Efficient traversal of beta-sheet protein folding pathways using ensemble models

Solomon Shenker[1], Charles W. O'Donnell[2], Srinivas Devadas[2], Bonnie Berger[2,3,⋆], and Jérôme Waldispühl[1,2⋆]

[1] School of Computer Science & McGill Centre for Bioinformatics, McGill University, Montreal, Canada,
[2] Computer Science and AI Lab, MIT, Cambridge, USA,
[3] Department of Mathematics, MIT, Cambridge, USA.

Abstract. Molecular Dynamics (MD) simulations can now predict ms-timescale folding processes of small proteins — however, this presently requires hundreds of thousands of CPU hours and is primarily applicable to short peptides with few long-range interactions. Larger and slower-folding proteins, such as many with extended β-sheet structure, would require orders of magnitude more time and computing resources. Furthermore, when the objective is to determine only which folding events are necessary and limiting, atomistic detail MD simulations can prove unnecessary. Here, we introduce the program `tFolder` as an efficient method for modelling the folding process of large β-sheet proteins using sequence data alone. To do so, we extend existing ensemble β-sheet prediction techniques, which permitted only a fixed anti-parallel β-barrel shape, with a method that predicts arbitrary β-strand/β-strand orientations and strand-order permutations. By accounting for all partial and final structural states, we can then model the transition from random coil to native state as a Markov process, using a master equation to simulate population dynamics of folding over time. Thus, all putative folding pathways can be energetically scored, including which transitions present the greatest barriers. Since correct folding pathway prediction is likely determined by the accuracy of contact prediction, we demonstrate the accuracy of `tFolder` to be comparable with state-of-the-art methods designed specifically for the contact prediction problem alone. We validate our method for dynamics prediction by applying it to the folding pathway of the well-studied Protein G. With relatively very little computation time, `tFolder` is able to reveal critical features of the folding pathways which were only previously observed through time-consuming MD simulations and experimental studies. Such a result greatly expands the number of proteins whose folding pathways can be studied, while the algorithmic integration of ensemble prediction with Markovian dynamics can be applied to many other problems.

⋆ Corresponding authors: `jeromew@cs.mcgill.ca`, `bab@mit.edu`.

1 Introduction

Protein folding and unfolding is a key mechanism used to control biological activity and molecule localization [1]. The simulation of folding pathways is thus helpful to decipher the cell behavior. Classical molecular dynamics (MD) methods [2] can produce reliable predictions but unfortunately the heavy computational load required by these techniques limits their application to inputs tens of amino acids long and prevents their application to large sequences (i.e. hundreds of amino acids). Recently, P. Faccioli *et al.* proposed an effective solution of the Fokker-Planck equation to compute dominant protein folding pathways [3], but the same size limitations remain.

The development of distributed computing technologies has dramatically extended the range of application of MD techniques. For instance, Pande and co-workers achieved a 1.5 millisecond folding simulation of a 39 residue protein NTL9 [4]. In spite of this achievement, this strategy still seems limited to small polypeptides (about 50 residues) and, more importantly, requires several months of parallel computing and typically thousands of GPU's.

In this paper, we introduce a complete methodology to address these computational complexity limitations. Our approach aims to complement the range of techniques already offered. Unlike MD simulations which use an all-atom description of structures together with a fine-tuned energy force field, here we use a residue-level representation of the structure with a statistical residue contact potentials. This simplification enable us to sample intermediate structures and build a coarse-grained model of the energy landscape and subsequently simulate folding processes.

Since the seminal work of Levitt and Warshel [5], it is widely acknowledged that simplified representations of protein structures and motions are required to circumvent computational limitations. A conceptual breakthrough came when Amato and co-workers applied motion planning techniques to the protein folding problem [6,7]. The method is much faster than classical MD techniques and enables the study of the folding of large proteins. However, this approach does not predict structures, rather it requires the three-dimensional structure of the native state to compute potential intermediate structures and unfolding pathways, on which the folding simulations are performed. It follows that the methodology cannot be applied to proteins with unknown structures and cannot be relied upon to study misfolding processes.

In fact, all the methods previously described face a difficulty common with MD: efficient sampling of the conformational landscape. MD algorithms explore the landscape through force-directed local search and progressive modification of the structure. However, the scalability and numerical efficiency when modeling large molecular structures remains problematic, limiting their application to small molecular systems. On the other hand, motion planning algorithms use a 3D structure of the native fold to predict distant structural intermediates. Accordingly, the accuracy of the method can suffer when intermediates sampled are far away from the native state. Recently, Hosur *et al.* [8] have combined efficient motion planning techniques with machine-learning to model proteins as an ensemble, but this approach is effective only in the local neighborhood of the input structure.

Such obstacles have been addressed for RNA molecules by the development of structural ensemble prediction algorithms [9,10], and the derivation of a finely-tuned energy model based on experimental data [11]. Combined together, these techniques enable us to compute the RNA secondary structure energy landscapes and sample structures from sequence information alone. Wolfinger *et al.* [12] further demonstrated how an RNA energy landscape can be constructed by connecting these samples together and estimating the transition rates between pairs of interconverting states. The resulting ordinary differential equation (ODE) system can be solved to predict and characterize RNA folding pathways. The method has since been improved to analyze the motion of large RNAs [13].

