Fast Protein Loop Sampling and Structure Prediction Using Distance-Guided Sequential Chain-Growth Monte Carlo Method

Ke Tang[1], Jinfeng Zhang[2]*, Jie Liang[1]*

[1] Department of Bioengineering, University of Illinois at Chicago, Chicago, Illinois, United States of America, [2] Department of Statistics, Florida State University, Tallahassee, Florida, United States of America

Abstract

Loops in proteins are flexible regions connecting regular secondary structures. They are often involved in protein functions through interacting with other molecules. The irregularity and flexibility of loops make their structures difficult to determine experimentally and challenging to model computationally. Conformation sampling and energy evaluation are the two key components in loop modeling. We have developed a new method for loop conformation sampling and prediction based on a chain growth sequential Monte Carlo sampling strategy, called Distance-guided Sequential chain-Growth Monte Carlo (DiSGro). With an energy function designed specifically for loops, our method can efficiently generate high quality loop conformations with low energy that are enriched with near-native loop structures. The average minimum global backbone RMSD for 1,000 conformations of 12-residue loops is 1.53 Å, with a lowest energy RMSD of 2.99 Å, and an average ensemble RMSD of 5.23 Å. A novel geometric criterion is applied to speed up calculations. The computational cost of generating 1,000 conformations for each of the x loops in a benchmark dataset is only about 10 cpu minutes for 12-residue loops, compared to *ca* 180 cpu minutes using the FALCm method. Test results on benchmark datasets show that DiSGro performs comparably or better than previous successful methods, while requiring far less computing time. DiSGro is especially effective in modeling longer loops (10–17 residues).

Citation: Tang K, Zhang J, Liang J (2014) Fast Protein Loop Sampling and Structure Prediction Using Distance-Guided Sequential Chain-Growth Monte Carlo Method. PLoS Comput Biol 10(4): e1003539. doi:10.1371/journal.pcbi.1003539

Editor: Roland L. Dunbrack, Fox Chase Cancer Center, United States of America

Received August 29, 2013; **Accepted** February 1, 2014; **Published** April 24, 2014

Funding: This work was supported by NSF DBI 1062328 and DMS- 0800257, http://www.nsf.gov/, and by NIH 1R21GM101552, NIH GM079804 and GM086145, http://www.nih.gov/. This work was also funded by the Chicago Biomedical Consortium with support from the Searle Funds at The Chicago Community Trust, http://chicagobiomedicalconsortium.org/. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: jinfeng@stat.fsu.edu(JZ); jliang@uic.edu (JL)

This is a *PLOS Computational Biology* Methods article.

Introduction

Protein loops connect regular secondary structures and are flexible regions on protein surface. They often play important functional roles in recognition and binding of small molecules or other proteins [1–3]. The flexibility and irregularity of loops make their structures difficult to resolve experimentally [4]. They are also challenging to model computationally [5,6]. Prediction of loop conformations is an important problem and has received considerable attention [5–27].

Among existing methods for loop prediction, template-free methods build loop structures *de novo* through conformational search [5–7,9,10,13,14,17,18,21,23,28]. Template-based methods build loops by using loop fragments extracted from known protein structures in the Protein Data Bank [11,19,27]. Recent advances in template-free loop modeling have enabled prediction of structures of long loops with impressive accuracy when crystal contacts or protein family specific information such as that of GPCR family is taken into account [14,23,25].

Loop modeling can be considered as a miniaturized protein folding problem. However, several factors make it much more challenging than folding small peptides. First, a loop conformation needs to connect two fixed ends with desired bond lengths and angles [8,12]. Generating quality loop conformations satisfying this geometric constraint is nontrivial. Second, the complex interactions between atoms in a loop and those in its surrounding make the energy landscape around near-native loop conformations quite rugged. Water molecules, which are often implicitly modeled in most loop sampling methods, may contribute significantly to the energetics of loops. Hydrogen bonding networks around loops are usually more complex and difficult to model than those in regular secondary structures. Third, since loops are located on the surface of proteins, conformational entropy may also play more prominent roles in the stability of near-native loop conformations [29,30]. Approaches based on energy optimization, which ignore backbone and/or side chain conformational entropies, may be biased toward those overly compact non-native structures. Despite extensive studies in the past and significant progress made in recent years, both conformational sampling and energy evaluation remain challenging problems, especially for long loops (*e.g.*, $n \geq 12$).

Author Summary

Loops in proteins are flexible regions connecting regular secondary structures. They are often involved in protein functions through interacting with other molecules. The irregularity and flexibility of loops make their structures difficult to determine experimentally and challenging to model computationally. Despite significant progress made in the past in loop modeling, current methods still cannot generate near-native loop conformations rapidly. In this study, we develop a fast chain-growth method for loop modeling, called Distance-guided Sequential chain-Growth Monte Carlo (DiSGro), to efficiently generate high quality near-native loop conformations. The generated loops can be used directly for downstream applications or as candidates for further refinement.

In this paper, we propose a novel method for loop sampling, called Distance-guided Sequential chain-Growth Monte Carlo (DiSGro). Based on the principle of chain growth [15,31,32,34,35], the strategy of sampling through sequentially growing protein chains allows efficient exploration of conformational space [15,34–37]. For example, the Fragment Regrowth via Energy-guided Sequential Sampling (FRESS) method outperformed previous methods on folding benchmark HP sequences [15,33]. In addition to HP model [15], sequential chain-growth sampling has been used to study protein packing and void formation [35], side chain entropy [29,38], near-native protein structure sampling [30], conformation sampling from contact maps [39], reconstruction of transition state ensemble of protein folding [40], RNA loop entropy calculation [37], and structure prediction of pseudo-knotted RNA molecules [41].

In this study, we first derive empirical distributions of end-to-end distances of loops of different lengths, as well as empirical distributions of backbone dihedral angles of different residue types from a loop database constructed from known protein structures. An empirical distance guidance function is then employed to bias the growth of loop fragments towards the *C*-terminal end of the loop. The backbone dihedral angle distributions are used to sample energetically favorable dihedral angles, which lead to improved exploration of low energy loop conformations. Computational cost is reduced by excluding atoms from energy calculation using REsidue-residue Distance Cutoff and ELLipsoid criterion, called Redcell. Sampled loop conformations, all free of steric clashes, can be scored and ranked efficiently using an atom-based distance-dependent empirical potential function specifically designed for loops.

Our paper is organized as follows. We first present results for structure prediction using five different test data sets. We show that DiSGro has significant advantages in generating native-like loops. Accurate loops can be constructed by using DiSGro combined with a specifically designed atom-based distance-dependent empirical potential function. Our method is also computationally more efficient compared to previous methods [8,9,18,22,42]. We describe our model and the DISGRO sampling method in detail at the end.

Results

Test set

We use five data sets as our test sets. Test Set 1 contains 10 loops at lengths four, eight, and twelve, for a total of $3\times10=30$ loops from 21 PDB structures, which were described in Table 2 of zRef. [8]. Test Set 2 consists of 53 eight, 17 eleven, and 10 twelve-residue loops from Table C1 of Ref. [42]. Several loop structures were removed as they were nine-residue loops but mislabeled as eight-residue loops: (1awd, 55–63; 1byb, 246–254; and 1ptf, 10–18). Altogether, there are 50 eight-residue loops. Test Set 3 is a subset of that of [5], which was used in the RAPPER and FALCm studies [10,22]. Details of this set can be found in the "Fiser Benchmark Set" section of Ref. [10]. Test Set 4 is taken from Table A1–A6 of Ref. [42]. Test Set 5 contains 36 fourteen, 30 fifteen, 14 sixteen and 9 seventeen-residue loops from Table 3 of Ref. [23]. Test Set 1 and 2 are used for testing the capability of DiSGro and other methods in generating native-like loops. Test Set 3, 4, and 5 are used for assessing the accuracy of predicted loops based on selection from energy evaluation using our atom-based distance-dependent empirical potential function. Our results are reported as global backbone RMSD, calculated using the N, C_α, C and O atoms of the backbone.

Loop sampling

To evaluate our method for producing native-like loop conformations, we use Test Set 1 and 2.

We generate 5,000 loops for each of the 10 loop structures in Test Set 1 at length 4, 8, and 12 residues, respectively. We compare our results with those obtained by CCD [8], CSJD [12], SOS [18], and FALCm [22]. The minimum RMSD among 5,000 sampled loops generated by DiSGro are listed in Table 1, along with results from the four other methods.

Accurate loops of longer length are more difficult to generate. For loops with 12 residues, DiSGro generates more accurate loops than other methods. Our method has a mean of 1.53 Å for the minimum RMSD, compared to 1.81 Å for FALCm, the next best method in the group [22]. The minimum RMSD of nine of the ten 12-residue loops have $\text{RMSD}\leq2$ Å, while five loops of the ten generated by FALCm have $\text{RMSD}>2$ Å. Compared to the CCD, CSJD, and SOS methods, our loops have significantly smaller minimum RMSD (1.53 Å *vs* 3.05, 2.34, and 2.25 Å, respectively, Table 1). The average minimum global backbone RMSD for 12-residue loops can be further improved when we increase the sample size of generated loop conformations. The minimum global RMSD is improved to 1.45 Å, 1.26 Å, and 0.96 Å when the sample size is increased to 20,000, 100,000, and 1,000,000, respectively. Further improvement would likely require flexible bond lengths and angles.

For loops with 8 residues, DiSGro has an average minimum RMSD value smaller than the CCD, CSJD, and SOS methods (0.81 Å *vs* 1.59 Å, 1.01 Å, and 1.19 Å, respectively, Table 1). In eight of the ten 8-residue loops, DiSGro achieves sub-angstrom accuracy ($\text{RMSD}<1$ Å), although the mean of minimum RMSD of 8-residue loops is slightly larger than that from FALCm (0.80 Å *vs* 0.72 Å).

For loops with 4-residue, the mean of the minimum RMSD (0.21 Å) by DiSGro is significantly smaller than those by the CSJD and the CCD methods (0.40 Å and 0.56 Å, respectively), and is similar to those by the SOS and FALCm methods(0.20 Å and 0.22 Å, respectively). Noticeably, three of the ten loops have $\text{RMSD}<0.1$ Å, indicating our sampling method has good accuracy for short loop modeling.

These loops can be generated rapidly. The computing time per conformation averaged over 5,000 conformations for 4, 8, and 12-residues is 4.4, 13, and 20 *ms* using a single AMD Opteron processor of 2 *GHz*. In addition to improved average minimum RMSD, DiSGro seems to take less time than CCD (31, 37, and 23 *ms* on an AMD 1800+ MP processor for the 4, 8, and 12-residue loops), and is as efficient as SOS (5.0, 13, and 19 *ms* for the 4, 8, and 12-residue loops on an AMD 1800+ MP processor).

Table 1. Minimum backbone RMSD values of the loops sampled by five different algorithms.

