BIOINFORMATICS

Vol. 26 ISMB 2010, pages i310–i317
doi:10.1093/bioinformatics/btq193

Fragment-free approach to protein folding using conditional neural fields

Feng Zhao, Jian Peng and Jinbo Xu*

Toyota Technological Institute, Chicago, IL 60637, USA

ABSTRACT

Motivation: One of the major bottlenecks with *ab initio* protein folding is an effective conformation sampling algorithm that can generate native-like conformations quickly. The popular fragment assembly method generates conformations by restricting the local conformations of a protein to short structural fragments in the PDB. This method may limit conformations to a subspace to which the native fold does not belong because (i) a protein with really new fold may contain some structural fragments not in the PDB and (ii) the discrete nature of fragments may prevent them from building a native-like fold. Previously we have developed a conditional random fields (CRF) method for fragment-free protein folding that can sample conformations in a continuous space and demonstrated that this CRF method compares favorably to the popular fragment assembly method. However, the CRF method is still limited by its capability of generating conformations compatible with a sequence.

Results: We present a new fragment-free approach to protein folding using a recently invented probabilistic graphical model conditional neural fields (CNF). This new CNF method is much more powerful than CRF in modeling the sophisticated protein sequence-structure relationship and thus, enables us to generate native-like conformations more easily. We show that when coupled with a simple energy function and replica exchange Monte Carlo simulation, our CNF method can generate decoys much better than CRF on a variety of test proteins including the CASP8 free-modeling targets. In particular, our CNF method can predict a correct fold for T0496_D1, one of the two CASP8 targets with truly new fold. Our predicted model for T0496 is significantly better than all the CASP8 models.

Contact: jinboxu@gmail.com

1 INTRODUCTION

Despite significant progress in recent years, *ab initio* protein folding is still one of the most challenging problems in computational structural biology. Fragment-based *ab initio* protein folding (Bowie and Eisenberg, 1994; Claessens *et al*., 1989; Jones and Thirup, 1986; Levitt, 1992; Simon *et al*., 1991; Sippl, 1993; Unger *et al*., 1989; Wendoloski and Salemme, 1992) and lattice-models (Kihara *et al*., 2001; Xia *et al*., 2000; Zhang *et al*., 2003) has been extensively studied. These two popular methods and their combination for protein modeling have achieved great success in critical assessment of structure prediction (CASP) competitions (Moult, 2005; Moult *et al*., 2003, 2005, 2007). For example, the widely-used fragment assembly program Rosetta (Misura *et al*., 2006; Simons *et al*., 1997) is one of the most successful *ab initio* protein folding programs. The TASSER program (Zhang and Skolnick, 2005) and its derivative Zhang-Server (Wu *et al*., 2007) have achieved outstanding performance in both CASP7 and CASP8 by combining lattice model and threading-generated fragments and distance restraints.

Although fragment-based *ab initio* protein folding demonstrates encouraging performance, several important issues remain with this method. First, there is no guarantee that the local conformations of a protein can be accurately covered by short structural fragments in the PDB since a protein with new fold is likely to be composed of some structural motifs that rarely occur in the PDB (Andras Fiser, CASP8 talk). Second, the conformation space defined by a fragment library is discrete in nature. This discrete nature may exclude the native fold from the conformational search space since even a slight change in backbone angles, especially in the middle region of a protein, can result in a totally different fold. To resolve these two limitations, this article will propose a fragment-free folding method that can efficiently explore protein conformations in a continuous space.

In literature there are quite a few fragment-free methods for *ab initio*. protein folding. For example, Joe *et al.* described an iterative folding method (DeBartolo *et al*., 2009), which folds a protein by mimicking folding pathway and explores the conformation space by directly sampling the backbone angles using a trimer library. Shakhnovich group also described a method that can directly sample backbone angles using a trimer library (Chen *et al*., 2007; Yang *et al*., 2007). Faraggi *et al*. (2009) first predict the backbone angles of a protein using a machine learning method and then explore protein conformation search space using a genetic algorithm, based upon the predicted backbone angles. Recently, Hamelryck *et al.* have developed two hidden Markov models (HMMs) (i.e. FB5-HMM and Torus-HMM) (Boomsma *et al*., 2008; Hamelryck *et al*., 2006) for fragment-free conformation sampling. Using a Torus-HMM model, they can generate local conformations as accurately as the fragment assembly method (Boomsma *et al*., 2008). However, these HMM models have not been applied to real-world *ab initio* folding yet. Recently, we have proposed a protein conformation sampling algorithm based on conditional random fields (CRF) (Zhao *et al*., 2008, 2009) and directional statistics. The CRF model is a generalization of the HMM models and much more powerful than HMM. Our CRF model can accurately describe the complex sequence-angle relationship and estimate the probability distribution of (virtual) backbone angles directly from sequence information and predicted secondary structure. We have shown that by using the CRF models, we can sample protein conformations with much better quality than FB5-HMM (Zhao *et al*., 2008). We have also shown that by coupling our CRF model with a simple energy function, our method compares favorably with fragment assembly in the CASP8 blind prediction (Zhao *et al*., 2009).

This article presents a new probabilistic graphical model conditional neural fields (CNFs) for *ab initio* protein folding. CNF is recently invented by our group for the modeling of sequential data. See Peng *et al*. (2009) for a detailed exposition. CNF is

*To whom correspondence should be addressed.

similar to but much more powerful than CRF in that CNF can naturally model the non-linear relationship between input and output while CRF cannot do so. Thus, CNF can model better the sophisticated relationship between backbone angles, sequence profile and predicted secondary structure, estimate the probability distribution of backbone angles more accurately and sample protein conformations more efficiently. In addition, this work also differs from our previous CRF method (Zhao *et al*., 2008, 2009) in that (i) instead of using a simulated annealing (SA) method for folding simulation, we developed a replica exchange Monte Carlo (REMC) method for folding simulation. The REMC method enables us to minimize energy function to a lower level and thus possibly produce better decoys. (ii) Our previous CRF method uses the position-specific frequency matrix (PSFM) generated by PSI-BLAST as the input. This work will use the position-specific scoring matrix (PSSM) generated by PSI-BLAST as the input of our CNF model. It has been proved that PSSM contains more information than PSFM for structure prediction such as secondary-structure prediction. We did not use PSSM with CRF because CRF cannot easily take PSSM as input. In contrast, we can easily feed PSSM into our CNF model. We will show that our new method is much more effective than our previous method and can dramatically improve sampling efficiency and we can generate much better decoys than before on a variety of test proteins.

