Computational Biology and Chemistry 70 (2017) 142–155

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Research Article

Protein β-sheet prediction using an efficient dynamic programming algorithm

Mostafa Sabzekar[a], Mahmoud Naghibzadeh[a,*], Mahdie Eghdami[a], Zafer Aydin[b]

[a] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Department of Computer Engineering, Abdullah Gul University, Kayseri, Turkey

ARTICLE INFO

Article history:
Received 5 March 2017
Received in revised form 25 July 2017
Accepted 18 August 2017
Available online 24 August 2017

Keywords:
β-sheet structure prediction
Dynamic programming
Repetitive calculation
Sheet-tree
Grouping-tree

ABSTRACT

Predicting the β-sheet structure of a protein is one of the most important intermediate steps towards the identification of its tertiary structure. However, it is regarded as the primary bottleneck due to the presence of non-local interactions between several discontinuous regions in β-sheets. To achieve reliable long-range interactions, a promising approach is to enumerate and rank all β-sheet conformations for a given protein and find the one with the highest score. The problem with this solution is that the search space of the problem grows exponentially with respect to the number of β-strands. Additionally, brute-force calculation in this conformational space leads to dealing with a combinatorial explosion problem with intractable computational complexity. The main contribution of this paper is to generate and search the space of the problem efficiently to reduce the time complexity of the problem. To achieve this, two tree structures, called *sheet-tree* and *grouping-tree*, are proposed. They model the search space by breaking it into sub-problems. Then, an advanced dynamic programming is proposed that stores the intermediate results, avoids repetitive calculation by repeatedly uses them efficiently in successive steps and reduces the space of the problem by removing those intermediate results that will no longer be required in later steps. As a consequence, the following contributions have been made. Firstly, more accurate β-sheet structures are found by searching all possible conformations, and secondly, the time complexity of the problem is reduced by searching the space of the problem efficiently which makes the proposed method applicable to predict β-sheet structures with high number of β-strands. Experimental results on the BetaSheet916 dataset showed significant improvements of the proposed method in both execution time and the prediction accuracy in comparison with the state-of-the-art β-sheet structure prediction methods Moreover, we investigate the effect of different contact map predictors on the performance of the proposed method using BetaSheet1452 dataset. The source code is available at http://www.conceptsgate.com/BetaTop.rar.

1. Introduction

Proteins are large and complex molecules and vital parts of living organisms. They are responsible for almost every task of cellular life. There is a close association between the protein structure and its function. Therefore, to understand the biological function of a protein, we should have knowledge about its three-dimensional (3D) structure. However, one of the challenging open problems in structural bioinformatics is protein structure prediction (PSP). Solving this problem would have a great positive effect on the fields of medicine (e.g. drug design), biotechnology (e.g. design of new enzymes) and would give better insight into protein-protein interactions. The main motivations for predicting the protein three-dimensional structure from its amino acid sequence can be highlighted as follows:

- The gap between the number of known protein sequences and known protein structures: 3D structure is unknown for a large number of proteins. As of November 2016, there are approximately 125,000 protein structures in the Protein Data Bank (PDB) (Berman et al., 2000) when compared to more than 73 million protein sequences in the RefSeq database (Pruitt et al., 2009).

* Corresponding author.
E-mail address: naghibzadeh@um.ac.ir (M. Naghibzadeh).

http://dx.doi.org/10.1016/j.compbiolchem.2017.08.011
1476-9271/

Hence, just a small fraction of all proteins (i.e., less than one percent) have known three-dimensional structure published in the PDB.
- The shortcomings of the experimental methods: X-ray crystallography and multi-dimensional magnetic resonance are two primary experimental methods for protein structure prediction. However, these methods are relatively expensive and time-consuming.
- Anfinsen's *dogma* (Anfinsen, 1973): suggests that protein tertiary structure information is completely determined by its primary structure.

Unfortunately, the prediction of tertiary structure of a protein from its sequence suffers from: 1) low accuracy and 2) exponential increase of the conformational space of the problem with the length of the primary sequence. One of the current promising and successful ways to reduce the complexity of this problem is to utilize auxiliary predictions such as secondary structures, contact maps or local structure predictions, which subsequently help in predicting the protein tertiary structure.

β-sheets are predominant structures found in proteins. More than 80% of all protein domains in the PDB contain β-sheets (Savojardo et al., 2013). They are formed by a set of β-strand segments which are held together by hydrogen bonds and are run in parallel or antiparallel direction on a continuous surface.

One of the most important and challenging intermediate steps towards the prediction of protein tertiary structure is β-sheet prediction which identifies how the β-strands are assembled into β-sheets. In a β-sheet, adjacent β-strands bring distant residues into close contact with one another, and constitute a specific mode of amino acid pairing. This mode of interactions occur widely in protein tertiary structures and play an important role in many human diseases (Zhang et al., 2010). The association of β-sheets has been implicated in formation of protein aggregates (Sgourakis et al., 2011) and fibrils observed in many diseases such as AIDS, cancer, mad cow, anthrax, Alzheimer's, etc. A list of such diseases can be found in Chiti and Dobson (2006) and continues to grow. For example, recently, Buhimschi et al. found a strong evidence of protein aggregation as a possible cause of preeclamsia, a pregnancy-specific disorder (Kuhlman et al., 2003; Kouza et al., 2017).

Any improvement in β-sheet prediction is important both as a stand-alone result and in relation to protein structure prediction (Sabzekar et al., 2017). It is also essential for reducing the search space of PSP methods (Steward and Thornton, 2002; Cheng and Baldi, 2005), designing new proteins (Kuhlman et al., 2003; Kortemme et al., 1998) and elucidating folding pathways (Mandel-Gutfreund et al., 2001; Merkel and Regan, 2000). To predict the structure of a protein β-sheet, we should predict the conformational arrangement of β-strands including:

1. Assignment of β-strands into β-sheets
2. Spatial ordering of β-strands in each sheet
3. The interaction types between each pair of β-strands (i.e., parallel or antiparallel)
4. Residue-residue contacts (i.e., contact maps).

For this purpose, we need to search the space of possible β-sheet conformations and find the optimum one based on a scoring function.

Although several methods have been proposed for predicting β-sheet structures, it is still an open challenge. There are two major problems with these methods. First, the protein β-sheet structures typically involve long-range interactions between several discontinuous regions. Unlike α-helices that are locally stabilized, β-sheets result from pairwise hydrogen bonding of two or more disjoint regions of the protein backbone. In addition, the patterns of amino acids in β-sheet formation are unpredictable which makes it difficult to predict specific contacts between strands (Jeong et al., 2008). This causes the β-residue contacts not to follow any clear and predefined rules. Consequently, it is a critical task to define the functions that accurately score the β-sheet conformations. Second, in proteins with higher number of β-strands the β-sheet formation search space is enormously big. Furthermore, as it will be shown in Section 2, there is an extra overhead in construction of repetitive pairings of β-strands and therefore in computation of their scores. Hence, finding the optimal conformation in this search space requires avoiding a combinatorial explosion problem with an intractable computational complexity. These challenges make the prediction of β-sheets more difficult than predicting helices and coils. Moreover, it is concluded that the protein β-sheet prediction is the primary bottleneck towards the three dimensional structure predictions, as evidenced through all CASP blind predictions (Aydin et al., 2011).

From the above, since the β-sheet structure of a protein cannot be predicted accurately using the available scoring functions (the first challenge), a reliable way of tackling this problem is to enumerate and rank all β-sheet conformations for a given protein and find the one with a highest score (Fonseca et al., 2011). But, as discussed earlier, the problem is that the conformational space grows exponentially (the second challenge) as the number of β-strands increase. It causes this approach to be applicable only to proteins with small number of β-strands (i.e., usually up to six (Aydin et al., 2011; Fonseca et al., 2011)). However, in real datasets, the majority of protein chains have more than six β-strands. For example, for BetaSheet916 (BetaSheet916, 2009) and CulledPDB (Pre-Compiled, 2008) datasets the percentage of proteins with more than six β-strands is calculated as 79.58 percent and 74.49 percent, respectively.

In this paper, we will predict β-sheet structure of a protein with more accuracy and less execution time. Our goal is to understand and provide better insight into the arrangement of β-strands in protein structure similar to many previous researches, such as Savojardo et al. (2013); Cheng and Baldi (2005); Aydin et al. (2011); Fonseca et al. (2011); Subramani and Floudas (2012); Eghdami et al. (2015); Ruczinski et al. (2002a); Carbonell (2003); Tsutsumi and Otaki (2011). The main contribution of this paper is to search the conformational space efficiently with the goal of reducing the time complexity of the problem. In this way, two tree structures called *sheet-tree* and *grouping-tree* are introduced to model the search space. In fact, they break the problem into sub-problems, and then a dynamic programming approach is proposed to find the optimal conformation. It helps us to store intermediate results and avoid brute-force calculations by reusing them. Furthermore, the construction of these trees performed in such a way that we can remove many nodes which will not be reused at higher levels and consequently the space requirements of the problem will be reduced. Therefore, the proposed method, called BetaTop, will be applicable to predict β-sheet structure of proteins with higher number of β-strands. Thus the achieved results would be utilized in the fields of structural biology of proteins, functional proteomics and Bioinformatics to improve the accuracy of protein structure prediction. Moreover, since BetaTop has implemented efficiently in terms of time complexity, it can overcome the combinatorial explosion challenge in predicting the β-sheet and consequently tertiary structure of a protein.