Length	Loop	CCD	CSJD	SOS	FALCm	DiSGro
12-res	1cruA_358	2.54	2.00	2.39	2.07	1.84
	1ctqA_26	2.49	1.86	2.54	1.66	1.36
	1d4oA_88	2.33	1.60	2.44	0.82	1.50
	1d8wA_46	4.83	2.94	2.17	2.09	1.17
	1ds1A_282	3.04	3.10	2.33	2.10	1.82
	1dysA_291	2.48	3.04	2.08	1.67	1.45
	1eguA_508	2.14	2.82	2.36	1.71	2.13
	1f74A_11	2.72	1.53	2.23	1.44	1.46
	1qlwA_31	3.38	2.32	1.73	2.20	0.79
	1qopA_178	4.57	2.18	2.21	2.36	1.77
	Average	3.05	2.34	2.25	1.81	**1.53**
8-res	1cruA_85	1.75	0.99	1.48	0.62	1.34
	1ctqA_144	1.34	0.96	1.37	0.56	0.70
	1d8wA_334	1.51	0.37	1.18	0.96	0.93
	1ds1A_20	1.58	1.30	0.93	0.73	0.62
	1gk8A_122	1.68	1.29	0.96	0.62	1.08
	1i0hA_145	1.35	0.36	1.37	0.74	0.80
	1ixh_106	1.61	2.36	1.21	0.57	0.39
	1lam_420	1.60	0.83	0.90	0.66	0.63
	1qopB_14	1.85	0.69	1.24	0.92	0.87
	3chbD_51	1.66	0.96	1.23	1.03	0.67
	Average	1.59	1.01	1.19	0.72	**0.80**
4-res	1dvjA_20	0.61	0.38	0.23	0.39	0.31
	1dysA_47	0.68	0.37	0.16	0.20	0.09
	1eguA_404	0.68	0.36	0.16	0.22	0.39
	1ej0A_74	0.34	0.21	0.16	0.15	0.09
	1i0hA_123	0.62	0.26	0.22	0.17	0.13
	1id0A_405	0.67	0.72	0.33	0.19	0.33
	1qnrA_195	0.49	0.39	0.32	0.23	0.19
	1qopA_44	0.63	0.61	0.13	0.30	0.39
	1tca_95	0.39	0.28	0.15	0.09	0.11
	1thfD_121	0.50	0.36	0.11	0.21	0.05
	Average	0.56	0.40	0.20	0.22	**0.21**

Minimum backbone RMSD values of the loops sampled by CCD, CSJD, SOS, FALCm and DiSGro for different loop structures. CCD result was obtained from Table 2 of Ref. [8]. CSJD result was obtained from Table 1 of Ref. [12]. SOS result was obtained from Table 1 of Ref. [18]. FALCm result was obtained from Table 2 of Ref. [22].
doi:10.1371/journal.pcbi.1003539.t001

Reducing the number of trial states in DiSGro can further reduce the computing time, with some trade-off in sampling accuracy. For example, when we take $(m,n) = (10,2)$, the computing time per conformation averaged over 5,000 conformations for 4, 8, and 12-residues is only 3.5, 5.0, and 5.8 *ms*, respectively, with the average minimum RMSDs comparable to those from SOS's (0.29 Å *vs* 0.20 Å, 1.15 Å *vs* 1.19 Å, and 2.24 Å *vs* 2.25 Å for the 4, 8, and 12-residue loops, respectively). Although the CSJD loop closure method has faster computing time (0.56, 0.68, and 0.72 *ms* on AMD 1800+ MP processor), the speed of DiSGro is adequate in practical applications.

We compare DiSGro in generating near-native loops with Wriggling [43], Random Tweak [44], Direct Tweak [42,45], $\mathrm{LOOPY}_{\mathrm{bb}}$ [45], and PLOP-build [13] using Test Set 2. The minimum RMSD among 5,000 loops generated by DiSGro are listed in Table 2, along with results from the other methods obtained from Table 2 in Ref. [42]. Direct Tweak and $\mathrm{LOOPY}_{\mathrm{bb}}$ from the LoopBuilder method and our DiSGro have better accuracy in sampling than Wriggling, Random Tweak, and PLOP-build methods. For loops with 11 and 12-residues, these three methods are the only ones that can generate near-native loop structures with minimal RMSD values below 2 Å. Among these, DiSGro outperforms $\mathrm{LOOPY}_{\mathrm{bb}}$ in generating loops at all three lengths: the average minimal RMSD ($R_{\mathrm{m}in}$) is 1.28 Å *vs*. 1.80 Å for length 12, 1.19 Å *vs*. 1.51 Å for length 11, and 0.80 Å *vs*. 0.89 Å for length 8, respectively. Compared to the Direct Tweak sampling method, DiSGro has improved $R_{\mathrm{m}in}$ for 12-residue loops (1.28 Å *vs* 1.48 Å), slightly improved $R_{\mathrm{m}in}$ for 11-residue loops (1.19 Å *vs* 1.20 Å) and inferior $R_{\mathrm{m}in}$ for 8-residue loops (0.80 Å *vs* 0.69 Å). Overall, these results show that DiSGro are very effective in sampling near-native loop conformations, especially when modeling longer loops of length 11 and 12.

Table 2. Comparison of R_{min} of the loop conformations sampled by DiSGro and six other methods using Test Set 2 used by Ref. [42].

Length	Average minimum backbone RMSD (R_{min})						
	Random Tweak	CCD	Wriggling	PLOP-build	Direct Tweak	LOOPY$_{bb}$	DiSGro
8	1.22	1.20	1.43	0.99	0.69	0.89	**0.80**
11	2.22	2.11	2.24	2.18	1.20	1.51	**1.19**
12	2.64	2.57	2.68	2.69	1.48	1.80	**1.28**

R_{min} denote the average minimum backbone RMSD of the loop ensemble. Random Tweak, CCD, Wriggling, PLOP-build, Direct Tweak and LOOPY$_{bb}$ results were obtained from Table 2 of Ref. [42].

doi:10.1371/journal.pcbi.1003539.t002

Our DiSGro method can generate accurate loops and has significant advantages for longer loops compared to previous methods. Using RMSD values calculated from three backbone atoms N, C_α, and C for all loop lengths lead to the same conclusion.

Loop structure prediction and energy evaluation

To assess the accuracy of loops selected by our specifically designed atom-based distance-dependent empirical potential function, we test DiSGro using Test Set 3 and follow the approach of reference [22] for ease of comparison. Because of the high content of secondary structures, these loops are very challenging to model. In the study of [22], 1,000 backbone conformations with the best scores evaluated by DFIRE potential function [46] were retained after screening 4,000 generated backbone conformations for each loop. Loop closure and steric clash removal were not enforced to the 4,000 conformations. We follow the same procedure, except the DFIRE potential function is replaced by our atom-based distance-dependent empirical potential function. The ensemble of the selected 1,000 backbone conformations are then subjected to the procedure of side-chain construction as described in the Section "Side-chain modeling and steric clash removal". The loop conformations with full side-chains are then scored and ranked by the atom-based distance-dependent empirical potential function. Our results are summarized in Table 3.

We measure the average minimum backbone RMSD R_{min}, the average ensemble RMSD R_{ave}, and the average RMSD of the lowest energy conformations R_{Emin} of the 1,000 loop ensemble with the same length. Overall, DiSGro performs significantly better than FALCm and RAPPER in R_{min}, R_{ave} and R_{Emin} for all loop lengths. Compared to FALCm, DiSGro shows significant advantages in R_{min} on sampling long loops of 10–12 residues. Our method has R_{min} of 1.15 Å compared to 1.45 Å for 10-residue loops, 1.39 Å compared to 1.47 Å for 11-residue loops, and 1.53 Å compared to 1.74 Å for 12-residue loops, respectively. For example, as can be seen in Figure 1, the lowest energy loop (red) of a 12-residue loop in the protein 1scs (residues 199–210) has a 0.9 Å RMSD to the native structure (white). The generated top five lowest energy loops are all very close to the native loop, yet are diverse among themselves.

DiSGro also generates loops with smaller R_{ave} compared to FALCm in loops with length ranging from 4 to 12, indicating DiSGro can generate ensemble of loop conformations with enriched near native conformations. Furthermore DiSGro achieves better modeling accuracy using the atom-based distance-dependent empirical potential function. Compared to FALCm, DiSGro has a R_{Emin} of 1.72 Å *vs* 1.87 Å for 8-residue loops, 1.82 Å *vs* 2.08 Å for 9-residue loops, 2.33 Å *vs* 3.09 Å for 10-residue loops, 2.98 Å *vs* 3.43 Å for 11-residue loops, and 2.99 Å *vs* 3.84 Å for 12-residue loops, respectively.

DiSGro is also much faster than other methods. The reported typical computational cost of FALCm is 180 cpu minutes for 8–12 residue loops on a Linux server of a 2.8 *GHz* 2-core Intel Xeon processor [47]. The computation cost for DiSGro method is only 6 and 10 cpu minutes for 10 and 12–residue loops on a single 2 *GHz* AMD Opteron processor, respectively. In addition, FALCm has a size restriction, and it only works with proteins with <500 residues. In contrast, the overall protein size has no effect on the computational efficiency of DiSGro since the numbers of atoms for energy calculation that are retained by the ellipsoid criterion are bounded.

The LOOPER method is an accurate and efficient loop modeling method using a minimal conformational sampling

Table 3. Comparison of $R_{\min}$, R_{ave} and R_{Emin} of the lowest energy conformations of the loops sampled by RAPPER, FALCm4 and DiSGro using Test Set 3.

Length	# of Targets	RAPPER			FALCm			DiSGro		
		R_{min}	R_{ave}	R_{Emin}	R_{min}	R_{ave}	R_{Emin}	R_{min}	R_{ave}	R_{Emin}
4	35	0.43	1.65	0.86	0.33	0.92	0.54	**0.21**	**0.66**	**0.48**
5	35	0.53	2.27	1.00	0.44	1.63	0.92	**0.25**	**1.11**	**0.84**
6	36	0.69	3.06	1.85	0.47	2.34	1.36	**0.44**	**1.74**	**1.22**
7	38	0.78	3.79	1.51	0.58	2.74	1.17	**0.55**	**2.23**	**1.08**
8	32	1.11	4.16	2.11	0.84	3.69	1.87	**0.80**	**2.87**	**1.72**
9	37	1.29	5.00	2.58	0.95	4.21	2.08	**0.94**	**3.64**	**1.82**
10	37	1.67	5.66	3.60	1.45	5.07	3.09	**1.15**	**3.96**	**2.33**
11	33	1.99	6.71	4.25	1.47	5.76	3.43	**1.39**	**4.96**	**2.98**
12	34	2.21	6.96	4.32	1.74	6.31	3.84	**1.53**	**5.23**	**2.99**

R_{min}, R_{ave} and R_{Emin} denote the average minimum backbone RMSD, the average ensemble RMSD and the average RMSD of the lowest energy conformations of the 1,000 loop ensemble with the same length, respectively.

doi:10.1371/journal.pcbi.1003539.t003

method combined with energy minimization [17]. The test set used in the LOOPER study is the original Fiser data set without removal of any loops. Therefore, it is different from Test Set 3 used in the RAPPER and FALCm studies [10,22]. For ease of comparison, we compare DiSGro to the LOOPER using the test set with 10–12-residue loops from [17]. Our results are summarized in Table 4.

