Bayesian Models and Algorithms for Protein Beta-Sheet Prediction

Zafer Aydin, *Student Member, IEEE,* Yucel Altunbasak, *Senior Member, IEEE,* and Hakan Erdogan, *Member, IEEE*

Abstract—Prediction of the three-dimensional structure greatly benefits from the information related to secondary structure, solvent accessibility, and non-local contacts that stabilize a protein's structure. We address the problem of β-sheet prediction defined as the prediction of β-strand pairings, interaction types (parallel or anti-parallel), and β-residue interactions (or contact maps). We introduce a Bayesian approach for proteins with six or less β-strands, in which we model the conformational features in a probabilistic framework by combining the amino acid pairing potentials with *a priori* knowledge of β-strand organizations. To select the optimum β-sheet architecture, we significantly reduce the search space by heuristics that enforce the amino acid pairs with strong interaction potentials. In addition, we find the optimum pairwise alignment between β-strands using dynamic programming, in which we allow any number of gaps in an alignment to model β-bulges more effectively. For proteins with more than six β-strands, we first compute β-strand pairings using the BetaPro method. Then, we compute gapped alignments of the paired β-strands and choose the interaction types and β-residue pairings with maximum alignment scores. We performed a 10-fold cross validation experiment on the BetaSheet916 set and obtained significant improvements in the prediction accuracy.

Index Terms—Protein β-sheets, open β-sheets, β-sheet prediction, contact map prediction, Bayesian modeling.

1 Introduction

A β-sheet is a set of β-strand segments, which are involved in hydrogen bonding interactions. The association of β-sheets has been implicated in the formation of protein aggregates and fibrils observed in many human diseases, including Alzheimer's and mad cow diseases [1]. β-sheets can be open, meaning that they have two edge strands (as in the flavodoxin fold or the immunoglobulin fold) or they can be closed β-barrels (such as the TIM barrel). Open β-sheets are the most common sheet types observed in cellular proteins. An example is shown in Fig. 1, where four β-strands interact pairwise to form an open β-sheet. The conformational arrangement of β-strands that form β-sheets can be described by the following components: the assignment (or grouping) of β-strands into β-sheets, the spatial ordering of β-strand segments in each sheet, the interaction types of β-strand segment pairs, and amino acid residue interactions also known as contact maps. For instance, in Fig. 1, four β-strands interact to form a single β-sheet. Here, the β-strand segments are ordered as (1-2-4-3) in the spatial direction, in which the numbers represent the sequential indices of the β-strands[1]. The interaction types of the segments are such that the first and the second segments make an anti-parallel interaction, the second and the fourth segments make the second anti-parallel interaction, while the third and the fourth segments make a parallel interaction. As the fourth component of the β-sheet formation, a contact map defines the amino acid pairs that make non-local interactions (or residue pairs). In Fig. 2, two possibilities are shown for the residue pairing pattern of a β-sheet with three β-strands. Both β-sheets have the same grouping, ordering and interaction type combination but their contact map is different. The β-sheet conformation of a protein is essential for understanding its structure [2]. Prediction of β-sheet conformation from amino acid sequence is

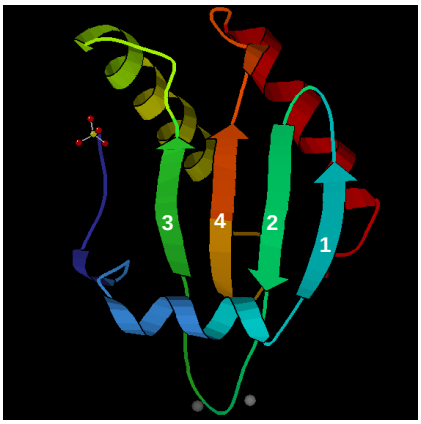


Fig. 1. Secondary structure of the Rnase P protein (PDB id: 1A6F). β-strands that form the β-sheet are numbered in sequential order.

- *Z. Aydin is with the Department of Genome Sciences, University of Washington, Seattle, WA, 98195. E-mail: zafer@u.washington.edu*
- *Y. Altunbasak is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332.*
- *H. Erdogan is with the Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey.*

1. For convenience, we start with the segment with smaller sequential index.

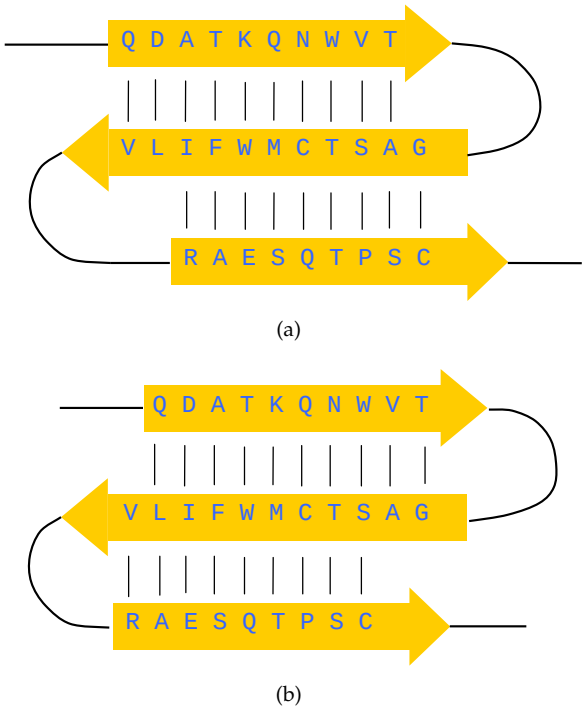


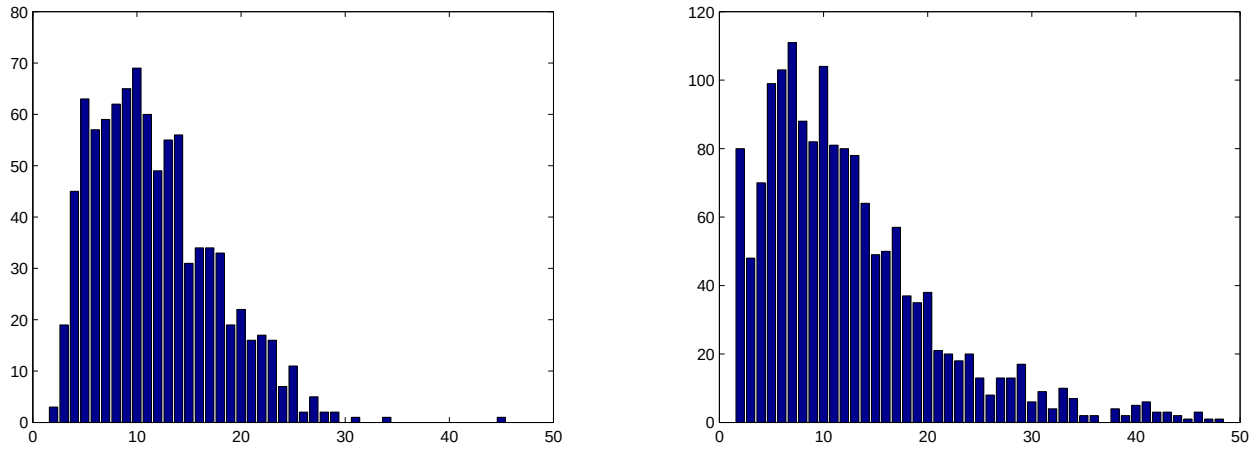
Fig. 2. Two possibilities for the residue pairing pattern of a β-sheet with three β-strands. The letters represent the amino acids in β-strand segments.

useful not only for predicting the tertiary structure [3], [4] but also for elucidating folding pathways [5], [6] and designing new proteins [7], [8]. Several methods have been proposed to understand and predict topological features of β-sheets. Methods that aim to improve our understanding of β-sheet formation analyzed the intrinsic and statistical propensities of amino acids [4], [9]–[12], their evolutionary conservation [3], [6] and the contribution of these factors to local structure and β-sheet stability [10], [13]–[15]. Methods that predict β-strand interactions and/or amino acid residue contacts utilize statistical potentials [12], [16]–[19], information theory [20] and machine learning [21]–[31]. Note that all these methods are developed for global proteins though similar ideas were also applied to predict contacts in specific folds [32] as well as transmembrane proteins that contain β-strand interactions [33], [34]. In this paper, we are concentrating on globular proteins only.

Cheng and Baldi [26] proposed BetaPro, which is a three stage modular approach that predicts and assembles the β-sheets of a native protein. BetaPro utilizes recursive neural networks followed by dynamic programming and graph theory to exploit global covariation and constraints characteristic of β-sheets. To derive the residue interaction propensities, BetaPro utilizes information from 10 surrounding residues instead of modeling each pair as independent. In a cross validation setting, BetaPro had 68% sensitivity and 61% positive predictive value (PPV) in the segment pairing category when true secondary structure and solvent accessibility information is used, which is a significant improvement over statistical data-driven approaches. BetaPro was followed by SVMcon, a new contact map predictor that uses support vector machines to predict medium- and long-range contacts [31]. Although SVMcon utilized a larger feature set, its performance was not better than BetaPro when evaluated on CASP datasets [31].

The BetaPro method [26] does not explicitly employ folding rules and does not discriminate between possible topological organizations. In other words, it treats possible groupings of β-strands into β-sheets, spatial ordering of β-strands within a sheet and interaction types of β-strand pairs equally. In a related study, Ruczinski *et al.* [4] showed that the organization of β-strands into β-sheets is not random and shows a distinct pattern. Some of the conformations are physically unstable and are never observed. For the remaining ones, there is a preference for particular orientations, which are favored more than the others. Another aspect of BetaPro is that it employs a simple greedy algorithm to compute β strand pairings and interaction types. This leaves room for more sophisticated algorithms to be developed. To address these problems, Jeong et al. [35] investigated two new algorithms for predicting β-strand partners. To make direct comparisons, they used the same scoring function as of BetaPro. The objective of the first algorithm is very similar to BetaPro. Instead of having a two-stage greedy selection heuristic, it poses the problem as integer linear programming optimization problem and solves it using the ILOG CPLEX$^{\text{TM}}$ package. The second approach is greedy and it explicitly encourages two simple folding rules. This is achieved by dynamically increasing the scores of strand pairs that are potential partners depending on the pairs predicted so far. The second algorithm performed better than the first one but the improvement over BetaPro was not drastic (a 0.7% improvement in sensitivity and 2.7% improvement in positive predictive value evaluated in the β-strand pairing category). Also they did not report the accuracy in interaction type and contact map predictions. Furthermore, their accuracy was not better than BetaPro for all separation distances between contacts. For some distances the accuracy decreased slightly. More importantly, although Jeong *et al.* [35] aimed to enforce physical constraints by incorporating folding rules into BetaPro, they considered only two simple folding rules. Therefore, for an elaborate treatment of the problem, one has to include a more comprehensive set of rules and physical preferences that guide the formation of β-sheets.

In this paper, we address the problem of β-sheet prediction defined as the prediction of β-strand pairings, interaction types (parallel or anti-parallel), and β-residue interactions (or contact maps). We analyze proteins according to the number of β-strands they contain. We consider two categories: (1) proteins with six or less β-strands; (2) proteins with more than six β-strands. In Fig. 3, histogram plots are shown for the number of β-strands in BetaSheet916 and CulledPDB datasets (see



(a) (b)

Fig. 3. Histograms for the number of β-strands in an amino acid chain: (a) BetaSheet916 set; (b) CulledPDB set.