The paper is organized as follows. Section 2 surveys related work about β-sheet prediction and describes the contributions in this field and their limitations. The proposed algorithm is presented in Section 3. Experimental results are reported in Section 4. Finally, Section 5 concludes the paper with a final discussion and suggests future research directions.

2. Literature review

Predicting the architecture of protein β-sheets from its amino acid sequence is still a challenging problem and an active research topic in bioinformatics. As discussed in the previous section, the β-sheet prediction algorithms suffer from low accuracy due to the presence of non-local inter-strand residue interactions. In the following, we survey and classify the most important studies in this regard and discuss the advantages and disadvantages of each approach. Although there are many efforts in the literature aimed at understanding and predicting topological features of β-sheets, we focus on those studies that predict all components of the conformational arrangement of β-strands, including: the assignment of β-strands into β-sheets, the spatial ordering of β-strands in each sheet, the interaction types of β-strand pairs (parallel or antiparallel), and amino acid residue interactions, also known as contact maps.

Cheng and Baldi (2005) established a standard for predicting protein β-sheets using a three-stage method called BetaPro, as shown in Fig. 1. This idea has been followed by recent researches.

In the first stage, inter-strands residue pairing is predicted using a contact map prediction algorithm. The provided information is then used to find the optimal pairwise alignments of β-strands in the next stage. The alignment score indicates a tendency for two strands to interact in a sheet. Finally, the β-sheet prediction problem can be modeled and then an optimization problem is defined to find the optimal conformation. There are three major approaches in the literature for modeling and solving such a problem in this stage: 1) graph-based approach; 2) Integer Linear Programming (ILP)-based approach; and 3) state-space search-based approach.

In the first category, a weighted complete graph is defined to model the problem. The vertices of the graph represent strands while the edges represent possible pairing relations. Moreover, the weight of each edge is the score of the optimal alignment between the two strands. To predict the β-sheet structure, the complete graph is pruned to derive a spanning sub-graph. The main advantages of this approach are that, firstly, it is computationally faster than other approaches due to its greedy nature, and secondly, it utilizes the well-understood graph algorithms such as graph matching. However, the major problem of this approach comes from the low accuracy due to the greedy nature of the final stage.

The second approach models the problem as an ILP optimization problem and finds the optimal solution according to the pairwise alignment scores of β-strands and appropriate constraints. The objective function maximizes the overall β-sheet sum of scores. The biological strand pairing limitations are modeled as the integer linear constraints. BeST (Subramani and Floudas, 2012), BCov (Savojardo et al., 2013) and BetaProbe (Eghdami et al., 2015) are examples of such efforts. Although this approach benefits from a mathematical foundation, it suffers from some drawbacks. First, there is no guarantee of optimality. Second, it is difficult to formulate the structural features of β-sheets and, finally, considering more strand pairing constraints makes solving the problem more complicated and time consuming.

Since the β-sheet structure of a protein cannot be predicted accurately due to the limitations of β-sheet formation (as discussed in Section 1), the third approach generates the entire search space of the possible β-sheet conformations, ranks all of them for a given protein and, finally, finds the one with a highest score as the true conformation. Aydin et al. (2011) and Fonseca et al. (2011) in two different studies utilized this approach. The number of possible arrangements (different ordering and directions) of m β-strands in a β-sheet is $m! \times 2^{m-2}$. Consequently, for a protein with n β-strands, the state space of the dynamic programming algorithm to find all possible β-sheets' structures is calculated as follows:

$$\sum_{m=2}^{n}\left(\binom{n}{m} \times m! \times 2^{m-2}\right). \quad (1)$$

Eq. (1) implies that increasing the number of β-strands for a protein lead to a large search space. However, the aim is to combine the β-sheets and form different conformations and then find the optimal one. Hence, the total number of conformations that are enumerated with this brute-force technique will be roughly equal to the number of subsets of possible β-sheets in Eq. (1). Finding the optimal conformation in this space leads to an intractable computational complexity. There are two ways to tackle this problem:

- Reduction the search space using some heuristics. For example, in Aydin et al. (2011), the search space was reduced by heuristics that enforce the residue pairs with strong interaction potentials derived from BetaPro. Their method, BetaZa, uses threshold parameters that are estimated experimentally and divide the residue pairs into high-score and mid-score categories. Then, they eliminate the segment pairs with the low scores by an algorithm. However, the problem is that when we have no accurate and reliable scoring functions, it is extremely critical to choose the thresholds and eliminate some β-strand pairs. Although BetaZa uses pruning to reduce the space of the problem, it still enumerates all the assignments and also the arrangements of β-strands into β-sheets. Additionally, the Bayesian model of BetaZa for proteins with more than six strands becomes less specific, and therefore, its discriminative power reduces. Thus, it is applied to proteins with only up to six β-strands.
- Construction of a sub-structure from the true conformation. For example, in Fonseca et al. (2011), for proteins with more than six strands, a subset of six strands is chosen, and the 23,800 corresponding conformations are added to the set. This process is repeated for all subsets of six strands. Thus, the total number of conformations for a protein with more than six strands is equal to:

$$\binom{n}{6} \times 23800. \quad (2)$$

Hence, the predicted conformation is a subset of the β-topologies and also it contains fewer β-strands than the true conformation. Although at least one conformation can be found that will be very similar to the true conformation, it significantly affects the accuracy of the method.

Another problem with this approach is that there is an overhead in construction of repetitive pairings of β-strands and, consequently, in recalculation of their scores. For more clarity, let us give an example. Suppose that we have a protein with seven β-strands which have been numbered according to the order they appear in the chain. Let B denote a β-sheet formed by three β-strands segments which are ordered as (1-2-3) interacting in parallel directions. Fig. 2 shows some of different possible conformations that contain the β-sheet B.

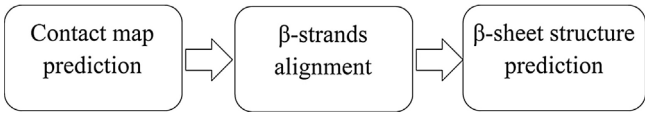

Fig. 1. Three-stage prediction of protein β-sheets.

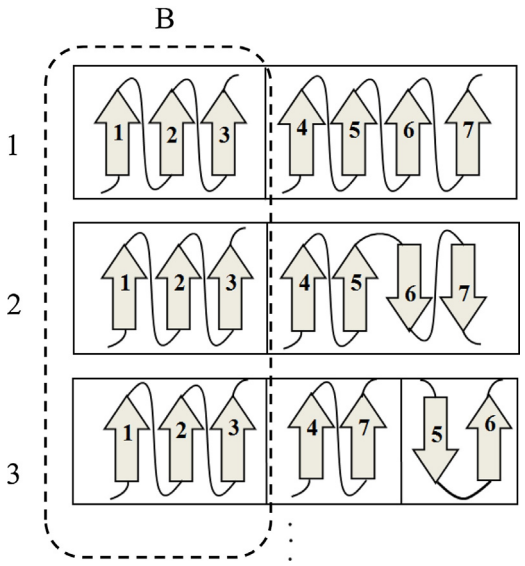

Fig. 2. The space of possible conformations for a protein with seven strands that contain β-sheet B formed by β-strand segments are ordered as (1-2-3) interacting in parallel direction.

The space of possible conformations containing B is obtained when the remaining four strands form one or two β-sheets. Therefore, based on the following calculations, the β-sheet B appears in 108 separate conformations:

$$\left[\binom{4}{2} \times 2! \times 2^0 \times \binom{2}{2} \times 2! \times 2^0\right]/2 + \left[\binom{4}{4} \times 4! \times 2^2\right] = 108.$$

Thus, there is an extra overhead in reconstruction and, as a result, in recalculation of its scores that leads to high time complexity.

It is important to point out that the majority of studies assume the secondary structure of proteins to be known as an inputs of the problem. However, there are several researches on predicting secondary structure of proteins, which their results can be utilized by β-sheet structure prediction methods. But the problem is that they have some inaccuracies, which affect the accuracy of β-sheet structure predictors. For example, the authors in Fonseca et al. (2011) used PSIPRED (Jones, 1999) for this purpose. However, there are more accurate secondary structure predictors were developed which can be applied to the problem. Lin et al. (2005) proposed a new approach that utilizes a single neural network. Their method, namely YASPIN, predicts the secondary structure elements in a 7-state local structure scheme and then optimizes the output using a hidden Markov model, which results in providing more information for the prediction. Magnan and Baldi (2014), besides the upgrade of existing predictors, study in detail the effectiveness of sequence-based structural similarity for secondary structure and relative solvent accessibility prediction alone, and how it can be combined with predictions derived by machine learning methods with profiles to improve the overall state of-the-art. The authors in Wang et al. (2016) proposed DeepCNF, which is a Deep Learning extension of Conditional Neural Fields (CNF). It can model not only complex sequence-structure relationship by a deep hierarchical architecture, but also interdependency between adjacent secondary structure labels, so it is much more powerful than CNF.