We denote $R_{\mathrm{B}kb,\mathrm{ave}}$ and $R_{\mathrm{B}kb,\mathrm{med}}$ as the mean and median of backbone RMSD of the lowest energy conformations with the same loop length. Similarly, we use $R_{\mathrm{A}tm,\mathrm{ave}}$, and $R_{\mathrm{A}tm,\mathrm{med}}$ to denote the mean and median RMSD values of all-heavy atoms. DiSGro shows improved prediction accuracy compared to LOOPER in both backbone and all-heavy atom RMSD. For the 40 loops of length 12, $R_{\mathrm{B}kb,\mathrm{ave}}$ is 3.20 Å compared to 4.08 Å, while the median $R_{\mathrm{B}kb,\mathrm{med}}$ is 2.39 Å compared to 3.80 Å. It also has better all-heavy atom RMSD of 3.39 Å/3.18 Å (mean/median), compared to 3.58 Å/3.35 Å for 10-residue loops, 3.58 Å/3.30 Å compared to 4.30 Å/3.60 Å for 11-residue loops, and 4.18 Å/3.60 Å compared to 5.22 Å/4.96 Å for 12-residue loops.

It is worth noting that DiSGro outperforms LOOPER in speed as well. For a loop with 10 residues, the time cost of DiSGro is 6 minutes using a 2 *GHz* CPU versus 40 cpu minutes using a 3 *GHz* processor according to Figure 7 in the LOOPER paper [17].

Prior publications also allowed us to compare results in loop structure predictions based on energy discrimination using Test Set 4 with results obtained using the LoopBuilder method [42]. Following [42], we generated 1,000 closed loop conformations for eight-residue loops, 2,000 for nine-residue loops, 5,000 for ten, eleven, and twelve-residue loops, and 8,000 for thirteen-residue loops. Energy calculations are carried out using our atom-based distance-dependent empirical potential function. The average RMSD of the lowest energy conformations, R_{Emin}, are then compared between these two methods. The results are summarized in Table 5.

Compared to LoopBuilder, DiSGro has better R_{Emin}: 1.83 Å *vs* 1.88 Å for 9-residue loops, 1.83 Å *vs* 1.93 Å for 10-residue loops, 2.38 Å *vs* 2.50 Å for 11-residue loops, 2.62 Å *vs* 2.65 Å for 12-residue loops, and 3.26 Å *vs* 3.74 Å for 13-residue loops, respectively. DiSGro has inferior performance in selecting R_{Emin} for 8-residue loops (1.59 Å *vs* 1.31 Å). The average time using LoopBuilder for twelve-residue loops was around 4.5 hours or 270 minutes, while the computational time using DiSGro is around 10 minutes. Overall, DiSGro has equal or slightly better performance than LoopBuilder in average prediction accuracy of loop structures with far less computing time.

To test the feasibility of DiSGro in modeling longer loops with length >12, we use the Fiser 13-residue loops data set to generate and select low energy loop conformations. 1,000 conformations with low energy are obtained. The mean of minimum backbone RMSD $R_{\mathrm{m}in}$ of 40 loops with 13-residue is 1.76 Å, and the median is 1.61 Å. The mean/median of the backbone RMSD $R_{Bkb,Emin}$, and all heavy atom RMSD $R_{\mathrm{A}tm,Emin}$ of the lowest energy conformations are 2.91 Å/2.53 Å and 3.84 Å/3.29 Å, respectively (Table 6).

With extensive conformational sampling using molecular mechanics force field, the Protein Local Optimization Program (PLOP) can predict highly accurate loops [13,14,23]. We tested DiSGro using Test Set 5 consisting of 89 loops with length 14–17 and compared results with those using PLOP. Here the sampling and scoring processes were similar to those used in Test Set 3, except 100,000 backbone conformations were generated. We measured the average minimum backbone RMSD $R_{\mathrm{m}in}$ and the

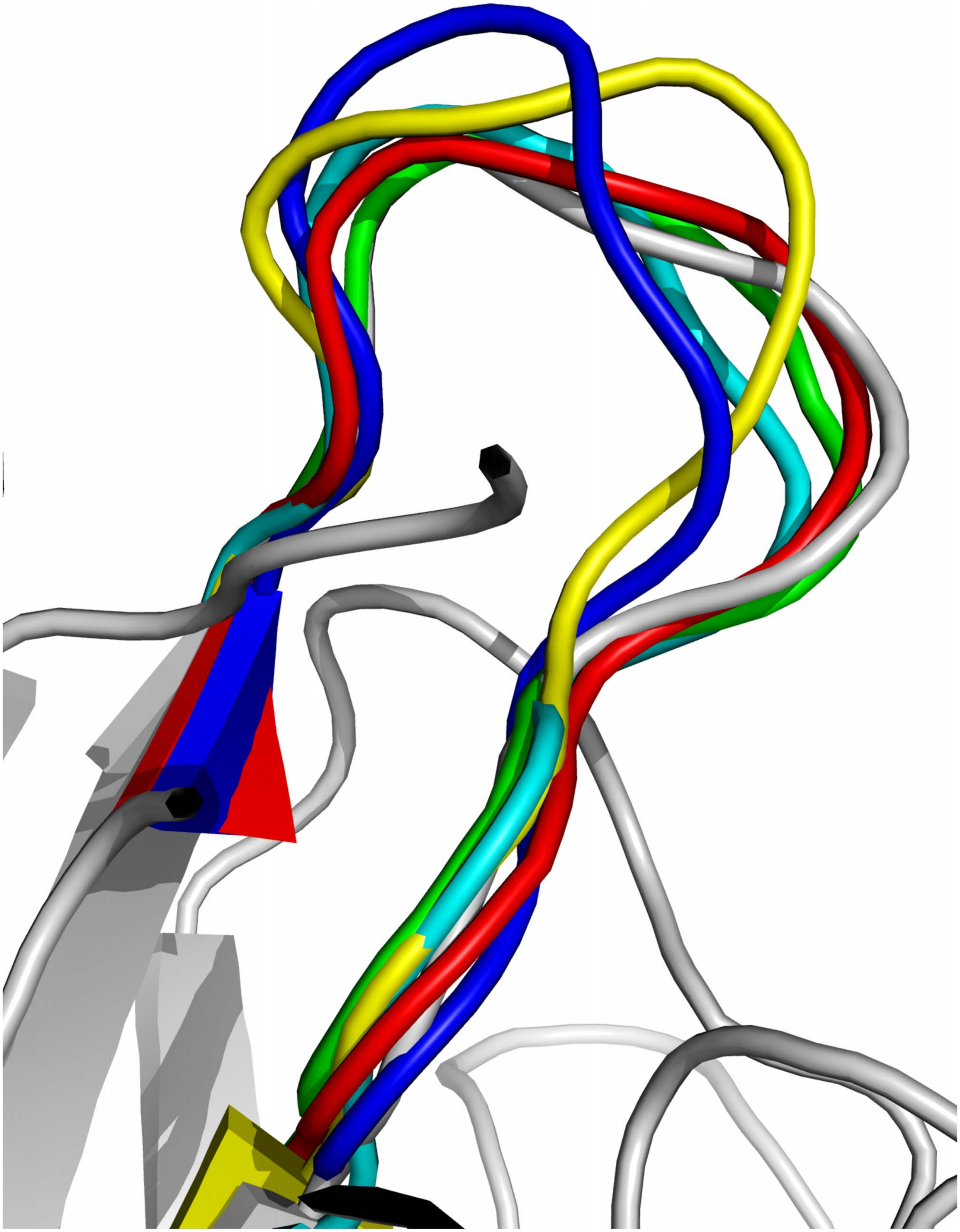

Figure 1. Top five lowest energy loops of length 12 for single-metal-substituted concanavalin A (pdb 1scs, residues 199–210). The lowest energy loop after side-chain construction is colored in red, and the native structure is in white.
doi:10.1371/journal.pcbi.1003539.g001

Table 4. Comparison of accuracy of modeled loops using the original Fiser data set of loops with 10–12 residues.

Length	Targets	DiSGro/LOOPER			
		$R_{Bkb,ave}$	$R_{Bkb,med}$	$R_{Atm,ave}$	$R_{Atm,med}$
10	40	**2.30**/2.66	**2.20**/2.39	**3.39**/3.58	**3.18**/3.35
11	40	**2.63**/3.35	**2.25**/2.76	**3.58**/4.30	**3.30**/3.60
12	40	**3.20**/4.08	**2.39**/3.80	**4.18**/5.22	**3.60**/4.96

The accuracy achieved by LOOPER and DiSGro at different loop length using the original Fiser data set of loops with 10–12 residues is listed. $R_{Bkb,ave}$, and $R_{Bkb,med}$ denote the mean and median of backbone RMSD, while $R_{Atm,ave}$, and $R_{Atm,med}$ denote the mean and median of all-heavy atoms RMSD of the lowest energy conformations with the same loop length.
doi:10.1371/journal.pcbi.1003539.t004

average RMSD of the lowest energy conformations R_{Emin}. Our results are summarized in Table 7.

Loops predicted by the PLOP method have smaller $R_{E_{\min}}$ compared to DiSGro [23], although DiSGro samples well and gives small $R_{\min}$ of 1.58 Å for 14-residue loops, 1.80 Å for 15-residue loops, 1.88 Å for 16-residue loops, and 2.18 Å for 17-residue loops. For loops of length 17, the R_{min} of 2.18 Å is less than the reported $R_{Emin} = 2.30$ Å using PLOP, although it is unclear whether the R_{min} of loops generated by PLOP is less than 2.18 Å. Overall, DiSGro is capable of successfully generating high quality near-native long loops, up to length 17. The accuracy of R_{Emin} of loops generated by DiSGro may be further improved by using a more effective scoring function.

We also compared the computational costs of the two methods. The average computing time for DiSGro is 0.73, 0.72, 0.81, and 0.95 hours for loops of lengths 14, 15, 16, and 17 using a single core AMD Opteron processor 2350, respectively, which is more than two orders of magnitude less than the time required for the PLOP method (216.0, 309.6, 278.4, and 408.0 hours for loops of length 14, 15, 16, and 17 residues, respectively).

Improvement in computational efficiency

We used a REsidue-residue Distance Cutoff and ELLipsoid criterion (Redcell) to improve the computational efficiency. To assess the effectiveness of this approach, we carry out a test using a set of 140 proteins (see discussion of the tuning set in Materials and Methods). We compared the time cost of energy calculation of generating a single loop, with and without this procedure. When the procedure is applied, we only calculate the pairwise atom-atom distance energy between atoms in loop residues and other atoms within the ellipsoid. When the procedure is not applied, we calculate energy function between atoms in loop residues and all other atoms in the rest of the protein. The computational cost of energy calculations for sampling single loops with 12 and 6-residues are shown in Figure 2A and Figure 2B, respectively.

From Figure 1, we can see that significant improvement in computational cost is achieved. The average time cost using our procedure is reduced from 82.3 *ms* to 6.0 *ms* for sampling 12-residue loops, and 39.4 *ms* to 2.0 *ms* for 6-residue loops. In addition, this approach makes the time cost of energy calculations independent of the protein size (Figure 2A and Figure 2B), whereas the computing time without applying this procedure increases linearly with the protein size. The improvement is especially significant for large proteins. For example, to generate a 15-residue loop in a protein with 1,114 residues, the computing time is improved from 93.7 *ms* to 1.8 *ms*, which is more than 50-fold speed-up. Detailed examination indicates that both distance cutoff and the ellipsoid criterion contribute to the computational efficiency. Furthermore, the full Redcell procedure has improved efficiency over using either "Ellipsoid Criterion Only" or "Cutoff Criterion Only". The computing time for generating a 15-residue loops is 2.0 *ms* when the full Redcell procedure is applied, compared to 5.3 *ms*, and 3.9 *ms*, when only the ellipsoid criterion and only the distance-threshold are used, respectively (Figure 2C). Furthermore, there is no loss of accuracy in energy evaluation. Overall, Redcell improves the computational cost by excluding many atoms from collision detections and energy calculations, with significant reduction in computation time, especially for large proteins.