2 METHODS

2.1 Continuous representation of conformations

In our previous work (Zhao *et al*., 2008, 2009), we used a simplified representation of a protein model and demonstrated that even with such a representation, we can achieve good folding performance. In this simplified representation only the main-chain and C_β atoms are considered. This work will continue to use such a simplified representation. That is, we assume that the distance between two adjacent C_α atoms is constant and represent the C_α-trace of a protein using a set of pseudo backbone angles (θ, τ). Given a residue at position i, its θ is defined as the pseudo bond angle formed by the C_α atoms at positions $i-1$, i and $i+1$; τ is a pseudo dihedral angle around virtual bond between $i-1$ and i and can be calculated from the C_α atoms at positions $i-2$, $i-1$, i and $i+1$. Therefore, given the first three C_α positions and sub-sequential (θ, τ) angles, we can build the C_α trace of a protein. Using the C_α trace, we then can build the coordinates for the main chain and C_β atoms using a method similar to BBQ (Gront *et al*., 2007). To employ the KMB hydrogen-bonding energy (Morozov *et al*., 2004) for β-containing proteins, we also build the backbone hydrogen atoms using a quick and dirty method (Branden and Tooze, 1999).

The preferred conformations of a residue in the protein backbone can be described as a probabilistic distribution of (θ, τ). Each (θ, τ) corresponds to a unit vector in the three-dimensional space (i.e. a point on a unit sphere surface). We can use the five-parameter Fisher–Bingham (FB5) distribution to model the probability distributions over unit vectors (Kent, 1982). FB5 is the analog on the unit sphere of the bivariate normal distribution with an unconstrained covariance matrix. The probability density function of the FB5 distribution is given by

$$f(u)=\frac{1}{c(\kappa,\beta)}\exp\left(\kappa\gamma_1\cdot u+\beta\left(\left(\gamma_2\cdot u\right)^2-\left(\gamma_3\cdot u\right)^2\right)\right),$$

where u is a unit vector variable and $c(\kappa,\beta)$ is a normalizing constant. The parameters κ and β determine the concentration of the distribution and the ellipticity of the contours of equal probability, respectively. The higher κ and β are, the more concentrated and elliptical the distribution is, respectively. The three vectors γ_1, γ_2 and γ_3 are the mean direction, the major and minor axes, respectively. The latter two vectors determine the orientation of the equal probability contours on the sphere, while the first vector determines the common center of the contours.

We cluster all the (θ, τ) angles in a set of ∼3000 non-redundant proteins with high-resolution X-ray structures into 100 groups. Then we calculate the FB5 distribution of each group using KentEstimator (Hamelryck *et al*., 2006). See Zhao *et al*. (2008) for a detailed description of how we calculate the FB5 distributions. Once we have the distribution of (θ, τ) at one residue, we can sample the real-valued (θ, τ) angles by probability and thus, explore protein conformations in a continuous space.

2.2 A second-order CNF model of conformation space

Previously we developed a CRF method for protein conformation sampling (Zhao *et al*., 2008, 2009). This CRF method uses a linear combination of input features (i.e. PSI-BLAST sequence profile and predicted secondary structure) to estimate the probability distribution of backbone angles. This kind of linear parameterization implicitly assumes that all the features are linearly independent, which contradicts with the fact that some input features are highly correlated. For example, the predicted secondary structure is correlated with sequence profiles since the former is usually predicted from the latter using tools such as PSIPRED (Jones, 1999). To model the correlation between predicted secondary structure and sequence profiles, an easy way is to explicitly enumerate all the possible combinations of secondary-structure type and amino acid identity in the linear CRF model. In fact, we can always combine some basic features to form a complex feature. However, explicitly defining complex features may introduce a number of serious issues. First, it will result in a combinatorial explosion in the number of complex features, and hence, in the model complexity. It is challenging to train a model with a huge number of parameters without overfitting. Second, explicit enumeration may miss some important complex features. For example, the CRF model presented in Zhao *et al*. (2008, 2009) does not accurately model the correlation among sequence information at several adjacent positions. Finally, explicit enumeration of complex features may also introduce a large number of unnecessary features, which will increase the running time of probability estimation.

Instead of explicitly enumerating all the possible non-linear combinations of the basic sequence and structure features, we can use a better graphical model to implicitly account for the non-linear relationship between sequence and structure. Very recently, we have developed a new probabilistic graphical model CNF (Peng *et al*., 2009), which can implicitly model non-linear relationship between input and output. As shown in Figure 1, CNF consists of at least three layers: one or more hidden layers, input (i.e. sequence profile and secondary structure) and output (i.e. backbone angles) while CRF consists of only two layers: input and output. The relationship between the backbone angles and the hidden layer is still linear. However, the hidden layer uses some gate functions to non-linearly transform the input features into complex features. Here we use $G_\theta(x)=\left(1/\left(1+\exp\left(-\theta^T x\right)\right)\right)$ as the gate function where θ is the parameter vector and x a feature vector. CNF can also be viewed as the seamless integration of CRF and neural networks (NN). The neurons in the hidden layer will automatically extract non-linear relationship among input features. Therefore, without explicit enumeration, CNF can directly model non-linear relationship between input and output. The training of a CNF model is similar to that of a CRF, but more complicated.

We have tested this CNF model for protein secondary-structure (SS) prediction from sequence profiles. Table 1 compares the performance of various machine learning methods for SS prediction. The results are averaged on a 7-fold cross-validation on the CB513 data set, except that SPINE uses 10-fold cross-validation. As shown in Table 1, by using only one hidden layer to model non-linear relationship between output and input, CNF achieves almost 10% relative improvement over CRF. CNF also outperforms other methods including SVMpro (Hua and Sun, 2001), SVMpsi (Kim and Park, 2003), YASSPP (Karypis, 2006), PSIPRED (Jones, 1999), SPINE (Dor and Zhou, 2007) and TreeCRFpsi (Dietterich *et al*., 2004). The linear CRF is the worst since it does not model non-linear relationship between secondary

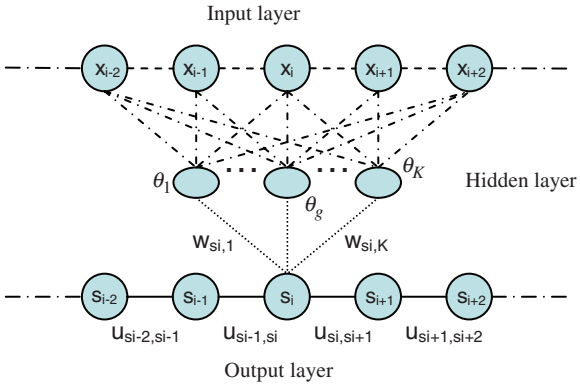


Fig. 1. A first-order CNF model consists of three layers: input, output and hidden layer. A second-order model is similar but not shown for the purpose of simplicity. In contrast, a CRF model consists of only input and output.