Section 2.3). The percentage of proteins with six or less β-strands is calculated as 20.41% in BetaSheet916 and 25.51% in CulledPDB. For proteins with six or less β-strands, we introduce a Bayesian approach, in which we model the conformational features in a probabilistic framework by combining the amino acid pairing potentials with *a priori* knowledge of β-strand organizations. Starting from the amino acid sequence, secondary structure, and the amino acid pairing probability matrix computed by BetaPro, we assign probability scores to possible β-sheet architectures by considering four structure levels: (1) groupings of β-strands into β-sheets; (2) spatial arrangement of β-strands in each β-sheet; (3) interaction types of β-strands (parallel or anti-parallel); (4) residue pairing patterns (or contact maps). For the first three levels, we utilize the results of Ruczinski *et al.* [4], who performed a statistical analysis of the frequency of β-strand groupings and β-sheet motifs. For the fourth level, we use the raw amino acid pairing probabilities that are derived from the DSSP database [36], [37][2]. This approach allows us to enforce a large set of physical rules that characterize the intrinsic preferences of β-sheet formation.

To select the optimum β-sheet architecture, we search the space of possible conformations by efficient heuristics. In our computations, we significantly reduce the search space by enforcing the amino acid pairs with strong interaction propensities derived from the residue pairing propensity matrix. On this reduced search space, we sample the first three levels using a brute-force sampling approach. To derive the optimum amino acid pairing combination (*i.e.*, the contact map), we apply dynamic programming and compute pairwise alignments of β-strand pairs. For this purpose, we employ an algorithm that finds the optimum pairwise alignment of β-strands. In this algorithm, we define match as well as gap scores and perform global alignments (Needleman-Wunsch algorithm). This is a more elaborate approach as compared to the earlier work by Cheng and Baldi [26] and Jeong *et al.* [35]. Although Cheng and Baldi [26] defined and introduced a gapped alignment algorithm, they did not implement gapped alignments in BetaPro. They simply ignored gaps by sliding one segment along with the other. On the other side, Jeong et al. [35] only allowed a single gap in an alignment. We further improved the dynamic programming approach by allowing any number of gaps. The gapped nature of the alignments enables us to model β-bulges more effectively.

For proteins with more than six β-strands, the discriminative power of the Ruczinski model reduces significantly due to an exponential increase in the number of possible β-strand organizations and insufficient training data to reliably represent such conformations. Therefore, for such proteins, we first use BetaPro to compute β-strand pairings. Then, we compute gapped alignments of the paired β-strands in parallel and anti-parallel directions and choose the interaction types and β-residue pairing patterns with maximum alignment scores.

2 METHODS

2.1 β-sheet Prediction for Proteins with $\leq$ 6 β-strands: A Bayesian Approach

We will formulate the β-sheet prediction in a probabilistic framework. Before providing the mathematical details, we first define our model parameters.

2.1.1 Model Parameters

The input parameters are the amino acid sequence, the secondary structure, and an amino acid pairing propensity matrix. The amino acid sequence is denoted by $\mathbf{R}$,

2. The BetaPro's pairing probability matrix is not used in scoring the conformations

where $\mathbf{R}_i$ is the i^{th} amino acid. Similarly, the secondary structure is represented by $\mathbf{SS}$, where $\mathbf{SS}_i$ is the secondary structure state of the i^{th} amino acid (whether it is α-helix, β-strand or loop). Note that, as a necessary condition for β-sheet formation, $\mathbf{SS}$ should contain at least two β-strand segments. The third parameter is denoted by $\mathbf{PP}$, where $\mathbf{PP}_{ij}$ is the probability of the i^{th} and j^{th} β-residues to make a pair (or contact). In this matrix, only the amino acids in β-strands or β-bridges are considered (*i.e.*, E or B states in the DSSP assignment [37]). The pairing probability matrix is computed using the BetaPro method [26] and is utilized to reduce the space of possible β-sheet conformations.

The output parameters are the grouping sequence $\mathbf{G}$, the ordering sequence $\mathbf{O}$, the interaction type sequence $\mathbf{I}$, and the contact map (or the residue pairing sequence) $\mathbf{C}$. We explain each parameter in more detail.

- $\mathbf{G}$ defines the number of β-sheets as well as the grouping of β-strands into β-sheets. In other words, $\mathbf{G}$ contains the information about which β-strands appear together in each β-sheet. Here, the ordering of β-strands is not important, therefore they are ordered in the sequential order to remove ambiguity. $\mathbf{G}$ is a 2D sequence, where $\mathbf{G}(p,l)$ is the sequence index of the l^{th} β-strand in the p^{th} β-sheet. For the β-sheet in Fig. 1(a), $\mathbf{G} = (1,2,3,4)$ meaning that all β-strands form a single β-sheet.
- $\mathbf{O}$ specifies the spatial ordering of β-strands within each β-sheet. $\mathbf{O}$ is a 2D sequence, where $\mathbf{O}(p,l)$ is the spatial order of the l^{th} β-strand in the p^{th} β-sheet. If the p^{th} β-sheet contains n_p β-strands, then $\mathbf{O}(p,:)$ (also denoted by $\mathbf{O}_p$) is simply a permutation of the sequence 1:n_p. Therefore, in this notation, $\mathbf{O}$ can be represented as the concatenation of $\mathbf{O}_p$'s. This is formulated as $\mathbf{O} = \Upsilon_p \mathbf{O}_p$, where Υ is the sequence concatenation operator. Note that, in our model, a permutation and its inverse represent the same spatial ordering because we only keep permutations, in which the sequential index of the first segment is lower than the index of the last segment. The spatial ordering information also specifies the β-strand segments that interact with each other. For the β-sheet in Fig. 1(a), $\mathbf{O} = (1,2,4,3)$ meaning that the first β-strand interacts with the second, the second with the fourth, and the fourth with the third. The pairwise interactions are bidirectional. Here, for simplicity, we assume that a segment can interact with up to two neighboring segments. The percentage of proteins that have six or less β-strands and that contain interactions with more than two segments is only 1.7% in the BetaSheet916 set (see Section 2.3.2). Extension of the model to characterize interactions with more than two neighbors is not a difficult task and is left as a future work (see Section 4). Note that, for proteins with more than six β-strands, we are not putting any restriction on the number of interactions a β-strand makes (see Section 2.2).
- $\mathbf{I}$ determines the interaction types (parallel or anti-parallel) of β-strand pairs in each sheet. $\mathbf{I}$ is a 2D sequence, where $\mathbf{I}(p,l)$ is the interaction type between the l^{th} and $(l+1)^{th}$ β-strands in the p^{th} β-sheet represented in the spatial order. We set $\mathbf{I}(p,l) = P$ if the l^{th} β-strand is parallel to the $(l+1)^{th}$ β-strand. If two neighboring β-strands are anti-parallel, we set $\mathbf{I}(p,l) = AP$. For the β-sheet in Fig. 1(a), $\mathbf{I} = (AP, AP, P)$. Similar to the ordering sequence, $\mathbf{I}$ can be decomposed into its subcomponents denoted by $\mathbf{I}_p = \mathbf{I}(p,:)$. This is formulated as $\mathbf{I} = \Upsilon_p \mathbf{I}_p$, where Υ is the sequence concatenation operator.
- $\mathbf{C}$ describes the non-local residue pairing pattern or the contact map arising from the amino acid interactions in each β-sheet. In our model, we assume that an amino acid can make a residue pairing interaction with up to 2 other amino acids. There can be various formats to represent the contact map. The first one is the classical representation, where a 2D sequence $\bar{\mathbf{C}}$ of size $n_R \times n_R$ is used. Here, n_R is the total number of amino acids labeled as β-strand and the amino acid residues in β-strand segments (β-residues) are indexed following the sequential order (*i.e.*, from the N-terminus to the C-terminus of the protein). $\bar{\mathbf{C}}(i,j)$ is set to 1 if the i^{th} β-residue interacts with the j^{th} β-residue. If there is no interaction between the residue pair, then $\bar{\mathbf{C}}(i,j)$ is set to 0. As an alternative representation, we can only keep the indices of the residue pairs that make residue pairing interaction and store them in a 2D sequence denoted by $\mathbf{C}$. In other words, we only keep the residue indices for which $\bar{\mathbf{C}}$ is 1. In this representation, each row of $\mathbf{C}$ corresponds to a β-sheet and contains the indices of the amino acid residue pairs that make chemical interactions. For instance, if the 5^{th} amino acid interacts with the 3^{rd} and 21^{th} amino acids, and if they all belong to the p^{th} β-sheet then $\mathbf{C}(p,:) = \mathbf{C}_p$ contains (3,5,5,21). This notation is equivalent to the classical contact map representation in the sense that given the secondary structure segmentation $\mathbf{SS}$, it is possible to convert one to the other. Similar to $\mathbf{O}$ and $\mathbf{I}$, $\mathbf{C}$ can be decomposed into subcomponents denoted by $\mathbf{C}_p$. In addition, we can decompose each $\mathbf{C}_p$ into its subcomponents designated by $\mathbf{C}_p^m$. Here, $\mathbf{C}_p^m$ contains the set of residue pairs that connect a pair of β-strands and m runs from 1 to $n_S^p - 1$, where n_S^p is the number of β-strand segments in the p^{th} β-sheet. In that case, $\mathbf{C}_p^m$ can be concatenated to form $\mathbf{C}_p$ and likewise $\mathbf{C}_p$ can be concatenated to form $\mathbf{C}$. This is expressed as $\mathbf{C}_p = \Upsilon_m \mathbf{C}_p^m$ and $\mathbf{C} = \Upsilon_p \mathbf{C}_p$, where Υ is the sequence concatenation operator.