In this paper, we propose to expand the methodology developed for RNAs to the more complicated case of proteins. First, we design an algorithm to sample the complete conformational landscape of large protein sequences given sequence data alone. Then, we use this sampling algorithm to build a coarse-grain representation of the energy landscape of a protein, from which we construct an ODE system modeling transition rates between folding intermediates that we solve to simulate protein folding.

We choose to address specifically β-sheet structures. The folding of these structures is particularly difficult to simulate. Indeed, β-sheets are stabilized by inter-strand residue interactions, and thus the folding and assembly of these structures is largely influenced by long-range interactions and global conformational rearrangements. For instance, Voeltz *et al.* recently showed that the rate-limiting step in the NTL9 fold was beta-sheet hairpin formation [4].

Since the original work of Mamitsuka and Abe [14], several groups have proposed models to predict general β-sheets [15,16,17]. However, none of these methods are capable of computing *ensembles* of β-sheet structures (i.e. perform an exact enumeration of all β-structures without duplicates) and therefore cannot be used to sample the β-sheet energy landscape.

We recently introduced a structural ensemble predictor for transmembrane β-barrel (TMB) proteins [18], continuing earlier work on molecular structure modeling [19,20]. However, TMBs are a special case of β-sheets where each strand pairs with its two sequence neighbors via an anti-parallel interaction (except the "closing" pair which involves the first and last strands). Here, we expand these techniques to allow any β-strand organization in the β-sheet, with parallel and anti-parallel orientations, and enable the sampling of general β-sheets. This algorithm is implemented in the program `tFolder`.

We use `tFolder` to sample the β-sheet conformational landscape and build a coarse-grain model of the energy landscape. More specifically, we cluster protein configurations according to contact distance metrics, and associate each cluster with an intermediate folding state. We use the difference between the ensemble free energies of the clusters to compute the transition rates and build an ODE system that models the energy landscape. Finally, we solve this system to estimate the distribution of conformations over folding time.

This methodology reconciles the MD and motion planning approaches for studying folding pathways. Using `tFolder`, we are now able to simulate in a couple of minutes on a single desktop the folding of large proteins, and to predict the folding pathways (as well as possible misfolding pathways) of proteins with unknown structures. Thus we are able to provide a broader range of applications, while offering computational efficiency comparable with motion planning techniques. Although we focus on β-sheet proteins, our method in principle could be extended to describe the folding pathways of a wider class of protein structures.

This paper is organized as follows. In section 2 we describe the `tFolder` algorithm and explain how we construct the coarse grained energy landscape model. Then, in section 3, we benchmark our methods. First, we evaluate the accuracy of `tFolder` for simple inter-strand residue contact prediction and show that it performs comparably with more sophisticated techniques specifically designed for this task. Importantly, our contact predictions are not dependent on the separation between the residue indices, which means an improved "very" long-range contact prediction

accuracy. Then, we illustrate the insights provided by our methods by analyzing the energy landscape of the extensively studied Protein G. We show that `tFolder` predicts the correct folding pathways, and interestingly, our simulation reveals a possible off-pathway structure. All these simulations can be performed on query sequences using our program `tFolder`, available at `http://csb.cs.mcgill.ca/tFolder`

2 Methods

To predict realistic protein folding pathways, we exploit well-established ensemble prediction algorithms [18] for their ability to accurately predict the energy scores of millions of feasible structural conformations from sequence alone. Our approach proceeds in two steps: (1) Given an arbitrary peptide sequence, we produce ensemble predictions of the energetic weight for all possible β-sheet structures and sub-structures, utilizing an enhancement to standard ensemble predictors which allows permutation. (2) Using each conformation's energetic score and metrics of conformational similarity, we derive the likelihood of dynamic state-to-state transitions and assemble a set of complete folding paths. In this way, we can identify and rank the most likely pathways from an unfolded conformation to a fully folded conformation based on predicted energy landscapes.

Modelling β-sheet Ensembles

We model the set of all possible β-sheet conformations a peptide can attain using a statistical-mechanical framework. Conceptually, each structure is described by the set of residue/residue contacts that form hydrogen bonds between β-strand backbones, and is assigned a Boltzmann-distributed pseudo-energy, determined by the specific residues involved in contacts. To characterize the energetic landscape of this ensemble, a partition function Z can be calculated over all structural states $S = \{1...n\}$ such that

$$Z = \sum_{i=1}^{n} e^{-\frac{E_{S_i}}{RT}},$$

with energies E_{S_i}, temperature T, and the Boltzmann constant R. For example, from this the relative abundance of a structure S_i can be easily derived:

$$p(S_i) = \frac{e^{-E_{S_i}}}{Z}.$$

Our energy model is based on statistical potentials and follows directly from prior prediction tools that have been shown to be accurate [18][21]. An energy $E_{i,j}$ is given to each residue/residue pair within the β-sheet fold following $E_{i,j} = -RT[\log(p(i,j)) - Z_c]$, where Z_c is a statistical recentering constant and $p(i,j)$ is the probability of these two residues appearing in a β-sheet environment, as observed across all non-sequence-homologous solved structures in the PDB [18]. Further, we assign separate probabilities based on the hydrophobicity of the environment on either face of a β-sheet.