Table 1. Secondary-structure prediction accuracy

Methods	Q3 (%)	Methods	Q3 (%)
CRF	72.3	**CNF**	**80.1**
TreeCRFpsi	77.6	YASSPP	77.8
SVMpro	73.5	PSIPRED	76.0
SVMpsi	76.6	SPINE	76.8

Bold in this table indicates the best performance.

structure and sequence profile. This result indicates that we can indeed benefit from modeling non-linear sequence-structure relationship. We expect that using CNF, we are able to more accurately model sequence–angle relationship and thus, to sample conformations more efficiently.

In the context of CNF, the PSI-BLAST sequence profile (i.e. PSSM) and predicted secondary structure are viewed as observations; the backbone angles and their FB5 distributions are treated as hidden states or labels. Let H denote the 100 groups (i.e. states or labels) generated from clustering of the backbone angles. Each group is described by an FB5 distribution. Given a protein with solved structure, we calculate its backbone angles at each position and determine one of the 100 groups (i.e. states or labels) to which the angles at each position belong. Let $S=\{s_1,s_2,...,s_N\}$ $(s_i \in H)$ denote such a sequence of states/labels (i.e. FB5 distributions) for this protein. We also denote the sequence profile of this protein as M and its secondary structure as X. As shown in Figure 1, our CNF model defines the conditional probability of S given M and X as follows:

$$P_\Lambda(S|M,X)=\frac{\exp\left(\sum_{i=1}^{N}F(S,M,X,i)\right)}{Z(M,X)}$$

where $\Lambda=\{\lambda_1,\lambda_2,...,\lambda_p\}$ is the model parameter and $Z(M,X)=\sum_S \exp\left(\sum_{i=1}^{N}F(S,M,X,i)\right)$ is a normalization factor summing over all the possible labels for the given M and X. $F(S,M,X,i)$ consists of two edge feature functions and one label feature function at position i. It is given by

$$F(S,M,X,i)=e_1(s_{i-1},s_i)+e_2(s_{i-1},s_i,s_{i+1})+\sum_{j=i-w}^{i+w}v(s_{i-1},s_i,M_j,X_j)$$

where $e_1(s_{i-1},s_i)$ and $e_2(s_{i-1},s_i,s_{i+1})$ are the first- and second-order edge feature functions, respectively, and $v(s_{i-1},s_i,M_j,X_j)$ is the label feature function. The edge functions describe the interdependency between two or three neighboring labels. CNF is different from CRF in the label feature function. In CRF, the label feature function is defined as a linear combination of features. In CNF, there is an extra hidden layer between the input and output, which consists of K gate functions (see Fig. 1). The K gate functions extract a K-dimensional implicit non-linear representation of input features. Therefore, CNF can be viewed as a CRF with its inputs being K homogeneous hidden feature-extractors at each position. The label feature function of CNF is defined as follows:

$$v(s_{i-1},s_i,X,M)=\sum_{g=1}^{K}w_{s_{i-1},s_i,g}G_{\theta_g}(f(X,M,i)).$$

That is, the label feature function is a linear combination of K gate functions G. In the above definition, w is the parameter vector and f is a vector of basic features at position i. In our current implementation, f contains 23×9 (=207) elements, corresponding to the sequence profile and secondary-structure information in a window of size nine centered at position i. We use PSIPRED to predict the secondary structure of a protein from its sequence profile. PSIPRED generates likelihood score of three secondary structure types for each residue, which is used as the input of our CNF model.

Similar to CRF, we use the maximum likelihood method to train the model parameters such that $P_\Lambda(S|M,X)$ is maximized. That is, we maximize the occurring probability of a set of ~3000 non-redundant high-resolution protein structures. Although both the output and hidden layers contain model parameters, all the parameters can be learned together by gradient-based optimization. We use LBFGS (Liu and Nocedal, 1989) as the optimization routine to search for the optimal model parameters. Since CNF contains a hidden layer of gate functions G, the log-likelihood function is not convex any more. Therefore, it is very likely that we can only obtain a local optimal solution of the model parameters. To achieve a good solution, we run the training algorithm several times and use the solution with the best objective function as the final solution of the model. See Peng *et al.* (2009) for a detailed description of training CNF.

2.3 Model parameter training

To do a fair comparison between our previous CRF model and this CNF model, we used exactly same data to train both CRF and CNF models. That is, we use a set of ~3000 non-redundant proteins to train the parameters in our CNF and CRF models. Any two proteins in the training set share no more than 30% sequence identity and the resolution of a training protein is at least 2.0 Å. To avoid overlap between the training data and the test proteins, we removed the following proteins from our training set: (i) the proteins sharing at least 25% sequence identity with our test proteins; (ii) the proteins in the same fold class as our test proteins according to the SCOP classification; and (iii) the proteins having a TM-score (Zhang and Skolnick, 2007) at least 0.5 with our test proteins. Finally, the training data was prepared before CASP8 started. Therefore, we can use our CRF/CNF models to test the CASP8 free-modeling targets without worrying about bias.

The training set is randomly divided into five sets of same size and then used for 5-fold cross validation. To train a CNF model, we shall determine the number of gate functions at the hidden layer. In addition, since the CNF model contains a very large number of model parameters, to avoid overfitting, we shall also control the model complexity. We achieve this by regularizing the L_2-norm of the model parameters using a regularization factor. We trained our CNF model by enumerating the number of gate functions (50, 100, 200 and 300) and different regularization factors: 25, 50, 100 and 200 to see which one yields the best F_1-value. F_1-value is widely-used to measure the prediction capability of a machine learning model. F_1-value is an even combination of precision p and recall r and defined as $2pr/(p+r)$. The higher the F_1-value is, the better the CNF model. Our CNF model achieves the best F_1-value (23.44%) when 200 gate functions are used with regularization factor 50. In contrast, the best F_1-value achieved by our previous CRF method is 22.0%. The F_1-value improvement achieved by CNF over CRF seems not to be very big, partially because in total 100 labels are used in our

models. Later we will show that CNF can do conformation sampling much better than CRF.

