Simultaneous Alignment and Folding of Protein Sequences

Jérôme Waldispühl[1,2], Charles W. O'Donnell[2], Sebastian Will[3], Srinivas Devadas[2], Rolf Backofen[3,†], Bonnie Berger[1,2,†]

[1] Department of Mathematics, MIT, Cambridge, USA,
[2] Computer Science and AI Lab, MIT, Cambridge, USA,
[3] Institut für Informatik, Albert-Ludwigs-Universität, Freiburg, Germany

[†] Corresponding author: bab@mit.edu, backofen@informatik.uni-freiburg.de

Abstract

One of the central challenges in computational biology is to develop accurate tools for protein structure analysis. Particularly difficult cases of this are sequence alignment and consensus folding of low-homology proteins. In this work, we present partiFold-Align, the first algorithm for simultaneous alignment and consensus folding of unaligned protein sequences; the algorithm's complexity is polynomial in time and space. Algorithmically, partiFold-Align additionally exploits sparsity in the set of likely super-secondary structure pairings and alignment candidates for each amino acid to achieve an effectively cubic running time for simultaneous pairwise alignment and folding. We demonstrate the efficacy of these techniques on transmembrane β-barrel proteins, an important yet difficult class of proteins with very few available three-dimensional structures. In tests on sequence alignments derived from structure alignments, partiFold-Align is significantly more accurate than current best approaches for pairwise sequence alignment in the difficult case of low sequence homology and improves secondary structure prediction when current approaches fail. Importantly, partiFold-Align does not require training on transmembrane β-barrel proteins. The generality of these techniques should allow them to be applied to a wide variety of protein structures.

1. INTRODUCTION

The consensus fold of proteins is an important consideration in structural bioinformatics analyses. In structure-function relationship studies, proteins that have the same consensus fold are likely to have the same function and be evolutionarily related [1]; in protein structure prediction studies, consensus fold predictions can guide tertiary structure predictors; and in sequence alignment algorithms [2], consensus fold predictions can improve alignments. The primary limitations in achieving accurate consensus folding, however, is the difficulty of obtaining reliable sequence alignments for divergent protein families and the inaccuracy of folding algorithms.

The specific problem we address is predicting consensus fold of proteins from their unaligned sequences. The consensus fold is the common minimum energy structure, given a sequence alignment, which is not to be confused with the agreed structure between unrelated predictors[cite]. We take an approach in which we *simultaneously* align and fold protein sequences. By optimizing unaligned protein sequences for both sequence homology and structural conservation concurrently, both higher fidelity sequence alignment and structure prediction can be obtained. For sequence alignment, this sidesteps the requirement of correct initial profiles (because the best sequence aligners require profile/profile alignment [3]). For structure prediction, this harnesses powerful evolutionary corollaries between structure.

While this class of problems has received a lot of attention in the RNA world [4], [5], [6], [7], [8], [9], [10], [11], [12], it has not yet been applied to proteins. Applying these techniques to proteins is more difficult and less defined. For proteins, the variety of structures is much more complicated and diverse than the RNA Turner model, requiring an initial step of constructing an abstract template for the structure. Moreover, for proteins, there is not a clear chemical basis for compensatory mutations [13], the energy models that define even β-strand pairings are more complex, and the larger residue alphabet vastly increases the complexity of the problem.

This class of problems is also different than any that have been attempted for structure analysis. The closest related structure-prediction methods rely on sequence profiles, as opposed to consensus folds. Current protein threading methods such as Raptor [14] often construct sequence profiles of the query sequence before threading it onto solved structures in the PDB; however, given two query sequences, even if they are functionally related, it will output two structure matches but does not try to form a consensus from these. There are β-structure specific methods that 'thread' a profile onto an abstract template representing a class of structures [15], [16], but do not generate consensus folds. Further, a new class of "ensemble" methods, e.g., partiFold TMB [17], "threads" a profile onto an abstract template, yet does not incorporate sequence alignment information nor generate consensus folds.

In this paper, we describe partiFold-Align, the first algorithm for simultaneous alignment and folding of pairs of unaligned protein sequences. Pairwise alignment is an important component in achieving reliable multiple alignments. Our strategy uses dynamic programming schemes to simultaneously enumerate the complete space of structures and sequence alignments and compute the optimal solution (as identified by a convex combination of ensemble-derived contact probabilities and sequence alignment matrices [18], [19], [20]). To overcome the intractability of this problem, we exploit sparsity in the set of likely amino acid pairings and aligned residues (inspired from the LocRNA algorithm [21]). partiFold-Align is thus able to achieve effectively cubic time and space in the length of the input sequences.

We demonstrate the efficacy of this approach by applying it to transmembrane β-barrel (TMB) proteins, one of the most difficult classes of proteins in terms of both sequence alignment and structure prediction [17], [16]. In tests on sequence alignments derived from structure alignments, we obtain significantly better pairwise sequence alignments, especially in the case of low homology. In tests comparing single-sequence versus consensus structure predictions, partifold-align obtains improved accuracy, considerably for cases where single-sequence results are poor. The methods we develop in this paper specifically target the difficult case of alignment of low homology sequences and aim to improve the accuracy of such alignments.

Contributions: The main contribution of this work is that we introduce the new concept of consensus folding of unaligned protein sequences. Our algorithm partiFold-Align is the first to perform simultaneous folding and alignment for protein sequences. We use this to provide better sequence alignments and structure predictions for the important and difficult TMB proteins, particularly in the case of low-homology. Given the broad generality of this approach and its proven impact on the RNA world, we hope that this will become a standard in protein structure prediction.

2. Approach

To design an algorithm for simultaneous alignment and folding we must overcome one fundamental problem: predicting a consensus fold (structure) of two unaligned protein sequences requires a correct sequence alignment on hand, however, the quality of any sequence alignment depends upon the underlying unknown structure of the proteins. We adopt our solution to this issue from the approach introduced by Sankoff [4] to solve this problem in the context of RNAs — by predicting *partial* structural information that is then aligned through a dynamic programming procedure.

For our consensus folding algorithm, we define this partial information using probabilistic contact maps (i.e., a matrix of amino acid pairs with a high likelihood of forming hydrogen bonding partners in a protein conformation), based on Boltzmann "ensemble" methods, which predict the likelihood of possible residue-residue interactions given all possible in-vivo protein conformations [16]. This is inspired by the recent LocARNA [21] algorithm, which improves upon Sankoff's through the use of such probabilistic contact maps. This technique is also somewhat related to the problem of *maximum contact map overlap* [22], although in such problems, contact maps implicitly signify the biochemical strength of a contact in a *solved* structured and not a well-distributed likelihood of interaction taken from a complete ensemble of possible structures.