Before finishing this section, different β-topology scoring methods needs to be mentioned. There are two methods in the literature (Fonseca et al., 2011): the *topology scoring method* (Ruczinski, 2002; Ruczinski et al., 2002b) and the *pair scoring method* (Sgourakis et al., 2011). The first method assigns a probability to each β-sheet topology based on several topological features. It performed a statistical analysis of the frequency of β-strand groupings and β-sheet motifs (Aydin et al., 2011). The second method utilizes the probabilities of pairing two amino acids which are obtained by the contact map prediction methods. The pseudo-energy of pairing two strands is directly calculated by them. Then, a score is assigned to a β-topology by taking the average or sum of pseudo-energies of all its β-pairs. Fonseca et al. in (Fonseca et al., 2011) compared these methods and concluded that the pair scoring method outperforms the topology scoring method.

3. The proposed method

To overcome the drawbacks of the state-space search-based methods, we generate and search the conformational space efficiently to reduce the time complexity and control the space complexity of the problem, simultaneously. In other words, the proposed method, named BetaTop, will be able to:

1. search the entire search space
2. break the problem into sub-problems and reuse the intermediate results to avoid repetitive calculations and therefore, reduce the time complexity of the problem,
3. try to reduce the space complexity of the problem.

In the following, two tree structures called *sheet-tree* and *grouping-tree* are introduced to model the search space. The former structure generates and scores all possible β-sheets and the later one generates and scores all possible conformations. Also, a dynamic programming approach is proposed to find the optimal conformation. It can be noted that the *sheet-tree* and the *grouping-tree* are constructed simultaneously but they are introduced separately for better understanding and more convenient description.

3.1. Generate and score possible β-sheets

To generate all possible β-sheets for a protein, we introduce a tree structure called *sheet-tree*. It is responsible for constructing all possible β-sheets. To find the optimal conformation, we will extend the *sheet-tree* to the *grouping-tree* and use a dynamic programming approach to solve the problem.

Before describing the *sheet-tree* construction algorithm (Algorithm 1), let us introduce some notations. Suppose that the target protein contains n β-strands $\beta_1, \beta_2, \ldots, \beta_n$. Moreover, each β-sheet is formed by m β-strands ($2 \le m \le n$). The first and the last β-strands in a β-sheet are denoted by β_f and β_r, respectively, and z is a set of intermediate β-strands (if any). Thus, each β-sheet is denoted by $S = [\beta_f, z, \beta_r]$ and, subsequently, $S.\beta_f$ and $S.\beta_r$ denote the first and the last strands in S, respectively, and also *s_score* is a function that calculates the score of a sheet. Furthermore, m_i represents the number of β-strands in sheet S_i, and finally each node of the tree represents a β-sheet. Algorithm 1 describes the details of the *sheet-tree* construction.

Algorithm 1. Construction of the sheet-tree

```
    // Constructing the first level
1   for i = 1 : n do
2       for j = 1 : n do
3           if (i ≠ j) then
4               Generate possible β-sheet formations using two β-strands β_i and β_j;
5               Calculate each sheet's score;  // equal to the pairwise alignment score
6           end if
7       end for
8   end for

      // Constructing other levels
9   for i = 2 : n-1 do
10      for each β-sheet S1 from the first level
11          for each β-sheet S2 from the (i-1)-th level
12              if (S1.βr = S2.βf) AND (S1 ∩ S2 = S1.βr) then
13                  Create β-sheet S= [S1.βf, S2.βf, S2.Z, S2.βr] with the length m2+1;
14                  s_score(S) = (m2*s_score(S2)+s_score(S1)) / (m2+1);  //reuse of previous calculations
15              end if
16          end for
17      end for
18      if (i ≠ 2) then
19          Remove the (i-1)-th level of the tree;    //reducing the space
20      end if
21  end for
```

The i-th level of the tree creates all possible β-sheets with i+1 β-strands and, hence, the tree will have n-1 levels. The first level of the tree creates all possible arrangements of two β-strands. In this level, the sheet score for each node is calculated as the pairwise alignment score of corresponding β-strands. The nodes of the i-th level of the tree ($i > 1$) are constructed as follows.

$$\left\{\left[S_1.\beta_f, S_2.\beta_f, S_2.z, S_2.\beta_r\right] | S_1 \in L_1, S_2 \in L_{i-1}, (S_1.\beta_r = S_2.\beta_f), \right.$$
$$\left. (S_1 \cap S_2 = S_1.\beta_r)\right\} \quad (3)$$

The sheet score for each node is calculated by taking the average of pseudo-energies of all its β-strands pairs. Thus, we can reuse the previous calculations (see line 11 of Algorithm 1). Notice that we utilize *pair scoring method* (Sgourakis et al., 2011) in our algorithms. Furthermore, since the i-th level of the tree is constructed only by using the nodes in the first and (i-1)-th levels, we can remove any intermediate levels (see line 12 of Algorithm 1). Therefore, we can significantly reduce the space complexity of the algorithm. It is clear that if we want to store all β-sheets, the last two lines of Algorithm 1 must be ignored. Fig. 3 shows an example of the construction of the tree for a protein with six strands. To avoid confusion, the construction of the tree is shown only for one node in each level.

Based on the above discussion, we make the following claims.

Theorem 1. The i-th level of the sheet-tree creates all possible β-sheets with i+1 strands.

Proof. We use strong induction on the number of levels in *sheet-tree*. The basic step for the first level is trivial because we create all combinations of two β-strands that form a β-sheet (see the steps of constructing the first level in Algorithm 1). Assume the result is true for all levels of the tree up to k-1. Consider the k-th level of the tree and assume that it does not create a sheet S=[β_1, β_2, . . . , β_{k+1}] with k+1 strands. We prove by contradiction to reject this assumption. Considering the construction of β-sheets in Algorithm 1 for this level, the sheet S may not be created due to one or both of the following causes:

1. The sheet S_1=[β_1, β_2] is not created in the first level. As discussed above, this assumption contradicts the basic step because the first level of the tree creates all β-sheets with two strands.
2. The sheet S_2=[β_2, β_3, . . . , β_{k+1}] is not created by the (k-1)-th level. This assumption also cannot be made since it contradicts the induction hypothesis.

Consequently, since S_1 and S_2 are created, respectively, in the first and (k-1)-th levels and also they have necessary and sufficient conditions for merging (line 9 of Algorithm 1), the assumption that the β-sheet S is not created by the k-th level of the tree must be false and hence proving the claim.

Lemma 1. The i-th ($i \geq 2$) level of the *sheet-tree* creates all possible β-sheets with i+1 strands using only the β-sheets in the first and i-1 levels.