2.4 Conformation sampling and resampling

Using the trained CNF model, we can sample the whole conformation of a protein or propose a new conformation from an existing one by resampling the local conformation of a segment. This procedure is very similar to the conformation sampling algorithm in our CRF method (Zhao *et al*., 2008, 2009). That is, we can use the forward–backward algorithm to first sample labels (i.e. angle distribution) by probability estimated from our CNF model and then sample real-valued angles from the labels. See Zhao *et al*. (2008) for a detailed description of the algorithm.

2.5 REMC simulation

The energy function we used for folding simulation consists of three items: DOPE (a pairwise statistical potential) (Fitzgerald *et al*., 2007; Shen and Sali, 2006), KMBhbond (hydrogen bonding energy) (Morozov *et al*., 2004) and ESP (a simplified solvent accessibility potential) (Fernandez *et al*., 2002). We use the weight factors previously trained for the CRF model for these three energy items. Therefore, the energy function is not biased towards our CNF method. The weight factor for DOPE is always fixed to 1, so only two weight factors shall be determined. See Zhao *et al*. (2009) for a detailed description of weight determination.

Previously we employ a SA algorithm to minimize energy function, based upon the algorithm proposed by Aarts and Korst (1991). In this work, we employ a REMC method (Earl and Deem, 2005; Swendsen and Wang, 1986) to minimize energy function. By using REMC, we can minimize energy function to lower values and thus produce better decoys for most of our test proteins. Our REMC method employs 20 replicas and the highest temperature is set to 100. The temperature for replica i ($i = 1, 2, \ldots, 20$) is set to $5i$. We have also tested other temperature assignment, but have not seen much difference in terms of folding performance. Each replica consists of 24 000 time steps. At each time step a new conformation is proposed and then accepted with probability $\min\{1, \exp(-\Delta E/T_i)\}$ where ΔE is the energy difference between the new and old conformations and T_i is the temperature for this replica. The conformations between two neighboring replicas are exchanged every 30 time steps. Therefore, in total 800 conformation exchange events will happen between two neighboring replicas during the whole folding simulation. It will make our simulation process very inefficient if we yield only the decoy with the lowest energy at the end of the folding simulation. To generate more decoys from a single folding simulation, we output the final decoy of each replica as long as it has an energy value within 15% of the lowest energy we can achieve. Experimental results indicate that on average, each folding simulation can generate ∼10 decoys.

3 RESULTS

Since in our previous work (Zhao *et al*., 2009), we have demonstrated that our CRF method compares favorably with the popular fragment-based Robetta server in the CASP8 blind prediction, in this article we will focus on the comparison between our CNF and CRF methods, and show that our nre method is indeed superior over our previous method.

We test our new method using two datasets and compare it with our previous method. These two datasets were used to evaluate our previous method before. The first dataset consists of 22 proteins: 1aa2, 1beo, 1ctfA, 1dktA, 1enhA, 1fc2C, 1fca, 1fgp, 1jer, 1nkl, 1pgb, 1sro, 1trlA, 2croA, 2gb1A, 4icbA, T052, T056, T059, T061, T064 and T074. These proteins have very different secondary-structure type and their sizes range from 40 to 120 residues. Some proteins (e.g. T052, T056, T059, T061, T064 and T074) in this dataset are very old CASP targets. Therefore, we denote this dataset as 'old testset'. The second dataset contains 12 CASP8 free-modeling targets: T0397_D1, T0405_D1, T0405_D2, T0416, T0443_D1, T0443_D2, T0465, T0476, T0482, T0496_D1, T0510_D3 and T0513_D2. These proteins are called free-modeling targets because a structurally similar template cannot be identified for them using a template-based method. We denote this dataset as 'CASP8 testset'. To avoid bias, we removed all the proteins similar to the first dataset from our training set (see Section 2.3). Since the training set was constructed before CASP8 started, there is no overlap between our training data and the CASP8 testset.

3.1 Performance on the old testset

As shown in Table 2, we evaluate our CNF and CRF methods in terms of their capability of generating good decoys. We run both methods on each test protein and generate similar number of decoys (5000–10 000). Each decoy is compared to its native structure and RMSD to the native is calculated for this decoy. Then we rank all the decoys of one test protein in an ascending order by RMSD. Finally we calculate the average RMSD of the top 1, 2, 5 and 10% decoys, respectively. We do not compare these two methods using the best decoys because they may be generated by chance and usually the more decoys are generated, the better the best decoys will be. In terms of the average RMSD of the top 5 or 10% decoys, our CNF method outperforms the CRF method on all test proteins except 1ctfA, 1dktA, 1fc2C and 1fgp. The CNF method reduces the average RMSD of top 10% decoys by at least 1 Å for many proteins such as 1aa2, 1beo, 1fca, 1pgb, 1sro, 2gb1A, 4icbA, T052, T056, T059, T061 and T064. Furthermore, our CNF method dramatically reduces the average RMSD of top 10% decoys for some proteins. For example, our CNF method reduces the average RMSD of top 10% decoys for 4icbA from 8.0 to 5.2 Å, for T056 from 11.1 to 7.2 Å and for T061 from 7.6 to 5.6 Å. Even for some test proteins (e.g. 1enhA, 1pgb and 2gb1A) on which the CRF method has already performed well, our CNF method still improves a lot.

3.2 Performance on the CASP8 testset

To further compare our CRF and CNF methods, we also evaluate them on the 12 CASP8 free-modeling (FM) targets, as shown in Table 3. During the CASP8 competition, structurally similar templates cannot be identified for these targets. Similarly, we evaluate both methods in terms of the average RMSD of the top 1, 2, 5 and 10% decoys, respectively. Compared to CRF, our CNF method does not significantly worsen the decoy quality of any of the 12 CASP8 targets. Instead, our CNF method outperforms the CRF method on 10 of the 12 targets and yields slightly worse performance on another two targets: T0397_D1 and T0482. In particular, our CNF method reduces the average RMSD of the top 10% decoys by at least 1 Å for the following seven targets: T0405_D1, T0405_D2, T0416_D2, T0443_D2, T0476, T0496_D1 and T0510_D3.

Our CNF method reduces the average RMSD of top 10% decoys for T0510_D3 from 9.1 to 6.3 Å and for T0496_D1 from 10.1 to 8.1 Å. Even for T0416_D2, a target on which our CRF method performed well, our CNF method improves the average RMSD of the top 10% decoys by 1 Å. We have also examined the average TM-score/GDT-TS of the top 10% decoys, on average our CNF method is better than the CRF method by ∼10% (data not shown due to space limitation).

