chosen as low as possible, because as b grows, the probability of sampling an independent subchain that admits closure drops off dramatically as b grows, particularly for "stretched out" conformations. Our implementation uses $b = 8$ blocks for proteins because even numbers align better with the (φ, ψ) angles of each residue. (Throughout this discussion we are assuming a 3D chain. In the planar case, at least 4 angles are needed, and the manifold of solutions is $(b-3)$-dimensional)

C. Metropolis-Hastings Sampling Step

The M-H detailed balance condition requires calculating the *sampling density* for the *sampling procedure* Sample-Block. We derive this density $Q_B(\mathbf{x}'_B|\mathbf{x}_C^{(k)})$ here, and introduce several concepts from differential geometry that are needed to do so.

Fix the endpoints of the block, and denote the $(b-6)$-dimensional manifold of loop-closing conformations as M. Let us call the $(b-6)$ angles of the independent subchain $\mathbf{y}$, which are sampled w.r.t. the density $P(\mathbf{y})$. Observe that the candidate sample $\mathbf{x}'_B$ is distributed according to a nonlinear transformation of $P(\mathbf{y})$ onto M. In fact, at non-singular conformations the independent subchain forms a local *chart* of M, which is a local bijection between $\mathbb{R}^{b-6}$ to M centered at $\mathbf{x}'_B$ (Figure 5).

Since there is a local bijection f between $\mathbf{y}$ and the point on the manifold $\mathbf{x}_B$, the sampling density over $\mathbf{x}_B$ is given by:

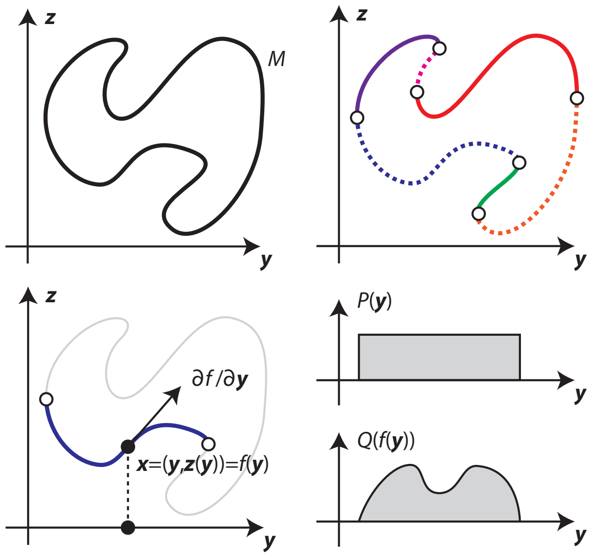

Fig. 5. Top: abstract illustration of how analytical IK implicitly decomposes a 1-parameter manifold M into a set of local bijections (charts). Bottom: the Jacobian of a chart must be taken into account when calculating the sampling distribution Q over M.

$$Q_B(\mathbf{x}_B|\mathbf{x}_C^{(k)}) = \frac{P(\mathbf{y})}{s\sqrt{\det G(\mathbf{y})}} \tag{7}$$

where s is the number of IK solutions at $\mathbf{y}$ and G is the *metric tensor* of the chart $\mathbf{x}_B = f(\mathbf{y})$ (see Appendix). The inclusion of the metric tensor is a natural consequence of transformation of variables. For example, for the case $b = 7$, the metric tensor is the squared arc length of the 1-dimensional parametrization of M (Figure 5, bottom). In general, G is given by

$$G(\mathbf{y}) = \left(\frac{\partial f}{\partial \mathbf{y}}(\mathbf{y})\right)^T W \left(\frac{\partial f}{\partial \mathbf{y}}(\mathbf{y})\right) \tag{8}$$

where $\frac{\partial f}{\partial \mathbf{y}}(\mathbf{y})$ is the Jacobian of the function f. Here we have also introduced a positive semidefinite weighting matrix W for the purpose of weighting the relative importance of matching the prior along certain axes. In the standard case, W is an identity matrix, but it can also be useful to choose a nonuniform diagonal matrix to account for heterogeneous units (e.g., angle vs. position variables).