2.1.2 Problem Definition

In β-sheet prediction, the goal is to predict the overall β-sheet conformation of the protein given the input

variables. Since the contact map $\mathbf{C}$ contains all the information in the parameter set $(\mathbf{G}, \mathbf{O}, \mathbf{I})$, the problem reduces to finding the optimum contact map or the residue pairing structure. This is formulated as

$$\mathbf{C}^{max} = \arg\max_{\mathbf{C}} P(\mathbf{C} \mid \mathbf{D}), \tag{1}$$

where $\mathbf{C}^{max}$ is the MAP estimator, which corresponds to the contact map (or equivalently the conformation) maximizing the *a posteriori* probability $P(\mathbf{C} \mid \mathbf{D})$, and $\mathbf{D}$ is a short-hand notation for $(\mathbf{R}, \mathbf{SS}, \mathbf{PP})$, *i.e.*, the set of input variables defined in Section 2.1.1. The posterior probability can be modeled as

$$\begin{aligned} P(\mathbf{C} \mid \mathbf{D}) &= P(\mathbf{C}, \mathbf{G}, \mathbf{O}, \mathbf{I} \mid \mathbf{D}) && (2)\\ &= P(\mathbf{G}, \mathbf{O}, \mathbf{I} \mid \mathbf{D}) \times P(\mathbf{C} \mid \mathbf{G}, \mathbf{O}, \mathbf{I}, \mathbf{D}) && (3)\\ &= P(\mathbf{G} \mid \mathbf{D}) \times P(\mathbf{O}, \mathbf{I} \mid \mathbf{G}, \mathbf{D}) && (4)\\ &\times\ P(\mathbf{C} \mid \mathbf{G}, \mathbf{O}, \mathbf{I}, \mathbf{D}), \end{aligned}$$

Given the grouping vector, which specifies the assignment of β-strands into β-sheets, we model the terms $P(\mathbf{O}, \mathbf{I} \mid \mathbf{G}, \mathbf{D})$ and $P(\mathbf{C} \mid \mathbf{G}, \mathbf{O}, \mathbf{I}, \mathbf{D})$ as

$$P(\mathbf{O}, \mathbf{I} \mid \mathbf{G}, \mathbf{D}) = \prod_k P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D}) \tag{5}$$

$$P(\mathbf{C} \mid \mathbf{G}, \mathbf{O}, \mathbf{I}, \mathbf{D}) = \prod_k P(\mathbf{C}_k \mid \mathbf{G}, \mathbf{O}_k, \mathbf{I}_k, \mathbf{D}), \tag{6}$$

where the vectors $\mathbf{O}_k$, $\mathbf{I}_k$, and $\mathbf{C}_k$ denote the ordering, interaction type and the contact map of the k^{th} β-sheet, respectively. With this formulation, we assume that β-sheets[3] are independent from each other. We further assume that

$$P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D}) = P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D}) \tag{7}$$

$$P(\mathbf{C}_k \mid \mathbf{G}, \mathbf{O}_k, \mathbf{I}_k, \mathbf{D}) = P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D}), \tag{8}$$

where the arrangements within a β-sheet is modeled as independent from the grouping vector $\mathbf{G}$.

2.1.3 Bayesian Models

In this section, we concentrate on the modeling of $P(\mathbf{G} \mid \mathbf{D})$, $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D})$, and $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$.

2.1.3.1 $P(\mathbf{G} \mid \mathbf{D})$: We model the grouping term using the distributions introduced in Ruczinski [38]:

$$P(\mathbf{G} \mid \mathbf{D}) = P(SD \mid n_{SH}, n_S) \times P(n_{SH} \mid n_S), \tag{9}$$

where n_S is the number of β-strand segments in $\mathbf{SS}$, n_{SH} is the number of β-sheets in $\mathbf{G}$, SD is the sheet decomposition term, which defines the assignment of β-strands into β-sheets. Analyzing the available data, Ruczinski [38] derived probability models for computing $P(SD \mid n_{SH}, n_S)$ and $P(n_{SH} \mid n_S)$ (see the thesis chapter of Ruczinski [38] for further details). In this paper, we used the same models as in Ruczinski [38] for $P(SD \mid n_{SH}, n_S)$ and $P(n_{SH} \mid n_S)$.

3. Note that β-strands are not assumed to be independent.

2.1.3.2 $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D})$: The vector $(\mathbf{O}_k, \mathbf{I}_k)$ defines a structural unit known as a β-sheet motif. Ruczinski *et al.* [4] developed probabilistic models to compute the motif-likelihood distribution. We model $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D})$ as

$$P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D}) = P(\mathbf{O}_k, \mathbf{I}_k \mid H, L), \tag{10}$$

where H is the helical status of the protein (helical or non-helical), and L is the connector lengths between the strands given as indicators (short or long). Here, a protein is considered to be helical if at least 20% of its amino acids are part of an α-helix, and a connector is defined as a set of segmental residues, which connect two β-strands. Note that, connectors can include α-helices and loops. In BetaZa, we used the same model as in Ruczinski [38] for the motif distribution term (see the thesis chapter of Ruczinski [38] for further details).

2.1.3.3 $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$: Let the k^{th} β-sheet contain r β-strand segments, which are represented by $\mathcal{B}_1$, $\mathcal{B}_2$,...,$\mathcal{B}_r$ in the spatial order. We model $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$ as

$$P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D}) = \frac{P(\mathbf{C}_k \mid \mathbf{D})}{\sum_{(\mathbf{C}'_k \mid \mathbf{O}_k, \mathbf{I}_k)} P(\mathbf{C}'_k \mid \mathbf{D})} \tag{11}$$

$$P(\mathbf{C}_k \mid \mathbf{D}) = \prod_{m=1}^{r-1} P(\mathbf{C}_k^m \mid \mathbf{D}) \tag{12}$$

$$\begin{aligned} P(\mathbf{C}_k^m \mid \mathbf{D}) &= \prod_p P(BP = 1 \mid R_m^p, R_{m+1}^p) && (13)\\ &\times \prod_q P(BP = 0 \mid R_m^q)\\ &\times \prod_r P(BP = 0 \mid R_{m+1}^r). \end{aligned}$$

Eqs. 13, and 12 simply compute $P(\mathbf{C}_k \mid \mathbf{D})$ and Eq. 11 normalizes it over the possible conformations to obtain the conditional probability. In Eq. 12, $\mathbf{C}_k^m$ is the residue pairing pattern of the m^{th} segment pair $(\mathcal{B}_m, \mathcal{B}_{m+1})$ of the k^{th} β-sheet. In other words, it is a subset of $\mathbf{C}_k$ and defines the interactions (or contacts) between $\mathcal{B}_m$ and $\mathcal{B}_{m+1}$. Concatenation of $\mathbf{C}_k^m$ with respect to m gives $\mathbf{C}_k$ (see the definition of $\mathbf{C}$ in Section 2.1.1). In Eq. 13, $P(BP = 1 \mid R_m^p, R_{m+1}^p)$ represents the probability of an amino acid pair to make a residue pairing interaction, and the terms $P(BP = 0 \mid R_m^q)$, $P(BP = 0 \mid R_{m+1}^r)$ represent the probability of an amino acid in one segment not to make an interaction with any amino acid in the opposite segment. In this formulation, BP is an indicator function that is set to 1 when a pair of amino acid residues make a residue pairing interaction, and 0 when an amino acid residue does not make any interaction with the opposing segment. The range of p, q, and r depends on the number of contacts defined in $\mathbf{C}_k^m$. Furthermore, the residue pairing interactions and residues not making any interaction with the opposing segment can be numbered in any order. $P(BP = 1 \mid R_m^p, R_{m+1}^p)$, $P(BP = 0 \mid R_m^q)$, and $P(BP = 0 \mid R_{m+1}^r)$ can be reliably estimated from the latest available data because they

$$\sum_{(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k)} P(\mathbf{C}_k \mid \mathbf{D}) = \sum_{(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k)} \prod_{m=1}^{r-1} P(\mathbf{C}_k^m \mid \mathbf{D}) \quad (14)$$

$$= \sum_{(\mathbf{C}_k^1 \mid \mathbf{O}_k, \mathbf{I}_k)} \ldots \sum_{(\mathbf{C}_k^{r-1} \mid \mathbf{O}_k, \mathbf{I}_k)} \prod_{m=1}^{r-1} P(\mathbf{C}_k^m \mid \mathbf{D}) \quad (15)$$

$$= \prod_{m=1}^{r-1} \sum_{(\mathbf{C}_k^m \mid \mathbf{O}_k, \mathbf{I}_k)} P(\mathbf{C}_k^m \mid \mathbf{D}). \quad (16)$$

do not contain any dependency to $(\mathbf{O}_k, \mathbf{I}_k)$. Eq. 12 computes the contact map score of the k^{th} β-sheet.

In Eq. 11, the sum of scores is computed for all possible residue pairing patterns that are realizable for a given $(\mathbf{O}_k, \mathbf{I}_k)$. This value can be efficiently computed as in Eqs. 14, 15, and 16. Eq. 14 follows from Eq. 12. In Eqs. 15 and 16, instead of sampling all possible $\mathbf{C}_k$ one by one, we sample all possible $\mathbf{C}_k^m$ for the segment pairs in $\mathbf{C}_k$ and take the product of sums to get the sum of $P(\mathbf{C}_k \mid \mathbf{D})$ values. The logic behind this approach can also be explained by the following equation, where the sums of products is converted to the product of sums.

$$\sum_i \sum_j \sum_k X_i Y_j Z_k = (\sum_i X_i) \times (\sum_j Y_j) \times (\sum_k Z_k). \quad (17)$$

The sum of the scores of all possible contact maps for a segment pair (*i.e.*, $\sum_{(\mathbf{C}_k^m \mid \mathbf{O}_k, \mathbf{I}_k)} P(\mathbf{C}_k^m \mid \mathbf{D})$) can be computed using dynamic programming, which is explained in Section 2.1.4.1.3.b. To illustrate how the term $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$ is computed, it is useful to consider the example shown in Fig. 2(a). Let the upper β-strand segment be the first segment of the sheet, which is denoted by $\mathcal{B}_1$. We need to first compute $P(\mathbf{C}_k \mid \mathbf{D})$ using Eqs. 13 and 12:

$$P(\mathbf{C}_k \mid \mathbf{D}) = P(\mathbf{C}_k^1 \mid \mathbf{D}) \times P(\mathbf{C}_k^2 \mid \mathbf{D}),$$

where $\mathbf{C}_k^1$ is the contact map (or the residue pairing interaction pattern) for the segment pair $(\mathcal{B}_1, \mathcal{B}_2)$, and $\mathbf{C}_k^2$ is the contact map for the segment pair $(\mathcal{B}_2, \mathcal{B}_3)$. The terms $P(\mathbf{C}_k^1 \mid \mathbf{D})$ and $P(\mathbf{C}_k^2 \mid \mathbf{D})$ become:

$$\begin{aligned} P(\mathbf{C}_k^1 \mid \mathbf{D}) &= P(BP = 1 \mid Q, V) \times P(BP = 1 \mid D, L) \\ &\quad \times \ldots \times P(BP = 0 \mid G) \\ P(\mathbf{C}_k^2 \mid \mathbf{D}) &= P(BP = 0 \mid V) \times P(BP = 0 \mid L) \\ &\quad \times P(BP = 1 \mid I, R) \times \ldots \times P(BP = 1 \mid G, C), \end{aligned}$$

where $P(BP = 1 \mid Q, V)$ is the probability of the amino acid Q to make a residue pairing interaction with the amino acid V in the second segment, and $P(BP = 0 \mid G)$ is the probability of the amino acid G not to make a residue pairing interaction with any amino acid in the second segment. Then, $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$ can be computed using Eqs. 11- 16. In the next section, we will explain the algorithms developed for efficient computation of the optimum β-sheet conformation.