Table 2. Performance of the CNF and CRF methods on the old testset

	s/t	M	Best	1%	2%	5%	10%
1aa2	108	N	6.0	7.0	7.6	8.5	9.2
	5α	R	7.1	9.0	9.4	10.0	10.4
1beo	98	N	5.5	6.1	6.5	7.4	8.3
	5α	R	5.6	7.2	7.8	8.7	9.3
1ctfA	68	N	3.6	4.5	4.8	5.4	6.1
	3α3β	R	3.3	3.9	4.1	4.6	5.2
1dktA	72	N	4.5	5.1	5.5	6.2	6.9
	4β	R	4.5	5.0	5.3	5.9	6.6
1enhA	54	N	1.5	2.0	2.1	2.3	2.4
	3α	R	2.1	2.6	2.7	2.9	3.0
1fc2C	43	N	2.0	2.3	2.4	2.5	2.6
	2α	R	2.1	2.3	2.3	2.4	2.4
1fca	55	N	3.2	3.9	4.2	4.6	5.0
	4β	R	5.0	5.6	5.8	6.2	6.4
1fgp	67	N	6.4	7.5	8.0	8.6	9.1
	6β	R	6.6	7.3	7.6	8.1	8.6
1jer	110	N	9.6	10.8	11.1	11.6	12.1
	2α6β	R	10.0	11.5	11.9	12.4	12.8
1nkl	78	N	1.8	2.5	2.6	2.8	3.0
	5α	R	2.3	2.8	2.9	3.2	3.4
1pgb	56	N	1.4	1.9	2.0	2.3	2.6
	1α4β	R	2.2	3.0	3.2	3.5	3.7
1sro	76	N	4.2	5.2	5.9	6.7	7.4
	6β	R	5.1	6.4	6.9	7.7	8.4
1trlA	62	N	3.2	3.6	3.7	3.9	4.1
	6α	R	3.9	4.2	4.4	4.5	4.7
2croA	65	N	1.8	2.2	2.3	2.4	2.5
	5α	R	2.2	2.5	2.6	2.7	2.8
2gb1A	56	N	1.7	1.9	2.0	2.3	2.6
	1α4β	R	1.9	3.1	3.3	3.6	3.8
4icbA	76	N	4.1	4.8	4.9	5.1	5.2
	4α	R	5.3	6.1	6.5	7.3	8.0
T052	98	N	7.6	8.1	8.5	9.1	9.6
	8β	R	8.6	9.6	10.0	10.7	11.3
T056	114	N	4.1	4.9	5.3	6.1	7.2
	6α	R	7.9	9.4	9.7	10.3	11.1
T059	71	N	5.7	6.9	7.3	7.7	8.1
	7β	R	6.9	8.4	8.7	9.2	9.6
T061	76	N	2.8	3.4	3.7	4.6	5.6
	4α	R	5.9	6.6	6.8	7.2	7.6
T064	103	N	6.5	7.0	7.2	7.5	7.9
	8α	R	5.9	7.1	7.5	8.2	8.9
T074	98	N	3.7	5.0	5.4	5.9	6.3
	4α	R	5.0	6.0	6.4	6.7	6.9

Column 's/t' lists the size and secondary-structure content of the test proteins. Column 'M' indicates methods. 'N' and 'R' represent the CNF and CRF methods, respectively. Column 'x%' lists the average RMSD (Å) of the decoys among the top x% of the generated decoys. Column 'best' lists the RMSD of the best decoys.

We have also examined the relationship between RMSD and energy. Due to space limitation, here we only visualize the RMSD-energy relationship for several typical targets: T0397_D1, T0416_D2, T0476, T0482, T0496_D1 and T0510_D3, as shown in Figure 2. Note that in the figure, we normalize the energy of a decoy by the mean and SD calculated from the energies of all the decoys of one target. By energy normalization, we can clearly see the energy difference between the decoys generated by the CNF/CRF methods. Figure 2 clearly demonstrates that our CNF method can generate

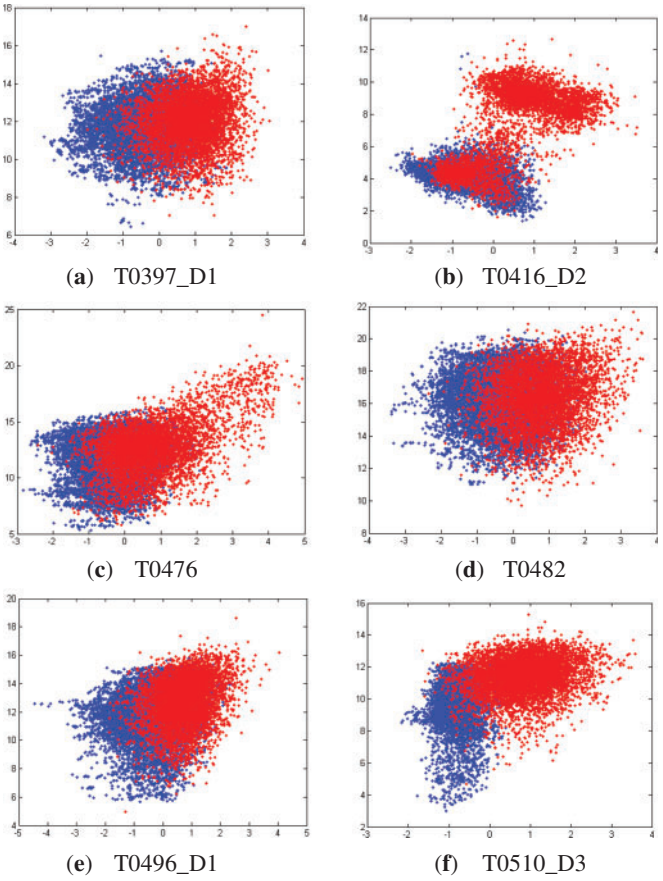


Fig. 2. The relationship between RMSD (y-axis) and energy (x-axis) for (**a**) T0397_D1, (**b**) T0416_D2, (**c**) T0476, (**d**) T0482, (**e**) T0496_D1 and (**f**) T0510_D3. The red and blue colors represent the CRF and CNF methods, respectively. See text for the energy normalization methods.

decoys with much lower energy than the CRF method. However, decoys with lower energy might not have better quality if the correlation between RMSD and energy is very weak. For example, our CNF method can generate decoys for T0397_D1 and T0482 with much lower energy, but cannot improve decoy quality for them. To improve the decoy quality for T0397_D1 and T0482, we have to improve the energy function. In contrast, the correlation between RMSD and energy is positive for T0416_D2, T0476, T0496_D1 and T0510_D3. Therefore, we can improve decoys quality for these four targets by generating decoys with lower energy.