Using such ensemble-based contact maps for simultaneous alignment and folding can be applied to other classes of proteins, however, in this work we describe our application to the class of transmembrane β-barrels. Unlike the RNA model used by Sankoff, TMB protein structure takes a complex form, with inclined, anti-parallel hydrogen-bonding β-strand forming a circular barrel structure, as depicted in Figure 1. Partitioning such diversity of structure presents an intractable problem, so we apply a fixed parameter approach to restrict structural elements such as β-strand length, coil size, and the amount of strand inclination to biologically meaningful sizes.

Broadly speaking, our simultaneous alignment and folding procedure begins by predicting the ensemble-based probabilistic contact map of two unaligned sequences through an algorithm extended from partiFold TMB [16]. Importantly, β-strand contacts below a parameterizable threshold are excluded to allow for

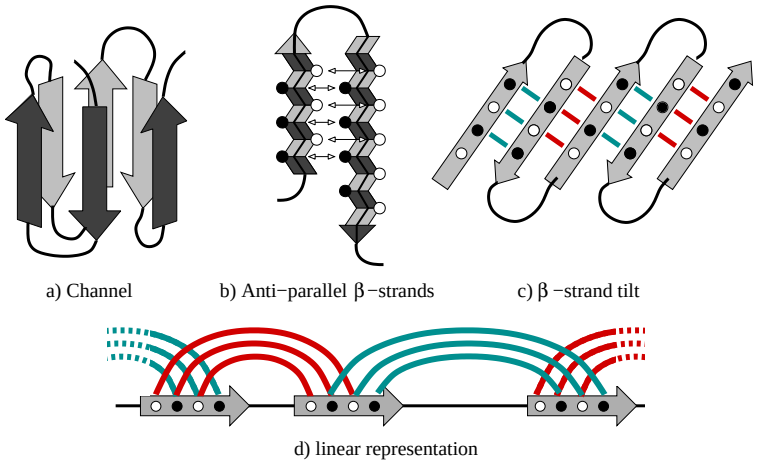


Fig. 1. Different structural elements of transmembrane β-barrels.

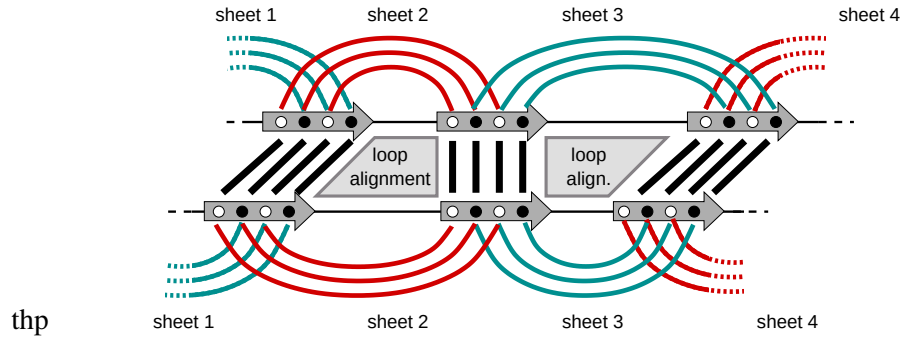


Fig. 2. Elements of TMB-alignment. The different coloring of amino acids in the sheet denotes the exposure to the membrane and to the channel, respectively. In a valid sheet alignment, only amino acids of the same type can be matched, whereas no further constraint (except length restriction) are applied to the loop alignment

an efficient alignment of the most likely interactions. Alignment is then broken into two structurally different parts: the alignment of β-sheets, and the alignment of coils (seen in Figure 2). Coil alignments can be performed independently at each position, however β-sheet alignments must respect residue pairs. Finally, to decompose the problem (Figure 3), we first consider the optimal alignment of a single β-sheet with a given inclination, including the enclosed coil alignment. For energetic considerations, we must note the orientation of the β-strand residues (core-facing or membrane-facing), as well as whether the coil extends into the extra-cellular or periplasmic side of the membrane. Once all single alignments have been found, we "chain" these subproblems to arrive at a single consensus alignment and structure.

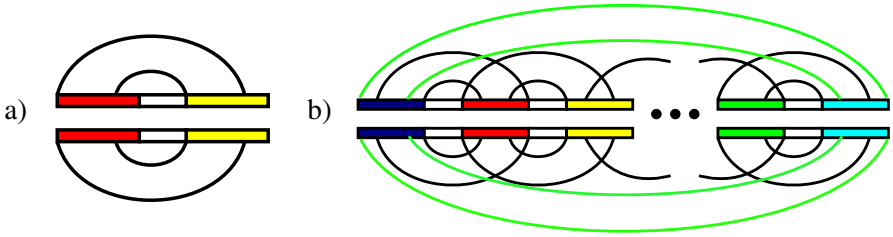


Fig. 3. Problem decomposition; a) alignment of a single sheet including the enclosed loop with positive shear; b) chaining of single sheet alignment to form a β-barrel. The sheet indicated with green lines connects the beginning and the end.

2.1. The TMB Alignment Problem

To formally describe this problem, we define an alignment $\mathcal{A}$ of two sequences a, b as a set of pairs $\{(p_1, p_2) \mid p_1 \in [1..|a|] \cup \{-\} \wedge p_2 \in [1..|b|] \cup \{-\}\}$ s.t. for all $(i,j),(i',j') \in (\mathcal{A} \cap [1..|a|] \times [1..|b|)$ we have $i < j \implies i' < j'$ (non-crossing) and there is no $i \in [1..|a|]$ (resp. $j \in [1..|b|]$) such that there are two different p, p' with $(i,p),(i,p') \in \mathcal{A}$ (resp. $(p,j),(p',j) \in \mathcal{A}$). Furthermore, for any position in both sequences, we must have an entry in $\mathcal{A}$. We say that $\mathcal{A}$ is a *partial alignment* if there are some positions for which there is no entry in $\mathcal{A}$. In this case, we denote with $\mathsf{def}(a, \mathcal{A})$ (resp. $\mathsf{def}(b, \mathcal{A})$) the set of positions in a (resp. b) for which an entry in $\mathcal{A}$ exists.

Now the result of the structure prediction is not a single structure, but a set of putative structural elements, namely the set of possible contacts for the β-strand. As indicated in Fig. 1, we have two different side chain orientations, namely to the channel (C) and to the membrane (M). Since contacts can form only if both amino acids share the same orientation, a *TMB probabilistic contact map* P *of a TMB* a is a matrix $P = (P(i,i',x))_{1 \le i <= i' \le |a|, x \in \{C,M\}}$ where $P(i,i',x) = P(i',i,x)$ and $\forall x \in \{C,M\}$: $\sum_i P(i,i',x) \le 1$. To overcome the intractability of this problem, we exploit sparsity in the set of likely amino acid pairings. Thus, we use only those entries in the matrix P which have a likelihood above a certain threshold.