2.1.4 Computational Methods

2.1.4.1 Sampling the Search Space and Computation of the Optimum Conformation: To determine the most likely β-sheet conformation, it is necessary to search the space of conformations using efficient algorithms. There can be many alternatives for grouping β-strands into β-sheets, ordering them spatially, defining their interaction types, and matching their amino acids. Although the number of possible conformations rises exponentially with the number of β-strands [4], we can reduce the computational cost by shrinking the search space to a reasonable subspace and applying efficient sampling algorithms. For the first objective, we impose β-strand segments as well as residue pairs that are predicted by the BetaPro method [26] as strong interactions. In addition, we eliminate motifs from the search space that have reasonably small motif scores. Details on space reduction methods can be found in Sections 2.1.4.1.1 and 2.1.4.1.2. For the second objective, we follow a hierarchical approach to sample the search space. We observed that if we sample the possible $\mathbf{C}$ patterns after sampling $(\mathbf{G}, \mathbf{O}, \mathbf{I})$, then we make redundant computations for some β-strand pairs. Therefore, given the amino acid sequence $\mathbf{R}$ and the secondary structure $\mathbf{SS}$, we first compute the optimum residue pairing interactions (or alignments) between all β-strand segment pairs in $\mathbf{SS}$, and store them together with their alignment scores in a table. For a protein with n_S β-strands, there are $n_S(n_S - 1)/2$ possible segment pairs. For each segment pair, we compute both parallel and anti-parallel alignments. Hence, the total number of segment alignments becomes $n_S(n_S - 1)$. The optimum alignment between two β-strand segments can be computed using the Needleman-Wunsch algorithm [39], [40], which is the global pairwise sequence alignment algorithm. Details of the Needleman-Wunsch implementation can be found in Section 2.1.4.1.3. After computing pairwise alignments of all possible segment pairs, we sample β-sheet conformations hierarchically. We first sample $\mathbf{G}$, and for each $\mathbf{G}$, we sample $(\mathbf{O}, \mathbf{I})$. Here, we assume that the β-sheets in $\mathbf{G}$ are independent and sample possible $(\mathbf{O}_k, \mathbf{I}_k)$ values for each β-sheet separately[4]. If a particular $(\mathbf{O}_k, \mathbf{I}_k)$ combination contradicts with the significant segment pairs and their directions derived using BetaPro, then

4. $k = 1, \ldots, r$, where r is the number of β-sheets in $\mathbf{G}$.

we eliminate that $(\mathbf{O}_k, \mathbf{I}_k)$ from the search space. For instance, if segments 1 and 2 have strong interaction propensity but $\mathbf{O}_k$ pairs segment 1 with segment 3, then we eliminate $\mathbf{O}_k$ from the search space. In the next step, for a given $(\mathbf{O}_k, \mathbf{I}_k)$ and $\mathbf{G}$, we simply select the best scoring residue pairing pattern $\mathbf{C}_k^*$ using the alignments we computed earlier. This is formulated as:

$$\mathbf{C}_k^* = \arg\max_{\mathbf{C}_k} P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{G}, \mathbf{D}). \quad (18)$$

Since $\mathbf{C}_k$ can be represented as the concatenation of $(\mathbf{C}_k^m)$, the following relation holds for the optimum contact map of the k^{th} β-sheet:

$$\mathbf{C}_k^* = \Upsilon_m(\mathbf{C}_k^m)^*, \quad (19)$$

where Υ is the concatenation operator, $(\mathbf{C}_k^m)$ is the subset of $(\mathbf{C}_k)$ that defines the contact map (or the alignment) between the m^{th} β-strand pair of the k^{th} β-sheet, and $(\mathbf{C}_k^m)^*$ is the optimum contact map for that segment pair. Hence, for a given $(\mathbf{O}_k, \mathbf{I}_k, \mathbf{G})$ combination, the optimum contact map of the k^{th} β-sheet is constructed by concatenating the optimum contact maps (or the alignments) of the individual β-strand pairs (see the definition of $\mathbf{C}$ in Section 2.1.1).

After computing the optimum contact map for a given $(\mathbf{O}_k, \mathbf{I}_k, \mathbf{G})$, we select the best scoring ordering and interaction pattern $(\mathbf{O}_k^*, \mathbf{I}_k^*)$ for the k^{th} β-sheet as

$$(\mathbf{O}_k^*, \mathbf{I}_k^*) = \arg\max_{(\mathbf{O}_k, \mathbf{I}_k)} \{P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D}) \times P(\mathbf{C}_k^* \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{G}, \mathbf{D})\}. \quad (20)$$

Let $\mathbf{C}_k^{**}$ be the optimum contact map for $(\mathbf{O}_k^*, \mathbf{I}_k^*)$. In other words, $\mathbf{C}_k^{**}$ is the optimum among $\mathbf{C}_k^*$ values derived for each $(\mathbf{O}_k, \mathbf{I}_k)$. In the next step, we can combine the optimum ordering, interaction and contact map of all β-sheets and obtain $(\mathbf{O}^*, \mathbf{I}^*, \mathbf{C}^*)$ for a given $\mathbf{G}$

$$(\mathbf{O}^*, \mathbf{I}^*, \mathbf{C}^*) = \Upsilon_{k=1}^r(\mathbf{O}_k^*, \mathbf{I}_k^*, \mathbf{C}_k^{**}) \quad (21)$$

Finally, the best scoring grouping $\mathbf{G}^{max}$ and the best scoring contact map $\mathbf{C}^{max}$ can be found as in Eq. 22. The algorithm for finding the optimum β-sheet conformation is summarized in Algorithm 4.

To reduce the number of computations, we applied various constraints and eliminated the low scoring conformations. In the next two sections, we explain space reduction techniques in more detail. Then, we explain how we compute the best scoring alignment between a pair of β-strands.

1. *Constraint Based Reduction of the Search Space*

To sample possible grouping combinations (*i.e.*, $\mathbf{G}$ values), we utilize a simple recursive algorithm and perform an exhaustive search. Similarly, for each β-sheet in $\mathbf{G}$, we sample every possible β-sheet motif, *i.e.*, $(\mathbf{O}, \mathbf{I})$ combinations. If the likelihood of a motif is less than the motif threshold ($P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D}) < t_1$), then we eliminate that motif from the search space and do not make any further computations. We chose $t_1 = 1e-20$, a number close to zero to eliminate unlikely motifs.

Algorithm 1: Computation of the Optimum β-Sheet Conformation

Input: Amino acid sequence $\mathbf{R}$, secondary structure $\mathbf{SS}$, BetaPro's residue pairing probability matrix PP, Bayesian model.
Output: Optimum β-sheet Conformation: $(\mathbf{G}^{max}, \mathbf{O}^{max}, \mathbf{I}^{max}, \mathbf{C}^{max})$

Extract β-strand segments and residue pairs with strong interaction propensities from PP;
Compute optimum pairwise alignments of β-strand segments both in parallel and anti-parallel orientation. Impose amino acid pairs with strong interaction propensities derived in step 1;
maximum overall score = 0;
for *each* $\mathbf{G}$ **do**
  grouping score = $P(\mathbf{G} \mid \mathbf{D})$;
  for *each* β*-sheet in* $\mathbf{G}$ **do**
    k = index of the β-sheet;
    maximum joint score$_k$ = 0;
    for *each* $(\mathbf{O}_k, \mathbf{I}_k)$ *in the* β*-sheet* **do**
      if $(\mathbf{O}_k, \mathbf{I}_k)$ *contradicts with the significant segment pairs and their directions derived in step 1* **then**
        continue with the next $(\mathbf{O}_k, \mathbf{I}_k)$;
      motif score = $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D})$;
      if *(motif score < motif threshold)* **then**
        continue with the next $(\mathbf{O}_k, \mathbf{I}_k)$;
      else
        Find $\mathbf{C}_k^*$, the optimum contact map of the β-sheet for a given $(\mathbf{O}_k, \mathbf{I}_k)$ using the table of alignments computed earlier.
        joint score = $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{G}, \mathbf{D}) \times P(\mathbf{C}_k^* \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{G}, \mathbf{D})$;
        if *(joint score > maximum joint score$_k$)* **then**
          $(\mathbf{O}_k^*, \mathbf{I}_k^*) = (\mathbf{O}_k, \mathbf{I}_k)$;
          maximum joint score$_k$ = joint score;
          $\mathbf{C}_k^{**} = \mathbf{C}_k^*$;
    all sheets score = $\prod_k$ maximum joint score$_k$;
    $(\mathbf{O}^*, \mathbf{I}^*, \mathbf{C}^*) = \Upsilon_{k=1}^r(\mathbf{O}_k^*, \mathbf{I}_k^*, \mathbf{C}_k^{**})$;
    overall score = grouping score $\times$ all sheets score;
    if *(overall score > maximum overall score)* **then**
      maximum overall score = overall score;
      $\mathbf{G}^{max} = \mathbf{G}$;
      $(\mathbf{O}^{max}, \mathbf{I}^{max}, \mathbf{C}^{max}) = (\mathbf{O}^*, \mathbf{I}^*, \mathbf{C}^*)$;

This approach allows us to reduce the set of candidate conformations. The same approach can also be applied when sampling the $\mathbf{G}$ values, particularly when the number of β-strands is reasonably high.

2. *Search Space Reduction using BetaPro*

To further reduce the space of configurations, we found it useful to utilize the amino acid pairs predicted by BetaPro [26] with significant scores. BetaPro generates a pairing probability matrix, for all β-strand residue pairs using secondary structure, solvent accessibility and PSSM profiles. In this table, each entry is a real value in the range [0, 1] and represents the propensity of an amino acid pair to make a contact. If the total number of amino acids that are labeled as β-strands is n_R, then the size of the pairing probability matrix becomes $n_R \times n_R$. We observed that when the residue pairing score is

$$\begin{aligned} \mathbf{G}^{max} &= \arg\max_{\mathbf{G}} \{P(\mathbf{G} \mid \mathbf{D}) \times P(\mathbf{O}^*, \mathbf{I}^* \mid \mathbf{G}, \mathbf{D}) \times P(\mathbf{C}^* \mid \mathbf{O}^*, \mathbf{I}^*, \mathbf{G}, \mathbf{D})\} \\ (\mathbf{O}^{max}, \mathbf{I}^{max}, \mathbf{C}^{max}) &= \arg \max_{(\mathbf{O}^*, \mathbf{I}^*, \mathbf{C}^*)} \{P(\mathbf{O}^*, \mathbf{I}^* \mid \mathbf{G}^{max}, \mathbf{D}) \times P(\mathbf{C}^* \mid \mathbf{O}^*, \mathbf{I}^*, \mathbf{G}^{max}, \mathbf{D})\} \end{aligned} \quad (22)$$

above a certain threshold, then with high confidence there is a contact between the pair. Let $S_{res-pair}$ denote the residue pairing score for a pair of amino acid residues. We consider two categories: (1) high scoring residue pairs ($S_{res-pair} > 0.16$); (2) mid scoring residue pairs ($0.02 < S_{res-pair} \leq 0.16$).[5]

We apply the following heuristics before aligning the β-strand segments. For each segment pair, we first select the corresponding sub-block from the BetaPro's pairing probability matrix and identify whether the segments form a significant pair. To align the i^{th} and j^{th} segments, we choose the sub-array in the pairing probability matrix where the rows of the sub-array correspond to the i^{th} segment and columns to the j^{th} segment. The size of this block becomes $n_r \times n_c$, where n_r and n_c are equal to the number of amino acid residues in the i^{th} and j^{th} segments, respectively. Then, we search the diagonals of the sub-block (both in parallel and anti-parallel directions) and check if there is a high or mid scoring residue pair (see Fig. 4). If the number of high scoring residue pairs in a diagonal is greater than equal to two, then we flag the segment pair as high scoring and store it in a table. If the total number of high scoring residue pairs in all diagonals is less than two, then we check if there is a mid-scoring residue pair. Similar to the high scoring case, we search the diagonals of the sub-block and identify mid-scoring residue pairs. If the number of mid scoring residue pairs in a diagonal is greater than equal to three and if the average score of such pairs is greater than or equal to 0.08, then we flag the segment pair as mid scoring and store it in a table. The average score for a set of amino acid pairs is computed simply as the sum of the residue pairing scores divided by the total number of residue pairs.