A naive approach to computing the partition function would thus be to enumerate all possible structures and compute each structure's contribution to the sum individually. However, as was previously shown for the special case of anti-parallel β-strands in transmembrane β-barrel proteins, a much more efficient method exists using dynamic programming [18]. We have generalized this approach to enable the computation of arbitrary single β-sheet fold topologies.

Permutable β-templates

We introduce the concept of permutable β-templates to enable the calculation of the partition function of a β-sheet with arbitrary β-strand topologies. This extends existing ensemble prediction techniques by allowing any combination of parallel and anti-parallel β-strands to be including within a single β-sheet fold, and by removing any sequence dependency between β-strand/β-strand pairing partners. Prior methods supported only all-anti-parallel β-strands and required β-strand/β-strand interactions to be separated only by coil (and not other strands) [18].

To efficiently encode these generic shapes, each strand is labeled $\{1...n\}$ to allow a stepwise permutation through β-strand ordering, and a signed permutation is defined such that each β-strand is assigned to be parallel or anti-parallel relative to the first strand in the sheet (Figure 1). Algorithmically, `tFolder` is capable of constructing a dynamic program over all such permutations to calculate the partition function. In practice, since such an encoding can result in unrealistic combinations of β-strand/β-strand pairings (such as if β-strands 1 and 4 had too short a coil between them in Figure 1), we impose that valid foldings must satisfy steric and biologically derived constraints. These include a minimum and maximum β-strand length, maximum shear between neighboring β-strands (the amount of inclination that causes the β-sheet to deviate from a perfect rectangle), and minimum inter-strand loop size. These constraints serve to limit the

exploration of unrealistic conformations, minimizing excess computation and allowing directed investigation into specific motifs.

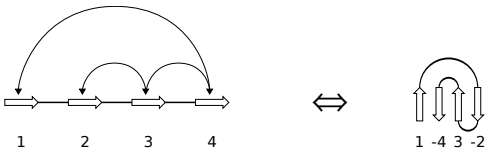
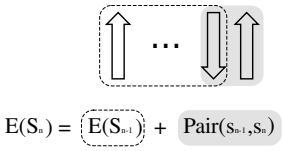
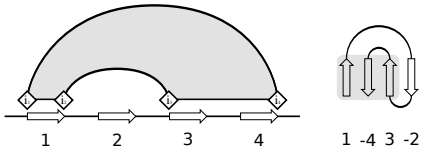


Fig. 1. An illustration of how a permutable β-template can be encoded as a signed permutation. The permutation lists the strands in the order that they occur in the sheet, with the sign indicating whether the strand is parallel (+) or anti-parallel (-) to the first strand.

The energy of a structure with n strands, can be recursively defined as $E(S_n) = E(S_{n-1}) + Pairing(s_{n-1}, s_n)$, where $E(S_{n-1})$ is the interaction energy between the first $n-1$ strands, and $Pairing(s_{n-1}, s_n)$ is the energy of the pairing of strand $n-1$ with strand n (See Figure 2(a)). `tFolder` exploits the shared structure between instances in the ensemble by computing this recursion using a dynamic programming algorithm. The result of each recursive call is stored in a table indexed by the parameters of the call. Subsequent recursive calls made with the same parameters perform a table lookup instead of re-computing the value of the recursion.



(a) β-sheet decomposition recursion



(b) Indices of intermediate structures

Fig. 2. (a) Illustrates how the energy function of a β-sheet can be recursively defined as the sum of the contribution of the last two strands with the contribution of the remaining structure. (b) Indicates the indices used to store the energies of intermediate structures for the recursion.

For a sheet of n strands, the table has n rows, where the kth row has entries corresponding to valid configurations of the first k strands. For the kth strand, these configurations are partitioned by the location of four indices k_1, k_2, k_3, k_4, which denote the boundaries of the region occupied

by the k strands (Figure 2(b)). To begin, the algorithm enumerates all possible positions of the first two strands, and for each stores the strand pair interaction energy in entry $E_{2_1 2_2 2_3 2_4}$ of the table. For each subsequent strand k, the value of $E_{k_1 k_2 k_3 k_4}$ is computed as:

$$E_{k_1 k_2 k_3 k_4} = \sum_{i_1 i_2 i_3 i_4} E_{i_1 i_2 i_3 i_4} + Pairing(i, k),$$

where i_1, i_2, i_3, i_4 are enumerated for all valid settings for the boundaries of the preceding strands, given the boundaries of the kth strand. Once the recursion has filled the table, the partition function Z is calculated by summing over all possible settings of n_1, n_2, n_3, n_4:

$$Z = \exp\left(- \sum_{n_1 n_2 n_3 n_4} E_{n_1 n_2 n_3 n_4} \right).$$

The table constructed to calculate the partition function can be used to sample the distribution of configurations of a given topology, utilizing the approach established by Ding and Lawrence for RNA secondary structure [22], and successfully applied previously by Waldispühl *et al.* to sample conformations of β-barrel proteins[20]. To do this, we perform a traceback through the table and, at ith step, sample the indices within which the first i strands are contained, according to the Boltzmann representation of these i-stranded structures (Figure 3).