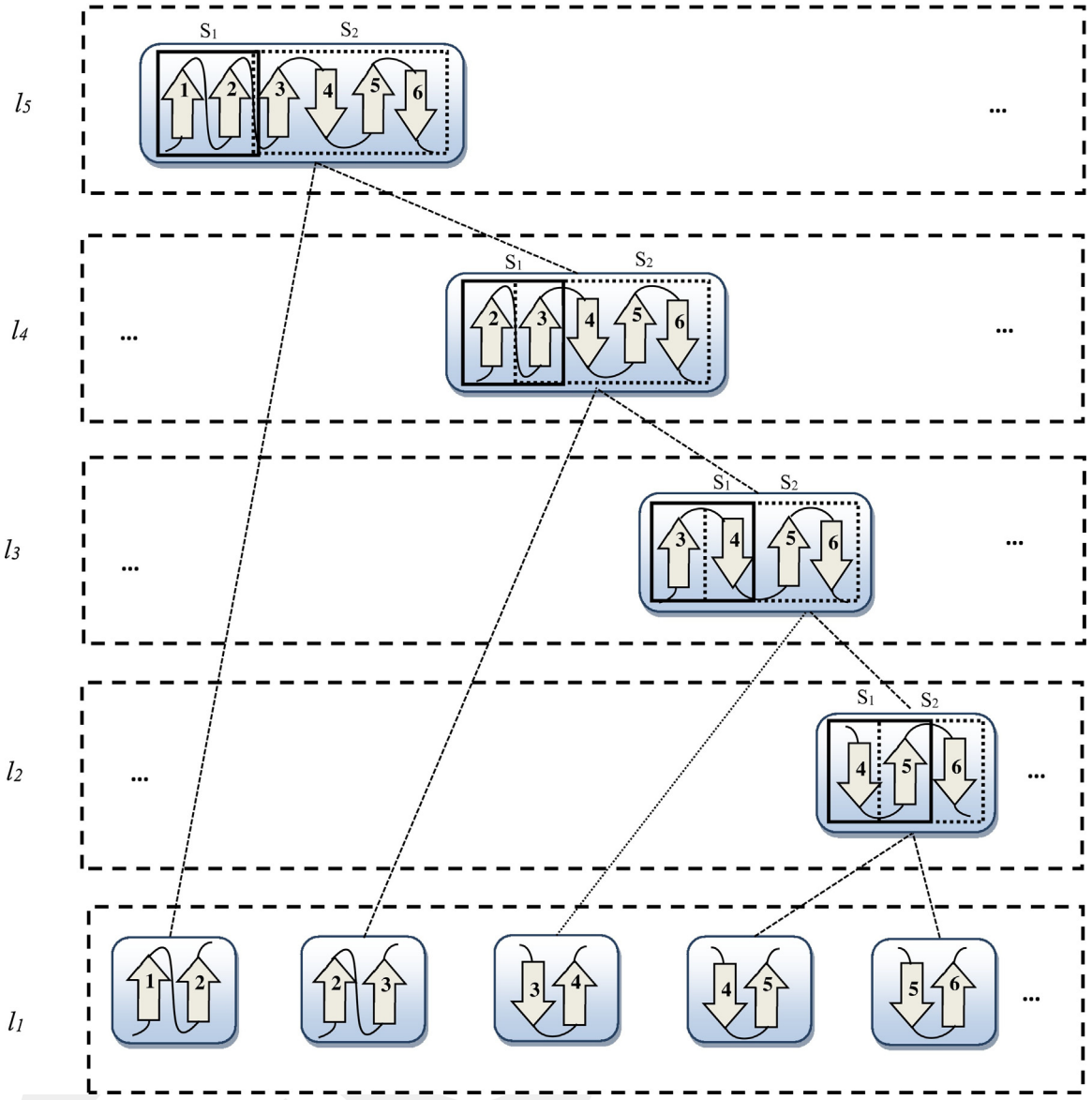

Fig. 3. Construction of the sheet-tree for a protein with six strands.

Proof. We use strong induction to prove our claim. The base case for the first level is trivial because in construction of the second level (i=2) of the tree, as described in Algorithm 1, S_1 and S_2 are selected to merge and form a β-sheet from only the first level. Additionally, based on the Theorem 1, all β-sheet formations with three strands are created in the second level. Now, as the induction hypothesis, let us assume the lemma is true for $i > 2$ up to k-1. We need to show the statement is true for the k-th level of the tree. In one hand, the Algorithm 1 is designed such that it creates a β-sheet S with k+1 strands with merging a two-strand β-sheet S_1 and k-strand β-sheet S_2 and there is no any other way in the algorithm to create the β-sheet S. On the other hand, based on the inductive hypothesis, the first and the (k-1)-th level of the tree create all possible β-sheet formations with two and k strands, respectively. Therefore, we can conclude that the statement is true.

Next, we summarize the main characteristics of the proposed *sheet-tree* construction algorithm as follows:

- It is responsible for constructing possible β-sheets.
- The calculated sheet scores are reused in higher levels of the tree and hence a considerable amount of calculations are avoided.
- The k-th level of the tree creates all possible β-sheets with k+1 β-strands using only the first and (k-1)-th level of the tree. It helps us to remove the intermediate nodes of the tree when constructing the *grouping-tree* (as it will be discussed later) and reduce the space of the problem.
- The total number of β-sheets at k-th level of the tree is equal to:

$$\prod_{i=0}^{k}(n-i) = \frac{n!}{(n-k-1)!}, \tag{4}$$

where n is the number of protein β-strands.

- Since in construction of each level we need the first and its previous level and also we can remove the intermediate levels, the maximum number of created nodes that needs to be stored in the memory are:

$$n(n-1) + \prod_{i=0}^{n-2}(n-i) + \prod_{i=0}^{n-1}(n-i) = 2n! + n(n-1) = O(n!), \tag{5}$$

where each of the terms in left side of Eq. (5) is the total number of nodes in the first, one before the last and the last levels, respectively. It is obvious that we use the storage space very efficiently in comparison with the case that we create and save all possible β-sheets (see Eq. (1)).

3.2. Generate and score possible conformations

The *sheet-tree* is responsible for constructing possible β-sheets but the goal is to find the optimal conformation. As discussed before, a β-sheet conformation partitions the protein β-strands into one or multiple β-sheets such that each β-sheet is formed by at least two β-strands which interact in parallel or antiparallel direction. Thus, we will extend the *sheet-tree* to the *grouping-tree* and propose a dynamic programming approach to solve the problem. The *grouping-tree* is responsible for constructing possible conformations. Before introducing the details of constructing the *grouping-tree* (Algorithm 2), we first introduce some notations. Each node of the tree is denoted by C which represents the optimal conformation that can be constructed by its β-strands. Moreover, *s_score* and *β_score* are the functions that calculate the score of a sheet and a conformation, respectively. Additionally, *C.structure* denotes the β-sheet structure of C, including the assignment and arrangement of its β-strands into β-sheets. The following recursive algorithm creates possible conformations and the optimal conformation will be placed at the last level of the tree.

Algorithm 2. Constructing the grouping-tree and finding the optimal conformation

Algorithm 2: Constructing the grouping-tree and finding the optimal conformation

```
for i = 1 : n-1 do
    Create all combinations C_j of n β-strands;
    for each combination C_j do
        for each β-sheet B_j do
            S* = the β-sheet in the i-th level of the sheet-tree with maximum
                 s_score that contains the β-strands in C_j;
            if (i ≤ 2) then
                C_j .structure = S*;
                β_score(C_j) = s_score(S*);
            else
                for all k possible pipe-position in B_j that forms C_jk do
                    Find_the_optimal_conformation C'_jk and C''_jk;
                      //reuse the previous calculations
                    C_jk .structure = merge C'_jk and C''_jk;
                    β_score(C_jk)= The average of s_score(C'_jk) and s_score(C''_jk);
                    //reuse the previous calculations
                end for
                C* = argmax_{s_score}(C_jk); //C_jk contains the β-strands in C_j
                if (β_score (C*) > s_score(S*)) then
                    C_j .structure = C*;
                    β-score(C_j) = β-score(C*);
                else
                    C_j .structure = S*;    // No pipe
                    β_score(C_j) = β_score(S*);
                end if
            end if
        end for
    end for
end for
The optimal conformation = the node at the last level;
```

The i-th level of the tree creates all optimal conformations that can be constructed by the i+1 β-strands. Each node stores the optimal arrangement of its β-strands and also its conformation score (β_score). Since each β-sheet should contain at least two

β-strands, each node in the first/second level includes only one β-sheet. Consequently, to find the optimal conformation for the nodes in these levels, it is sufficient to find the β-sheet with maximum *s_score* in the *sheet-tree* which is formed by the same β-strands. However, each conformation at higher levels of the tree can be formed by more than one sheet. For example, a conformation with four strands can be formed by a four-strand β-sheet or two of two-strand β-sheets. To solve the problem, we utilize a dynamic programming approach. The problem of finding the optimal conformation for each C_{jk} of the tree is broken into two sub-problems C'_{jk} and C''_{jk} (see Fig. 4), whose optimal arrangement was found recursively at lower levels of the tree (line 10 of Algorithm 2). The score of the conformation C_{jk} is calculated by taking the average β_score of C'_{jk} and C''_{jk}. The break point is denoted by the *pipe* operator. In fact the pipe position indicates that C'_{jk} and C''_{jk} are located into two different β-sheets, however, they can be formed by some β-sheets, independently. For each C_j with m strands, there is m-2 pipe position with regard to the biological strand pairing limitations. Then, among all possible conformations which are derived by different pipe positions plus no pipe, the conformation with maximum β_score (C^*) is assigned to C_j by defining the *argmax* operator. Finally, the conformation at the last level of the tree is returned as the output of the algorithm. Fig. 4 shows an example of the construction of the *grouping-tree* for a protein with six strands. To avoid confusion, the construction of the tree is shown only for one node in each level.