After assigning a segment pair to the high or mid scoring category, we select the high and mid scoring residue pairs for those segments. If the segment pair is in the high scoring category, we first find the diagonal on the probability matrix that has the highest average residue pairing score and select the significant residue pairs on that diagonal. Then, we eliminate the diagonals that share the same rows and columns with the best scoring diagonal. Finally, we select the diagonals that are immediate neighbors of the best diagonal. The steps of the selection process is illustrated in Fig. 5. This approach ensures that each amino acid residue makes at most one contact with the partner β-strands and allows gapped alignments. For example, the diagonals (a) and (d) in Fig. 5 should generate the alignment shown in Fig. 6.

5. All the thresholds used in this section are found empirically.

For mid scoring residue pairs, we scan the diagonals of the sub-block (both in parallel and anti-parallel directions) and store the residue pairs for which the average diagonal score is the highest (see Fig. 7). Here, we do not consider a second neighboring diagonal because for mid-scoring segments the residue pairing probabilities take lower values and hence the signal to noise ratio is smaller. However, we still allow gapped alignments for the mid-scoring case. The only difference is gapped alignments are not imposed by residue pairs derived from BetaPro as in the high scoring case. The average score of a diagonal is again computed as the sum of the mid scoring residue pairs on that diagonal divided by the total number of residue pairs.

After storing segment and residue pairs with significant scores, we sort the segment pairs according to the average residue pair score. Then, we eliminate segment pairs that contribute to a cycle using a simple cycle detection algorithm from the graph theory. This step is necessary because our model does not cover β-barrels which are characterized by cyclic segment graphs. As an example to a cyclic pairing graph we can consider the following segment pairs 1-2, 2-3, 3-1, in which the segments 1 to 3 form a cyclic interaction graph. Our cycle elimination algorithm is as follows. We first check if the stored segment pairs form a cycle. This could be achieved using a simple cycle detection algorithm [41]. If there is a cycle, then we remove a segment pair with the lowest average residue pair score and check for cycles again. If there is no cycle, we terminate. If there is still a cycle, then we insert the removed segment pair back to the table and remove the second lowest segment pair. We continue until no cycle condition is satisfied. If no cycle

	M	K	T	V	D	A	S	D	P
H									
D									
V									
S									
K							(b)		
R		(a)							
S									

Fig. 4. A sub-block of the BetaPro's residue pairing probability matrix. Each entry represents the probability of an amino acid pair to make a contact. In this figure, the segments being compared are HDVSKRS and MKTV-DASDP. Diagonals of the sub-block are searched for high and mid scoring residue pairs: (a) a diagonal in parallel direction, (b) a diagonal in anti-parallel direction.

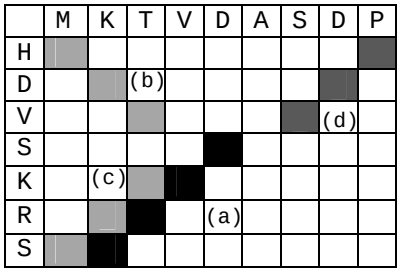


Fig. 5. Identifying high-scoring residue pairs for a high scoring segment pair. (a): The diagonal with the best average residue pairing score. (b) and (c): Diagonals that are eliminated for sharing the same rows and columns with the best scoring diagonal. (d): A neighbor of the top scoring diagonal. The selected residue pairs are: H-P, D-D, V-S, S-D, K-V, R-T, S-K.

condition is not satisfied by removing a single segment pair, then this means that there is more than one cycle. In that case, we explicitly identify the cycles including their edges and vertices and remove from each cycle the lowest scoring segment pair. Details of the heuristics applied in this section is summarized in Algorithm 18.

After identifying segments and residue pairs that are going to be imposed in subsequent steps, we align every possible segment pair considering the residue pairs with significant scores. This is explained in the next section.

3. *Pairwise Alignment of Segments using the Needleman-Wunsch Algorithm*

We used the Needleman-Wunsch algorithm [39], [40] to compute the optimum alignment between a pair of β-strand segments. The classical implementation of the algorithm uses dynamic programming and consists of three steps: (1) initialization, (2) forward pass, (3) backtracking. In Needleman-Wunsch algorithm, the score of a path is computed by adding match or gap scores since

```
H D V - S K R S -
| | |   | | | |
P D S A D V T K M
```

Fig. 6. The alignment expected from the high scoring residue pairs for the sub-block of the example pairing probability matrix.

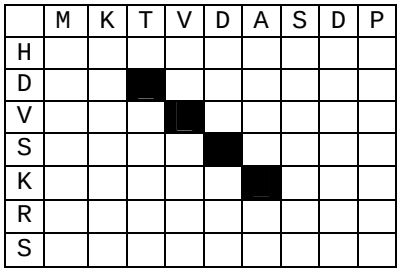


Fig. 7. Identifying mid-scoring residue pairs for a mid-scoring segment pair. Only the residue pairs on the diagonal that have the highest average score are selected.

Algorithm 2: Selecting The Significant β-Strand Segments and Residue Pairs

Input: β-strand segments (segment of amino acids) in **SS** and BetaPro's residue pairing probability matrix PP. Number of segments is n_S.
Output: β-strand segments and residue pairs with strong interaction propensities.

```
for i = 1 : n_S do
    for j = 1 : n_S do
        if i = j then
            continue;
        else
            Extract the (i, j)^th sub-block of PP;
            Count the number of high scoring residue pairs (n_high) in parallel and anti-parallel diagonals of the sub-block;
            if (∃ a diagonal with n_high ≥ 2) then
                Flag (i, j) as a high scoring segment pair;
                Select the high scoring residue pairs;
            else
                Count the number of mid scoring residue pairs (n_mid) in parallel and anti-parallel diagonals of the sub-block;
                if (∃ a diagonal with n_mid ≥ 3) AND (average score > 0.08) then
                    Flag (i, j) as a mid scoring segment pair;
                    Select the mid scoring residue pairs;
                else
                    continue;
Drop segment pairs that are part of a cycle. Start eliminating the segment pairs with the lowest average score;
```

they are essentially log-odds values. For the β-strand alignment problem, we used a similar approach. We first initialized the dynamic programming matrix at position $(0,0)$ to 0. We then set $s(i,j)$ to $\log P(BP = 1 \,|\, R_i, R_j)$, which is the match/mismatch score for aligning the amino acid R_i to R_j (see Section 2.1.3.3). For gap scores, we chose $d(i)$ as $\log P(BP = 0 \,|\, R_i)$ and $d(j)$ as $\log P(BP = 0 \,|\, R_j)$, where $d(i)$ is the gap penalty score for aligning the i^{th} amino acid of the first sequence to a gap symbol, and $d(j)$ is the gap score for aligning the j^{th} amino acid of the second sequence to a gap symbol. The dynamic programming matrix is then computed by adding the match/mismatch and gap scores as formulated in Eqs. 23 to 25.

$$M(i,0) = M(i-1,0) + d(i) \quad 1 \le i \le l_1 \tag{23}$$

$$M(0,j) = M(0,j-1) + d(j) \quad 1 \le j \le l_2 \tag{24}$$

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(i,j) \\ M(i-1,j) + d(i) \\ M(i,j-1) + d(j) \end{cases} \tag{25}$$

After computing the dynamic programming matrix, we start from the cell indexed as (l_1, l_2), and perform

backtracking to find the optimum alignment path. For this purpose, we used the same backtracking algorithm as in the classical implementation of the Needleman-Wunsch algorithm [39], [40], [42]. The alignment score is then converted to a probability value by computing its exponential.

a. *Enforcing High and Mid Scoring Residue Pairs in the Alignment*

As explained in Section 2.1.4.1, the alignment between a pair of β-strand segments is computed using the Needleman-Wunsch algorithm. After identifying high and mid scoring residue pairs, we need to make sure that the optimum alignment path passes through such pairs. This can be achieved by a simple modification of the Needleman-Wunsch algorithm. Let the β-strand segments that will be aligned have l_1 and l_2 amino acids, respectively. Also, let the m^{th} amino acid of the first segment and the n^{th} amino acid of the second segment have a significant residue pairing probability score. In the classical implementation of the Needleman-Wunsch algorithm, first, a dynamic programming matrix, which contains the alignment scores of sub-paths up to a certain residue pair is computed in the forward pass. Since our alignment should pair the m^{th} amino acid of the first segment to the n^{th} amino acid of the second segment, we need to make sure that the alignment path makes a transaction from $(m-1, n-1)$ to (m, n). This can be easily guaranteed by setting the scores of the cells $(m, 0), (m, 1), ..., (m, n-1)$ and $(0, n), (1, n), ..., (m-1, n)$ to 0 as shown in Fig. 8 during the forward pass. When this step is repeated for all residue pairs in the high or mid scoring category, they are guaranteed to appear in the resulting alignment.

b. *The Sum of the Alignment Scores*

In Eq. 16, for each segment pair in a given β-sheet, the sum of the alignment scores of all possible residue pairing combinations has to be computed. This can be performed efficiently using a dynamic programming approach. Let M_{sum} denote a dynamic programming matrix, similar to the M matrix used in the Needleman-Wunsch algorithm. The only difference is that M_{sum} includes the sum of the scores of alignment paths instead of the maximum scores. The initialization of M_{sum} is the same as that of the M matrix. On the other hand, the forward pass equation takes the following form:

$$\begin{aligned} M_{sum}(i,j) &= \log\{e^{M_{sum}(i-1,j-1)+s(i,j)} \\ &+ e^{M_{sum}(i-1,j)+d(i)} \\ &+ e^{M_{sum}(i,j-1)+d(j)}\}, \end{aligned} \tag{26}$$

where e is the exponential. Therefore, at each position, instead of choosing the maximum score, we compute the sum of scores. Then, the sum of the scores of all possible alignments expressed as $\sum_{(\mathbf{C}_k^m \mid \mathbf{O}_k, \mathbf{I}_k)} P(\mathbf{C}_k^m \mid \mathbf{D})$ becomes equal to $exp(M_{sum}(l_1, l_2))$. This can be easily proved using Eq. 17, which is omitted here for simplicity.

	W	Y	L	I	T	E	S
A				0			
K			↘	0			
V	0	0	0				
D							
Q							

Fig. 8. Modification of the dynamic programming matrix during the forward pass of the Needleman-Wunsch algorithm. The segments being aligned are AKVDQ and WYLITES. The amino acid residues V and I are detected as a significant residue pair. To ensure the alignment path matches V to I, the cells shown are assigned to zero. This discards all the paths that do not pair V with I.

2.1.4.2 Computation Times: The BetaPro method has three modular blocks. The first block generates a pairing probability matrix using the amino acid sequence, secondary structure, solvent accessibility and PSSM profiles. The second and third blocks compute the optimum β-sheet conformation by dynamic programming. Computationally, the first block is more intensive as compared to the second and third blocks due to the derivation of PSSM profiles using the PSI-BLAST algorithm. On average, the last two blocks take at most a couple of seconds to execute, whereas the first block's execution time is on the order of minutes.