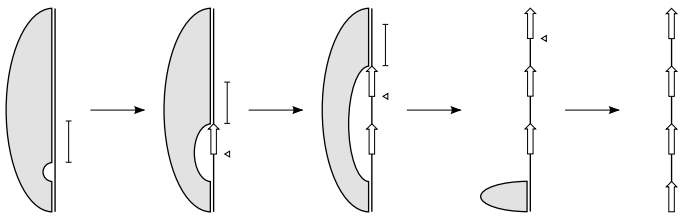


Fig. 3. Illustration of of how the sampling procedure performs a traceback through the table, over the indices of intermediate structures. During each step of the sampling procedure, the location of a single strand is sampled from the region indicated by the vertical bars. The triangles denote the location of the strand sampled during the previous step.

2.1 Predicting Folding Dynamics

Conceptually, we model the folding process as a path through a graph of varyingly folded conformations of a protein. In this graph, different protein conformations are represented as states, and two states that interconvert in a folding pathway are connected by an edge, analogous to work with RNA described previously [12]. The `tFolder` algorithm provides a means to efficiently sample the energetically accessible conformations that make up the states of this graph. We further propose a means to determine the connectivity between states and demonstrate how this can be applied to calculate the dynamics of the folding process.

Since we do not know the final structure, we begin by sampling configurations from all possible permutations of β-sheet topology, as described above. For every pair of states, we add an edge between two states if (1) the states have compatible topologies, and further, (2) the states show structural similarity.

Two templates are compatible if they are identical to each other, modulo the addition or removal of a single strand pairing. This operation can result in the growth of a core structure, or the nucleation of an independent strand pair (see Figure 4). Note that the requirements for satisfying the second criterion of structural similarity depends on the metric used to estimate structural similarity between two conformations. In practice, we use a contact based metric and deem two structures to be structurally similar if the metric is below the transition threshold.

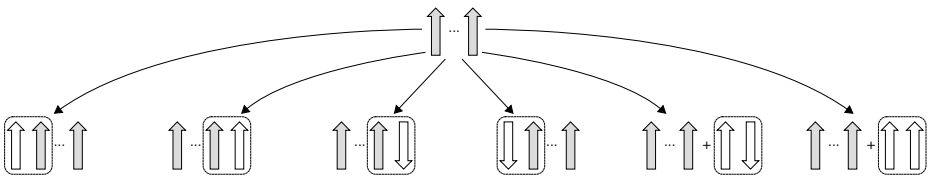


Fig. 4. The topologies that are compatible with a given state (shaded gray) result from the addition of a single pairing between strands (dashed box). The '+' indicates that there is no pairing between the gray structure and the white strand pair.

Given the graph constructed according to these two criteria, the change in the probability of the system being in state i at time t is calculated from the total flux into and out of state i,

$$\frac{dp_i}{dt} = \sum_{j \epsilon X} r_{ij} p_j(t),$$

where p_i is the probability of state i, X is the state space, and r_{ij} is the rate of transition from state i to state j. Given that two states are connected in the graph, the rate at which two states inter-convert is proportional to the difference between free energies of the states(ΔG); the system tends toward energetically favorable states. We calculate the transition rate r_{ij} between states i and j using the Kawasaki rule (with parameter r_0 to scale the time dimension):

$$r_{ij} = r_0 \exp\left(-\Delta G_{ij}/2RT\right).$$

The dynamics of the system are calculated by treating the folding process as a continuous time discrete state Markov process. Given the matrix of folding rates R, where $R_{ij} = r_{ij}$ and initial state density $\boldsymbol{p}(0)$, the distribution over states $\boldsymbol{p}(t)$ of the system at time t is given by the explicit solution to the system of linear differential equations,

$$\boldsymbol{p}(t) = \exp\left(Rt\right)\boldsymbol{p}(0).$$

Since we sample hundreds of states from each β-strand topology, we partition the state space into macro states using clustering, in order to work with a tractably sized system. Under this approximation, we consider two clusters the graph to be connected if the minimum distance between any two states from each cluster are connected. We define the ensemble free energy difference ΔG_{ij} between two macrostates i and j by summing over the states from which they are composed.

$$\Delta G_{ij} = E(\chi_i) - E(\chi_j) = \sum_{x \epsilon \chi_i} E(x) - \sum_{x \epsilon \chi_j} E(x).$$

Although this approximation lessens the computational burden, it represents a trade off. The granularity achievable by our simulation is at the level of the macrostates. Note, energy barriers are not explicitly incorporated into the model, since entire β-strands are either added or removed between states without partially-formed intermediates.