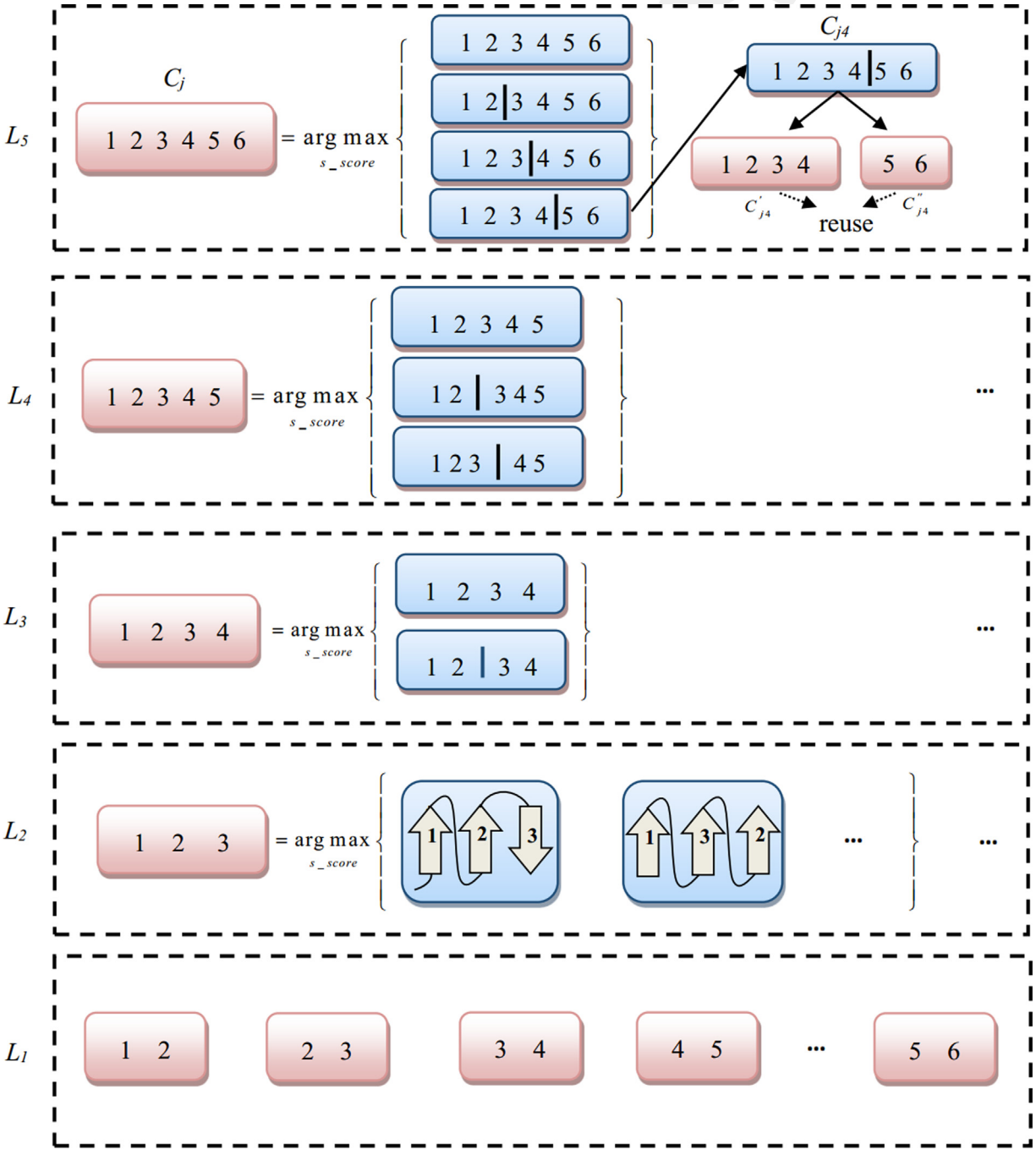

Fig. 4. Construction of the grouping-tree for a protein with six strands.

Our proposed approach for solving the problem (BetaTop) has the sufficient conditions that an optimization problem must have in order for dynamic programming to be applicable, namely, overlapping sub-problems and optimal sub-structure. First, we formulated the problem as composed of smaller sub-problems and the solutions are combined to get the final result. Second, as discussed in Algorithm 2, the optimal arrangement of β-strands in β-sheets is obtained for each sub-structure. In the following, let us summarize the highlights of the *grouping-tree* as follows:

- It is responsible for constructing the possible conformations and finding the optimal solution.
- The *sheet-tree* and the *grouping-tree* are constructed, simultaneously. In fact, we extended the *sheet-tree* and introduced the *grouping-tree.*
- The *k*-th level of the tree creates all optimal conformations that can be constructed by the *k*+1 β-strands.
- The total number of conformations in *k*-th level of the tree is equal to the $\binom{n}{k+1}$, where *n* is the number of protein strands.
- Since the *sheet-tree* and *grouping-tree* are constructed simultaneously, in addition to all possible conformations, all possible β-sheets also are created to calculate and find the optimal conformation. However, after construction the *k*-th level of the tree, as discussed in Section 3.1, all of the β-sheets can be removed and only the optimal conformations are retained. Hence, the algorithm searches the conformational space efficiently and subsequently reduces the time and space complexities of the problem.
- The total number of calculations for all of the conformations is:

$$\left[\sum_{k=2}^{n-1}(k-1)\times\binom{n}{k+1}\right]+\binom{n}{2}, \tag{6}$$

where *k* denotes the level of the tree and the term $\binom{n}{2}$ shows the total number of conformations at the first level.

3.3. Time and space complexities

To find the optimal conformation for a protein with *n* β-strands, the proposed BetaTop utilizes a tree structure. In each level of the tree, all possible conformations are generated by introducing the pipe operator. For each conformation, all possible β-sheets are generated and the structure with the maximum score is stored. In Eqs. (4) and (6), for each level of the tree, we calculated the number of possible β-sheets and possible conformations, respectively. Therefore, the computation time and the time complexity of the proposed algorithm can be calculated by merging (4) and (6) as follows:

$$\begin{aligned}&\left[\sum_{k=2}^{n-1}(k-1)\times\frac{n!}{(n-k-1)!}\right]+\binom{n}{2}\\&=n!\left(\frac{1}{(n-3)!}+\frac{2}{(n-4)!}+\ldots+\frac{n-2}{1}\right)+\frac{n(n-1)}{2}\\&=n(n-1)(n-2)+2n(n-1)(n-2)(n-3)+\ldots\\&+(n-2)n!+\frac{n(n-1)}{2}=O(n!).\end{aligned}$$

It is worthy to mention here that this total number of calculations is much less than the brute-force one (followed by state-space search-based methods) which is roughly equal to the number of subsets of possible β-sheets in Eq. (1).

To find the total required memory space of BetaTop, the following points should be noted:

1) The total number of stored conformations is equal to:

$$\sum_{k=1}^{n-1}\binom{n}{k+1}=2^n-n-1. \tag{8}$$

2) Based on Eq. (5), the maximum number of possible β-sheets that needs to be stored in the memory are *2n!+n(n−1).*

Thus, the space complexity of BetaTop can be calculated as the required space for storing all nodes of the tree which were obtained in Eqs. (7) and (8):

$$\begin{aligned}\sum_{k=1}^{n-1}\binom{n}{k+1}+2n!+n(n-1)&=(2^n-n-1)+(2n!+n(n-1))\\&=2n!+2^n+n^2-2n-1=O(n!).\end{aligned} \tag{9}$$

At the end of this section, we summarize the main advantages of the proposed algorithm as follows. First, it prevents repetitive calculations and, therefore, reduce the time complexity of the problem utilizing a dynamic programming approach. Thus, it can solve the β-sheet prediction problem for proteins with higher number of β-strands. The second strength of the proposed algorithm is that it removes many nodes which will not be reused at higher levels of the tree and consequently reduces the memory space requirements of the problem. Hence, we proposed an efficient dynamic programming approach for solving the problem that reduces the time complexity and control the space of the problem, simultaneously. In the next section, the performance of the BetaTop will be assessed and the obtained experimental results will be discussed.

4. Experimental results

4.1. Data

Here, the BetaSheet916 dataset is used for evaluating the proposed BetaTop. The dataset was first introduced by Cheng and Baldi (2005) and adopted by Savojardo et al. (2013); Aydin et al. (2011); Eghdami et al. (2015); Lippi and Frasconi (2009). It contains 916 protein chains which is extracted from the PDB (Berman et al., 2000). Secondary structure labels are assigned to residues by the DSSP (Kabsch and Sander, 1983). The set is divided into 10-folds for cross-validation. Table 1 shows the statistics of the dataset, while Table 2 contains the number of proteins in each fold with respect to the number of β-strands. Furthermore, histogram plot for the number of β-strands in the BetaSheet916 dataset are shown in Fig. 5.

4.2. Experimental settings

In this paper, a three-stage approach is adopted for protein β-sheet prediction (Fig. 1). In the first stage, residue-residue

Table 1
Statistics of the BetaSheet916.

Feature	Number
Protein chains	916
Total residues	187,516
β-residues	48,996
β-strands	10,745
β-strand pairs	8172
Parallel β-strand pairs	4519
Antiparallel β-strand pairs	2214
β-sheets	2533

Table 2
The number of protein chains in each fold of the BetaSheet916.