Our CNF method dramatically improves the decoy quality on T0416_D2 over the CRF method, as shown in Figure 2b. The underlying reason is that our CNF method can estimate the backbone angle probability more accurately. Around half of the decoys generated by the CRF method for T0416_D2 are the mirror images of the other half. These mirror images are introduced by the non-native-like backbone angles around residue #31, as shown in Figure 3. We calculated the marginal probability of the 100 angle states at these residues and found out the native-like angle states have much higher marginal probability in the CNF model than in the CRF model. Thus, our CNF method can sample native-like angles at these residues more frequently than the CRF method and avoid generating a large number of mirror images. In addition to the CNF sampling method,

Fig. 3. Two typical mirror images generated by the CRF method for T0416_D2. The decoys in blue and gold represent the lower and upper regions in Figure 2b, respectively.

Table 3. Performance of our CNF and CRF methods on the CASP8 testset

	s/t	M	Best	1%	2%	5%	10%
T0397_D1	70	N	6.4	8.2	8.5	9.0	9.4
	7β	R	7.0	8.0	8.3	8.9	9.4
T0405_D1	80	N	5.0	5.4	5.5	5.7	5.9
	4α	R	5.7	6.6	6.8	7.1	7.4
T0405_D2	112	N	7.1	9.0	9.5	10.1	10.5
	3α6β	R	8.5	10.1	10.5	11.0	11.5
T0416_D2	57	N	1.4	1.9	2.1	2.3	2.6
	4α	R	1.6	2.6	2.8	3.3	3.6
T0443_D1	86	N	4.8	6.0	6.4	7.2	7.9
	6α	R	5.6	7.1	7.7	8.3	8.7
T0443_D2	114	N	9.3	10.6	10.9	11.5	11.9
	2α8β	R	10.4	11.9	12.3	12.9	13.4
T0465	157	N	11.0	11.8	12.2	12.9	13.5
	5α8β	R	10.2	12.2	12.7	13.4	13.9
T0476	108	N	5.3	6.3	6.8	7.4	8.0
	4α6β	R	5.9	7.8	8.2	8.7	9.3
T0482	120	N	10.7	11.9	12.2	12.8	13.2
	3α5β	R	8.8	10.9	11.5	12.3	13.0
T0496_D1	110	N	5.7	6.2	6.6	7.3	8.1
	3α6β	R	6.3	8.2	8.7	9.5	10.1
T0510_D3	44	N	3.0	4.0	4.5	5.3	6.3
	1α3β	R	4.7	7.2	7.7	8.6	9.1
T0513_D2	77	N	7.5	8.4	8.7	9.1	9.5
	2α4β	R	8.0	9.3	9.6	10.0	10.4

Column 's/t' lists the size and secondary-structure content of the test proteins. Column 'M' indicates methods. 'N' and 'R' represent the CNF and CRF methods, respectively. Column 'x%' lists the average RMSD (Å) of the decoys among the top x% of the generated decoys. Column 'best' lists the RMSD of the best decoys.

our energy function also helps improve the occurring frequency of native-like angles at these residues.

Table 4. Clustering result of the 12 CASP8 free-modeling targets

Target	First cluster			Best cluster		
	GDT	CASP8 rank	Internal rank (%)	GDT	CASP8 rank	Internal rank (%)
T0397_D1	25.7	12/60	50.6	28.6	28/262	18.8
T0405_D1	39.2	6/63	41.6	48.4	14/285	6.5
T0405_D2	27.0	10/62	72.3	34.6	19/280	5.1
T0416_D2	69.3	1/53	5.4	76.8	1/242	3.5
T0443_D1	46.9	3/64	38.2	49.2	6/253	19.7
T0443_D2	24.8	26/59	35.3	27.9	73/252	12.1
T0465	31.3	12/65	12.6	31.3	34/286	12.6
T0476	34.2	4/66	17.5	35.6	15/287	10.0
T0482	34.2	34/65	4.3	34.2	132/279	4.3
T0496_D1	30.5	1/59	30.3	49.1	1/266	0.4
T0510_D3	47.7	1/54	15.7	51.7	2/244	3.3
T0513_D2	57.7	5/50	3.8	57.7	17/225	3.8

Column 'GDT' lists the GDT-TS of the first and best cluster centroids. Column 'CASP8 rank' lists the rank of the #1 cluster centroid or the best cluster centroid among the first CASP8 server models or all the CASP8 server models, respectively. Column 'Internal rank' lists the percentile ranking (%) of a cluster centroid among all the decoys we generated for the target.

3.3 Comparison with CASP8 models

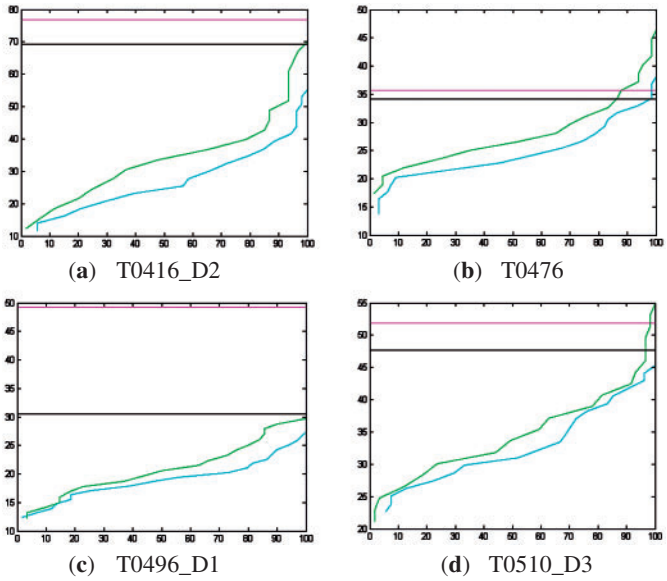
In order to compare our method with the CASP8 results, we use MaxCluster[1] to cluster the decoys of the 12 CASP8 FM targets. We ran MaxCluster so that for a given target, the first cluster contains ∼30% of all the decoys and the top five clusters in total cover ∼70% of the decoys. We examine only the top five clusters because CASP8 evaluated at most five models for a FM target. As shown in Table 4, we list the GDT-TS of a cluster centroid, its rank among the CASP8 models and its percentile ranking among all the decoys we generated. As shown in this table, our method did pretty well on T0405_D1, T0416_D2, T0443_D1, T0476, T0496_D1, T0510_D3 and T0513_D2; reasonably well on T0397_D1, T0405_D2 and T0465; and badly on T0443_D2 and T0482. Roughly speaking, our method can do well on mainly-alpha or small beta proteins, but not well on large beta proteins. This is expected since our CNF method can model well local sequence-structure relationship, but cannot model long-range hydrogen bonding.