Our method uses the pairing probability matrix of BetaPro to extract the amino acid pairs that have strong interaction propensities. Therefore, we first execute the first block of BetaPro and then sample possible conformations using efficient algorithms. Since we reduce the space of conformations significantly through the utilization of BetaPro's pairing probability matrix, our computations are significantly reduced. On average our method computes the optimum conformation of a protein with six or less β-strands in 0.31 seconds. For proteins with four β-strands it takes approximately 1 second to compute the optimum conformation. This is the same for proteins with five or six β-strands. Therefore, our method is computationally efficient and the computation time does not rise exponentially with the number of β-strands. Note that, we implemented our method on a Windows XP OS, with an Intel Pentium III Xeon processor, 930 MHz CPU and 640MB RAM. BetaPro and PSI-BLAST on the other hand are implemented on a 32-bit GNU/Linux machine with Intel Pentium IV processor, 3.0 GHz CPU and 2GB RAM.

2.2 β-Sheet Prediction for Proteins with $>$ 6 β-strands

The Bayesian nature of the Ruczinski model requires sufficient amount of training data to reliably estimate probability distributions. As the number of β-strands increase, the number of possible motifs rise exponentially. For proteins with more than four β-strands, Ruczinski model reduces the feature set (or dimensions) by grouping proteins according to their structural properties. In

our simulations we observed that, for proteins with more than six β-strands, the model becomes less specific and therefore its discriminative power reduces (result not shown). For such proteins, instead of utilizing a Bayesian approach, we simply choose the same β-strand pairing predictions as BetaPro. Then, we compute gapped alignments of the paired β-strands both in parallel and anti-parallel directions. Here, for simplicity, we set the gap scores to zero and compute the score of an alignment by taking the sum of the residue pairing probability values derived using BetaPro. Finally, we select the interaction type and the residue pairing patterns with maximum alignment scores.

2.3 Datasets

2.3.1 CulledPDB

The CulledPDB set is compiled from the PDB [43] by the Dunbrack lab [44]. In this paper and in the work by Ruczinski *et al.* [4] the set with sequence identity percentage cut-off 25% and resolution cut-off 2.5Å is used. Since the CulledPDB lists are updated periodically, the datasets grow in time. Therefore the version used by Ruczinski *et al.* is smaller in size (approximately 2000 non-homologous chains) than the one we downloaded in May 2007, which contains 2234 chains. The latest version of this dataset can be obtained from the PISCES server [45].

2.3.2 BetaSheet916

The BetaSheet916 set is extracted from the PDB as of May 2004 by Cheng and Baldi [26]. This dataset contains 916 chains with an HSSP threshold of 0, which corresponds to a sequence identity of 15-20%. The set is splitted randomly and evenly into 10 folds (subsets) to perform cross validation. Details of how the set is compiled can be found in Cheng and Baldi [26] and the set can be downloaded from [46].

2.4 BetaPro and PSI-BLAST

We downloaded and installed the BetaPro method from [46]. BetaPro uses PSI-BLAST version 2.2.8 to generate PSSM profiles. In our simulations, we used the latest versions of the PSI-BLAST (version 2.2.18) and the NR database (as of July 2008), which are obtained from the NCBI's archives [47].

3 Results

3.1 Accuracy Measures

To assess the prediction performance, we used the sensitivity (TP/(TP+FN)) and the positive predictive value (TP/(TP+FP)) as the accuracy measures. We evaluated the predictions in the following categories: β-strand pairing, pairing direction (parallel or anti-parallel), and amino acid residue pairing (contact map). In each category, we computed the sensitivity and positive predictive value measures separately. For instance, the contact map sensitivity is computed as the total number of correctly predicted amino acid pairs divided by the total number of amino acid pairs in the dataset.

3.2 Experimental Settings

For the BetaPro method, we used the greedy graph algorithm to predict the β-sheet topology. Similar to the paper by Cheng and Baldi [26], we used true (native) secondary structure assignments and solvent accessibility measures, which are available in the DSSP database [36]. Hence, the results reported in this work serve as an upper bound on the performance obtained by predicted versions of secondary structure and solvent accessibility.

3.2.1 Model Training

The following distributions were learned from the training data: grouping distribution $P(\mathbf{G} \mid \mathbf{D})$, motif distribution $P(\mathbf{O}_k, \mathbf{I}_k \mid \mathbf{D})$, and contact map distribution $P(\mathbf{C}_k \mid \mathbf{O}_k, \mathbf{I}_k, \mathbf{D})$. The parameters used in modeling the grouping and motif distributions were estimated by Ruczinski [38] from the Culled PDB database, which is a database of non-homologous proteins (see Section 2.3.1). In the Culled PDB release used by Ruczinski, there were 1602 two stranded β-sheets, and 872 four stranded β-sheets (the number of three stranded β-sheets is not provided). Out of 96 possible four stranded motifs, Ruczinski observed only 48 motifs in the database. Among those, 18 motifs were observed only once and less than 20 motifs were observed ten times or more. Ruczinski used 8 bins for two stranded, 96 bins for three stranded and 1536 bins for four stranded β-sheets to estimate the probability distributions of motifs conditioned on the helical status and the connector lengths state between β-strands. Therefore, each bin represents a different configuration (or folding topology) including the motif type, helical status and connector lengths state. Ruczinski also used pseudo-counts and performed bin collapsing when the number of counts in bins were significantly low. This prevents the model to overfit to particular configurations. In the following sections, we provide details on the estimation of the parameters in our model.

3.2.1.1 Grouping Distribution: We used the same parameters as in Ruczinski [38] for $P(\mathbf{G} \mid \mathbf{D})$. We computed the term $\#[crossings(SD, n_{SH}, n_S)]$ in Eq. 9.13 of Ruczinski [38] using the Culled PDB dataset as it was not available in [38].

3.2.1.2 Motif Distribution: We used the estimated values in [4], [38] for proteins with two and three β-strands. For four stranded proteins, only the frequency information of the most common motifs is available in [4], [38]. Here, we used those frequencies as the probability values of the most common motifs and we assigned equal conditional probabilities to the remaining less common motifs. For example, the most frequent motif for a non-helical protein having short

connectors (column L1 of Figure 9.9(b) in [38] or column L1 of Figure 3 in [4]) was $\mathbf{O}$ =(1-2-3-4) and $\mathbf{I} = (AP, AP, AP)$. In our model, the probability of this motif is represented by $P(\mathbf{O}_k, \mathbf{I}_k \,|\, H = 0, L = (SSS))$ (see Eq. 10) and this probability was estimated by Ruczinski as 0.85. To the remaining 95 possible motifs, we assigned equal conditional probability values *i.e.*, $P(\mathbf{O}_k, \mathbf{I}_k \,|\, H = 0, L = (SSS)) = 0.15/95$. For proteins with higher number of β-strands, we estimated the parameters $P(P_p, J \,|\, n, H, L, F)$ and $k_{n,L}(P_p, P_p^s, J, J^s, F)$ of the Ruczinski's model using the CulledPDB dataset as of May 2007 (see Section 2.3.1) as the estimated values were not available in [38]. For the remaining two parameters of the Ruczinski's model (*i.e.*, $P(F \,|\, H, L)$ and $P(P_p^s, J^s \,|\, n, H, L, F, P_p, J)$), we used the estimated values in [38].

3.2.1.3 Contact Map Distribution: Contact map distribution depends on the parameters $P(BP = 1 \,|\, R_m^p, R_{m+1}^p)$, $P(BP = 0 \,|\, R_m^q)$, $P(BP = 0 \,|\, R_{m+1}^r)$ in Eq. 13. In this paper, we estimated those probability distributions from the BetaSheet916 dataset (see Section 2.3.2) for which, the secondary structure assignments are taken from the DSSP database [36]. In the cross validation experiment, we only used the folds that form the training set. To estimate those parameters, we used the maximum-likelihood estimation procedure where we count the observed number of occurrences, and apply a proper normalization factor to compute probability values.

3.3 10 Fold Cross Validation on BetaSheet916

In the first set of simulations, we performed a 10 fold cross validation on the BetaSheet916 set, which contains 916 proteins extracted from the Protein Data Bank (PDB) (see Section 2.3.2 for details). In a cross validation experiment, at each step, a fold is selected as a test data and remaining folds form the training set. Then predictions are computed for proteins in the test set with the models trained on the training set. This process is repeated until all proteins in the original set are tested. Once the predictions are complete, then prediction accuracy is computed.

3.3.1 Performance for Proteins with $\leq$ 4 β-Strands

In this simulation, we performed a 10 fold cross validation experiment on BetaSheet916 and evaluated BetaZa (our method) and BetaPro for proteins with less than or equal to four β-strands. In each fold of the BetaSheet916, we only considered proteins with less than or equal to four β-strands (a total of 67 proteins). Furthermore, since the current version of our model allows up to two β-strand partners for proteins with six or less β-strands, we eliminated proteins that had β-strand segments interacting with more than 2 segments. Among the 67 proteins, there was only one protein with more than 2 segmental partner. Therefore, the total number of proteins tested from all folds becomes 66, which contain a total of 163 β-strand pairs and 1846 β-residue pairs.

Comparing the performances of BetaPro and BetaZa, we obtained the results summarized in Tables 1, and 2, for sensitivity and positive predictive value (PPV), respectively. From these results, we can conclude that BetaZa significantly outperforms BetaPro for proteins with less than or equal to four β-strands.

TABLE 1
Sensitivity measures, evaluated on the BetaSheet916 set. Proteins with four or less β-strands and less than three segmental partners are used as test data.

Sensitivity (%)	BetaPro	BetaZa
Strand Pairing	81.595	90.798
Pairing Direction	79.755	88.344
Contact Map	72.264	82.232

TABLE 2
Positive predictive value measures, evaluated on the BetaSheet916 set. Proteins with four or less β-strands and less than three segmental partners are used as test data.

Positive Predictive Value (%)	BetaPro	BetaZa
Strand Pairing	85.807	90.244
Pairing Direction	83.871	87.805
Contact Map	73.702	81.965

3.3.2 Performance for Proteins with $\leq$ 6 β-Strands

In the next step, we extended our test set to include proteins with six or less β-strands and repeated the 10 fold cross validation experiment performed in Section 3.3.1. There were a total of 187 such proteins in BetaSheet916. Among those, 16 had β-strands with more than 2 segmental partners. Eliminating those, our test data contained 171 proteins from all folds with 586 β-strand pairs and 5838 β-residue pairs.

The sensitivity and positive predictive value measures are shown in Tables 3, and 4. For proteins with six or less β-strands, BetaZa is significantly more accurate than BetaPro. This is also validated by evaluating the accuracy for proteins with a fixed number of β-strands. Tables 5, and 6 show the performance for proteins with five β-strands, whereas Tables 9, and 10 show the performance for proteins with six β-strands. Although the positive predictive value measure of BetaPro is slightly better than BetaZa in segment pairing and interaction type categories, BetaZa performs better in sensitivity measure and especially in the contact map category. BetaPro's higher positive predictive value measure is caused by its tendency to generate less number of predictions instead of generating higher true positives.

TABLE 3
Sensitivity measures, evaluated on the BetaSheet916 set. Proteins with six or less β-strands and less than three segmental partners are used as test data.

Sensitivity (%)	BetaPro	BetaZa
Strand Pairing	79.010	83.27
Pairing Direction	77.133	80.37
Contact Map	71.634	77.66

TABLE 4
Positive predictive value measures, evaluated on the BetaSheet916 set. Proteins with six or less β-strands and less than three segmental partners are used as test data.