We weight the alignments with a scoring function summing a folding energy term $\mathcal{E}$ with an alignment score $\mathcal{W}$, where the energy term $\mathcal{E}$ corresponds to the sum of the folding energies of the consensus

structure mapped on the two sequences. To allow a convex optimization of this function, we introduce a parameter α distributing the weights of the two terms. Then, given two sequences a, b, an alignment $\mathcal{A}$ and a consensus TMB structure $\mathcal{S}$ of length $|\mathcal{A}|$, the score of the alignment is:

$$\text{score}(\mathcal{A}, \mathcal{S}, a, b) = (1-\alpha) \cdot \mathcal{E}(\mathcal{A}, \mathcal{S}, a, b) + \alpha \cdot \mathcal{W}(\mathcal{A}, a, b) \tag{1}$$

Let $E_{ct}(x, y)$ be the energy value of a pairwise residue contact. Since by definition of the consensus structure these contacts are aligned, we define energy component as:

$$\mathcal{E}(\mathcal{A}, \mathcal{S}, a, b) = \sum_{\substack{\binom{i}{j} \in \mathcal{A}, \binom{i'}{j'} \in \mathcal{A} \\ (i,i') \in \mathcal{S}^{\text{arcs}_a}, (j,j') \in \mathcal{S}^{\text{arcs}_b}}} \tau(i, i', j, j'), \; where \; \tau(i, i', j, j') = E_{ct}(i, i') + E_{ct}(j, j') \tag{2}$$

In practice, we have implemented in partiFold-Align a stacking pair energy model as described in [17]. However, for the simplicity of the description, we use here only pairwise residue contact potentials.

Now, let $\sigma(x, y)$ be the substitution score of the amino acids x by y, and $g(x)$ an insertion/deletion cost. Then, the score of the sequence alignment component is given by:

$$\mathcal{W}(\mathcal{A}, a, b) = \sum_{\binom{i}{j} \in \mathcal{A}} \sigma(a_i, a_j) \; + \sum_{\binom{i}{-} \in \mathcal{A}} g(a_i) \; + \sum_{\binom{-}{j} \in \mathcal{A}} g(a_j) \tag{3}$$

Again, in practice, a penalty for opening gaps is added but not described here for the clarity of the discussion. Finally, the problem we solve in this paper is, given two sequences a and b:

$$\underset{\substack{\mathcal{A} \text{ TMB alignment of } a \text{ and } b, \\ \mathcal{S} \text{ TMB structure of length } |\mathcal{A}|}}{\arg\max} \text{score}(\mathcal{A}, \mathcal{S}, a, b).$$

To account for the side-chain orientation of residues in TM β-strands toward the channel or the membrane, the recursion equations require a slightly more detailed version of the scoring introduced in Eq. 2 and 3. An additional condition is that contacts only happen between amino acids with the same orientation, and that this orientation alternate between consecutive contacts. Hence, we introduce in τ an additional parameter *or* standing for this side-chain orientation feature. The same holds for the edit scores σ and g, where the orientation can also be the loop environment. For the strands, we use $\sigma_s(i, j, or)$, while for loops, we distinguish inner from outer loops (indicated by the loop type *lt*) and the amino acids in the loops are scored using $\sigma_l(i, j, lt)$. The gaps function is treated analogously.

2.2. Decomposition

We now define the dynamic programming tables used for the decomposition of our problem. The alignment of a single anti-parallel strand pair as shown in Fig. 3a) has nested arcs and an outdegree of at most one. We introduce for this configuration a table ShA$(\cdot)$ aligning pairs of subsequences $a_{i..i'}$ and $b_{j..j'}$. Another parameter to account for is the shear number which represents the inclination of the strands in the TM β-barrel. Since the strand pair alignments also include a loop alignment, and the scoring function of this loop depends on the loop type (inner/outer loop), we need to set the loop type as an additional parameter. Similarly, we need to know the orientation of the final contact to ensure the succession of channel and membrane orientation. Given an orientation of a contact *or*, the term next$_c$(*or*) return the orientation of the following contact. Thus, we have a table ShA$(i, i'; j, j'; or; lt; s)$ (where ShA stands for *sheet alignment*) with the following recursion:

$$\text{ShA}(i, i'; j, j; or; lt; s) = \max \begin{cases} \textit{ShAgap}(i, i'; j, j; or; lt; s) & \\ \textit{ShAshear}(i, i'; j, j; or; lt; s) & \text{if } s \neq 0 \\ \textit{ShAcontact}(i, i'; j, j; or; lt) & \text{if } s = 0 \\ \text{LA}(i, i'; j, j; lt) & \text{if } s = 0 \end{cases} \tag{4}$$

where

$$\begin{aligned}
\textit{ShAcontact}(i,i';j,j;\textit{or};\textit{lt}) = \;& \mathrm{ShA}(i+1,i'-1;j+1,j'-1;\mathrm{next}_c(\textit{or});\textit{lt};0)\\
&+\tau(i,i';j,j';\textit{or})+\sigma_\mathrm{s}(a_i,b_j,\textit{or})+\sigma_\mathrm{s}(a_{i'},b_{j'},\textit{or})
\end{aligned}$$

$$\textit{ShAgap}(i,i';j,j;\textit{or};\textit{lt};s) = \max\begin{cases}\mathrm{ShA}(i+1,i';j,j;\textit{or};\textit{lt};s)+g_\mathrm{s}(a_i,\textit{or})\\ \mathrm{ShA}(i,i'-1;j,j;\textit{or};\textit{lt};s)+g_\mathrm{s}(a_{i'},\textit{or})\\ \mathrm{ShA}(i,i';j+1,j;\textit{or};\textit{lt};s)+g_\mathrm{s}(b_j,\textit{or})\\ \mathrm{ShA}(i,i';j,j-1;\textit{or};\textit{lt};s)+g_\mathrm{s}(b_{j'},\textit{or})\end{cases}$$

$$\textit{ShAshear}(i,i';j,j;\textit{or};\textit{lt};s) = \max\begin{cases}\mathrm{ShA}(i+1,i';j+1,j';\textit{or};\textit{lt};s+1)\\ \quad+\sigma_\mathrm{s}(a_i,b_j,\textit{or}) & \text{if } s<0\\ \mathrm{ShA}(i,i'-1;j,j'-1;\textit{or};\textit{lt};s-1)\\ \quad+\sigma_\mathrm{s}(a_{i'},b_{j'},\textit{or}) & \text{if } s>0\end{cases}$$