A remaining issue is that it is often difficult to explicitly compute the Jacobian of the IK function involved in f. In other words, if $\mathbf{z} \equiv \mathbf{z}(\mathbf{y})$ is the 6 angles in the dependent chain that are determined by IK, it is difficult to evaluate $\partial\mathbf{z}/\partial\mathbf{y}$. So, we compute it through the implicit form of the constraints $C(\mathbf{x}_B) = 0$. These vector-valued constraints state that the difference between the terminal frame of the subchain and the desired frame is zero. We have the constraint equation:

$$0 = C(\mathbf{x}_B) = C(\mathbf{y}, \mathbf{z}) \tag{9}$$

Performing the derivative of both sides of (9) with respect to $\mathbf{y}$ we get:

$$0 = \frac{\partial C}{\partial \mathbf{y}} + \frac{\partial C}{\partial \mathbf{z}}\frac{\partial \mathbf{z}}{\partial \mathbf{y}} \tag{10}$$

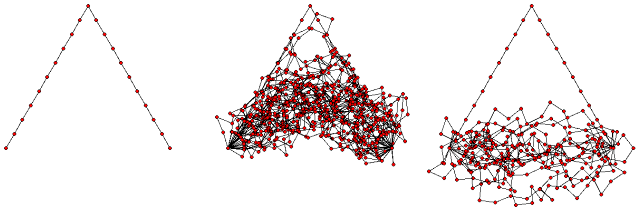

Fig. 6. Sampling conformations of a planar 20-link chain, anchored at the endpoints, with a uniform prior. (Left) Starting from a deliberately bad initial conformation. (Middle) The sequence mixes relatively quickly, but the first 40 samples are biased by the initial conformation and autocorrelate strongly. (Right) A sequence that takes every 40'th sample does not significantly autocorrelate.

and hence

$$\frac{\partial \mathbf{z}}{\partial \mathbf{y}} = -\left(\frac{\partial C}{\partial \mathbf{z}}(\mathbf{x}_B)\right)^{-1} \frac{\partial C}{\partial \mathbf{y}}(\mathbf{x}_B) \tag{11}$$

holds as long as $\frac{\partial C}{\partial \mathbf{z}}$ is invertible, which is true everywhere except at singular conformations. Each derivative of C in the above expression is a submatrix of the Jacobian and can be computed using standard techniques.

Finally, since

$$f(\mathbf{y})^T = \left[\mathbf{y}^T, \mathbf{z}^T, T_i, \dots T_{i+b-2}\right] \tag{12}$$

we obtain the Jacobian

$$\frac{\partial f^T}{\partial \mathbf{y}} = \left[I, \frac{\partial \mathbf{z}}{\partial \mathbf{y}}^T, \frac{dT_i}{d\mathbf{y}}, \dots, \frac{dT_{i+b-2}}{d\mathbf{y}}\right] \tag{13}$$

in which I is the identity matrix and all frame derivatives are calculated using the chain rule $\frac{dT_j}{d\mathbf{y}} = \frac{\partial T_j}{\partial \mathbf{y}} + \frac{\partial T_j}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{y}}$. These partial derivatives are calculated using standard techniques.

D. Implementation Details

We have only described the sampling step for intermediate blocks, which is sufficient for sampling closed loops. Free-endpoint chains require separate sampling subroutines for the start and end blocks. Standard MC methods are employed here.

It is also important to examine efficient methods for computing the M-H acceptance probability. In experiments on a 10-residue chain (1AMP181-190) the probability that a move is clash-free is 78%, but only 0.28% of clash-free moves are accepted. Since collision checking is 60 times more expensive than calculating the remainder of the terms in Φ, we consider clash detection *after* determining whether it will be accepted. This method achieves an order of magnitude speedup over the naive method.