Discussion

In this study, we presented a novel method Distance-guided Sequential chain-Growth Monte Carlo (DiSGro) for generating

Table 5. Comparison of R_{Emin} of the loop conformations sampled by Loop Builder and DiSGro using Test Set 4 taken from the Loop Builder study [42].

		Average prediction accuracy (R_{Emin})	
Length	# of Targets	LoopBuilder	DiSGro
8	63	1.31	**1.59**
9	56	1.88	**1.83**
10	40	1.93	**1.83**
11	54	2.50	**2.38**
12	40	2.65	**2.62**
13	40	3.74	**3.26**

R_{Emin} denote the average RMSD of the lowest energy conformations of the loop ensemble. Results of LoopBuilder were obtained from Table 5 of Ref. [42].
doi:10.1371/journal.pcbi.1003539.t005

Table 6. Accuracy of modeled loops by DiSGro using the original Fiser data set of loops with 13 residues.

Target	PDB	Start	End	Sequence	$R_{\mathbf{m}in}$	$R_{\mathbf{en},Ave}$	$R_{\mathbf{B}kb,Emin}$	$R_{\mathbf{A}tm,Emin}$
1	154l	21	33	akpeglsycgvsa	2.14	7.08	2.42	3.20
2	1aba	5	17	ygydsnihkcgpc	1.83	4.37	5.08	5.58
3	1amp	252	264	nprihttqdtlan	2.12	6.03	2.88	4.32
4	1art	160	172	yydaenhtldfda	2.67	7.28	5.62	6.80
5	1byb	139	151	vdnepifhgrtai	2.85	9.42	3.98	5.02
6	1cbn	32	44	ciiipgatcpgdy	1.50	6.34	2.12	3.28
7	1cgt	38	50	aydatcsnlklyc	2.38	6.90	4.47	4.74
8	1clc	258	270	mqypdgsgrvahk	2.25	5.17	4.45	5.72
9	1ctm	34	46	evpqavlpdtvfe	0.92	4.98	1.21	1.98
10	1fas	4	16	yshtttsrailtn	1.56	8.21	1.73	2.57
11	1fnd	47	59	kitgddapgetwh	1.41	10.25	1.51	2.02
12	1frd	37	49	lpfschsgscssc	1.56	8.78	5.83	6.46
13	1fus	91	103	thtgasgnnfvgc	2.36	6.61	3.33	4.78
14	1gof	70	82	mlprqdgnqngwi	1.17	3.21	2.66	2.95
15	1ivd	429	441	grkqetrvwwtsn	2.48	5.76	2.82	5.60
16	1l58	50	62	igrncngvitkde	1.62	12.03	2.13	3.30
17	1msc	33	45	issnsrsqaykvt	2.20	6.74	5.33	6.18
18	1osa	55	67	vdadgngtidfpe	0.82	10.27	0.82	1.35
19	1pca	204	216	ypygyktqspadk	2.49	7.25	5.99	6.30
20	1php	59	71	hlgrpkgkvveel	1.40	8.61	1.40	2.16
21	1prn	213	225	ydnglstagdqvt	1.41	5.32	1.55	1.72
22	1rec	159	171	fgkkdddklteke	1.18	9.66	1.18	2.35
23	1srp	46	58	wngykvfgqpvkl	1.02	7.58	1.90	2.24
24	1thg	493	505	dpnvgtnllqwdq	1.39	5.42	2.11	2.97
25	1thw	102	114	isnikgfnvpmdf	1.31	3.75	1.87	2.86
26	1trb	222	234	lrdtqnsdniesl	1.56	4.42	1.64	2.83
27	1xif	99	111	fkdggftandrdv	1.21	7.02	3.23	4.23
28	2ctc	204	216	ypygyttqsipdk	2.57	6.42	4.28	4.76
29	3cyr	34	46	hhlvdgkesyakc	2.18	5.89	5.62	6.17
30	2exo	51	63	tepsqnsfsfgag	2.64	7.58	2.64	3.98
31	2pia	58	70	slcndsqernryv	1.29	3.83	2.80	4.12
32	2por	170	182	idspdtalmadme	1.27	6.20	1.60	2.23
33	2sil	86	98	iyndrvnsklsrv	1.19	5.82	3.37	4.00
34	3grs	129	141	haaftsdpkptie	1.61	9.05	1.78	3.03
35	4icb	53	65	ldkngdgevsfee	1.84	9.06	2.01	2.95

Table 6. Cont.

Target	PDB	Start	End	Sequence	$R_{\min}$	$R_{en,Ave}$	$R_{Bkb,Emin}$	$R_{Atm,Emin}$
36	5fx2	93	105	cgdssyeyfcgav	2.27	5.15	4.61	6.57
37	5p21	115	127	gnkcdlaartves	1.79	7.09	2.81	3.79
38	5pti	9	21	pytgpckariiry	1.15	5.66	1.42	3.07
39	7rsa	86	98	etgsskypncayk	1.84	11.38	1.84	2.73
40	8dfr	166	178	padiqeedgiqyk	1.97	10.08	2.21	2.68
Mean					**1.76**	**7.04**	**2.91**	**3.84**
Median					**1.61**	**6.82**	**2.53**	**3.29**

R_{min} and $R_{en,Ave}$ are the minimum backbone RMSD and the average backbone RMSD of the 1,000 sampled conformations, respectively. $R_{Bkb,Emin}$ and $R_{Atm,Emin}$ are the backbone and all heavy atoms RMSD of the lowest energy conformations in the ensemble.
doi:10.1371/journal.pcbi.1003539.t006

protein loop conformations and predicting loop structures. Ensembles of near-native loop conformations can be efficiently generated using the DiSGro method. DiSGro has better average minimum backbone RMSD, $R_{\min}$, compared to other loop sampling methods. For example, $R_{\min}$ is 1.53 Å for 12-residue loops when using DiSGro, while the corresponding values are 3.05 Å, 2.34 Å, 2.25 Å, and 1.81 Å when using the CCD, CSJD, SOS, and the FALCm method.

DiSGro also performs well in identifying native-like conformations using atom-based distance-dependent empirical potential function. In comparison with other similar loop modeling methods, DiSGro demonstrated improved modeling accuracy, in terms of an average RMSD of the lowest energy conformations R_{Emin} for the more challenging task of sampling longer loops of 10–13 residues. For example, DiSGro outperforms FALCm [22] (2.33 Å *vs* 3.09 Å) and LOOPER [17] (2.30 Å *vs* 2.66 Å) in predicting 10-residue loops, while taking less computing time (6 minutes *vs* 180 minutes for FALCm and 40 minutes for LOOPER. Compared to LoopBuilder [42], DiSGro also has better R_{Emin}: For 13-residue loops, the R_{Emin} is 3.26 Å using DiSGro, but is 3.74 Å when using the Loop Builder. The average computing time is also faster when using DiSGro: it takes about 6 minutes to predict structures of 10-residue loops and 10 minutes for 12-residue loops. DiSGro also works well for short loops, although this may be largely a reflection of the underlying analytical closure method [12].

There are a number of directions for further improvement. DiSGro can be further improved by adding fragments of peptides when growing loops instead of adding individual residues. Fragment-based approach has been widely used in protein structure prediction [48–51] and specifically in loop structure prediction [21]. It is straightforward to apply the strategy described in this study for fragment-based growth, and it will likely lead to improved sampling efficiency further and enable longer loops to be modeled. Furthermore, the energy function employed here can be further improved by optimization such as those obtained by training with challenging decoy loops using nonlinear kernel [52], and/or using rapid iterations through a physical convergence function [53,54]. In addition, DiSGro is compatible with different loop closure methods [8,12,22], and experimenting with other closure strategy may also lead to further improvement.

An efficient loop sampling method such as DiSGro can help to improve overall modeling of loop structures. Currently, the hierarchical approach of the Protein Local Optimization Program (PLOP) [13,14,23] gives excellent accuracy in protein loop modeling, but requires significant computational time. The average time cost of modeling a 13-residue loop is about 4–5 days [23]. Kinematic closure (KIC) method can also make very accurate predictions of 12-residue loops [21]. However, KIC also requires substantial computation, with about 320 CPU hours on a single 2.2 *GHz* Opteron processor for predicting 12-residue loops [21]. As suggested earlier by Spassov et al [17], an efficient loop modeling method combined with energy minimization may overcome the obstacle of high computational cost. By generating high quality initial structures using DiSGro, near native conformations of loops can be used as candidates for further refinement.

Materials and Methods

Protein structures representation

All heavy atoms in the backbone and side chain of a protein loop are explicitly modeled. The bond lengths b and angles θ are taken from standard values specific to residue and atom type [55].

Table 7. Comparison of $R_{\min}$, R_{Emin} and *Time* of the loop conformations sampled by PLOP and DiSGro using Test Set 5.

Length	# of Targets	PLOP			DiSGro		
		R_{min}	R_{Emin}	*Time* (hours|days)	R_{min}	R_{Emin}	*Time* (hours)
14	36	NA	1.19	216.0|9.0	**1.58**	**3.73**	**0.73**
15	30	NA	1.55	309.6|12.9	**1.80**	**3.91**	**0.72**
16	14	NA	1.43	278.4|11.6	**1.88**	**4.16**	**0.81**
17	9	NA	2.30	408.0|17.0	**2.18**	**4.46**	**0.95**

R_{min} and R_{Emin} denote the average minimum backbone RMSD and the average RMSD of the lowest energy conformations of the loop ensemble.
doi:10.1371/journal.pcbi.1003539.t007

The backbone dihedral angles (ϕ,ψ,ω) and side chain dihedral angles χ constitute all the degrees of freedom (DOFs) in our model.

Distance-guided Sequential chain-Growth Monte Carlo (DiSGro)

In order to efficiently generate adequate number of native-like loop conformations, we have developed a Distance-guided Sequential chain-Growth Monte Carlo (DiSGro) method.

Let the loop to be modeled begins at residue t and ends at residue l. The sequence of the positions of backbone heavy atoms from C atom of residue t to C_α (CA) atom of residue l are unknown and need to be generated. We assume that the backbone atoms before and after this fragment are known. Coordinates of side chain atoms are also unknown and need to be generated if the coordinates of the CA atoms they are attached to are unknown.

At each step of the chain growth process, we generate three consecutive backbone atoms continuing from the backbone atom sampled at the previous step. At the $(i-t)$-th growth step $(t \le i < l)$, the three backbone atoms are C atom of residue i, N atom of residue $i+1$, and C_α atom of residue $i+1$ (Figure 3). The coordinates of the three atoms, C_i, N_{i+1} and CA_{i+1}, are denoted as $\mathbf{x}_{C,i}$, $\mathbf{x}_{N,i+1}$, and $\mathbf{x}_{CA,i+1}$, respectively. The ω dihedral angles that determine the coordinate of C_α atoms are sampled from a normal distribution with mean 180° and standard deviation 4°. In the next section, we describe in detail in sampling of the dihedral angles (ϕ,ψ), which determine the coordinates of the C and the N atoms.