ShAgap, *ShAcontact* and *ShAshear* are just introduced for better readability and will not be tabulated. The matrix $\mathrm{LA}(i,i';j,j';\textit{lt})$ is the alignment of $a_{i..i'}$ with $b_{j..j'}$ as inner or outer loop of a sheet (as indicated by the loop type *lt*). This table can be calculated using the usual sequence alignment recursion on the right ends i',j', starting with each left end position pair i,j in the sequence. Thus, we have

$$\mathrm{LA}(i,i';j,j';\textit{lt}) = \begin{cases}\mathrm{LA}(i,i'-1;j,j';\textit{lt})+g_\mathrm{l}(a_{i'},\textit{lt})\\ \mathrm{LA}(i,i';j,j'-1;\textit{lt})+g_\mathrm{l}(b_{j'},\textit{lt})\\ \mathrm{LA}(i,i'-1;j,j'-1;\textit{lt})+\sigma_\mathrm{l}(a_{i'},b_{j'},\textit{lt})\end{cases} \tag{5}$$

As we have already mentioned in the definition of a contact map, we use a threshold on the probability to reduce both space and time complexity of the alignment problem, in a similar way as is done in the LocARNA-approach [21]. Hence, we will calculate and tabulate only values for the ShA-matrix for those positions i,i' and j,j' where there are contacts above the threshold in both sequences connecting between positions in a narrow range *r* of i,i' and j,j'. I.e., we consider only those i,i' where there is a contact $(p,p')\in P$ above the threshold s.t. $|i-p|\le r$ and $|i'-p'|\le r$, and analogously for j,j'. We write $\mathrm{ShA}(i,i';j,j';\textit{lt};s)\downarrow$ if $i,i';j,j'$ for which an entry is calculated. Similarly, only related values for $\mathrm{LA}(i,i';j,j';\textit{lt})$ will be tabulated with the exception of values $\mathrm{LA}(i,|a|;j,|b|;\textit{lt})$, which will be needed for the chaining. Again $\mathrm{LA}(i,i';j,j';\textit{lt})\downarrow$ indicates an defined value.

2.3. Chaining

The next problem is to chain the different single sheet alignments, as indicated by the Fig. 3b). To build a valid overall alignment, we have to guarantee that the sub-alignments agree on the overlapping part. A *strand alignment* $\mathcal{A}_\mathrm{s}$ is just a partial alignment. The solution is to extend the matrices for sheet alignments by an additional entry for the alignment of the strand regions. Albeit there are exponentially many alignments in general, there are several restrictions on the set of allowed alignments since they are alignments of strand regions. In the case of TMB-barrels, we assume no strand bulges since they are a rare event. Hence, one can insert or delete only a complete contact instead of a single amino acid. When chaining the sheet alignments, the gap in one strand is then (by the agreement of sub-alignment) transferred to the chained sheet.

The first step is to extend the matrices for the sheet alignment by a parameter for the strand alignments. Since every entry in the recursion scheme corresponds to an alignment operation, we only have to check for the compatibility of the operation with the alignment edge. Note that although the alignment is fixed for the strands of a sheet, the scoring is not since we could still differentiate between a match of two bases or a match of a contact. Thus, the new matrix is $\mathrm{ShA}(i,i';j,j';\textit{or};\textit{lt};s;\mathcal{A}_\mathrm{s})$, where we enforce $\mathcal{A}_\mathrm{s}$ to satisfy $\mathsf{def}(a,\mathcal{A}_\mathrm{s}) = [i..l_1]\cup[r_1..i']$ and $\mathsf{def}(b,\mathcal{A}_\mathrm{s}) = [j..l_2]\cup[r_2..j']$ for some $i<l_1<r_1<i'$ and

$j < l_2 < r_1 < j'$. The new version of Eq. 4 is simply

$$\mathrm{ShA}(i,i';j,j';or;lt;s;\mathcal{A}_\mathrm{s}) = \max \begin{cases} \mathit{ShAgap}(i,i';j,j';or;lt;s;\mathcal{A}_\mathrm{s}) & \\ \mathit{ShAshear}(i,i';j,j';or;lt;s;\mathcal{A}_\mathrm{s}) & \text{if } s \neq 0 \\ \mathit{ShAcontact}(i,i';j,j';or;lt;\mathcal{A}_\mathrm{s}) & \text{if } s = 0 \\ \mathrm{LA}(i,i';j,j';lt) & \text{if } s = 0 \end{cases} \tag{5}$$

$\mathrm{LA}(i,i';j,j';lt)$ does not get the additional parameter since the agreement for sub-alignment in chaining is restricted to the strand. For *ShAgap*, *ShAcontact* and *ShAshear*, we have now to check whether the associated alignment operations are compatible with $\mathcal{A}_\mathrm{s}$. Thus, the new definition of *ShAcontact* is

$$\mathit{ShAcontact}(i,i';j,j';or;lt;\mathcal{A}_\mathrm{s}) = \max \begin{cases} \mathrm{ShA}(i+1,i'-1;j+1,j'-1;or;lt;0;\mathcal{A}_\mathrm{s}) & \text{if } (i,j) \in \mathcal{A}_\mathrm{s} \\ \quad + \tau(i,i';j,j';or) + \sigma_\mathrm{s}(a_{i'},b_{j'},or) & \text{and } (i',j') \in \mathcal{A}_\mathrm{s} \\ -\infty & \text{else} \end{cases}$$

If all entries are incompatible with $\mathcal{A}_\mathrm{s}$, then $-\infty$ is returned. Note that we add an amino acid match score only for a single specified end of the contact. Thus, $\sigma_\mathrm{s}(a_i,b_j)$ is skipped. The reason is simply that otherwise, we would add this score twice in course of chaining. The new definition of *ShAshear* is then

$$\mathit{ShAshear}(i,i';j,j;or;lt;s,\mathcal{A}_\mathrm{s}) = \max \begin{cases} \mathrm{ShA}(i+1,i';j+1,j';or;lt;s+1;\mathcal{A}_\mathrm{s}) & \text{if } s < 0 \wedge (i,j) \in \mathcal{A}_\mathrm{s} \\ \mathrm{ShA}(i,i'-1;j,j'-1;or;lt;s-1;\mathcal{A}_\mathrm{s}) & \text{if } s > 0 \wedge (i',j') \in \mathcal{A}_\mathrm{s} \\ \quad + \sigma_s(a_{i'},b_{j'},or) & \end{cases}$$

The new variant of *ShAgap*() is defined analogously. Now we can define the matrix $Dchain()$ for chaining the strand pair alignments. At the end of its construction, the sheet is closed by pairing its first and last strands to create the barrel. To process this construction, we need to keep track of the leftmost and rightmost strand alignments $\mathcal{A}_s^{\text{chain}}$ and $\mathcal{A}_s^{\text{cyc}}$ of the sheet. We add to these two parameters, a variable *ct* used to determine in the closing strand pair has been added or not. Here, $ct = c$ means that the sheet is not closed while $ct = l_f$ indicates that the barrel has been built. Finally, to control the number of strand in the barrel, we add another variable nos storing the number of strands in the sheet.