3 Results

Evaluation of Contact Prediction

To evaluate the contact prediction performance of `tFolder`, we tested it using a 31 protein benchmark[4]. Proteins were selected from the Protein Data Bank that had dominantly beta structure and low sequence

[4] Complete benchmark results available at the `tFolder` website

homology. From each of these the β-topology was extracted and used as input for `tFolder`, along with the amino-acid sequence and fixed strand length of 4–6 residues. Since predicting folding dynamics involves a permutation over all β-topologies, this demonstrates the expected accuracy of each folding state along the pathway. We sample 500 configurations of each protein, and use these ensembles to compute a stochastic contact map and distribution of strand locations (See Figure 5(a) and 5(b) for example). The contact map represents the probability of observing a given contact, and predicted contacts are the set of all contacts with probability above a threshold value t. The selection of t influences the measured performance (Figure 5(c)), so to objectively set the threshold, we chose a t that maximizes the F-measure. We evaluated the quality of our contact maps based on the Accuracy ($\frac{no.\ of\ correctly\ predicted\ contacts}{no.\ of\ predicted\ contacts}$), Coverage ($\frac{no.\ of\ correctly\ predicted\ contacts}{no.\ of\ observed\ contacts\ contacts}$), and F-measure ($\frac{2 \cdot Accuracy \cdot Coverage}{Accuracy + Coverage}$) of our predictions. We calculated these measures in terms of β-contacts, which we defined as residues located withing β-strands less than 8Å apart (between C_α atoms) in the PDB structure. A summary of `tFolder` performance on Protein G, as well as average performance on the 31 protein dataset, is presented in Table 1. Here we distinguish between results for long range contacts, greater than 0, 12, or 24 residues apart. Thus, `tFolder` maintains reasonable predictive accuracy even with large contact separations

	Protein G						31 protein benchmark					
	Exact			$\pm$ 2			Exact			$\pm$ 2		
	≥ 0	≥ 12	≥ 24	≥ 0	≥ 12	≥ 24	≥ 0	≥ 12	≥ 24	≥ 0	≥ 12	≥ 24
Accuracy	13.3	10.6	14.0	52.1	54.1	58.3	8.6	7.0	8.4	24.2	32.1	39.8
Coverage	56.3	53.8	37.5	97.9	61.5	87.5	9.1	11.6	11.9	43.5	44.5	51.1
F-measure	21.5	17.7	20.4	68.0	57.6	70.0	8.3	9.3	19.0	27.3	29.7	45.2

Table 1. The performance of `tFolder` for contact prediction is evaluated based on the Accuracy, Coverage, and F-measure of experimentally observed contacts. These performance metrics are reported for contacts that are more than 0, 12, and 24 residues apart, showing that `tFolder` maintains reasonable predictive accuracy even with large contact separations. Additionally, these metrics are evaluated when predicted contacts are within ± 2 residues of an observed contact.

In order to evaluate the performance of `tFolder` with respect to other approaches for contact prediction, results on this protein dataset are presented in Table 2 along with a comparison with two leading contact prediction alogrithms, SVMcon and BETApro. The method SVMcon used

ten of the proteins in this dataset for the training of their SVM, so they were excluded from the evaluation for the comparison of methods. It can be seen that `tFolder` is able to perform comparably, in particular for the F-measure of contacts with sequence separation greate than 24 residues. Although these methods sometimes perform better for contact prediction, it is important to note that the predictive performance of `tFolder` is less sensitive to the distance of contact separation. Since critical protein folding steps can involve both short-range and long-range β-sheet contacts, it is especially important for long-range contacts to be predicted correctly to allow an accurate folding pathways to be reconstructed. Furthermore, since we cannot apply cross validation techniques to BETApro and SVMcon, we also indicate their performance for CASP 7.

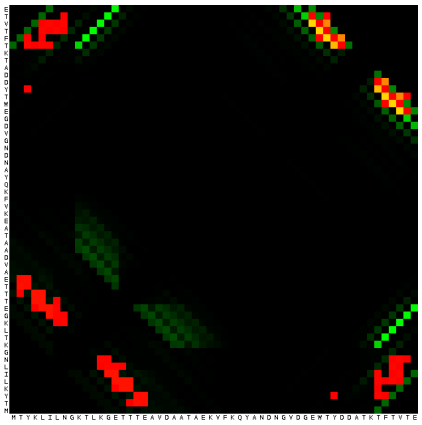
	≥ 12			≥ 24		
Method	F-measure	Accuracy	Coverage	F-measure	Accuracy	Coverage
`tFolder`	9.0 (24.0)	6.7 (29.4)	8.8 (37.2)	20.1 (41.8)	11.7 (41.0)	15.1 (44.8)
BETApro	10.8 (28.1)	41.5 (78.1)	4.8 (16.0)	6.2 (22.8)	28.0 (57.7)	1.1 (7.2)
(CASP 7)		35.4	5.1		19.7	3.2
SVMcon	27.8 (55.7)	26.7 (69.7)	32.9 (48.4)	19.9 (40.0)	15.6 (54.0)	29.1
(CASP 7)		27.7	4.7		13.1	2.8

Table 2. Comparison of the performance of `tFolder` contact prediction with contact prediction algorithms SVMcon and BETApro. The methods are evaluated based on their ability to perform contact prediction for contacts greater than 12 and 24 residue separation respectively. The metric values for contacts within ± 2 residues of an observed contact are reported in parentheses.