Sampling backbone (ϕ,ψ) angles. Without loss of generality, we describe the sampling procedure for C_i and N_{i+1} atoms at the $(i-t)$-th growth step. C_i is generated first, followed by N_{i+1}. Denote the distance between $\mathbf{x}_{CA,i}$ and $\mathbf{x}_{C,l}$ as $d_{CA_i,C_l} = |\mathbf{x}_{C,l} - \mathbf{x}_{CA,i}|$, and the distance between $\mathbf{x}_{C,i}$ and $\mathbf{x}_{C,l}$ as $d_{C_i,C_l} = |\mathbf{x}_{C,i} - \mathbf{x}_{C,l}|$. Since the bond angle $\theta_{C,i}$ formed by the $N_i - CA_i$ and $CA_i - C_i$ bonds is fixed, and the bond length b_{CA_i,C_i} is also fixed, C_i will be located on a circle $\mathbf{C_C}$ (Figure 3):

$$\begin{aligned}\mathbf{C_C} = \{\mathbf{x} \in \mathbb{R}^3 | \text{such that} ||\mathbf{x} - \mathbf{x}_{CA,i}|| = b_{CA_i,C_i} \\ \text{and } (\mathbf{x} - \mathbf{x}_{CA,i}) \cdot (\mathbf{x}_{CA,i} - \mathbf{x}_{N,i}) = \cos\theta_{C,i}\}.\end{aligned} \quad (1)$$

Given a fixed d_{C_i,C_l}, C_i can be placed on two positions $\mathbf{x}_{C,i}$ and $\mathbf{x}_{C',i}$ on circle $\mathbf{C_C}$ (Figure 3, $\mathbf{x}_{C,i}$ and $\mathbf{x}_{C',i}$ are labeled as C_i and C'_i, respectively.) As the probability for placing C_i on either position is about equal based on our analysis, we randomly select one position to place atom C_i.

In principle, sampling from the empirical distributions of d_{C_i,C_l} and mapping back to C_i should encourage the growth of loops to connect to the terminal C_l atom. Further analysis of the empirical distribution of d_{C_i,C_l} given d_{CA_i,C_l} shows that d_{CA_i,C_l} can be very informative for sampling d_{C_i,C_l} in some cases. This lead us to design the sampling of $\mathbf{x}_{C_i}$ based on the conditional distribution of $\pi(d_{C_i,C_l}|d_{CA_i,C_l})$. See below for details.

Generating atom N_{i+1} is similar to generating C_i, only N_{i+1} instead of C_i is placed on a circle $\mathbf{C_N}$:

$$\begin{aligned}\mathbf{C_N} = \{\mathbf{x} \in \mathbb{R}^3 | \text{such that} ||\mathbf{x} - \mathbf{x}_{C,i}|| = b_{C_i,N_{i+1}} \\ \text{and } (\mathbf{x} - \mathbf{x}_{C,i}) \cdot (\mathbf{x}_{C,i} - \mathbf{x}_{CA,i}) = \cos\theta_{N,i+1}\},\end{aligned} \quad (2)$$

where $b_{C_i,N_{i+1}}$ is the bond length between atom C_i and atom N_{i+1}, and the distance between $\mathbf{x}_{N,i+1}$ and $\mathbf{x}_{C,l}$ is $d_{N_{i+1},C_l} = |\mathbf{x}_{N,i+1} - \mathbf{x}_{C,l}|$. Similarly, atom N_{i+1} is placed by

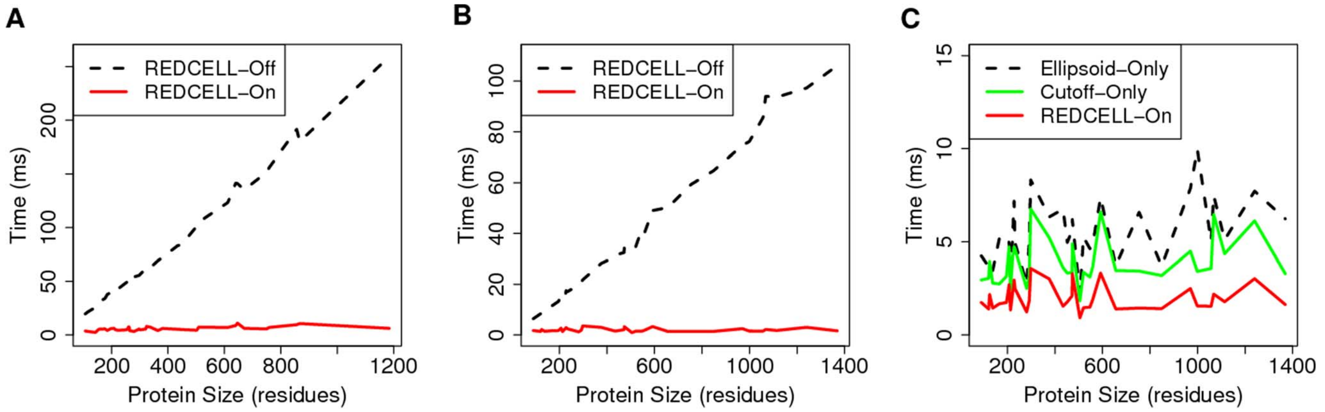

Figure 2. The time cost of energy calculations for generating one single loop. (A) The plot of computing time versus protein size show a large time saving of "Redcell-On" (red solid curve) compared to "Redcell-Off" (black dashed curve) for 12-residue loops, and (B) The plot of 6-residue loops. (C) Plot of computing time versus protein size show "Redcell-On" (red solid curve) has significantly improved computational time cost compared to "Ellipsoid-Only" (black dashed curve) and "Cutoff-Only" (green solid curve).
doi:10.1371/journal.pcbi.1003539.g002

sampling d_{N_{i+1},C_l} condition on d_{C_i,C_l} from the empirical conditional density $\pi(d_{N_{i+1},C_l}|d_{C_i,C_l})$. We repeat this process m times to generate m trial positions of C_i, N_{i+1}, and CA_{i+1}.

Sampling d_{C_i,C_l} and d_{N_{i+1},C_l} from conditional distributions. We sample d_{C_i,C_l} from the conditional distribution $\pi(d_{C_i,C_l}|d_{CA_i,C_l})$ to obtain the location of C_i atom. We first construct the empirical joint distribution $\pi(d_{CA_i,C_l},d_{C_i,C_l})$ by collecting $(d_{CA_i,C_l},d_{C_i,C_l})$ pairs over all loops in a loop database derived from the CulledPDB database (version 11118, at 30% identity, 2.0 Å resolution, and with $R=0.25$) [56]. From the 6,521 protein structures in the CulledPDB, we remove 7 PDB structures which appear in our test data set. For the rest of 6,514 protein structures, loop regions were identified using the secondary structure information either directly from the PDB records or from classification provided by the DSSP software [57]. All random coil regions, including α-helices and β-strands with length <4 amino acids, are included in our database. In total, we have 49,336 loop structures.

For each set of loops with the same residue separation $(l-i)$, $(d_{CA_i,C_l},d_{C_i,C_l})$ are Winsorised at 99.9% level [58]. Specifically, the extreme values above 99.9% are replaced by the values at the 99.9 percentile. We then use a nonparametric two-dimensional Gaussian kernel density estimator to construct a smooth bivariate distribution $\pi(d_{CA_i,C_l},d_{C_i,C_l})$ based on collected data. To estimate the probability density at a point $\mathbf{u}=(d_{CA_i,C_l},d_{C_i,C_l})\in\mathbb{R}^2$, we use the observed n pairs of data from the database $(\mathbf{x}_1,\cdots\mathbf{x}_n)=((d_{CA_i,C_l,1},d_{C_i,C_l,1}),\cdots(d_{CA_i,C_l,n},d_{C_i,C_l,n}))$ to derive the density function $\pi(\mathbf{u})$, which takes the form of:

$$\pi(\mathbf{u})=\frac{1}{n}\sum_{i=1}^{n}|\mathbf{H}|^{-\frac{1}{2}}\mathbf{K}[\mathbf{H}^{-\frac{1}{2}}\cdot(\mathbf{u}-\mathbf{x}_i)], \tag{3}$$

where $\mathbf{H}$ is the symmetric and positive definite bandwidth 2×2 matrix, $\mathbf{K}$ is a bivariate gaussian kernel function:

$$\mathbf{K}(x)=\frac{e^{(-\frac{1}{2}x^T x)}}{2\pi}. \tag{4}$$

To construct the bandwidth matrix $\mathbf{H}$, we calculate the standard deviation $\sigma_{d_{CA_i,C_l}}$ of the n pairs of $(d_{CA_i,C_l},d_{C_i,C_l})$. The corresponding entry $h_{d_{CA_i,C_l}}$ in the bandwidth matrix $\mathbf{H}$ is set as $h_{d_{CA_i,C_l}}=\sigma_{d_{CA_i,C_l}}(\frac{1}{n})^{\frac{1}{6}}$. Similarly, $h_{d_{C_i,C_l}}$ is set as $h_{d_{C_i,C_l}}=\sigma_{d_{C_i,C_l}}(\frac{1}{n})^{\frac{1}{6}}$. The bandwidth matrix $\mathbf{H}$ is then assembled as [59]:

$$\mathbf{H}=\begin{pmatrix} h_{d_{CA_i,C_l}} & h_{d_{C_i,C_l}} \\ h_{d_{C_i,C_l}} & h_{d_{CA_i,C_l}} \end{pmatrix}. \tag{5}$$

We partition the domain of $(d_{CA_i,C_l},d_{C_i,C_l})$ into a grid with 32 grid points in each direction. $\pi(d_{CA_i,C_l},d_{C_i,C_l})$ are estimated at the grid points, and interpolated by a bilinear function elsewhere. Conditional distribution $\pi(d_{C_i,C_l}|d_{CA_i,C_l})$ is constructed from the joint distribution $\pi(d_{CA_i,C_l},d_{C_i,C_l})$ when d_{CA_i,C_l} is fixed. d_{C_i,C_l} is sampled from $\pi(d_{C_i,C_l}|d_{CA_i,C_l})$. We follow the same procedure to construct $\pi(d_{N_{i+1},C_l}|d_{C_i,C_l})$, which is used to sample d_{N_{i+1},C_l}.

Backbone dihedral angle distributions from the loop database. Although the empirical conditional distributions can efficiently guide chain growth to generate properly connected loop conformations, the dihedral angles of the loops are often not energetically favorable. As a result, conditional distributions described above alone are not sufficient in generating near native loop conformations.

The problem can be alleviated by an additional step of selecting a subset of n loops with low-energy dihedral angles from generated samples. We use empirical distributions of the loop dihedral angles obtained from the loop database. Specifically, for the m sampled positions of the current residue i of type α_i with dihedral angles $(\phi^1,\psi^1),..(\phi^m,\psi^m)$, we select $n<m$ samples following an empirically derived backbone dihedral angle distribution $\pi(\phi_i,\psi_i,\alpha_i)$. Here $\pi(\phi_i,\psi_i,\alpha_i)$ is derived from the same protein loop structure database for conditional distance distributions and constructed by counting the frequencies of (ϕ,ψ) pairs for each residue type.