Positive Predictive Value (%)	BetaPro	BetaZa
Strand Pairing	83.877	84.28
Pairing Direction	81.884	81.34
Contact Map	73.575	79.54

TABLE 5
Sensitivity measures, evaluated on the BetaSheet916 set. Proteins with five β-strands and less than three segmental partners are used as test data.

Sensitivity (%)	BetaPro	BetaZa
Strand Pairing	80.349	83.843
Pairing Direction	78.603	80.349
Contact Map	74.803	77.865

TABLE 6
Positive predictive value measures, evaluated on the BetaSheet916 set. Proteins with five β-strands and less than three segmental partners are used as test data.

Positive Predictive Value (%)	BetaPro	BetaZa
Strand Pairing	88.462	86.099
Pairing Direction	86.534	82.511
Contact Map	78.947	79.181

TABLE 7
Sensitivity measures, evaluated on the BetaSheet916 set. Proteins with six β-strands and less than three segmental partners are used as test data.

Sensitivity (%)	BetaPro	BetaZa
Strand Pairing	75.258	76.289
Pairing Direction	73.196	73.711
Contact Map	66.706	72.333

TABLE 8
Positive predictive value measures, evaluated on the BetaSheet916 set. Proteins with six β-strands and less than three segmental partners are used as test data.

Positive Predictive Value (%)	BetaPro	BetaZa
Strand Pairing	77.249	76.684
Pairing Direction	75.132	74.093
Contact Map	66.628	77.125

3.3.3 Overall Performance

In this simulation, we evaluated the accuracy on the full set of proteins by performing a 10 fold cross validation experiment on BeteSheet916. This set contains a total of 8172 β-strand pairs and 31638 β-residue pairs. For proteins with six or less β-strands, we computed the predictions as described in Section 2.1, and for proteins that contain more than six β-strands as in Section 2.2. Among proteins with six or less β-strands, we eliminated those with more than two segmental interactions (removing only 16 proteins). For the remaining proteins, we allowed a β-strand to interact with more than two segments because we used BetaPro to compute β-strand pairing predictions. Hence, the overall accuracy of BetaZa is not significantly different from that of BetaPro in the first two categories. However, due to gapped alignments of β-strands, the β-residue pairing accuracy of BetaZa is better than BetaPro by 3% both in sensitivity and positive predictive value measures.

TABLE 9
Sensitivity measures, evaluated on the BetaSheet916 set. Only 16 proteins that had: (1) $\leq$ 6 β-strands and (2) at least one β-strand with more than two segmental interactions are excluded from the test data.

Sensitivity (%)	BetaPro	BetaZa
Strand Pairing	68.903	69.075
Pairing Direction	66.072	66.244
Contact Map	63.411	66.477

TABLE 10
Positive predictive value measures, evaluated on the BetaSheet916 set. Only 16 proteins that had: (1) $\leq$ 6 β-strands and (2) at least one β-strand with more than two segmental interactions are excluded from the test data.

Positive Predictive Value (%)	BetaPro	BetaZa
Strand Pairing	61.921	61.911
Pairing Direction	59.376	59.373
Contact Map	54.373	57.211

3.3.4 Performance of BetaZa for Individual Configurations

The analysis performed by Ruczinski *et al.* [4] shows that a handful of β-sheet configurations are much more frequent than the others. This means that higher probability values will be assigned to such configurations. In that case, it becomes important to verify that our method is capable of generating accurate predictions for less frequent configurations. To understand this, we analyzed the performance on individual proteins. For this purpose, we considered proteins with less than or equal to four β-strands as in Section 3.3.1. There are 66 proteins and 39 distinct configurations in this test data. Here, a configuration is represented by the following features: β-sheet motif (including spatial ordering and

TABLE 11
Performance of BetaZa for individual configurations. Proteins with ≤ 4 β-strands are evaluated. The protein that contains a β-strand with more than two segmental interactions is excluded.

Spatial Ordering	Interaction Types	Helical Status	Connecting Lengths	Motif Probability	Strand Pairing Sensitivity (%)	Strand Pairing PPV (%)	Interaction Type Sensitivity (%)	Interaction Type PPV (%)	Contact Map (%) Sensitivity (%)	Contact Map(%) PPV (%)
1-2	A	H	S	0.9900	100.0	100.0	100.0	100.0	100.0	100.0
1-2|3-4	A|A	NH	SSS	0.9801	100.0	100.0	100.0	100.0	100.0	100.0
1-2|3-4	A|A	H	SSS	0.9801	100.0	100.0	100.0	100.0	94.118	100.0
1-2-3	A-A	H	SS	0.8970	100.0	100.0	100.0	100.0	97.753	95.604
1-2-3	A-A	NH	SS	0.8970	100.0	100.0	100.0	100.0	96.0	96.0
1-2	A	NH	L	0.8700	100.0	100.0	100.0	100.0	100.0	100.0
1-4|2-3	A|A	NH	LSL	0.8613	75.0	60.0	75.0	60.0	53.846	43.750
1-2-3-4	A-A-A	H	SSS	0.8500	100.0	100.0	100.0	100.0	87.500	87.500
1-4|2-3	A|A	H	LSL	0.7227	50.0	33.333	50.0	33.333	42.857	33.333
1-2|3-4	A|A	H	LSS	0.7227	100.0	100.0	100.0	100.0	100.0	100.0
1-3-2	A-A	H	LS	0.5472	100.0	100.0	100.0	100.0	96.429	93.103
1-2-4-3	A-A-A	H	SLS	0.5100	83.333	90.909	83.333	90.909	76.923	75.757
2-3-1-4	A-A-A	H	LSL	0.3800	100.0	100.0	100.0	100.0	76.667	76.033
2-1-3-4	P-P-P	H	LLL	0.3600	100.0	100.0	100.0	100.0	100.0	100.0
2-1-3-4	P-A-A	H	LLS	0.2800	66.667	100.0	66.667	100.0	66.667	100.0
1-2	P	H	L	0.2700	100.0	50.0	100.0	50.0	100.0	85.714
1-2|3-4	P|A	H	LSS	0.2673	100.0	100.0	100.0	100.0	100.0	100.0
1-2-3	A-P	H	SL	0.2622	100.0	100.0	100.0	100.0	92.307	92.307
2-1-3	A-A	H	SL	0.2587	50.0	50.0	50.0	50.0	46.154	54.545
1-4-2-3	A-A-A	H	LSL	0.2400	66.667	66.667	66.667	66.667	60.0	64.286
1-4-3-2	A-A-A	H	LSS	0.1800	100.0	100.0	100.0	100.0	72.500	70.732
2-3-1-4	A-A-A	NH	LSL	0.1800	100.0	100.0	100.0	100.0	84.0	75.0
2-1-3	A-A	H	LL	0.1525	100.0	100.0	100.0	100.0	83.333	83.333
1-2-4-3	A-A-A	H	LLS	0.1200	100.0	100.0	66.667	66.667	77.143	75.0
1-2-4-3	P-A-A	H	LLS	0.1000	100.0	100.0	100.0	100.0	82.692	81.132
2-3-1-4	A-A-A	H	LLL	0.0800	100.0	100.0	100.0	100.0	61.905	65.0
2-1-4-3	A-P-A	H	SLS	0.0800	100.0	100.0	66.667	66.667	78.378	76.316
1-2-3	**P-P**	**H**	**LL**	**0.0491**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
2-1-3	A-P	NH	SS	0.0279	50.0	50.0	50.0	50.0	87.500	87.500
1-2-3-4	**A-A-A**	**H**	**LSS**	**0.0090**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
2-1-4-3	A-A-A	NH	SLL	0.0049	66.667	66.667	66.667	66.667	70.588	80.0
1-4-3-2	A-A-P	H	LLL	0.0042	66.667	100.0	66.667	100.0	41.667	62.500
1-4-2-3	A-A-P	H	LLL	0.0042	66.667	66.667	66.667	66.667	66.667	72.727
1-2-3-4	**A-A-A**	**H**	**LLL**	**0.0042**	**100.0**	**100.0**	**100.0**	**100.0**	**92.307**	**88.889**
3-2-1-4	P-A-A	H	LLL	0.0042	66.667	66.667	66.667	66.667	76.923	71.429
1-3-4-2	A-A-P	H	LLS	0.0036	66.667	66.667	66.667	66.667	64.286	64.286
1-2-4-3	**A-A-P**	**H**	**SSL**	**0.0025**	**100.0**	**100.0**	**100.0**	**100.0**	**86.667**	**92.857**
1-4-3-2	A-A-A	NH	SSS	0.0015	66.667	66.667	66.667	66.667	91.667	91.667
1-3-4-2	**P-A-A**	**NH**	**SSS**	**0.0015**	**100.0**	**100.0**	**100.0**	**100.0**	**80.0**	**80.0**

interaction types), helical status of the protein, and the length states of the segments that connect β-strands. We defined a configuration as less frequent when the motif probability assigned by the model is less than 0.05. The motif frequencies can be found in Ruczinski *et al.* [4].

Table 11 shows the sensitivity and positive predictive value of individual β-sheet configurations. In each row, the features that characterize the configuration as well as the motif probabilities conditioned on the helical status and connecting lengths states are listed. In this table, the symbol "|" is used to separate β-sheets in the spatial ordering representation. For instance, 1-2|3-4 means that the first and the second β-strands form the first β-sheet; the third and the fourth β-strands form the second β-sheet; and there is no interaction between the second and the third segments. Alternatively, 1-2-3-4 shows that all four β-strands form a single β-sheet. The helical status and the connecting length states are defined in Section 2.1.3.2. Here NH stands for non-helical and H for helical protein. Similarly, S denotes a short connector and L represents a long connector. A connector is a set of helix and/or loop segments that are in between β-strand pairs adjacent in sequence representation. From this table, we can observe that although the prediction accuracy of less frequent configurations is in general lower than the frequent ones, our method was able to generate highly accurate predictions for five configurations that have significantly low probability scores (marked in boldface). This clearly demonstrates that our method is able to predict less frequent motifs with high accuracy and the increase in the performance is not simply because of an affinity towards for more frequent motifs or an imbalance of the training data.

4 Conclusion

In this paper, we have shown that elaborate mathematical models combined with efficient algorithms bring significant improvements to β-sheet prediction. The performance of predictions can be improved even further. First of all, sophisticated methods for the estimation of residue pairing propensities will definitely improve the accuracy and quality of the predictions. For this purpose, one can incorporate additional informative features such as HMM profiles, contact potentials, residue types, segment window information, and protein-level information [31]. In a second avenue, one can develop more elaborate models for an enhanced scoring of β-strand organizations. We introduced a Bayesian model for proteins with six or less β-strands and allowed each β-strand to interact with at most two other segments. Extension of the model to characterize higher order segmental interactions can easily be achieved by estimating their probabilities and sampling them in the search space. For proteins with more than six β-strands, it is possible to incorporate a richer set of folding rules as in [35]. Finally, as new proteins are added to the structure database it will be possible to extend the motif distribution to model longer proteins with many β-strands and extend the coverage of the Bayesian model. Advances in β-sheet

prediction will contribute substantially to the accurate prediction of the three dimensional structure.

ACKNOWLEDGMENTS

This project has been supported by grant CCF-0514903 from the NSF-SPS.