We first initialize the array $Dchain$ for every i,j and any strand alignment $\mathcal{A}_s^{\text{cyc}}$ of the initial strand where $\mathsf{def}(a,\mathcal{A}_s^{\text{cyc}}) = [i..i']$ and $\mathsf{def}(b,\mathcal{A}_s^{\text{cyc}}) = [j..j']$. This stands for the alignment of the C-terminal sequences. Then

$$Dchain(i,j;\mathcal{A}_s^{\text{cyc}};\mathcal{A}_s^{\text{cyc}};c;lt;1) = \mathrm{LA}(i',|a|;j',|b|;lt;1),$$

where *lt* is an allowed loop-type for this sheet. Note that the strand alignment is not yet scored. This will be done at the next step when chaining the first sheet. We introduce for this purpose a function $\mathrm{ShA}(\mathcal{A},nos)$ returning the cost of the alignment if the rightmost strands if $nos = 2$ (which indicates to the first strand pair of the sheet) and 0 otherwise. A function $prev()$ returning the previous loop type is also used to alternates the loop environment on both sides of the membrane. In addition, given two strand alignments $\mathcal{A}_\mathrm{s}, \mathcal{A}'_\mathrm{s}$, we say that $\mathcal{A}_\mathrm{s}, \mathcal{A}'_\mathrm{s}$ agree on the strands $i..i'$ in the first and $j..j'$ in the second sequence (written $agr(\mathcal{A}'_\mathrm{s};\mathcal{A}_\mathrm{s};i,i';j,j')$ if $\mathcal{A}_\mathrm{s} \cup \mathcal{A}'_\mathrm{s}$ is an partial alignment. With these notations, the recursion used to build the unclosed sheets is:

$$Dchain(i,j;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};c;lt;nos) = \tag{6}$$
$$\max_{\substack{i',j',\mathcal{A}'_{\rm s},s,lt',or \\ \text{with ShA}(i,i';j,j';lt';s;\mathcal{A}'_{\rm s})\downarrow, \\ \mathsf{def}(a,\mathcal{A}_{\rm s})=[i..l_1]\cup[r_1..i'], \\ \mathsf{def}(b,\mathcal{A}_{\rm s})=[j..l_2]\cup[r_2..j'], \\ \text{and } agr(\mathcal{A}'_{\rm s};\mathcal{A}_{\rm s};i,l;j,l')}} \left(\begin{array}{l} \text{ShA}(i',i;j',j;or;lt';s;\mathcal{A}'_{\rm s}) \\ \quad + \ Dchain(r_1,r_2;\mathcal{A}'_{\rm s};\mathcal{A}_s^{\rm cyc};c;prev(lt);nos-1) \\ \quad + \ strandal(\mathcal{A}'_{\rm s},nos) \end{array} \right)$$

We conclude this section by giving the recursions used to close the barrel and perform the sequence alignment of the N-terminal sequences. Since the anti-parallel or parallel nature of the closing strand pair depends of the number of strands in the barrel, we introduce here a function *ShAclose* which returns the folding energy of the parallel strand pairings of the leftmost and rightmost strands of the sheet if the number of strands nos is odd, and folding energy of the anti-parallel strand pairings if nos is even.

$$Dchain(i,j;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};l_f;lt) = \tag{7}$$
$$\max \begin{cases} \max \begin{cases} Dchain(i+1,j;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};l_f;lt)+g_l(a_i,lt) \\ Dchain(i,j+1;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};l_f;lt)+g_l(b_j,lt) \\ Dchain(i+1,j+1;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};l_f;lt)+\sigma_l(a_i,b_j,lt) \end{cases} \\ \max\limits_{i',j',or,nos} \begin{cases} Dchain(i,i';\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};c;lt) \\ \quad + \ \textit{ShAclose}(i,i';j,j';or;s;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};dir(nos)) \end{cases} \end{cases}$$

The final value of the consensus folding problem is then found in $Dchain(1,1;\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};l_f;lt)$ for some *lt* and $\mathcal{A}_{\rm s},\mathcal{A}_s^{\rm cyc}$ with $agr(\mathcal{A}_{\rm s};\mathcal{A}_s^{\rm cyc};1,i;1,j)$, where $\mathsf{def}(a,\mathcal{A}_{\rm s})=[1..i]\cup[r..i']$ and $\mathsf{def}(b,\mathcal{A}_{\rm s})=[1..j]\cup[r..j']$. The solutions are built using the classical backtracking procedures.

Here, Equations 6 and 7 assumes that the strand inclination modeled using the shear number s for successive strand pairs are independent. However, in practice this parameter must be used to determine when a strand pair can be concatenated at the end of an existing sheet to ensure the coherency of the barrel structure and conserve a constant inclination of the strands (see Fig. 1).

3. Results

Here we demonstrate the benefits of the partiFold-Align algorithm when applied to the problems of pairwise sequence alignment and structure prediction of transmembrane β-barrel proteins. Our sequence alignment performance greatly improves upon comparable alignment techniques, and surpasses state-of-the-art alignment tools (which use additional algorithmic filters) in the case of low homology sequences. It is also shown that a partiFold-Align consensus fold can better predict secondary structure when aligning proteins within the same superfamily. We begin with a description of our test dataset and scoring metrics applied as well as the partiFold-Align parameters chosen for the analysis, followed by our specific sequence alignment and structure prediction results.

3.1. Dataset and evaluation technique

By implementing our algorithmic framework to align and fold transmembrane β-barrels, we highlight how this approach can significantly improve the alignment accuracy of protein classes with which current alignment tools have difficulty. Specifically, few TMB structures have been solved through X-ray crystallography or NMR (less-than 20 non-homologous to-date), and often known TMB sequences exhibit very low sequence homology (e.g. less-than 20%), despite sharing structure and function.