Determining the number of trial states at each growth step for backbone torsion angles. It is important to determine the appropriate size of trial states m and n for generating backbone conformations, as small m and n values may lead to insufficient sampling, resulting in inaccurate loop conformations. On the other hand, very large m and n values will require significantly more computational time, without significant gain in accuracy.

We use a data set, denoted as *tuning-set* to determine the optimal values of parameters m and n for sampling backbone conforma-

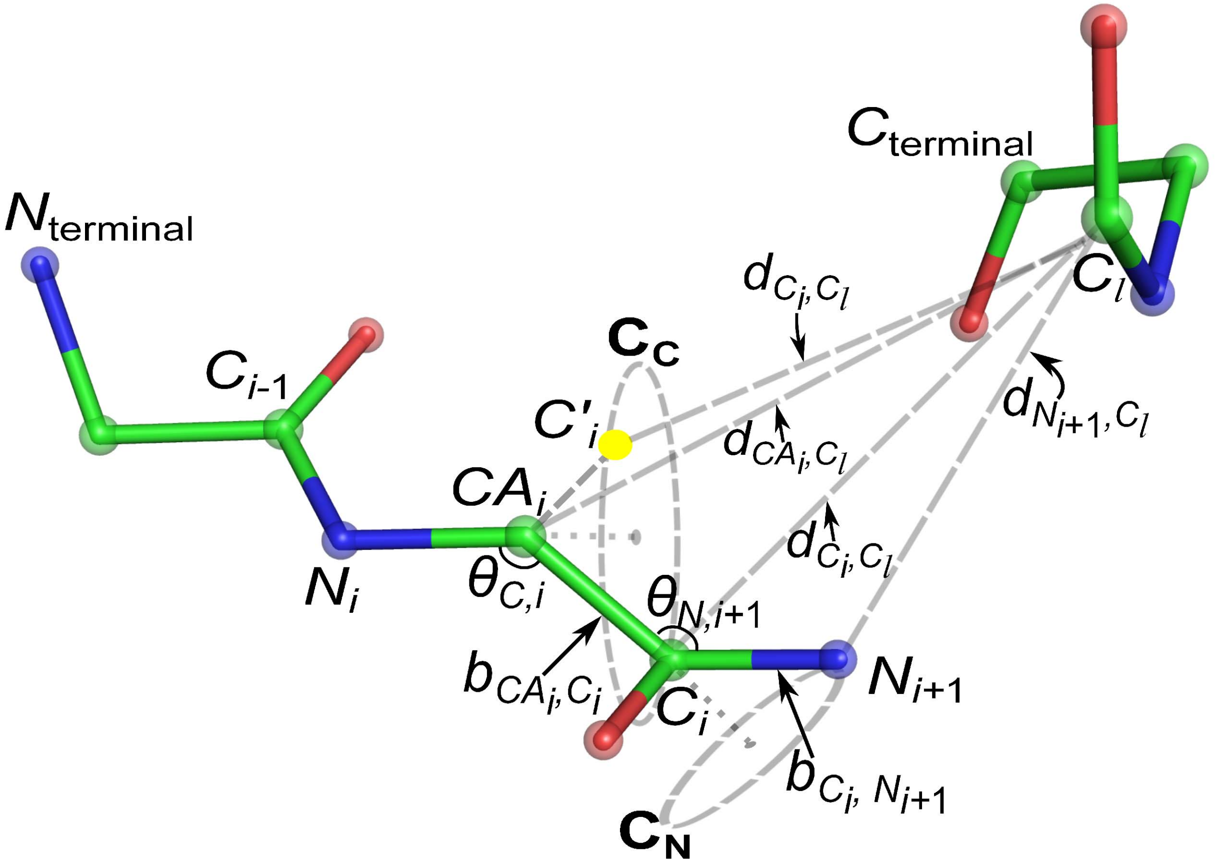

Figure 3. Schematic illustration of placing C_i and N_{i+1} atoms. Atom C_i has to be on the circle $\mathbf{C_C}$. The position $\mathbf{x}_{C,i}$ of the C_i atom of residue i is determined by d_{C_i,C_l}, which is based on known distance d_{CA_i,C_l} and the conditional distribution of $\pi(d_{C_i,C_l}|d_{CA_i,C_l})$. Once d_{C_i,C_l} is sampled, C_i can be placed on two positions with equal probabilities. Here $\mathbf{x}_{C,i}$ is the selected position of C_i. C'_i (yellow ball) is placed at the position $\mathbf{x}_{C',i}$ alternative to $\mathbf{x}_{C,i}$. Similarly, the N_{i+1} atom has to be on the circle $\mathbf{C_N}$ and its position $\mathbf{x}_{N,i+1}$ is determined by d_{N_{i+1},C_l} in a similar fashion.
doi:10.1371/journal.pcbi.1003539.g003

tions. Part of this data set comes from that of Soto *et al* [42]. The rest are randomly selected from pre-compiled CulledPDB (with $\leq$20% sequence identity, $\leq$1.8 Å resolution, and $R \leq 0.25$). It contains a total of 140 loops, with 35 loops of length 6, 35 of length 8, 35 of length 10, and 35 of length 12.

The optimal values of m and n are determined as $(m=160, n=32)$ according to the test result on tuning-set (Figure 4).

Placement of backbone atoms. From the n sampled dihedral angle pairs $(\phi^1,\psi^1),\cdots,(\phi^n,\psi^n)$, we can calculate the coordinates of atom C_i and N_{i+1} for all of the n trials. CA_{i+1} atoms are sampled by generating random ω dihedral angles from a normal distribution with mean 180° and standard deviation of 4°. Calculating the coordinates of backbone O atoms using standard bond length and angle values is straightforward.

The coordinates of backbone atoms of the n samples at this particular growth step can be denoted as $(\mathbf{x}^1_{C_i},\mathbf{x}^1_{O_i},\mathbf{x}^1_{N_{i+1}},\mathbf{x}^1_{CA_{i+1}},\cdots,\mathbf{x}^k_{C_i},\mathbf{x}^k_{O_i},\mathbf{x}^k_{N_{i+1}},\mathbf{x}^k_{CA_{i+1}},\cdots,\mathbf{x}^n_{C_i},\mathbf{x}^n_{O_i},\mathbf{x}^n_{N_{i+1}},\mathbf{x}^n_{CA_{i+1}},)$. For simplicity, we denote the coordinates of the four atoms at residue i as S_i and the k-th sample as S^k_i. We sample one of them using an energy criterion. The probability for S^k_i is defined by

$$\pi(S^k_i|S_t,S_{t+1},\cdots,S_{i-1}) \sim \exp(-E(S^k_i)/T),$$

where $T=1$ is the effective temperature, and $E(S^k_i)$ is the interaction energy of the four atoms defined by S^k_i with the remaining part of the protein, including those loop atoms sampled in previous steps. The energy function E is an atomic distance-dependent empirical potential function constructed from the loop database, which is effective in detecting steric clashes and efficient to compute. Fragments with steric clashes are rarely drawn because of their high energy values. In summary, the coordinates of the four backbone atoms, $S_i=(C_i,O_i,N_{i+1},CA_{i+1})$, is drawn from the following joint distribution at this step:

$$S_i \sim \pi(d_{C_i,C_l}|d_{CA_i,C_l})\cdot\pi(d_{N_{i+1},C_l}|d_{C_i,C_l})\cdot\pi(\omega)\cdot\pi(\phi^i,\psi^i,\alpha_i) \cdot\pi(S_i|S_t,S_{t+1},\cdots,S_{i-1}). \tag{6}$$

Altogether, $(l-t)$ backbone dihedral angle combinations need to be sampled. When the growing end is three residues away from the C-terminal anchor atom of the loop, C_l, we apply the CSJD analytical closure method to generate coordinates of the remaining backbone atoms [12]. Small fluctuations of bond lengths, angles, and ω dihedral angles are introduced to the analytical closure method to increase the success rate of loop closure.

Improving computational efficiency

To reduce computational cost of calculating atom-atom distances in energy evaluation, we use a procedure, REsidue-

residue Distance Cutoff and ELLipsoid criterion (Redcell) to reduce computational time.

Residue-residue distance cutoff. The residue-residue distance cutoff d_R is used to exclude residues far from the loop energy calculation. Instead of a universal cutoff value, such as the 10 Å $C_\beta - C_\beta$ distance used in reference [51], we use a residue-dependent distance cutoff value. The residue-residue distance cutoff d_R is assigned to be $r_i + r_j + c$, where r_i and r_j are the effective radii of residue i and j, respectively. For one residue type, effective radii is the distance between residue geometrical center and the heavy atom which is farthest away from the residue geometrical center. c is a constant set to 8 Å. For a residue i in the loop region and residue j in the non-loop region, we calculate the residue-residue distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the geometric centers of residue i and j, respectively. If $d_{ij} > d_R$, all of the atoms in residue j are excluded from energy calculation. This residue-dependent cutoff is more accurate and ensures close residues are included.

Ellipsoid criterion. The basic idea of ellipsoid criterion is to construct a symmetric ellipsoid such that all atoms that need to be considered for energy calculation during loop sampling are enclosed in the ellipsoid. Atoms that are outside of the ellipsoid can then be safely excluded. The starting and ending residues of a loop naturally serve as the two focal points of the ellipsoid. Intuitively, all backbone atoms of a loop must be within an ellipsoid. Formally, we define a set of points $\{\mathbf{x}\}$, the sum of whose distances to the two foci is less than L, defined as the sum of the backbone bond lengths b_{C-C} of the loop of length l:

$$\{\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3 | \ \|\mathbf{x} - \mathbf{x}_1\| + \|\mathbf{x} - \mathbf{x}_2\| \le L\},$$

$$L = 2a = \sum^{l} b_{C-C},$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the two focal points of the ellipsoid. The symmetric ellipsoid ($b = c$) can be written as:

$$\frac{{x_1}^2}{a^2} + \frac{{x_2}^2}{b^2} + \frac{{x_3}^2}{b^2} = 1, \qquad (7)$$

where $a = L/2$ and $b = [(L/2)^2 - (\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{2})^2]^{1/2}$ correspond to the semi-major axis and semi-minor axis of the symmetric ellipsoid, respectively. To incorporate the effects of side chain atoms, we enlarge the ellipsoid by the amount of the maximum side-chain length s. Furthermore, we assume that any atom can interact with a loop atom if it is within a distance cut-off of k. As a result, the overall enlargement of the ellipsoid is $(s+k)$. The final definition of the enlarged ellipsoid for detecting possible atom-atom interactions is given by Eqn (7), with

$$a = (\|\mathbf{x}_1 - \mathbf{x}_2\|/2)\sec\alpha_2, \qquad (8)$$

and

$$b = (\|\mathbf{x}_1 - \mathbf{x}_2\|/2)\tan\alpha_1 + s + k, \qquad (9)$$

where α_1 is determined by the equation $\sec\alpha_1 = \frac{L}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$, and α_2 by $\tan\alpha_2 = \frac{(s+k) + (\|\mathbf{x}_1 - \mathbf{x}_2\|/2)\tan\alpha_1}{\|\mathbf{x}_1 - \mathbf{x}_2\|/2}$ (see Figure 5B).