REFERENCES

[1] E. Koh, T. Kim, and H. S. Cho, "Mean curvature as a major determinant of beta-sheet propensity," *Bioinformatics.*, vol. 22, pp. 297–302, 2006.
[2] C. Zhang and S. Kim, "The anatomy of protein beta-sheet topology," *J. Mol. Biol.*, vol. 299, pp. 1075–1089, 2000.
[3] S. M. Zaremba and L. M. Gregoret, "Context-dependence of amino acid residue pairing in antiparallel β-sheets," *J. Mol. Biol.*, vol. 291, pp. 463–479, 1999.
[4] I. Ruczinski, C. Kooperberg, R. Bonneau, and D. Baker, "Distributions of beta sheets in proteins with application to structure prediction," *Proteins: Structure, Function and Genetics*, vol. 48, pp. 85–97, 2002.
[5] J. S. Merkel and L. Regan, "Modulating protein folding rates in vivo and in vitro by side-chain interactions between the parallel beta strands of green fluorescent protein," *J. Biol. Chem.*, vol. 275, pp. 29 200–29 206, 2000.
[6] Y. Mandel-Gutfreund, S. M. Zaremba, and L. M. Gregoret, "Contributions of residue pairing to beta-sheet formation: conservation and covariation of amino acid residue pairs on antiparallel beta-strands," *J. Mol. Biol.*, vol. 305, pp. 1145–1159, 2001.
[7] T. Kortemme, M. Ramirez-Alvarado, and L. Serrano, "Design of a 20-amino acid, three-stranded β-sheet protein," *Science*, vol. 281, pp. 253–256, 1998.
[8] B. Kuhlman, G. Dantas, G. Ireton, G. Varani, B. Stoddard, and D. Baker, "Design of a novel globular protein fold with atomic-level accuracy," *Science*, vol. 302, pp. 1364–1368, 2003.
[9] S. Lifson and C. Sander, "Specific recognition in the tertiary structure of beta-sheets of proteins," *J. Mol. Biol.*, vol. 139, pp. 627–639, 1980.
[10] D. L. Minor and S. Kim, "Context is a major determinant of beta-sheet propensity," *Nature*, vol. 371, pp. 264–267, 1994.
[11] M. A. Wouters and P. M. G. Curmi, "An analysis of side chain interactions and pair correlations within antiparallel beta-sheets: the differences between backbone hydrogen bonded and non-hydrogen bonded residue pairs," *Proteins Struct. Func. Genet.*, vol. 22, pp. 119–131, 1995.
[12] H. Zhu and W. Braun, "Sequence specificity, statistical potentials, and three-dimensional structure prediction with selfcorrecting," *Protein Sci.*, vol. 8, pp. 326–342, 1999.
[13] D. N. Woolfson, P. A. Evans, E. G. Hutchinson, and J. M. Thornton, "On the conformation of proteins: The handedness of the connection between parallel β-strands," *J. Mol. Biol.*, vol. 110, pp. 269–283, 1977.
[14] C. K. Smith and L. Regan, "Guidelines for protein design: The energetics of β sheet side chain interactions," *Science*, vol. 270, pp. 980–982, 1995.
[15] E. G. Hutchinson, R. B. Sessions, J. M. Thornton, and D. N. Woolfson, "Determinants of strand register in antiparallel beta-sheets of proteins," *Protein Sci.*, vol. 7, pp. 287–300, 1998.
[16] T. J. Hubbard, "Use of β-strand interaction pseudo-potentials in protein structure prediction and modelling," in *Proceedings of the Biotechnology Computing Track Protein Structure Prediction MiniTrack of the 27th HICSS*, R. H. Lathrop, Ed. New York: IEEE Computer Society Press, 1994, pp. 336–354.
[17] T. J. Hubbard and J. Park, "Fold recognition and ab initio structure predictions using hidden markov models and β-strand pair potentials," *Proteins: Struct. Func. Genet.*, vol. 23, pp. 398–402, 1995.
[18] M. Asogawa, "Beta-sheet prediction using inter-strand residue pairs and refinement with hopfield neural network," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 5, 1997, pp. 48–51.
[19] B. Rost, J. Liu, D. Przybylski, R. Nair, K. Wrzeszczynski, H. Bigelow, and Y. Ofran, "Prediction of protein structure through evolution," in *Handbook of ChemoinformaticsFrom Data to Knowledge.*, J. Gasteiger and T. Engel, Eds. New York: Wiley, 2003, pp. 1789–1811.
[20] R. E. Steward and J. M. Thornton, "Prediction of strand pairing in antiparallel and parallel beta-sheets using information theory," *Proteins Struct. Funct. Genet.*, vol. 48, pp. 178–191, 2002.
[21] K. Karplus, C. Barrett, and R. Hughey, "Hidden Markov models for detecting remote protein homologies," *Bioinformatics*, vol. I4, pp. 846–856, 1998.
[22] P. Baldi, G. Pollastri, C. A. F. Andersen, and S. Brunak, "Matching protein β-sheet partners by feedforward and recurrent neural networs," in *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, 2000, pp. 25–36.
[23] G. Pollastri and P. Baldi, "Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners," *Bioinformatics*, vol. I8 (Suppl. I), pp. S62–S70, 2002.
[24] N. Hamilton, K. Burrage, M. Ragan, and T. Huber, "Protein contact prediction using patterns of correlation," *Proteins*, vol. 56, pp. 679–684, 2004.
[25] R. MacCallum, "Striped sheets and protein contact prediction," *Bioinformatics*, vol. 20 (Supplement 1), pp. i224–i231, 2004.
[26] J. Cheng and P. Baldi, "Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms," *Bioinformatics*, vol. 21 (Suppl. 1), pp. i75–i84, 2005.
[27] J. Cheng, A. Randall, M. Sweredoski, and P. Baldi, "SCRATCH: A protein structure and structural feature prediction server," *Nucleic Acids Res.*, vol. 33, pp. w72–w76, 2005.
[28] M. Punta and B. Rost, "PROFcon: novel prediction of long-range contacts," *Bioinformatics*, vol. 21, pp. 2960–2968, 2005.
[29] A. Vullo, I. Walsh, and G. Pollastri, "A two-stage approach for improved prediction of residue contact maps," *BMC Bioinformatics*, vol. 7, no. 180, 2006.
[30] D. Bau, A. Martin, C. Mooney, A. Vullo, I. Walsh, and G. Pollastri, "Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins," *BMC Bioinformatics*, vol. 7, no. 402, 2006.
[31] J. Cheng and P. Baldi, "Improved residue contact prediction usign support vector machines and a large feature set," *BMC Bioinformatics*, vol. 8, no. 113, 2007.
[32] P. Bradley, L. Cowen, M. Menke, J. King, and B. Berger, "BETAWRAP: Successful prediction of parallel β-helices from primary sequence reveals an association with many pathogens," *PNAS*, vol. 98, pp. 14 819–14 824, 2001.
[33] J. Waldispuhl, B. Berger, P. Clote, and J. M. Steyaert, "Predicting transmembrane β-barrels and interstrand residue interactions from sequence," *PROTEINS: Structure, Function, and Bioinformatics*, vol. 65, pp. 61–74, 2006.
[34] A. Randall, J. Cheng, M. Sweredoski, and P. Baldi, "TMBpro: secondary structure, β-contact and tertiary structure prediction of transmembrane β-barrel proteins," *Bioinformatics*, vol. 24, pp. 513–520, 2008.
[35] J. Jeong, P. Berman, and T. Przytycka, "Bringing folding pathways into strand pairing prediction," in *The Workshop on Algorithms in Bioinformatics WABI*, vol. 4645, 2007, pp. 38–49.
[36] "FTP access to the DSSP files at the CMBI," ftp://ftp.cmbi.kun.nl/pub/molbio/data/dssp.
[37] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features," *Biopolymers*, vol. 22, pp. 2577–2637, 1983.
[38] I. Ruczinski, "Logic regression and statistical issues related to the protein folding problem," Ph.D. dissertation, Department of Statistics, University of Washington, Seattle, WA, 2000. [Online]. Available: http://biostat.jhsph.edu/~iruczins/sheets/scoring.pdf
[39] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.
[40] O. Gotoh, "An improved algorithm for matching biological sequences," *J. Mol. Biol.*, vol. 264, pp. 823–838, 1982.
[41] L. Parker, "CS302 lecture notes: Topological sort/cycle detection," http://www.cs.utk.edu/~parker/Courses/CS302-fall03/Notes/GraphIntro/.
[42] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.
[43] "The protein data bank," http://www.rcsb.org/pdb.
[44] "Dunbrack Lab," http://dunbrack.fcc.edu.
[45] "Pre-compiled CulledPDB lists from PISCES," http://dunbrack.fcc.edu/Guoli/pisces_download.php#culledpdb.

[46] "BetaSheet916 set," http://www.ics.uci.edu/∼baldig/betasheet_data.html.
[47] "NCBI BLAST Downloads," http://www.ncbi.nlm.nih.gov/BLAST/download.shtml.

Zafer Aydin (S'06) received the B.S. and M.S. degrees in electrical engineering from Bilkent University, Ankara, Turkey, in 1999, and 2001, respectively. He received the Ph.D. degree in electrical engineering from Georgia Institute of Technology, Atlanta, GA USA in 2008. His research interests include bioinformatics, computational biology, pattern recognition, and machine learning.

Yucel Altunbasak (SM'02) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY. He joined Hewlett-Packard Research Laboratories in July 1996. Meanwhile, he taught at Stanford and San Jose State Universities as a consulting Assistant Professor.

Dr. Altunbasak served as the technical program chair for ICIP-2006. He was an associate editor for IEEE Transactions on Image Processing, IEEE Transactions on Signal Processing, Signal Processing: Image Communication, and for the Journal of Circuits, Systems and Signal Processing. He served as the lead guest editor on two Signal Processing: Image Communication special issues on wireless video and video networking, respectively. He also served as a guest editor for the new IEEE Journal on Selected Topics on Signal Processing for the special issue Network-aware multimedia processing and communications". He served as the vice-president for the IEEE Communications Society MMC Technical Committee. He has been elected to the IEEE Signal Processing Society IMDSP, MMSP, and BISP Technical Committees. He has served as a co-chair for "Advanced Signal Processing for Communications" Symposia at ICC'03. He also served as a track chair at ICME'03 and ICME'04, as a panel sessions chair at ITRE'03, and as a session chair at various international conferences. He is a co-author for the article that received the most cited paper award of the EURASIP journal of Signal Processing: Image Communication in 2008. He is also a co-author for two conference papers that received the best student paper awards at ICIP'03 and VCIP'06. He also co-authored a conference paper that has been selected as design finalist at EMBS'2004. He received the National Science Foundation (NSF) CAREER Award in 2002. He received the 2003 Outstanding Junior Faculty Member Award from the School of Electrical and Computer Engineering.

Hakan Erdogan (M'92) received his B.S. degree in Electrical Engineering and Mathematics in 1993 from METU, Ankara and his M.S. and Ph.D. degrees in Electrical Engineering: Systems from the University of Michigan, Ann Arbor in 1995 and 1999, respectively. He was with the Human Language Technologies group at IBM T.J. Watson Research Center, NY between 1999 and 2002. His research interests are in developing and applying probabilistic methods and algorithms for multimedia information extraction and bioinformatics.