E. Mixing and Autocorrelation

In any MCMC methods it is important to empirically examine the mixing rate of the Markov Chain. First, it can potentially take many iterations to “forget” the effects of a poor initialization. For proteins, this is not a significant problem because we initialize the chain with the predicted structure in PDB, which is typically quite good.

Second, subsequent samples are highly autocorrelated, and many conformations must be skipped to obtain a sequence with low autocorrelation. This is a serious concern because autocorrelation grows stronger as more variables are included in the conformation (Figure 6). In practice, one must determine the skip length empirically in order to obtain a *quasi-independent* sampling sequence, which is defined as a sequence with autocorrelation below some given threshold (0.2 is used in our experiments).

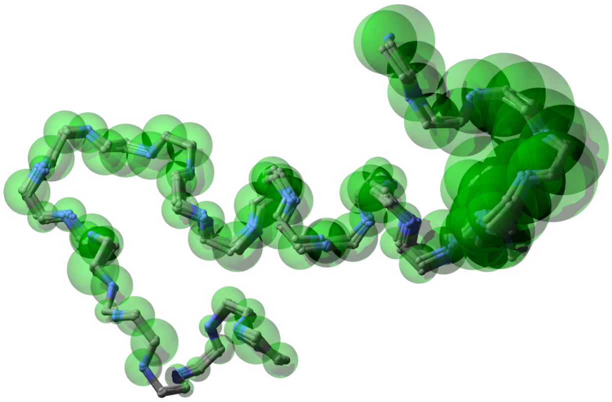

Fig. 7. 12 samples of a 30-residue subchain of protein 1B8C selected from first 300 consecutive samples with skip length 25. Transparent balls depict the 3σ spread of the atom position prior derived from its B-factor. Atoms with low B factors near the end of the chain increase the difficulty for standard Metropolis-Hastings method exploring the conformation space.

V. Experiments

Our sampler is implemented with auxiliary functions from the software package LoopTK [19]. All experiments are run on a Intel i7 2.7 GHz computer with 4 GB RAM.

A. Scalability tests on free-endpoint chains

Our first experiments study subchains of chain A in protein 1B8C, which is involved in calcium binding. We compared our sampler against a standard Metropolis-Hastings algorithm that samples backbone angles according to a Gaussian distribution with 1° standard deviation. Both methods sample from a joint probability that includes steric clashes, Ramachandran plots, and B-factors.

Two methods are tested on loops with a variety of lengths. Our sampler generates each sample for a 30-residue chain in 1 s (Figure 7) and is able to generate each sample for the entire 108-residue 1B8C protein in approximately 4 s (Figure 8).

Figure 9 displays the average time needed to obtain one quasi-independent sample over ten 30-minute runs for different chain lengths. The skip lengths are determined empirically for each run. This data suggests that our method achieves a cost per quasi-independent sample that is nearly linear to the length of the chain. In contrast, the likelihood that standard MH accepts a sample drops dramatically as the number of residues increases, leading to exponentially growing cost per sample.

B. Tests on closed-chains

We now use a 10-residue closed loop 1AMP181-190, which is a representative loop segment for testing loop reconstruction

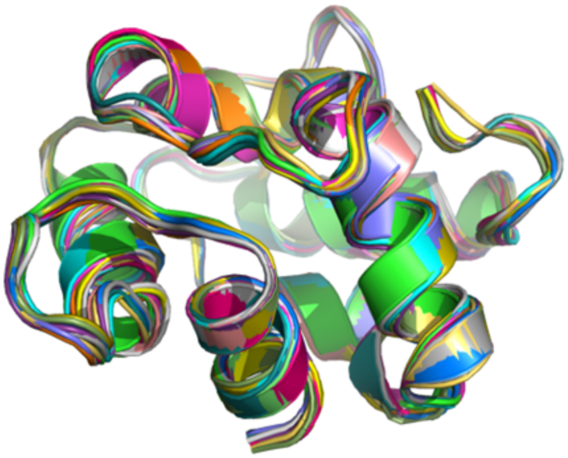

Fig. 8. 17 samples of 1B8C chain A (108 residues) selected from 170 consecutive samples with skip length 10. Each conformation is drawn in a distinct color.