	$Fold_1$	$Fold_2$	$Fold_3$	$Fold_4$	$Fold_5$	$Fold_6$	$Fold_7$	$Fold_8$	$Fold_9$	$Fold_{10}$	Total
Up to 6 strands	16	24	15	15	25	16	20	21	18	17	187
7 strands	6	9	2	5	7	10	5	4	6	5	59
8 strands	3	6	5	7	2	10	9	6	6	8	62
9 strands	9	6	6	6	5	2	4	8	7	12	65
10 strands	4	6	8	7	7	7	7	9	8	6	69
>10 strands	54	41	56	52	46	47	46	43	46	43	474

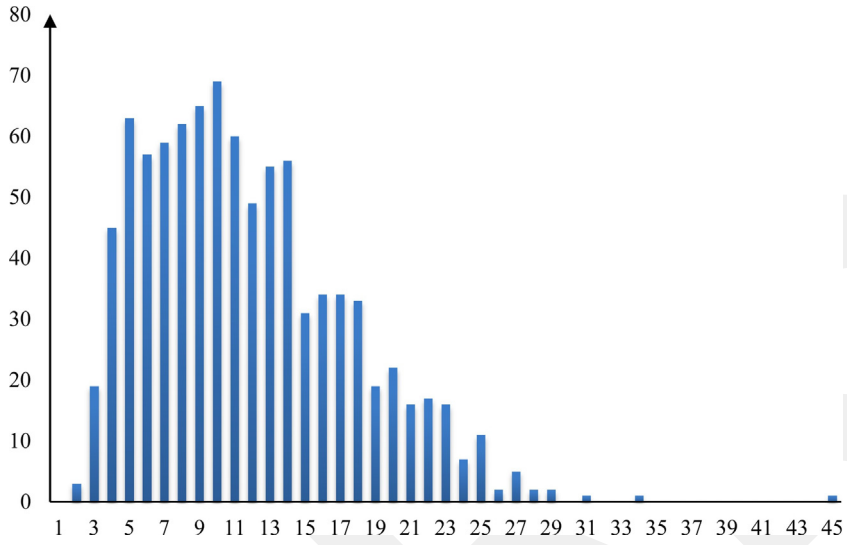

Fig. 5. Histogram for the number of β-strands in dataset.

contacts are predicted using a 2D-RNN, which is introduced in Cheng and Baldi (2005). Then, the optimal pairwise alignments of β-strands are obtained by the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), and finally the proposed BetaTop predicts the optimal β-sheet structure of a protein. In the third stage, we need two scoring functions to estimate the biological significance of a β-sheet and also a conformation. According to our discussion at the end of Section 2, the pair scoring method is chosen for this purpose. As shown in Algorithm 1 and Algorithm 2, the sheet score (*s_score*) and the conformation score (*β_score*) are calculated by taking the average of pseudo-energies of all β-strand pairs in a sheet and all β-sheets in a conformation, respectively.

4.3. Computational time of the prediction

The state-space search-based approaches cannot be applied to predict the β-sheet structure for proteins with more than six β-strands due to the combinatorial explosion problem. However, as shown in Table 2 and Fig. 5, the percentage of proteins with more than six β-strands in BetaSheet916 dataset is calculated as 79.58 percent. The BetaZa method utilized a greedy algorithm for proteins with more than six β-strands (just like BetaPro) and Fonseca et al. (2011) found a subset of conformation with less than six β-strands in facing this problem. Therefore, to evaluate the computational time of the proposed method, we compare it with a brute-force search (BFS) method that, similar to our proposed BetaTop, searches all the space of the problem. However, BetaTop performs this function, efficiently. Table 3 shows the obtained results on the BetaSheet916 dataset. The experiments are performed on a 2.50 GHz Intel Core i7 processor and 8.0 GB of DDR3 memory.

As shown in Table 3, the proposed BetaTop has reduced the prediction time, significantly. For proteins with more than eight strands, the optimal conformation were not calculated in a

Table 3
Computational Time of BetaTop (seconds).

	Up to 6 strands		7 strands		8 strands		9 strands		10 strands	
#Fold	BFS[a]	BetaTop	BFS	BetaTop	BFS	BetaTop	BFS	BetaTop	BFS	BetaTop
1	4.12	0.07	105.09	3.14	1505.45	25.76	–	504.08	–	2172.48
2	9.36	0.16	190.28	4.53	4592.77	41.45	–	317.46	–	3588.52
3	13.18	0.15	49.73	0.90	3840.02	49.35	–	292.67	–	4390.54
4	13.23	0.12	106.69	2.52	4211.02	55.94	–	301.76	–	4165.44
5	16.78	0.16	130.30	3.64	733.29	11.97	–	247.60	–	4008.70
6	5.94	0.07	186.02	5.39	12331.90	84.93	–	78.36	–	4083.24
7	3.62	0.06	108.77	2.51	9274.79	78.90	–	185.68	–	4150.39
8	17.83	0.21	94.31	2.00	7213.87	43.87	–	465.32	–	5557.89
9	13.64	0.14	133.38	3.04	4115.33	56.03	–	398.70	–	4777.30
10	9.47	0.11	109.98	2.46	10772.30	72.60	–	745.68	–	3424.10

[a] Brute-force search.

Table 4
Average runtime (seconds) for each protein.

Method	Up to 6 strands	7 strands	8 strands	9 strands	10 strands
BFS	0.573	20.59	945.012	–	–
BetaTop	0.007	0.511	8.400	54.420	584.327

reasonable computation time by the BFS approach. For example, for a protein with nine strands (PDB id: 1A9O), the BFS method did not report the result after more than eight hours of running time, whereas the BetaTop obtained the optimal conformation in less than one minute. Table 4 shows the average runtime results for each protein with respect to its number of β-strands.

It is evident from Table 4 that the proposed method decreases the computational time of protein β-sheet structure prediction, significantly. Consequently, it can be applied to proteins with more than six β-strands in order to search the space of possible conformations and consequently find more accurate results.

4.4. Performance of prediction

We compared the accuracy of the proposed BetaTop at the following levels: strand pairing, pairing direction (parallel or antiparallel) and amino acid residue pairing (contact map) levels. In either level, to evaluate the prediction accuracy, the following well-known statistical metrics were used:

$$\text{Precision (P)} = \frac{TP}{TP + FP} \times 100, \tag{10}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \times 100, \tag{11}$$

$$\text{F1} - \text{score (F1)} = \frac{2 \times P \times R}{P + R}, \tag{12}$$

where TP, FP and FN are true positive, false positive and false negative, respectively. In Table 5, we compare the performance of BetaTop at β-strand pairing level with the state-of-the-art methods on the BetaSheet916 dataset with respect to the number of β-strands of proteins.

From the obtained results, we can conclude that BetaTop outperforms the other methods. There are three important points regarding the results that must be mentioned. First, for proteins with more than six strands, Fonseca et al. choose a subset of six strands. This process is repeated for all subsets of six strands and finally the maximum conformation is selected as the result. This strategy causes that the predicted pairs of β-strands to be relevant (high precision) but there are many relevant pairs of β-strands that may be not considered by this method (low recall). Thus, as shown in Table 4, the recall and consequently F1-score of the method decreases with increases in the number of β-strands in proteins. Second, as stated in Aydin et al. (2011), the Bayesian model of BetaZa for proteins with more than six strands becomes less specific, and therefore, its discriminative power reduces. For such proteins, BetaZa simply chooses the same β-strand pairing predictions as BetaPro. Third, BetaPro allows each β-strand to have at most three strand partners, whereas in BetaTop, each β-strand can interact with at most two other β-strands. For this reason, BetaPro has reported higher prediction performance measures in some experiments at strand pairing level (Table 5), pairing direction (Table 6) and residue levels (Table 7). Fig. 6 shows the comparisons of the F1-scores obtained with different methods at pairing strands level with respect to the number of β-strands in proteins on BetaSheet916 dataset.

Table 5
BetaTop performance at strand pairing level.

	Up to 6 strands			7 strands			8 strands			9 strands			10 strands		
Method	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BetaZa	81.34	80.37	80.85	–	–	–	–	–	–	–	–	–	–	–	–
BetaPro	82.06	77.30	79.61	71.79	71.79	71.79	73.80	72.25	73.02	62.69	63.36	63.02	61.69	66.73	64.11
Fonseca et al.	78.26	66.38	71.83	72.80	53.20	61.47	76.49	46.85	58.11	77.95	41.16	53.87	77.22	38.02	50.95
BetaTop	86.01	79.37	82.55	76.34	71.43	73.80	82.05	75.17	78.46	72.12	69.00	70.52	70.69	68.79	69.73

Table 6
BetaTop performance at pairing direction level.

	Up to 6 strands			7 strands			8 strands			9 strands			10 strands		
Method	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BetaZa	81.35	80.37	80.86	–	–	–	–	–	–	–	–	–	–	–	–
BetaPro	74.23	85.97	79.67	72.08	66.03	68.92	79.45	63.73	70.73	62.84	57.94	60.29	65.81	62.01	63.85
Fonseca et al.	80.81	68.26	74.01	75.00	54.96	63.44	51.38	34.07	40.97	76.14	41.74	53.92	79.88	41.03	54.22
BetaTop	77.16	83.62	80.26	68.75	71.37	70.04	71.48	78.02	74.61	57.81	63.43	60.49	62.19	62.01	62.10

Table 7
BetaTop performance at residue pairing level.