For any atom in the protein, if the sum of its distances to the two foci points is greater than $2a$, this atom is permanently excluded from energy calculations. The computational cost to enforce this criterion depends only on the loop length and is independent of the size the protein, once the rest of the residues have been examined using the ellipsoid criterion. This improves our computing efficiency significantly, especially for large proteins. This criterion also helps to prune chain growth by terminating a growth attempt if the placed atoms are outside the ellipsoid.

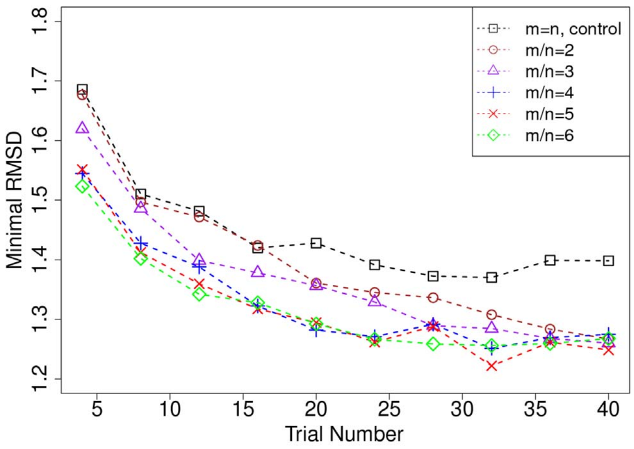

Figure 4. Mean of minimum backbone RMSD values for 140 protein loops. We generated 5,000 samples for each loop. The mean value of the minimum RMSD of the 140 loops (y-axis) is plotted against the size of trial samples n (x-axis) for different choices of m. For control, results obtained without sampling torsion angles ($m = n$, control) are also plotted. The backbone (N, C_a, C and O atoms) RMSD in this paper is calculated by fixing the rest of the protein body.
doi:10.1371/journal.pcbi.1003539.g004

Side-chain modeling and steric clash removal

Side chains are built upon completion of backbone sampling of a loop. For the i-th residue of type a_i, we denote the degrees of freedom (DOFs) for its side chain as $s_{(a_i)}$. DOFs of side chain residues depend on the residue types, *e.g.* Arg has four dihedral angles ($\chi_1, \chi_2, \chi_3, \chi_4$), with ($s_{(ARG)} = 4$). Val only has one dihedral angle (χ_1), with ($s_{(VAL)} = 1$). Each DOFs is discretized into bins of 4°, and only bins with non-zero entries for all loop residues in the loop database are retained.

We sample n_{sc} trial states of side chains from the empirical distribution $\pi(\chi_1 \cdots \chi_{s_{(a_i)}})$ obtained from the loop database. One of n_{sc} trials is then chosen according to the probability calculated by the empirical potential. Denote the side chain fragment for the i-th residue as $\mathbf{z}_i$, we select $\mathbf{z}_i$ following the probability distribution:

$$\pi_i(\mathbf{z}_i) \sim \exp(-E(\mathbf{z}_i)/T),$$

where $E(\mathbf{z}_i)$ is the interaction energy of the newly added side chain fragment $\mathbf{z}_i$ with the remaining part of the protein, and T is the effective temperature.

When there are steric clashes between side chains, we rotate the side-chain atoms along the $C_\alpha - C_\beta$ axis for all residue types except Pro. For Pro, we use the $N - C_\alpha$ axis for rotation. We consider two atoms to be in steric clash if the ratio of their distance to the sum of their van der Waals radii is less than 0.65 [13].

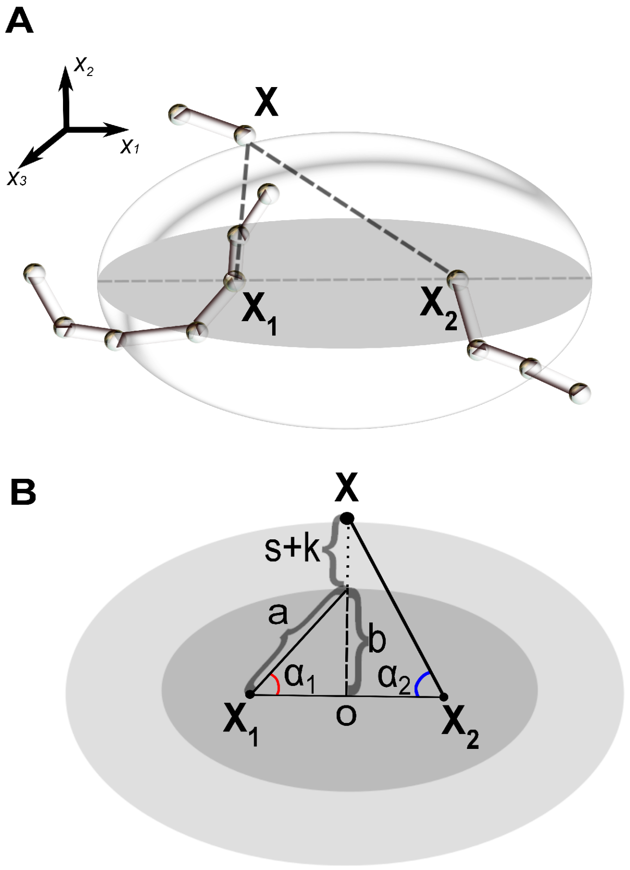

Figure 5. Schematic illustration of ellipsoid criterion. (A) Three dimensional view of a point $\mathbf{x}$ locating on the ellipsoid constructed from the total loop length L and the two foci $\mathbf{x}_1$ and $\mathbf{x}_2$. (B) Two dimensional view along through the x_3-axis of the ellipsoid, with $a = L/2$ and $b = c = [(L/2)^2 - (\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{2})^2]^{1/2}$ (dark gray). c is along x_3-axis, not shown. The maximum side-chain length is denoted as s and the distance cut-off of interaction is k. The enlarged ellipsoid, which has updated a and b, is also shown (light gray).
doi:10.1371/journal.pcbi.1003539.g005

Potential function

To evaluate the energy of loops, we develop a simple atom-based distance-dependent empirical potential function, following well-established practices [46,52,60–66]. Empirical energy functions developed from databases have been shown to be very effective in protein structure prediction, decoy discrimination, and protein-ligand interactions [54,63,64,67–71]. As our interest is modeling the loop regions, the atomic distance-dependent empirical potential is built from loop structures collected in the PDB [72].

Instead of using detailed 167 atom types associated with the 20 amino acids, we group all heavy atoms into 20 groups, similar to the approach used in Rosetta [50]. The 16 side-chain atom types comprise six carbon types, six nitrogen types, three oxygen types, and one sulfur type. The 4 backbone types are N, C_α, C, and O. This simplified scheme helps to alleviate the problem of sparsity of observed data for certain parameter values. For an atom i in the loop region of atom type a_i and an atom j of atom type a_j, regardless whether j is in the loop region, the distance-dependent interaction energy $E_{(a_i,a_j;d_{ij})}$ is calculated as :

$$E_{(a_i,a_j;d_{ij})} = -\ln\frac{\pi(a_i,a_j;d_{ij})}{\pi'(a_i,a_j;d_{ij})}, \tag{10}$$

where $E_{(}a_i,a_j;d_{ij})$ denotes the interaction energy between a specific atom pair (a_i,a_j) at distance d_{ij}, $\pi(a_i,a_j;d_{ij})$ and $\pi'(a_i,a_j;d_{ij})$ are the observed probability of this distance-dependent interaction from the loop database and the expected probability from a random model, respectively.

The observed probability $\pi(a_i,a_j;d_{ij})$ is calculated as:

$$\pi(a_i,a_j;d_{ij}) = \frac{n(a_i,a_j;d_{ij})}{n_{total}}, \tag{11}$$

where $n(a_i,a_j;d_{ij})$ is the observed count of (a_i,a_j) pairs found in the loop structures with the distance d_{ij} falling in the predefined bins. We use a total of 60 bins for d_{ij}, ranging from 2 Å to 8 Å, with the bin width set to 0.1 Å. d_{ij} ranging from 0 Å to 2 Å is treated as one bin. Here $n(a_i,a_j;d_{ij}) = \sum_{k=1}^{N} n(a_i,a_j,d_{ij}(k))$, where N is the number of loops in our loop database, $n(a_i,a_j,d_{ij}(k))$ is the observed number of (a_i,a_j) pairs at the distance of d_{ij} in the k-th loop. n_{total} is the observed total number of all atom pairs in the loop database regardless of the atom types and distance, namely, $n_{total} = \sum_{d_{ij}}\sum_{a_j}\sum_{a_i} n(a_i,a_j;d_{ij})$.

The expected random distance-dependent probability of this pair $\pi'(a_i,a_j;d_{ij})$ is calculated based on sampled loop conformations, called decoys. It is calculated as:

$$\pi'(a_i,a_j;d_{ij}) = \frac{n'(a_i,a_j;d_{ij})}{n'_{total}}, \tag{12}$$

where $n'(a_i,a_j;d_{ij}) = \sum_{k=1}^{N}(\frac{\sum_{x=1}^{M} n'(a_i,a_j,d_{ij}(x,k))}{M})$ is the expected number of $(a_i,a_j;d_{ij})$ pairs averaged over all decoy loop conformations of all target loops in the loop database. Here $n'(a_i,a_j,d_{ij}(x,k))$ is the number of (a_i,a_j) pairs at distance d_{ij} in the x-th generated loop conformations for the k-th loop. M is the number of decoys generated for a loop, which is set to 500. N is the number of loops in our loop database. n'_{total} is the total number of all atom pairs in the reference state, $n'_{total} = \sum_{d_{ij}}\sum_{a_j}\sum_{a_i} n'(a_i,a_j;d_{ij})$.

Tool availability

We have made the source code of DiSGro available for download. The URL is at: tanto.bioengr.uic.edu/DiSGro/.

Supporting Information

Text S1 Results of modeled loops on Test Set 2–5, calculated using DiSGro. Table 1–3 are tables for Test Set 2. Table 4–12 are tables for Test Set 3. Table 13–18 are tables for Test Set 4. Table 19–22 are tables for Test Set 5.
(PDF)

Acknowledgments

We thank Drs. Youfang Cao, Joe Dundas, David Jimenez Morales, Hammad Naveed, Hsiao-Mei Lu, and Gamze Gursoy, Meishan Lin, Yun Xu, Jieling Zhao for helpful discussions.