To judge how well partiFold-Align aligns proteins in this difficult class, we select 13 proteins from five superfamilies of TMBs found in the Orientation of Proteins in Membranes (OPM) database [23]

(using the OPM database definition of "class," "superfamily," and "family"). This constitutes all solved TMB proteins with a single, transmembrane, β-barrel domain, and excludes proteins with significant extracellular or periplasmic structure and limits the sequence length to a computationally-tractable maximum of ~ 300 residues. With the assumption that structural alignment best mimics the intended goal of identifying evolutionary and functional similarities, we perform structural alignments between all pairs of proteins within large superfamilies, and across smaller superfamilies (28 alignments, see Appendix for illustration of breakdown), and consider this the "correct" pairwise alignment. For structural alignments, the *Matt* [24] algorithm is used, which has demonstrated state-of-the-art structural alignment accuracy. Resulting alignments can then be sorted by relative sequence identity [1] (assuming the *Matt* alignment) [25], [26].

Our partiFold-Align alignments are then compared against structural alignments using the Q_{Cline} [27], [28] scoring metric, restricted to transmembrane regions as defined by the OPM (since structural predictions in the algorithm only contribute to transmembrane β-strand alignments; coils are effectively aligned on sequence-alone). Q_{Cline} can be considered a percentage accuracy, and resembles the simplistic $Q_{combined}$ score [2], measuring combined under- and over-prediction of aligned pairs, but more fairly accounts for off-by-n alignments. Such shifts often occur from energetically-favorable off-by-n β-strand pairings that remain good alignments. The Q_{Cline} parameter ϵ is chosen to be 0.2, which allows alignments displaced by up to five residues to contribute (proportionally) toward the total accuracy. The higher the Q_{Cline} score, the more closely the alignments match (ranging $[-\epsilon, 1]$).

To judge the accuracy of a partiFold-Align consensus structure against a structure predicted from single-sequence alone, we test against the same OPM database proteins described above. For all 13 proteins, a structure prediction is computed using the exact same ensemble structure prediction methodology as in partiFold-Align, only applied to a single sequence. The transmembrane-region Q_2 secondary structure prediction score between the predicted structures and the solved PDB structure (annotated by STRIDE [29]) can then be computed; where $Q_2 = (TP + TN)/(sequence\ length)$.

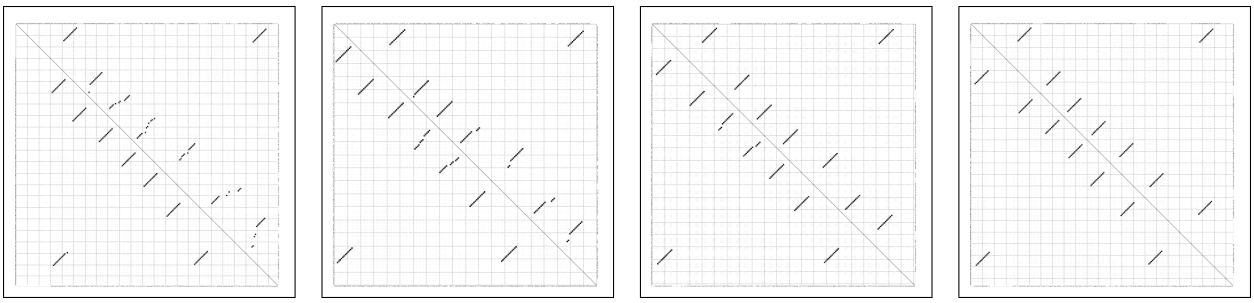
3.2. Model parameter selection

For our analyses, parameters must be chosen for the abstract structural template defined in Section 2. For transmembrane β-barrels, the choice of allowable (minimum and maximum) β-strand and coil region lengths, as well as shear numbers can be assigned based on biological quantities such as membrane thickness, etc. (Even in the absence of all other information, the sequence length alone of a putative transmembrane β-barrel can suggest acceptable ranges.) Other algorithmic parameters, such as the pairwise contact threshold (which filters which β-strand pairs are used in the alignment), the Boltzmann Z constant (found within $E_{ct}(\cdot)$ in Equation 2, effecting the structural energy score [17]), the gap penalty, the choice of substitution matrix, and the "α" balance parameter require selection without as clear a biological interpretation.

For results presented in this paper, one of three sets of structural parameters were chosen according to protein superfamily, with a fairly wide range of values permitted. To determine the algorithmic parameters listed above in a principled manner, we chose a single set of algorithmic parameters for all alignments, with the exception of varying the β-strand pair probability "threshold" used in the initial step of the algorithm, and the α score-balancing parameter. In all cases, our choices are made obliviously to the known structures in our testing sets. Interestingly, the substitution matrix we use is a combination of the BATMAS [18] matrix for transmembrane regions, and BLOSUM [19] for coils. For alignments with a sequence homology *below* 10%, we chose a higher probability threshold value ($1x10^{-5}$ versus $1x10^{-10}$) to restrict alignments to highly-likely β-strand pairs, to reduce signal degradation from low-likelihood β-strand pairs with very distant sequence similarities. For alignments with sequence homology *below*

[1] $Sequence\ Identity\ \% = \frac{Identical\ positions}{aligned\ positions + internal\ gap\ positions}$

[2] $Q_{combined} = \frac{\#\ correct\ pairs}{\#\ unique\ pairs\ in\ sequence\ \&\ structure\ alignments}$



(a) α=1.0 (b) α=0.66 (c) α=0.33 (d) α=0.0

Fig. 4. Stochastic contact maps from a partiFold-Align run on the proteins 1BXW and 2F1V. For each of the four plots, the sequence of 1BXW and 2F1V is given on the axes (with gaps), and high probability residue-residue interactions indicated for 1BXW on the lower left half of the graph and 2F1V on the upper right half (i.e., the single-sequence probabilistic contact maps). Structural contact map alignment can be judged by how well the the plot is mirrored across the diagonal. Subfigure (a) ($\alpha = 1.0$) shows an alignment which ignores the contribution of the structural contact map, while (d) ($\alpha = 0.0$) shows an alignment wholly-dependent on the structural contact map, and ignorant of sequence alignment information.

10%, we chose a lower α parameter (0.6 versus 0.7) to boost the contribution of the structural prediction to the overall solution when less sequence homology could be exploited. As seen in Figure 4, consensus predictions from lower α parameters more closely resemble predictions based solely on structural scores, and thus, an optimal alignment should correlate α with sequence homology.

Admittedly, this naive, single (or few) parameter solution does not enable the full potential of our algorithm. A protein-specific machine learning approach would allow for a better parameter fit, and is the focus of ongoing research.