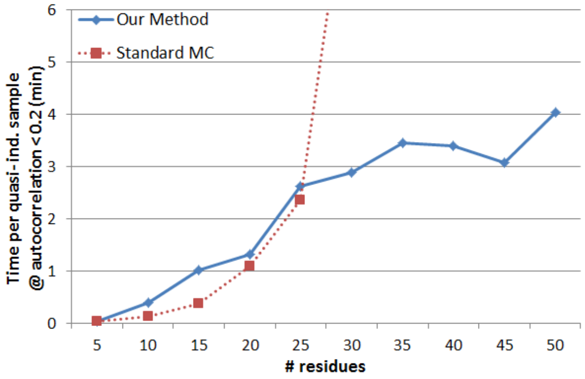

Fig. 9. Time required to obtain one quasi-independent sample on chains with a variety of lengths, for our method and standard Metropolis-Hastings. Standard MH did not generate even one sample for chain lengths above 30 after 30 minutes.

process [9], to test our method's ability to generate high-quality conformations. The Ramachandran plot (Figure 10) from 2000 consecutive samples shows that the dihedral angles from the generated samples are all within high probability regions and also explore a large allowable space.

We also compare our sampler with a sample-then-select method that first samples a set of clash-free conformations and then extracts the top scoring ones. The LoopTK configuration sampling method [19] was used here. Given 300 s cutoff time, LoopTK generates 888 conformations, while our method generates 1,922 (705 of which are unique). Figure 11 shows how the top 20 samples of LoopTK compare with every 100th sample of our method. The upper two sets are generated by the two methods using original B factors. Clearly, our method matches the prior information far more closely and obviates the need for postprocessing using expensive numerical optimization. The spread of the distribution can also be controlled precisely using our method. By enlarging B factors, we can obtain a greater spread of conformations that trusts the B-factor information less (Figure 11, bottom).

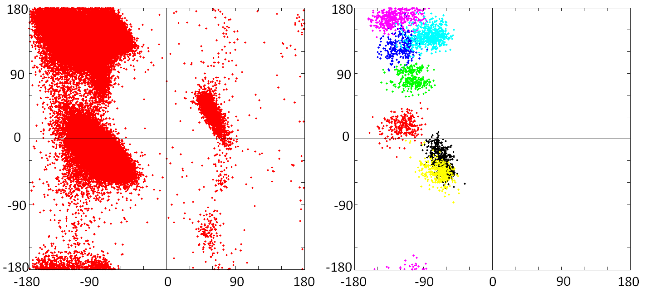

Fig. 10. The Ramachandran plot of generic residues from a database which includes 500 high-resolution proteins [12] used as a prior (left). The Ramachandran plot for the generic residues in our 10-residue test protein (1AMP 181-190) generated from 2,000 consecutive samples (right). Each color represents one residue.