Note that we generated decoys using domain definition we decided during the CASP8 season. Therefore, our domain definition may not be consistent with the CASP8 official definition. In this case, we calculate the GDT-TS of a model using the native structure common to our domain definition and CASP8 definition. The GDT-TS of a model is calculated using the TM-score program and may be slightly different from the CASP8 official GDT-TS.

3.4 Specific examples

In CASP8, we did prediction using the CRF method for T0476, T0496_D1 and T0510_D3, but not for T0416_D2 because our CRF method was not ready at the beginning of CASP8. The server model generated by our CRF method for T0510_D3 is among the

[1]http://www.sbg.bio.ic.ac.uk/∼maxcluster/index.html.



(**a**) T0416_D2 (**b**) T0476 (**c**) T0496_D1 (**d**) T0510_D3

Fig. 4. Ranking of our CNF predictions for (**a**) T0416_D2, (**b**) T0476, (**c**) T0496_D1 and (**d**) T0510_D3 (*x*-axis is percentile ranking and *y*-axis GDT-TS). Our first and best cluster centroids are plotted in black and magenta lines, respectively. The #1 models submitted by the CASP8 server are ordered by their GDT-TS and their percentile ranking is displayed as a cyan curve, so are the best models from each server but as a green curve.

best CASP8 server models.[2] Our CNF method further improves predictions for these four targets over the CRF method.

3.4.1 1 T0416_D2 The first and best cluster centroids have GDT-TS 69.3 and 76.8, respectively. As shown in Figure 4a, the best cluster centroid is better than all the CASP8 server models. In fact the best cluster centroid is also better than all the CASP8 human models (data not shown). The best cluster centroid also has a small RMSD 2.7 Å.

3.4.2 T0476 The first and best cluster centroids have GDT-TS 34.2 and 35.6, respectively. Our first and best cluster centroids for T0476 are ranked No. 4 out of 66 and No. 15 out of 287 CASP8 server models, respectively. The best human model for T0476 has GDT-TS 48.3 and RMSD 7.8 Å. Our best cluster centroid also has RMSD 7.8 Å.

3.4.3 T0496_D1 According to Grishin group, T0496_D1 is one of the only two CASP8 targets representing new folds (Shi *et al*., 2009). Our first and best cluster centroids have GDT-TS 30.5 and 49.1, respectively. As shown in Figure 4c, the best cluster centroid is significantly better than all the CASP8 server models. In fact the best cluster centroid is also significantly better than all the CASP8 human models. The best CASP8 model has GDT-TS only 33.96. The smallest RMSD among the CASP8 models with 100% coverage is 11.34 Å. Our best cluster centroid has a pretty good RMSD 6.2 Å considering that this target has more than 100 residues. In summary, our CNF method can predict an almost correct fold for this target.

[2]CASP8 results are available at http://predictioncenter.org/casp8/results.cgi.

3.4.4 T0510_D3 The first and best cluster centroids have GDT-TS 47.7 and 51.7, respectively. The best cluster centroid has RMSD 6.9 Å. As shown in Figure 4d, our first cluster centroid is better than all the #1 models submitted by the CASP8 servers. If all the 321 CASP8 models are considered, our first cluster centroid is worse than only three of them[3] and our best centroid is ranked No. 2.

4 CONCLUSION

This article has presented a new fragment-free approach to protein *ab initio* folding by using a recently-invented probabilistic graphical model CNF. Our fragment-free approach can overcome some limitations of the popular fragment assembly method. That is, this new method can sample protein conformations in a continuous space while the fragment-based methods cannot do so. This CNF method is also better than our previous CRF method in that (i) this method can easily model non-linear relationship between protein sequence and structure; and (ii) we can also minimize energy function to lower values. Experimental results indicate that our CNF method clearly outperforms the CRF method on most of the test proteins. Previously, we have compared our CRF method with the popular fragment-based Robetta server in the CASP8 blind prediction and shown that our CRF method is on average better than Robetta on mainly-alpha or small beta proteins (Zhao *et al*., 2009). This article further confirms our advantage on mainly-alpha or small beta proteins. Since CNF is better than CRF in modeling non-linear sequence-structure relationship, we are going to incorporate more information (such as amino acid physical–chemical property profile) to our model so that we can improve sampling efficiency further. We will also extend our CNF method so that long-range hydrogen bonding can also be modeled.

ACKNOWLEDGEMENTS

This work was made possible by the facilities of SHARCNET (http://www.sharcnet.ca) and the Open Science Grid Engagement VO. The authors are also grateful to Dr John McGee and Mats Rynge for their help with computational resources.

Funding: TTI-C research funding and National Institute of Health R01GM081642-01.

Conflict of Interest: none declared.

REFERENCES

Aarts,E. and Korst,J. (1991) *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, New York.

Boomsma,W. *et al*. (2008) A generative, probabilistic model of local protein structure. *Proc. Natl Acad. Sci. USA*, **105**, 8932–8937.

Bowie,J.U. and Eisenberg,D. (1994) An evolutionary approach to folding small α-helical proteins that uses sequence information and an empirical guiding fitness function. *Proc. Natl Acad. Sci. USA*, **91**, 4436–4440.

Branden,C.-I. and Tooze,J. (1999) *Introduction to Protein Structure*. Garland Publishing, New York, London.

Chen,W.W. *et al*. (2007) A knowledge-based move set for protein folding. *Proteins-Struct. Funct. Bioinformatics*, **66**, 682–688.

Claessens,M. *et al*. (1989) Modelling the polypeptide backbone with 'spare parts' from known protein structures. *Protein Eng.*, **2**, 335–345.

[3]There are very few human predictions for T0510_D3.

DeBartolo,J. *et al*. (2009) Mimicking the folding pathway to improve homology-free protein structure prediction. *Proc. Natl Acad. Sci. USA*, **106**, 3734–3739.