References

1. Bajorath J, Sheriff S (1996) Comparison of an antibody model with an x-ray structure: The variable fragment of BR96. Proteins: Structure, Function, and Bioinformatics 24: 152–157.
2. Streaker E, Beckett D (1999) Ligand-linked structural changes in the escherichia coli biotin repressor: The significance of surface loops for binding and allostery. Journal of molecular biology 292: 619–632.
3. Myllykoski M, Raasakka A, Han H, Kursula P (2012) Myelin 2′, 3′-cyclic nucleotide 3′-phosphodiesterase: active-site ligand binding and molecular conformation. PloS one 7: e32336.
4. Lotan I, Van Den Bedem H, Deacon A, Latombe J (2004) Computing protein structures from electron density maps: The missing loop problem. In: Workshop on the Algorithmic Foundations of Robotics (WAFR). pp. 153–68.
5. Fiser A, Do R, Šali A (2000) Modeling of loops in protein structures. Protein science 9: 1753–1773.
6. Sellers B, Zhu K, Zhao S, Friesner R, Jacobson M (2008) Toward better refinement of comparative models: predicting loops in inexact environments. Proteins: Structure, Function, and Bioinformatics 72: 959–971.
7. van Vlijmen H, Karplus M (1997) PDB-based protein loop prediction: parameters for selection and methods for optimization1. Journal of molecular biology 267: 975–1001.
8. Canutescu A, Dunbrack Jr R (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Science 12: 963–972.
9. de Bakker P, DePristo M, Burke D, Blundell T (2003) Ab initio construction of polypeptide fragments: Accuracy of loop decoy discrimination by an all-atom statistical potential and the amber force field with the generalized born solvation model. Proteins: Structure, Function, and Bioinformatics 51: 21–40.
10. DePristo M, de Bakker P, Lovell S, Blundell T (2003) Ab initio construction of polypeptide fragments: efficient generation of accurate, representative ensembles. Proteins: Structure, Function, and Bioinformatics 51: 41–55.
11. Michalsky E, Goede A, Preissner R (2003) Loops In Proteins (LIP)–a comprehensive loop database for homology modelling. Protein engineering 16: 979–985. michalsky2003
12. Coutsias E, Seok C, Jacobson M, Dill K (2004) A kinematic view of loop closure. Journal of computational chemistry 25: 510–528.
13. Jacobson M, Pincus D, Rapp C, Day T, Honig B, et al. (2004) A hierarchical approach to all-atom protein loop prediction. Proteins: Structure, Function, and Bioinformatics 55: 351–367.
14. Zhu K, Pincus D, Zhao S, Friesner R (2006) Long loop prediction using the protein local optimization program. Proteins: Structure, Function, and Bioinformatics 65: 438–452.
15. Zhang J, Kou S, Liu J (2007) Biopolymer structure simulation and optimization via fragment regrowth monte carlo. The Journal of chemical physics 126: 225101.
16. Cui M, Mezei M, Osman R (2008) Prediction of protein loop structures using a local move monte carlo approach and a grid-based force field. Protein Engineering Design and Selection 21: 729–735.
17. Spassov V, Flook P, Yan L (2008) LOOPER: a molecular mechanics-based algorithm for protein loop prediction. Protein Engineering Design and Selection 21: 91–100.
18. Liu P, Zhu F, Rassokhin D, Agrafiotis D (2009) A self-organizing algorithm for modeling protein loops. PLoS computational biology 5: e1000478.
19. Hildebrand P, Goede A, Bauer R, Gruening B, Ismer J, et al. (2009) Superlooper–a prediction server for the modeling of loops in globular and membrane proteins. Nucleic acids research 37: W571–W574.
20. Karmali A, Blundell T, Furnham N (2009) Model-building strategies for low-resolution x-ray crystallographic data. Acta Crystallographica Section D: Biological Crystallography 65: 121–127.
21. Mandell D, Coutsias E, Kortemme T (2009) Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. Nature methods 6: 551–552.
22. Lee J, Lee D, Park H, Coutsias E, Seok C (2010) Protein loop modeling by using fragment assembly and analytical loop closure. Proteins: Structure, Function, and Bioinformatics 78: 3428–3436.
23. Zhao S, Zhu K, Li J, Friesner R (2011) Progress in super long loop prediction. Proteins 79(10):2920–35
24. Arnautova Y, Abagyan R, Totrov M (2011) Development of a new physics-based internal coordinate mechanics force field and its application to protein loop modeling. Proteins: Structure, Function, and Bioinformatics 79: 477–498.
25. Goldfeld D, Zhu K, Beuming T, Friesner R (2011) Successful prediction of the intra-and extracellular loops of four g-protein-coupled receptors. Proceedings of the National Academy of Sciences 108: 8275–8280.

Author Contributions

Conceived and designed the experiments: KT JZ JL. Performed the experiments: KT JZ. Analyzed the data: KT JZ JL. Wrote the paper: KT JZ JL.

26. Subramani A, Floudas C (2012) Structure prediction of loops with fixed and flexible stems. The Journal of Physical Chemistry B 116: 6670–6682.
27. Fernandez-Fuentes N, Fiser A (2013) A modular perspective of protein structures: application to fragment based loop modeling. Methods in molecular biology (Clifton, NJ) 932: 141.
28. Bruccoleri R, Karplus M (1987) Prediction of the folding of short polypeptide segments by uniform conformational sampling. Biopolymers 26: 137–168.
29. Zhang J, Liu J (2006) On side-chain conformational entropy of proteins. PLoS computational biology 2: e168.
30. Zhang J, Lin M, Chen R, Liang J, Liu J (2007) Monte carlo sampling of near-native structures of proteins with applications. PROTEINS: Structure, Function, and Bioinformatics 66: 61–68.
31. Rosenbluth M, Rosenbluth A (1955) Monte carlo calculation of the average extension of molecular chains. The Journal of Chemical Physics 23: 356.
32. Grassberger P (1997) Pruned-enriched rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. Physical Review E 56: 3682.
33. Wong SWK (2013) Statistical computation for problems in dynamic systems and protein folding. PhD dissertation, Harvard University.
34. Liu J, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. Journal of the American statistical association : 1032–1044.
35. Liang J, Zhang J, Chen R (2002) Statistical geometry of packing defects of lattice chain polymer from enumeration and sequential monte carlo method. The Journal of chemical physics 117: 3511.
36. Liu J (2008) Monte Carlo strategies in scientific computing. Springer Verlag.
37. Zhang J, Lin M, Chen R, Wang W, Liang J (2008) Discrete state model and accurate estimation of loop entropy of RNA secondary structures. The Journal of chemical physics 128: 125107.
38. Zhang J, Chen Y, Chen R, Liang J (2004) Importance of chirality and reduced flexibility of protein side chains: A study with square and tetrahedral lattice models. The Journal of chemical physics 121: 592.
39. Lin M, Lu H, Chen R, Liang J (2008) Generating properly weighted ensemble of conformations of proteins from sparse or indirect distance constraints. The Journal of chemical physics 129: 094101.
40. Lin M, Zhang J, Lu H, Chen R, Liang J (2011) Constrained proper sampling of conformations of transition state ensemble of protein folding. Journal of Chemical Physics 134: 75103.
41. Zhang J, Dundas J, Lin M, Chen R, Wang W, et al. (2009) Prediction of geometrically feasible three-dimensional structures of pseudoknotted RNA through free energy estimation. RNA 15: 2248–2263.
42. Soto C, Fasnacht M, Zhu J, Forrest L, Honig B (2008) Loop modeling: Sampling, filtering, and scoring. Proteins: Structure, Function, and Bioinformatics 70: 834–843.
43. Cahill S, Cahill M, Cahill K (2003) On the kinematics of protein folding. Journal of computational chemistry 24: 1364–1370.
44. Shenkin P, Yarmush D, Fine R, Wang H, Levinthal C (1987) Predicting antibody hypervariable loop conformation. i. ensembles of random conformations for ringlike structures. Biopolymers 26: 2053–2085.
45. Xiang Z, Soto C, Honig B (2002) Evaluating conformational free energies: the colony energy and its application to the problem of loop prediction. Proceedings of the National Academy of Sciences 99: 7432–7437.
46. Zhou H, Zhou Y (2002) Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Science 11: 2714–2726.
47. Ko J, Lee D, Park H, Coutsias E, Lee J, et al. (2011) The FALC-loop web server for protein loop modeling. Nucleic acids research 39: W210–W214.
48. Simons K, Kooperberg C, Huang E, Baker D, et al. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. Journal of molecular biology 268: 209–225.
49. Rohl C, Strauss C, Misura K, Baker D, et al. (2004) Protein structure prediction using rosetta. Methods in enzymology 383: 66.
50. Sheffler W, Baker D (2010) Rosettaholes2: A volumetric packing measure for protein structure refinement and validation. Protein Science 19: 1991–1995.
51. Leaver-Fay A, Tyka M, Lewis S, Lange O, Thompson J, et al. (2011) Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545–574.
52. Hu C, Li X, Liang J (2004) Developing optimal non-linear scoring function for protein design. Bioinformatics 20: 3080–3098.
53. Thomas P, Dill K (1996) An iterative method for extracting energy-like quantities from protein structures. Proceedings of the National Academy of Sciences 93: 11628–11633.
54. Huang S, Zou X (2011) Statistical mechanics-based method to extract atomic distance-dependent potentials from protein structures. Proteins: Structure, Function, and Bioinformatics 79: 2648–2661.

55. Engh R, Huber R (1991) Accurate bond and angle parameters for x-ray protein structure refinement. Acta Crystallographica Section A: Foundations of Crystallography 47: 392–400.
56. Wang G, Dunbrack R (2003) Pisces: a protein sequence culling server. Bioinformatics 19: 1589–1591.
57. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22: 2577–2637.
58. Lewis D (2008) Winsorisation for estimates of change. SURVEY METHODOLOGY BULLETIN-OFFICE FOR NATIONAL STATISTICS- 62: 49.
59. Bowman A, Azzalini A (1997) Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations, volume 18. Oxford University Press, USA.
60. Sippl M (1990) Calculation of conformational ensembles from potentials of mena force. Journal of molecular biology 213: 859–883.
61. Miyazawa S, Jernigan R, et al. (1996) Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. Journal of molecular biology 256: 623–644.
62. Lu H, Skolnick J (2001) A distance-dependent atomic knowledge-based potential for improved protein structure selection. Proteins: Structure, Function, and Bioinformatics 44: 223–232.
63. Li X, Hu C, Liang J (2003) Simplicial edge representation of protein structures and alpha contact potential with confidence measure. Proteins: Structure, Function, and Bioinformatics 53: 792–805.
64. Zhang J, Chen R, Liang J (2005) Empirical potential function for simplified protein models: Combining contact and local sequence–structure descriptors. Proteins: Structure, Function, and Bioinformatics 63: 949–960.
65. Shen M, Sali A (2006) Statistical potential for assessment and prediction of protein structures. Protein Science 15: 2507–2524.
66. Li X, Liang J (2007) Knowledge-based energy functions for computational studies of proteins. In: Computational methods for protein structure prediction and modeling, Springer. pp. 71–123.
67. Samudrala R, Moult J (1998) An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. Journal of molecular biology 275: 895–916.
68. Zhang J, Chen R, Liang J (2004) Potential function of simplified protein models for discriminating native proteins from decoys: Combining contact interaction and local sequence-dependent geometry. In: Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE. IEEE, volume 2, pp. 2976–2979.
69. Zhang C, Liu S, Zhou Y (2004) Accurate and efficient loop selections by the DFIRE-based all-atom statistical potential. Protein science 13: 391–399.
70. Huang S, Zou X (2006) An iterative knowledge-based scoring function to predict protein–ligand interactions: I. derivation of interaction potentials. Journal of computational chemistry 27: 1866–1875.
71. Zimmermann M, Leelananda S, Gniewek P, Feng Y, Jernigan R, et al. (2011) Free energies for coarse-grained proteins by integrating multibody statistical contact potentials with entropies from elastic network models. Journal of structural and functional genomics 12: 137–147.
72. Bernstein F, Koetzle T, Williams G, Meyer Jr E, Brice M, et al. (1977) The protein data bank: a computer-based archival file for macromolecular structures. Journal of molecular biology 112: 535–542.