	Up to 6 strands			7 strands			8 strands			9 strands			10 strands		
Method	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BetaZa	79.50	77.62	78.55	–	–	–	–	–	–	–	–	–	–	–	–
BetaPro	73.57	71.63	72.59	70.31	73.96	72.09	71.00	71.93	71.46	58.62	64.14	61.26	59.69	67.68	63.43
Fonseca et al.	70.84	57.89	63.71	69.77	52.73	60.04	74.61	46.97	57.65	66.82	41.75	51.39	69.76	41.89	52.35
BetaTop	77.67	82.08	79.81	67.22	70.88	69.00	73.31	77.48	75.34	61.89	64.36	63.10	61.81	66.71	64.17

Tables 6 and 7 show the performance of the proposed BetaTop at strand pairing direction and residue pairing levels, respectively, with respect to the number of β-strands of proteins. Moreover, Figs. 7 and 8 shows the comparisons of the F1-scores of different methods obtained from Tables 5 and 6, respectively. As discussed earlier, BetaPro has shown better performance in comparison with BetaTop in some experiments because it allows each β-strand to have up to three strand partners.

Moreover, we may want to know the number of predicted proteins which have exactly the same β-sheet structure as the target proteins both at strand pairing and pairing direction levels. This number can reveal how much a β-sheet structure prediction method is reliable. In this regard, Table 8 compares the BetaTop results with the others.

As shown in Table 8, BetaTop significantly outperforms the other methods. For example, the β-sheet structure for 52.41% of proteins with up to six β-strands were predicted correctly by BetaTop, whereas these percentages were reported as 39.75%, 27.81 and 18.18% for BetaZa, BetaPro and Fonseca et al., respectively.

For the last experiment, we will study the effect of contact map prediction methods on the performance of the proposed BetaTop. Ma et al. (2015) classify existing contact map prediction methods to roughly two categories: (1) evolutionary coupling (EC) analysis methods, and (2) supervised machine learning (ML) methods. Experiments show that due to use of more information, supervised learning may outperform EC methods for proteins with few sequence homologs (Wang and Xu, 2013). In fact, the supervised ML methods, as opposed to EC analysis, have not limited to specific category of proteins (with known homologues sequences). In the previous experiments, a supervised machine learning method is chosen for predicting the contacts between β-residues in a protein. However, we know that significant improvements in protein structure prediction has been recently reported by using analysis of residue co-evolution or deep learning approach. Therefore, we have investigated whether these methods improve the performance of our proposed method. Furthermore, we have performed this experiment on a recently proposed dataset. This new dataset, BetaSheet1452, which was introduced by Savojardo et al. (2013) in 2013, consists of 1452 protein chains containing 56,552 β-residue contacts. In this experiment, we utilized an evolutionary coupling analysis method, namely PSICOV (Jones et al., 2012), and also a deep-learning method (Wang et al., 2017) as the first step of BetaTop. Table 9 compares the performance of BetaTop when it

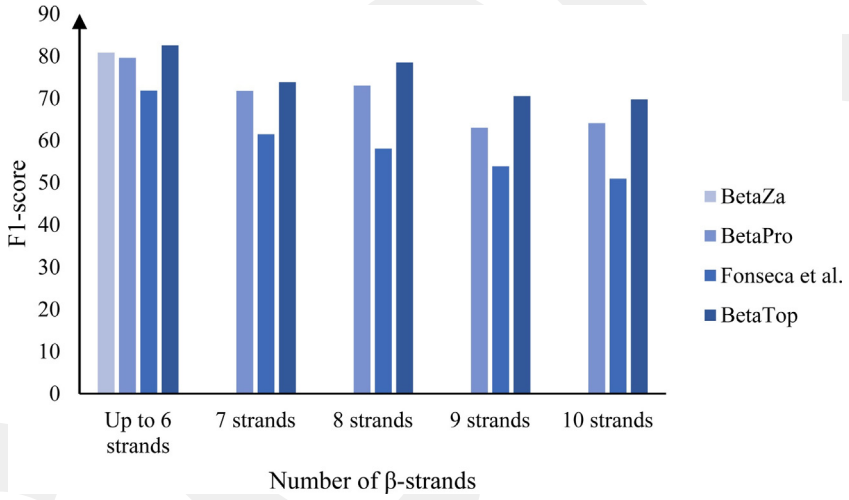

Fig. 6. F1-scores of different methods at pairing strands level with respect to the number of β-strands in proteins.

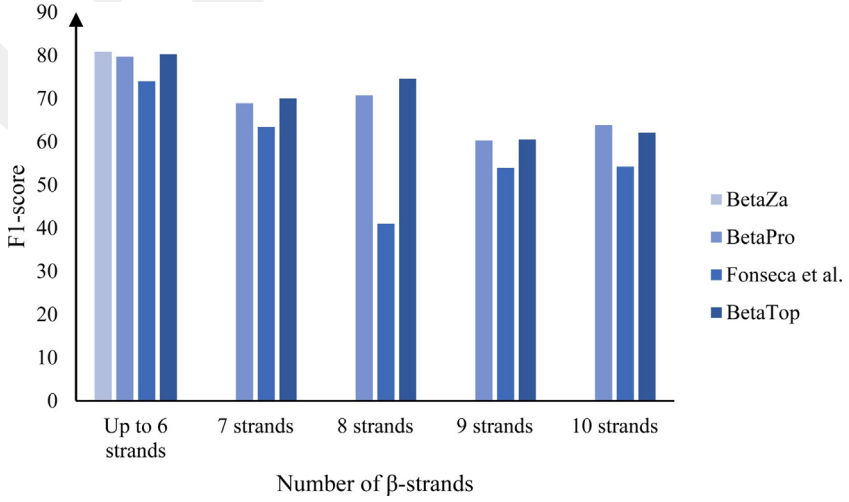

Fig. 7. F1-scores of different methods at pairing direction level with respect to the number of β-strands in proteins.

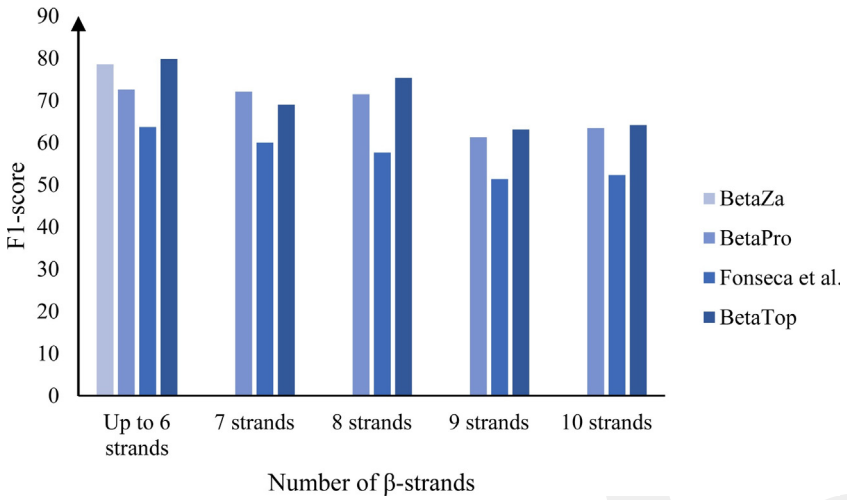

Fig. 8. F1-scores of different methods at residue level with respect to the number of β-strands in proteins.

Table 8
The number of correctly predicted proteins both at strand pairing and pairing direction levels.

Method	Up to 6 strands	7 strands	8 strands	9 strands	10 strands
BetaZa	74	–	–	–	–
BetaPro	52	3	2	1	3
Fonseca et al.	34	0	0	0	0
BetaTop	98	12	11	2	4

utilizes different contact map prediction methods on BetaSheet1452 dataset for proteins up to 10 β-strands at different prediction levels.

As shown in Table 8, there is no method that produce the best results in all prediction levels. However, regardless of pairing direction level results, utilizing a deep learning contact map predictor seems a better choice. The other reason for its better performance is related to this fact that it predicts contacts by integrating both evolutionary coupling and sequence conservation information through an ultra-deep neural network formed by two deep residual neural networks (Wang et al., 2017). Although better performance is achieved, it is worthy to note that deep learning methods also have problems such as memory limitations, high computational time required to train or optimally setting the weights within a network, complex theoretical analysis and mathematical techniques, parameters tuning problem (time-consuming and fine tuning problem) and lack of theory surrounding them. In the other hand, deep learning methods are often looked at as a black box, with most confirmations done empirically, rather than theoretically. Finally, the accuracy of the best contact map predictors for long-range contacts is still close to 30%. Hence, we cannot expect significant improvements in the performance of β-sheet structure prediction methods by applying contact map predictors due to their low accuracy for β-residue contacts.