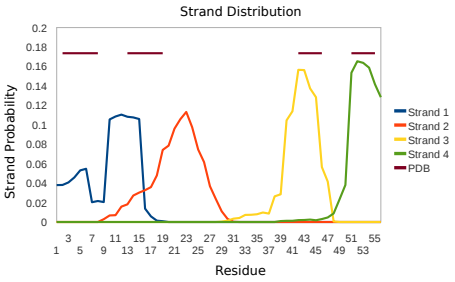
Predicting the folding pathways of the B1 domain of Protein G

To demonstrate the efficacy of our techniques for predicting protein folding pathways, we reconstruct the folding landscape of the B1 domain of Protein G — a well-studied protein for which the pathway has been elucidated through many experimental studies and MD simulations. To do this, all possible permutations of a 4-strand β-sheet topology were sampled and clustered. For each of these sets of structures, the cluster with the highest probability of being observed was selected to be representative of each topology.

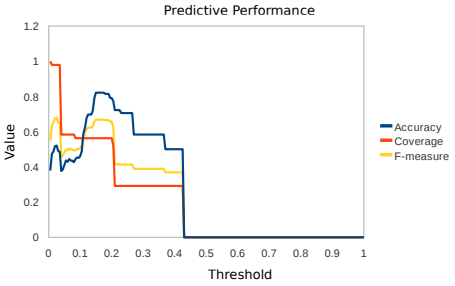
The graph of the folding pathway was constructed by considering all pairs of clusters. If the minimum distance beween two clusters was less



(a) contact map



(b) Per-residue Contact Probability



(c) Relationship between contact threshold and performance metrics

Fig. 5. Summary of the distribution of structures predicted by `tFolder` for Protein G (a) Shows the contact probability predicted by `tFolder` between all pairs of amino acids. Green squares indicate contacts predicted by `tFolder`, whereas red squares represent pairs of amino acids that are less than 8Å from each other in the observed structure. A higher intensity of green indicates a higher predicted probability of the contact, and yellow squares are an indication of agreement between prediction and observed contacts. (b) Shows the probability of the location of each strand, computed from the ensemble of sampled structures.The bars at the top of the plot indicate the location of the strands from the experimentally determined structure. (c) Shows the relationship between the threshold used to determine a contact from the contact map, and the values of the three metrics Accuracy, Coverage, and F-measure. The threshold can be set to maximize the value of the F-measure, representing a reasonable trade-off between Coverage and Accuracy.

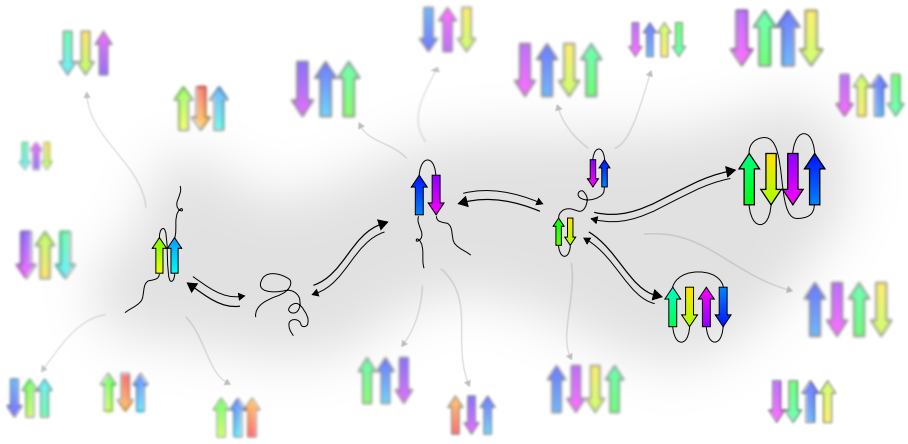
than the transition treshold, we considered that there was exchange between the two states. We tried several metrics, including segment overlap, mountain metric, and a contact based metric [24] [25], selecting the contact based metric, because it performed best empirically. The resulting graph of protein conformations is illustrated in Figure 6(a). Inspection of this graph, along with the folding dynamics computed from this graph in Figure 6(b), reveals folding intermediates consistent with those previously reported by Song *et al.* [26]. It should also be noted that although we compute other configurations of the sequence that are energetically favorable (faded states), they are not predicted to form because they are unreachable from the unfolded state. Interestingly, a four-stranded off-pathway structure is predicted to form, which has not been observed previously. Furthermore, our results agree with the work of Hubner *et al.*, who show that the anti-parallel beta-hairpin, predicted to form an interaction between residues 39–44 and 50–55, center around known nucleation points W43, Y50, F54 [27].