3.3. Alignment accuracy of low sequence identity TMBs

To compare the accuracy of alignments generated by partiFold-Align against current sequence alignment algorithms, we perform the same TMB pairwise sequence alignments using partiFold-Align, EMBOSS (Needleman-Wunsch) [20], and MUSCLE [30], [31]. EMBOSS may be considered the best Needleman-Wunsch style global sequence alignment algorithm (a straight-forward, widely applicable method of alignment), while MUSCLE is widely thought *the* state-of-the-art alignment tool, though it also incorporates a more sophisticated "k-tuple" selection method during pairwise alignments which can improve alignment accuracy under some circumstances. [3] Since the partiFold-Align algorithm utilizes Needleman-Wunsch style dynamic programming, comparisons between EMBOSS and partiFold-Align represent a fair analysis of what simultaneous folding and alignment algorithms specifically contribute to the problem. Comparisons with MUSCLE alignment scores necessitate inclusion to portray the practical benefits partiFold-Align provides. However, no technical reason prevents MUSCLE's k-tuple methods to be incorporated with partiFold-Align; this stands as future work.

Figure 5 presents transmembrane Q_{Cline} accuracy scores for EMBOSS, MUSCLE, and partiFold-Align across 27 TMB pairwise alignments. (The absent $28^{\mathbf{th}}$ alignment, between 1BXW and 2JMM (50% sequence-homologous), is aligned with a nearly-perfect Q_{Cline} score of 0.98 by all three algorithms). Results are separated into the 3 categories according to the number of circling strands within a protein's β-barrel: seven 8-stranded "OMPA-like" proteins account for 21 alignments, two 10-stranded "OMPT-like" proteins account for one alignment, and finally, four 12-stranded "Autotransporters," "OM phospholipases," and "Nucleoside-specific porins" make up the final six alignments (a full summary can be found in Table II of the Appendix). Equal-sized clusters of pairwise alignments are then formed and ordered according to sequence identity, with cluster mean Q_{Cline} and standard deviation reported. All

[3]We note, that while EMBOSS uses only the BLOSUM substitution matrix, and partiFold-Align a combination of BATMAS and BLOSUM, Forrest et al. [3] show that BATMAS-style matrices do not show improvement for EMBOSS-style algorithms.

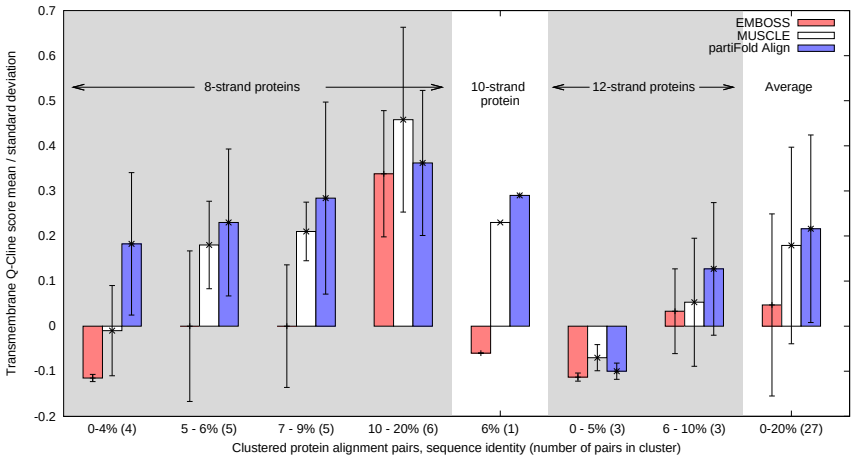


Fig. 5. Mean and standard deviation Q_{Cline} scores for 8-, 10-, and 12- stranded TMBs. Each of the 3 categories of proteins are clustered and ordered according to sequence identity, with the number of alignments in each cluster in parentheses. Note: By definition, Q_{Cline} scores range between $-\epsilon$ and 1.0, where $\epsilon = 0.2$; negative values indicate very poor alignments.

individual alignment-pair statistics, as well as alternative accuracy metrics (e.g. $Q_{combined}$) can be found in the Appendix.

Across all TMBs, partiFold-Align alignments are more accurate than EMBOSS alignments by an average Q_{Cline} of 16.9% (4.5x). Most importantly, partiFold-Align significantly improves upon the EMBOSS Q_{Cline} score for all alignments with a sequence identity lower than 9% (by a Q_{Cline} average of 28%), and roughly matches or improves $24/28$ alignments overall. Excluding the 12-strand alignments, which align proteins across different superfamilies, our intra-superfamily alignments exhibit even higher improvements in average Q_{Cline}, besting EMBOSS by 20.3% (27.4% versus 7.1%). Even compared with MUSCLE alignments, partiFold-Align is able to achieve a 4% increased Q_{Cline} on average, despite its lack of the k-tuple method employed by MUSCLE.

3.4. Secondary structure prediction accuracy of consensus folds

Here we investigate how the consensus structure resulting from our simultaneous alignment and folding algorithm can improve structure prediction accuracy over a prediction computed from a single sequence alone. We report in Table I Q_2 accuracies computed from alignments of all pairs of TMB sequences within the same n-stranded category. For each protein, the Q_2 score from the single sequence minimum folding energy (m.f.e.) structure is given (as done in [16]), and compared against: the Q_2 score from the best alignment partner, and the average Q_2 score obtained when aligning that protein with all others in its category.

The results for 8- and 10-stranded categories show a clear improvement (more than 8%) by the best consensus fold in $6/9$ instances (1P4T, 2F1V, 1THQ, 2ERV, 1K24, 1I78), and roughly equivalent results for the remaining 3 (2F1V, 1K24, I178). Further, on average, nearly all proteins show equivalent or improved scores when aligned with any other protein, with the exception of 1BXW. However, the single sequence structure prediction Q_2 for 1BXW is not only high, but significantly higher than all other 8-stranded proteins; the contact maps of any other aligning partner may simply add noise, diluting accuracy. Conversely, the proteins which have poor single sequence structure prediction benefit the greatest from alignment (e.g. 2F1V). This relationship is certainly not unidirectional, though, as we see that the consensus fold of 1K24 and 1I78 improves upon both proteins' single sequence structure prediction.