5. Conclusion

In this paper, we presented BetaTop, a new method which efficiently searches the conformational space of a protein to predict its β-sheet structure. Unlike other state-space search-based methods, BetaTop can deal with the proteins with higher number of β-strands. The main advantages of the proposed method can be summarized as follows:

- Reducing the time complexity of the problem: The search space of the β-sheet structure prediction problem enormously increases with respect to the number of protein β-strands. This causes the methods that search the conformational space to lose their ability to solve the problem in a reasonable time. In addition to this, repetitive calculation in the conformational space leads to dealing with a combinatorial explosion problem with intractable computational complexity. However, the proposed BetaTop breaks the problem into sub-problems and reuses the intermediate results utilizing a dynamic programming approach to avoid repetitive calculations and therefore, reduces the time complexity of the problem.
- Controlling the space of the problem: One of the interesting aspects of the proposed method is that its tree structure is designed in such a way that we can remove many nodes which will not be reused at higher levels of the tree and consequently reduce the space requirements of the problem.
- Dealing with proteins which have higher number of β-strands: Since BetaTop reuses the intermediate results and avoids repetitive computations and also removes many nodes which will not be used at higher levels of the tree, it has a reasonable

Table 9
BetaTop performance using different contact map predictors on BetaSheet1452 dataset.

	strand pairing level			pairing direction level			residue pairing level		
Method	P	R	F1	P	R	F1	P	R	F1
BetaTop	72.40	81.44	76.65	52.02	60.78	56.06	39.63	37.17	38.36
BetaTop+PSICOV	70.95	82.13	76.13	55.51	64.25	59.56	43.29	35.14	38.79
BetaTop+DL[a]	76.36	80.33	78.29	51.23	54.55	52.84	50.63	44.45	47.34

[a] Deep Learning.

time and space complexities in comparison with other state-space search-based method. Consequently, it will be applicable to predict β-sheet structure of proteins with higher number of β-strands. The experimental results showed that the BetaTop can easily predict the β-sheet structure of proteins up to 10 β-strands.

- Improving the prediction performance: Since BetaTop searches the entire space for proteins with higher number of β-strands, in comparison with other methods, it can find more accurate β-sheet structures.

The performance comparisons with the previous studies indicated the superiority of BetaTop in different prediction levels. However, there are some directions for further improvements. First, with increasing the number of β-strands, the total number of nodes of the tree and consequently the computation time of the problem increases. Thus, pruning the search space can be a promising way to tackle the problem. For example, we can consider the amino acids as well as β-strands with significant scores and eliminate the conformations related to the others from the search space and do not make any further computations. Second, in the proposed method, each β-strand can interact with at most two other β-strands. Hence, we will extend our method in order to perform a fair comparison with methods, such as BetaPro, that allow a β-strand to have more than two partners.

Acknowledgements

The authors gratefully thank Dr. Castrense Savojardo and Dr. Sheng Wang for providing the results of their research papers Savojardo et al. (2013) and Wang et al. (2017), respectively, which are used in some of our experiments.

References

Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. Science (80-.) 181, 223–230.
Aydin, Z., Altunbasak, Y., Erdogan, H., 2011. Bayesian models and algorithms for protein beta-sheet prediction. IEEE/ACM Trans. Comput. Biol. Bioinforma. 8 (2), 395–409.
Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. The protein data bank. Nucleic Acids Res. 28, 235–242.
BetaSheet916 Set, 2009. [Online]. Available: http://www.ics.uci.edu/~baldig/betasheet_data.html.
Carbonell, Y.L.J., 2003. Prediction of parallel and anti-parallel beta-sheets using Conditional Random Fields. Inst. Softw. Res. 24, 191–208.
Cheng, J., Baldi, P., 2005. Three-stage prediction of protein beta-sheets by neural networks, alignments and graph algorithms. Bioinformatics 1, i75–84.
Chiti, F., Dobson, C.M., 2006. Protein misfolding, functional amyloid and human disease. Annu. Rev. Biochem. 75 (1), 333–366.
Eghdami, M., Dehghani, T., Naghibzadeh, M., 2015. BetaProbe: a probability based method for predicting beta sheet topology using integer programming. 2015 5th International Conference on Computer and Knowledge Engineering (ICCKE) 152–157.
Fonseca, R., Helles, G., Winter, P., 2011. Ranking beta sheet topologies with applications to protein structure prediction. J. Math. Model. Algorithms 10 (4), 357–369 LA–English.
Jeong, J., Berman, P., Przytycka, T.M., 2008. Improving strand pairing prediction through exploring folding cooperativity. IEEE/ACM Trans. Comput. Biol. Bioinform 5 (4), 484–491.
Jones, D.T., Buchan, D.W., Cozzetto, D., Pontil, M., 2012. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28.
Jones, D., 1999. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292, 195–202.
Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22 (12), 2577–2637.
Kortemme, T., Ramirez-Alvarado, M., Serrano, L., 1998. Design of a 20-amino acid, three-stranded beta-sheet protein. Science 281 (July (5374)), 253–256.
Kouza, M., Banerji, A., Kolinski, A., Buhimschi, I.A., Kloczkowski, A., 2017. Oligomerization of FVFLM peptides and their ability to inhibit beta amyloid peptides aggregation: consideration as a possible model. Phys. Chem. Chem. Phys. 19 (4), 2990–2999.
Kuhlman, B., Dantas, G., Ireton, G.C., Varani, G., Stoddard, B.L., Baker, D., 2003. Design of a novel globular protein fold with atomic-level accuracy. Science 302 (November (5649)), 1364–1368.
Lin, K., Simossis, V.A., Taylor, W.R., Heringa, J., 2005. A simple and fast secondary structure prediction algorithm using hidden neural networks. Bioinformatics 21, 152–159.
Lippi, M., Frasconi, P., 2009. Prediction of protein β-residue contacts by Markov logic networks with grounding-specific weights. Bioinforma 25 (18), 2326–2333.
Ma, J., Wang, S., Wang, Z., Xu, J., 2015. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. Bioinformatics 31 (November (21)), 3506–3513.
Magnan, C.N., Baldi, P., 2014. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 30 (September (18)), 2592–2597.
Mandel-Gutfreund, Y., Zaremba, S.M., Gregoret, L.M., 2001. Contributions of residue pairing to beta-sheet formation: conservation and covariation of amino acid residue pairs on antiparallel beta-strands. J. Mol. Biol. 305 (5), 1145–1159.
Merkel, J.S., Regan, L., 2000. 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. 275 (38), 29200–29206.
Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48 (3), 443–453.
Pre-Compiled CulledPDB Lists from PISCES 2008. [Online]. Available: http://dunbrack.fccc.edu/Guoli/pisces_download.php.
Pruitt, K.D., Tatusova, T., Klimke, W., Maglott, D.R., 2009. NCBI reference sequences: current status, policy and new initiatives. Nucleic Acids Res. 37 (Database issue), D32–D36.
Ruczinski, I., Kooperberg, C., Bonneau, R., Baker, D., 2002a. Distributions of beta sheets in proteins with application to structure prediction. Proteins Struct. Funct. Genet. 48 (1), 85–97.
Ruczinski, I., Kooperberg, C., Bonneau, R., Baker, D., 2002b. Distributions of beta sheets in proteins with application to structure prediction. Proteins Struct. Funct. Bioinforma. 48 (1), 85–97.
Ruczinski, I., 2002. Logic Regression and Statistical Issues Related to the Protein Folding Problem. University of Washington.
Sabzekar, M., Naghibzadeh, M., Sadri, J., 2017. Efficient dynamic programming algorithm with prior knowledge for protein β-strand alignment. J. Theor. Biol. 417, 43–50.
Savojardo, C., Fariselli, P., Martelli, P.L., Casadio, R., 2013. BCov: a method for predicting β-sheet topology using sparse inverse covariance estimation and integer programming. Bioinformatics 29 (December (24)), 3151–3157.
Sgourakis, N.G., Merced-Serrano, M., Boutsidis, C., Drineas, P., Du, Z., Wang, C., Garcia, A.E., 2011. Atomic-level characterization of the ensemble of the Aβ(1-42) monomer in water using unbiased molecular dynamics simulations and spectral algorithms. J. Mol. Biol. 405 (2), 570–583.
Steward, R.E., Thornton, J.M., 2002. Prediction of strand pairing in antiparallel and parallel β-sheets using information theory. Proteins Struct. Funct. Bioinforma. 48 (2), 178–191.
Subramani, A., Floudas, C.A., 2012. β-sheet topology prediction with high precision and recall for β and mixed α/β proteins. PLoS One 7 (3), e32461.
Tsutsumi, M., Otaki, J.M., 2011. Parallel and antiparallel β-strands differ in amino acid composition and availability of short constituent sequences. J. Chem. Inf. Model. 51 (6), 1457–1464.
Wang, Z., Xu, J., 2013. Predicting protein contact map using evolutionary and physical constraints by integer programming. Bioinforma 29 (13), i266–i273.
Wang, S., Peng, J., Ma, J., Xu, J., 2016. Protein secondary structure prediction using deep convolutional neural fields. Sci. Rep. 6 (January), 18962.
Wang, S., Sun, S., Li, Z., Zhang, R., Xu, J., 2017. Accurate de novo prediction of protein contact map by ultra-deep learning model. PLOS Comput. Biol. 13 (1), e1005324.
Zhang, N., Duan, G., Gao, J., Ruan, J., Zhang, T., 2010. Prediction of the parallel/antiparallel orientation of beta-strands using amino acid pairing preferences and support vector machines. J. Theor. Biol. 263 (3), 360–368.