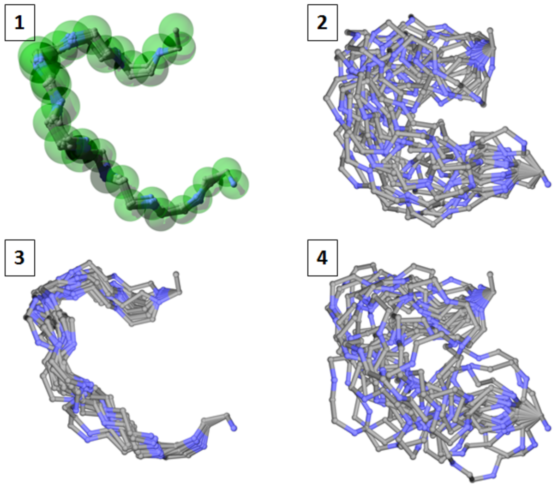

Fig. 11. Left: samples generated by our method with a skip length of 100. Right: samples generated by post-selecting the top 20 scoring samples generated from the LoopTK sampler [19]. Transparent balls depict the 3σ spread of the atom position prior derived from its B-factor. The top row uses the original B-factors, while the bottom row enlarges B-factors by 10.

VI. Conclusion

We propose a novel Monte Carlo method for loop conformation sampling. This method integrates a blocked Gibbs sampler and Metropolis-Hastings method on a sparse model that yields efficient inference. We derive the mathematical conditions necessary for Metropolis-Hastings to sample unbiased protein conformations according to a prior distribution restricted to a nonlinear manifold represented implicitly by equality constraints. Experiments show that our method can quickly generate conformations that match prior information for chains over 100 residues.

Appendix

We review a fundamental statement about densities under transformations of variables.

Suppose $\mathbf{u} \in \mathbb{R}^m$ *and* $\mathbf{v} \in \mathbb{R}^n$ *are multivariate random variables related by* $\mathbf{v} = f(\mathbf{u})$*, where* f *is differentiable and injective. Denote the image of* $A \subseteq \mathbb{R}^m$ *as* $M = f(A) \subseteq \mathbb{R}^m$*. If* g_u *is a density with support over* A*, then the corresponding density over* M*, with respect to the* m*-volume measure, is*

$$g_v(\mathbf{v}) = g_u(f^{-1}(\mathbf{v}))/\sqrt{\det G(f^{-1}(\mathbf{v}))} \tag{14}$$

where $G(\mathbf{u})$ *is the metric tensor:*

$$G(\mathbf{u}) = \left(\frac{\partial f}{\partial \mathbf{u}}(\mathbf{u})\right)^T \left(\frac{\partial f}{\partial \mathbf{u}}(\mathbf{u})\right). \tag{15}$$

More precisely, g_v *as defined above satisfies:*

$$\int_{f(U)} g_v(\mathbf{v}) d\mu = \int_U g_u(\mathbf{u}) d\mathbf{u} \tag{16}$$

for any subset $U \subseteq A$*, where* $d\mu$ *is the* m*-volume element of* M*.*

From change of variables we have:

$$\int_{f(U)} g_v(\mathbf{v}) d\mu = \int_U g_v(f(\mathbf{u})) X(\mathbf{u}) d\mathbf{u} \tag{17}$$

where $X(\mathbf{u})$ is the m-volume of the parallelotope spanned by the axes of the coordinate chart f centered at $\mathbf{u}$: $\frac{\partial f}{\partial u_1}(\mathbf{u}), \ldots \frac{\partial f}{\partial u_m}(\mathbf{u})$.

We now use the fact that the m-volume V of the parallelotope spanned by m vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ is given by the determinant:

$$V^2 = \det \begin{pmatrix} \mathbf{v}_1^T\mathbf{v}_1 & \mathbf{v}_1^T\mathbf{v}_2 & \cdots & \mathbf{v}_1^T\mathbf{v}_m \\ \mathbf{v}_2^T\mathbf{v}_1 & \mathbf{v}_2^T\mathbf{v}_2 & \cdots & \mathbf{v}_2^T\mathbf{v}_m \\ \vdots & & \vdots & \\ \mathbf{v}_m^T\mathbf{v}_1 & \mathbf{v}_m^T\mathbf{v}_2 & \cdots & \mathbf{v}_m^T\mathbf{v}_m \end{pmatrix}. \tag{18}$$

Note that this can be expressed more compactly as $\det(A^T A)$ where A is the matrix with $\mathbf{v}_1, \ldots, \mathbf{v}_m$ as its columns. Hence, $X(\mathbf{u}) = \sqrt{\det G(\mathbf{u})}$. Finally, substituting g_u in the r.h.s. of (17) gives the desired result.