Dietterich,T. *et al*. (2004) Training conditional random fields via gradient tree boosting. In *Proceedings of the 21th International Conference on Machine Learning (ICML)*, ACM, New York, pp. 217–224.

Dor,O. and Zhou,Y.Q. (2007) Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins-Struct. Funct. Bioinformatics*, **66**, 838–845.

Earl,D.J. and Deem,M.W. (2005) Parallel tempering: theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.*, **7**, 3910–3916.

Faraggi,E. *et al*. (2009) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure*, **17**, 1515–1527.

Fernandez,A. *et al*. (2002) Dynamics of hydrogen bond desolvation in protein folding. *J. Mol. Biol.*, **321**, 659–675.

Fitzgerald,J.E. *et al*. (2007) Reduced Cbeta statistical potentials can outperform all-atom potentials in decoy identification. *Protein Sci.*, **16**, 2123–2139.

Gront,D. *et al*. (2007) Backbone building from quadrilaterals: a fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates. *J. Comput. Chem.*, **28**, 1593–1597.

Hamelryck,T. *et al*. (2006) Sampling realistic protein conformations using local structural bias. *PLoS Comput. Biol.*, **2**.

Hua,S.J. and Sun,Z.R. (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.*, **308**, 397–407.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol Biol.*, **292**, 195–202.

Jones,T.A. and Thirup,S. (1986) Using known substructures in protein model building and crystallography. *EMBO J.*, **5**, 819–823.

Karypis,G. (2006) YASSPP: better kernels and coding schemes lead to improvements in protein secondary structure prediction. *Proteins-Struct. Funct. Bioinformatics*, **64**, 575–586.

Kent,J.T. (1982) The Fisher-Bingham distribution on the sphere. *J. Royal Statist. Soc.*, **44**, 71–80.

Kihara,D. *et al*. (2001) TOUCHSTONE: an ab initio protein structure prediction method that uses threading-based tertiary restraints. *Proc. Natl Acad. Sci. USA*, **98**, 10125–10130.

Kim,H. and Park,H. (2003) Protein secondary structure prediction based on an improved support vector machines approach. *Protein Eng.*, **16**, 553–560.

Levitt,M. (1992) Accurate modeling of protein conformation by automatic segment matching. *J Mol. Biol.*, **226**, 507–533.

Liu,D.C. and Nocedal,J. (1989) On the limited memory method for large scale optimization. *Math. Program. B*, **45**, 503–528.

Misura,K.M. *et al*. (2006) Physically realistic homology models built with ROSETTA can be more accurate than their templates. *Proc. Natl Acad. Sci. USA*, **103**, 5361–5366.

Morozov,A.V. *et al*. (2004) Close agreement between the orientation dependence of hydrogen bonds observed in protein structures and quantum mechanical calculations. *Proc. Natl Acad. Sci.*, **101**, 6946–6951.

Moult,J. (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr. Opin. Struct. Biol.*, **15**, 285–289.

Moult,J. *et al*. (2003) Critical assessment of methods of protein structure prediction (CASP)-round V. *Proteins: Struct. Funct. Genet.*, **53**, 334–339.

Moult,J. *et al*. (2005) Critical assessment of methods of protein structure prediction (CASP)-round 6. *Proteins: Struct. Funct. Bioinformatics*, **61**(Suppl 7), 3–7.

Moult,J. *et al*. (2007) Critical assessment of methods of protein structure prediction-Round VII. *Proteins: Struc. Funct. Bioinformatics*, **69**, 3–9.

Peng,J. *et al*. (2009) Conditional neural fields. In Bengio,Y. *et al*. (eds) *Advances in Neural Information Processing Systems (NIPS)*. NIPS foundation, Vancouver, Canada, pp. 1419–1427.

Shen,M.Y. and Sali,A. (2006) Statistical potential for assessment and prediction of protein structures., *Protein Sci.*, **15**, 2507–2524.

Shi,S. *et al*. (2009) Analysis of casp8 targets, predictions and assessment methods. *Database* [E-pub ahead of print, doi:10.1093/database/bap003, 2009].

Simon,I. *et al*. (1991) Calculation of protein conformation as an assembly of stable overlapping segments: application to bovine pancreatic trypsin inhibitor. *Proc. Natl Acad. Sci. USA*, **88**, 3661–3665.

Simons,K.T. *et al*. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, **268**, 209–225.

Sippl,M. (1993) Recognition of errors in three-dimensional structures of proteins. *Proteins: Struct. Funct. Bioinformatics*, **17**, 355–362.

Swendsen,R.H. and Wang,J.S. (1986) Replica Monte-Carlo simulation of spin-glasses. *Phys. Rev. Lett.*, **57**, 2607–2609.

Unger,R. *et al*. (1989) A 3D building blocks approach to analyzing and predicting structure of proteins. *Proteins: Struct. Funct. Genet.*, **5**, 355–373.

Wendoloski,J.J. and Salemme,F.R. (1992) PROBIT: a statistical approach to modeling proteins from partial coordinate data using substructure libraries. *J. Mol. Graphics*, **10**, 124–126.

Wu,S. *et al*. (2007) Ab initio modeling of small proteins by iterative TASSER simulations. *BMC Biol.*, **5**, 17.

Xia,Y. *et al*. (2000) Ab initio construction of protein tertiary structures using a hierarchical approach. *J. Mol. Biol.*, **300**, 171–185.

Yang,J.S. *et al*. (2007) All-atom ab initio folding of a diverse set of proteins. *Structure*, **15**, 53–63.

Zhang,Y. and Skolnick,J. (2005) The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl Acad. Sci. USA*, **102**, 1029–1034.

Zhang,Y. and Skolnick,J. (2007) Scoring function for automated assessment of protein structure template quality. *Proteins-Struct. Funct. Bioinformatics*, **57**, 702–710; erratum in **68**, 1020.

Zhang,Y. *et al*. (2003) TOUCHSTONE II: a new approach to ab initio protein structure prediction. *Biophys. J.*, **85**, 1145–1164.

Zhao,F. *et al*. (2008) Discriminative learning for protein conformation sampling. *Proteins: Struct. Funct. Bioinformatics*, **73**, 228–240.

Zhao,F. *et al*. (2009) A probabilistic graphical model for ab initio folding. In Batzoglou,S. (ed.), *Research in Computational Molecular Biology*. Springer Berlin Heidelberg, Tucson, Arizona, pp. 59–73.