Algorithm running time

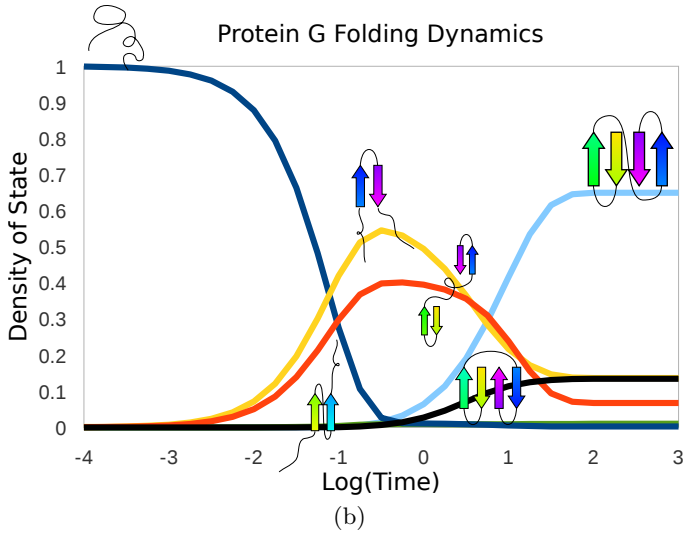
The computational bottleneck of our approach is the computation of the partition function of a template. The primary factors influencing this calculation are the length of sequence and the number of strands in the β-topology (the depth of the recursion). The partition function for sequences between 40–130 residues and 4–6 strands was calculated using a single 2.66GHz processor with 512 MB of RAM. The effect of these two parameters on the computation time is depicted in Figure 7 below. Further, computing the parition function across multiple β-templates is trivially parallelizable. The ability to formulate quick, coarse-grained predictions in a matter of minutes, rather than days of atomistic-detail simulation, is a fundamental benefit of our technique.

4 Discussion

We present `tFolder`, a novel approach for quickly predicting protein folding pathways through the accurate prediction of the conformational landscape of arbitrary β-sheet proteins. What distinguishes `tFolder` from other computational approaches that attempt to probe protein folding processes is that `tFolder` does not require vast computational resources; in fact, it can be ran on a single personal computer. To achieve this performance we use a simplified model for protein folding, allowing us to



(a)



(b)

Fig. 6. (a)The graph of the folding landscape of Protein G predicted by `tFolder` is illustrated above. The gray shaded region indicates the states predicted to be reachable from the unfolded state. The dark arrows indicate transitions between states, and the size of the arrow indicates the favored direction of transition along each edge. Faded arrows are drawn between states that have compatible topologies but do not reach the transition threshold. The size of each state indicates its relative representation at equilibrium. The faded structures indicate states that are unreachable from the unfolded state. (b)The folding dynamics of Protein G shows how the probability of observing any of the reachable states changes over the time the protein folds. Each line is annotated with an image of the state it represents.

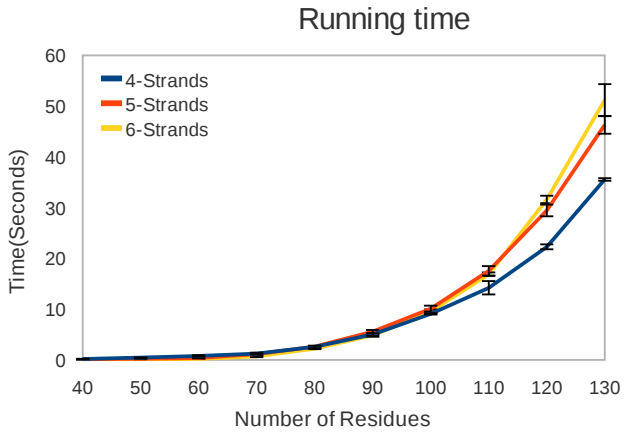


Fig. 7. The time required to compute the partition function increases with increasing size of amino acid sequence, and number of strands. The time was computed by averaging over n=3 trials, for sequences ranging from 40–130 residues in length, with 4–6 strands.

very rapidly compute a coarse-grained picture of the folding of a protein from sequence information alone. This contrasts with methods that attempt to determine folding mechanisms by trying to unfold proteins from their native state. Such methods require the *a priori* knowledge of the native structure, and as such are not applicable to study protein sequences with unknown structures. When computing protein folding pathways, our method explores all possible β-sheet configurations, and thus does not face such limitations. Interestingly, this independence from known structures could provide insights into off-pathway kinetics, such as the aggregation of proteins into amyloid structures.

Although `tFolder` only predicts coarse folding pathway transitions in β-sheet proteins, its strength lies in its ability to quickly separate conformational transitions that are critical to folding from those transitions that could simply result from minor structural fluctuations. This complements the use of MD simulations as the MD can be used to explore the nuanced structural interactions that certainly occur near a transition highlighted by `tFolder`. Further, although we are able to produce good results using a fairly simplistic energy model, a more complicated formulation, such as one including entropic forces, would clearly improve `tFolder`'s analysis. More advanced heuristics also exist [13] that more efficiently extract folding pathway information, which could be applied to `tFolder`.

Understanding the folding dynamics of β-sheet proteins, especially which β-strand contacts drive folding and conformational stability, could help create better models of hierarchical folding, protein aggregation, and evolutionary pressure. Significant overlap likely exists between many proteins' folding pathways to even permit a classification of common transition elements (e.g. [28]); however, creating such a database would only be possible with sufficiently fast and accurate algorithms. `tFolder` takes a step toward this end by demonstrating techniques for efficiently predicting ensembles of arbitrary β-sheet proteins, and for combining these predictions to construct accurate protien folding transition landscapes.