References

[1] D. Bouzida, S. Kumar, and R. H. Swendsen. Efficient monte carlo methods for the computer simulation of biological molecules. *Phys. Rev. A*, 45(12):8894–8901, Jun 1992.

[2] A. Canutescu and R. Dunbrack Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12:963–972, 2003.

[3] J. Cortés, T. Siméon, M. Remaud-Siméon, and V. Tran. Geometric algorithms for the conformational analysis of long protein loops. *Journal of Computational Chemistry*, 25(7):956–967, 2004.

[4] E. Coutsias, C. Soek, M. Jacobson, and K. Dill. A kinematic view of loop closure. *J. Computational Chemistry*, 25:510–528, 2004.

[5] M. A. DePristo, P. I. W. de Bakker, S. C. Lovell, and T. L. Blundell. Ab initio construction of polypeptide fragments: Efficient generation of accurate, representative ensembles. *Proteins: Structure, Function, and Bioinformatics*, 51(1):41–55, 2003.

[6] M. A. DePristo, P. I. W.de Bakker, S. C. Lovell, and T. L. Blundell. Ab initio construction of polypeptide fragments: Efficient generation of accurate, representative ensembles. *PROTEINS: Structure, Function, and Genetics*, 51:41 – 55, 2003.

[7] A. Fiser, R. K. G. Do, and A. Šali. Modeling of loops in protein structures. *Protein Science*, 9(9):1753–1773, 2000.

[8] U. H. Hansmann and Y. Okamoto. New monte carlo algorithms for protein folding. *Current Opinion in Structural Biology*, 9(2):177 – 183, 1999.

[9] M. Jamroz and A.Kolinski. Modeling of loops in proteins: a multi-method approach. *BMC Structural Biology*, 10:5, 2010.

[10] K. Lasker, M. Topf, A. Sali, and H. J. Wolfson. Inferential optimization for simultaneous fitting of multiple components into a cryoem map of their assembly. *Journal of Molecular Biology*, 388(1).

[11] Z. Li and H. A. Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences*, 84(19):6611–6615, 1987.

[12] S. C. Lovell and et al. Structure validation by calpha geometry: phi,psi and cbeta deviation. *Proteins: Structure, Function, and Bioinformatics*, 50, Issue 3:437 –450, 2003.

[13] D. Mandell, E. Coutsias, , and T. Kortemme. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature Methods*, 6:551–552, 2009.

[14] N. Rathore and J. J. de Pablo. Monte carlo simulation of proteins through a random walk in energy space. *J. Chem. Phys.*, 116(7225), 2002.

[15] A. Shehu, C. Clementi, and L. Kavraki. Modeling protein conformational ensembles: From missing loops to equilibrium fluctuations. *Proteins: Structure, Function, and Bioinformatics*, 65:164–179, 2006.

[16] S. C. Tosatto, E. Blindewald, J. Hesser, and R. Männer. A divide and conquer approach to fast loop modeling. *Protein Engineering*, 15(4):279 – 286, 2002.

[17] H. van den Bedem, I. Lotan, J.-C. Latombe, and A. Deacon. Real-space protein-model completion: an inverse-kinematics approach. *Acta Crystallography*, 61(1):2–13, Jan 2005.

[18] C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. *Journal of Computational Biology*, 15(7):899 – 911, 2008.

[19] P. Yao, A. Dhanik, N. Marz, R. Propper, C. Kou, G. Liu, H. van den Bedem, J.-C. Latombe, I. Halperin-Landsberg, and R. Altman. Efficient algorithms to explore conformation spaces of flexible protein loops. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 5(4):534 –545, oct.-dec. 2008.