References

1. Dobson, C.M.: Protein folding and misfolding. Nature **426**(6968) (Dec 2003) 884–90
2. Karplus, M., McCammon, J.A.: Molecular dynamics simulations of biomolecules. Nat Struct Biol **9**(9) (Sep 2002) 646–52
3. Faccioli, P., Sega, M., Pederiva, F., Orland, H.: Dominant pathways in protein folding. Phys Rev Lett **97**(10) (Sep 2006) 108101
4. Voelz, V.A., Bowman, G.R., Beauchamp, K., Pande, V.S.: Molecular simulation of ab initio protein folding for a millisecond folder NTL9(1-39). J Am Chem Soc **132**(5) (Feb 2010) 1526–8
5. Levitt, M., Warshel, A.: Computer simulation of protein folding. Nature **253**(5494) (Feb 1975) 694–8
6. Tapia, L., Thomas, S., Amato, N.M.: A motion planning approach to studying molecular motions. Communications in Information and Systems **10**(1) (2010) 53–68
7. Amato, N.M., Song, G.: Using motion planning to study protein folding pathways. J Comput Biol **9**(2) (2002) 149–68
8. Hosur, R., Singh, R., Berger, B.: Sparse estimation for structural variability. Algorithms Mol Biol (2011)
9. McCaskill, J.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers **29** (1990) 1105–1119
10. Ding, Y., Lawrence, C.E.: A bayesian statistical algorithm for RNA secondary structure prediction. Comput Chem **23**(3-4) (Jun 1999) 387–400
11. Turner, D.H., Mathews, D.H.: NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res **38**(Database issue) (Jan 2010) D280–2
12. Wolfinger, M.T., Andreas Svrcek-Seiler, W.A., Flamm, C., Hofacker, I.L., Stadler, P.F.: Efficient computation of RNA folding dynamics. Journal of Physics A: Mathematical and General **37**(17) (2004)
13. Tang, X., Thomas, S., Tapia, L., Giedroc, D.P., Amato, N.M.: Simulating RNA folding kinetics on approximated energy landscapes. J Mol Biol **381**(4) (Sep 2008) 1055–67
14. Mamitsuka, H., Abe, N.: Predicting location and structure of beta-sheet regions using stochastic tree grammars. In: ISMB. (1994) 276–284

15. Chiang, D., Joshi, A.K., Searls, D.B.: Grammatical representations of macromolecular structure. J Comput Biol **13**(5) (Jun 2006) 1077–100
16. Kato, Y., Akutsu, T., Seki, H.: Dynamic programming algorithms and grammatical modeling for protein beta-sheet prediction. J Comput Biol **16**(7) (Jul 2009) 945–57
17. Tran, V.D., Chassignet, P., Sheikh, S., Steyaert, J.M.: Energy-based classification and structure prediction of transmembrane beta-barrel proteins. In: Proceedings of the First IEEE International Conference on Computational Advances in Bio and medical Sciences (ICCABS). (2011)
18. Waldispühl, J., O'Donnell, C.W., Devadas, S., Clote, P., Berger, B.: Modeling ensembles of transmembrane beta-barrel proteins. Proteins **71**(3) (May 2008) 1097–112
19. Waldispühl, J., Steyaert, J.M.: Modeling and predicting all-alpha transmembrane proteins including helix-helix pairing. Theor. Comput. Sci. **335**(1) (2005) 67–92
20. Waldispühl, J., Berger, B., Clote, P., Steyaert, J.M.: Predicting transmembrane beta-barrels and interstrand residue interactions from sequence. Proteins **65**(1) (Oct 2006) 61–74
21. Cowen, L., Bradley, P., Menke, M., King, J., Berger, B.: Predicting the beta-helix fold from protein sequence data. J. Comput. Biol (2001) 261–276
22. Ding, Y., Lawrence, C.E.: A statistical sampling algorithm for RNA secondary structure prediction. Nucleic Acids Res. **31** (Dec 2003) 7280–7301
23. Cheng, J., Baldi, P.: Improved residue contact prediction using support vector machines and a large feature set. BMC Bioinformatics **8** (2007) 113
24. Zemla, A., Venclovas, C., Fidelis, K., Rost, B.: A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. Proteins **34**(2) (1999) 220–3
25. Moulton, V., Zuker, M., Steel, M., Pointon, R., Penny, D.: Metrics on RNA secondary structures. J Comput Biol **7** (2000) 277–292
26. Song, G., Thomas, S., Dill, K.A., Scholtz, J.M., Amato, N.M.: A path planning-based study of protein folding with a case study of hairpin formation in protein G and L. Pac Symp Biocomput (2003) 240–51
27. Hubner, I.A., Shimada, J., Shakhnovich, E.I.: Commitment and nucleation in the protein G transition state. J. Mol. Biol. **336** (2004) 745–761
28. Fulton, K.F., Devlin, G.L., Jodun, R.A., Silvestri, L., Bottomley, S.P., Fersht, A.R., Buckle, A.M.: PFD: a database for the investigation of protein folding kinetics and stability. Nucleic Acids Res **33**(Database issue) (Jan 2005) D279–83