In contrast, the results compiled on the 12-strands category do not show any clear change in the secondary structure accuracy. However, recalling that this category covers 3 distinct superfamilies in

Category	PDB id	single seq.	consensus	
			best	average
8-stranded	1BXW	72	70(-2)	63(-9)
	1P4T	60	68(+8)	58(-2)
	1QJ8	65	68(+3)	66(+1)
	2F1V	47	63(+22)	62(+15)
	1THQ	50	69(+13)	52(+2)
	2ERV	57	67(+10)	59(+2)
	2JMM	62	65(+3)	62(+0)
10-stranded	1K24	60	69(+9)	69(+9)
	1I78	76	83(+7)	83(+7)
12-stranded	1QD6	54	61(+7)	56(+2)
	1TLY	59	59(+0)	58(-1)
	1UYN	56	56(+0)	53(-3)
	2QOM	51	55(+4)	53(+2)

TABLE I

Secondary structure assignment accuracy. Percentage Q_2 of secondary structure prediction correctly assigned residues (transmembrane and non-transmembrane regions). Third column reports the performance of a single strand folding (no alignments). Fourth and fifth columns report respectively the best and the average Q_2 scores of a consensus structure.

the OPM database, such results may make sense. The "Autotransporter," "OM phospholipase," and "Nucleoside-specific porin" families all exhibit reasonably different structures, and perform quite unrelated tasks. Further, unlike the original partiFold TMB algorithm [17], the abstract structural template used in this work does not take into account β-strands that extend far beyond the cell membrane (since our alignments focus on membrane regions). This may also effect the structure prediction accuracy of more complex TMBs.

we can conclude from this benchmark that the consensus folding approach can be used to improve the structure prediction of low homology sequences, provided they both belong to the same superfamily. However, we emphasize the importance parameter selection may play in these results; a different parameter selection method may enable accuracy improvement for higher-level classes of proteins.

4. Conclusions

We have presented partiFold-Align, a new approach to the analysis of proteins, which simultaneously aligns and folds pairs of unaligned protein sequences into a consensus to achieve both improved sequence alignment and structure prediction accuracy. To demonstrate the efficacy of this approach, we designed and tested the algorithm for the difficult class of transmembrane β-barrel, low sequence homology proteins. However, we believe this technique to be generally applicable to many classes of proteins where the structure can be defined through a "chaining" procedure as described in Section 2 (e.g., most β-sheet structures). This could open new areas of analysis that were previously unattainable given current tools' poor ability to construct functional alignments on low sequence homology proteins.

While we have shown that consensus folds can significantly improve upon pairwise sequence alignment, we believe this approach can also translate into considerable improvements in multiple sequence alignments. This is because many multiple alignment procedures use pairwise alignment information at their core [27]. Such an extension would be an obvious next step for our approach to be added in combination with other, more elaborate techniques found in sequence alignment algorithms (e.g., MUSCLE).

Similarly, we believe that the effectiveness of partiFold-Align can be enhanced significantly by a well-formulated machine learning approach to parameter optimization as has been applied to the case of RNA [5], [32]. Supporting this notion, we experimented with parameters selected based on a known test set, and saw pairwise sequence alignment accuracies with an average Q_2 accuracy $\sim 20\%$ greater than MUSCLE (versus the reported $\sim 4\%$ improvement for test-set blind parameter selections). However, for the case of TMBs, one notable problem that would need to be overcome is the relatively small set of know structure or alignments with which to use for training.

REFERENCES

[1] Shakhnovich, B.E., Deeds, E., Delisi, C., Shakhnovich, E.: Protein structure and evolutionary history determine sequence space topology. Genome Res **15**(3) (2005 Mar) 385–392

[2] Edgar, R.C., Batzoglou, S.: Multiple sequence alignment. Curr Opin Struct Biol **16**(3) (2006 Jun) 368–373

[3] Forrest, L.R., Tang, C.L., Honig, B.: On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. Biophys J **91**(2) (2006 Jul 15) 508–517

[4] Sankoff, D.: Simultaneous solution of the rna folding, alignment and protosequence problems. SIAM J. Comput. **45**(5) (1985) 810–825

[5] Do, C.B., Foo, C.S., Batzoglou, S.: A max-margin model for efficient simultaneous alignment and folding of rna sequences. Bioinformatics **24** (2008) i68–0i76

[6] Hofacker, I.L., Bernhart, S.H.F., Stadler, P.F.: Alignment of rna base pairing probability matrices. Bioinformatics **20**(14) (2004 Sep 22) 2222–2227

[7] Mathews, D.H., Turner, D.H.: Dynalign: an algorithm for finding the secondary structure common to two rna sequences. J Mol Biol **317**(2) (2002 Mar 22) 191–203

[8] Harmanci, A.O., Sharma, G., Mathews, D.H.: Efficient pairwise rna structure prediction using probabilistic alignment constraints in dynalign. BMC Bioinformatics **8** (2007) 130

[9] Dowell, R.D., Eddy, S.R.: Efficient pairwise rna structure prediction and alignment using sequence alignment constraints. BMC Bioinformatics **7** (2006) 400

[10] Havgaard, J.H., Torarinsson, E., Gorodkin, J.: Fast pairwise structural rna alignments by pruning of the dynamical programming matrix. PLoS Comput Biol **3**(10) (2007 Oct) 1896–1908

[11] Gorodkin, J., Heyer, L.J., Stormo, G.D.: Finding the most significant common sequence and structure motifs in a set of rna sequences. Nucleic Acids Res **25**(18) (1997 Sep 15) 3724–3732

[12] Backofen, R., Will, S.: Local sequence-structure motifs in rna. J Bioinform Comput Biol **2**(4) (2004 Dec) 681–698

[13] Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. Proteins **Suppl 5** (2001) 157–162

[14] Xu, J., Li, M., Kim, D., Xu, Y.: RAPTOR: Optimal protein threading by linear programming. Journal of Bioinformatics and Computational Biology (JBCB) (2003)

[15] Bradley, P., Cowen, L., Menke, M., King, J., Berger, B.: Betawrap: Successful prediction of parallel beta-helices from primary sequence reveals an association with many microbial pathogens. Proceedings of the National Academy of Sciences **98**(26) (2001) 14819–14824

[16] Waldispuhl, J., Berger, B., Clote, P., Steyaert, J.M.: Predicting transmembrane beta-barrels and interstrand residue interactions from sequence. Proteins **65**(1) (2006 Oct 1) 61–74

[17] Waldispuhl, J., O'Donnell, C.W., Devadas, S., Clote, P., Berger, B.: Modeling ensembles of transmembrane beta-barrel proteins. Proteins **71**(3) (2008 May 15) 1097–1112

[18] Sutormin, R.A., Rakhmaninova, A.B., Gelfand, M.S.: Batmas30: amino acid substitution matrix for alignment of bacterial transporters. Proteins **51**(1) (2003 Apr 1) 85–95

[19] Henikoff, S., Henikoff, J.: Amino acid substitution matrices from protein blocks. PNAS **89** (1992) 10915–10919

[20] Rice, P., Longden, I., Bleasby, A.: Emboss: the european molecular biology open software suite. Trends Genet **16**(6) (2000 Jun) 276–277

[21] Will, S., Reiche, K., Hofacker, I.L., Stadler, P.F., Backofen, R.: Inferring noncoding rna families and classes by means of genome-scale structure-based clustering. PLoS Comput Biol **3**(4) (2007 Apr 13) e65

[22] Caprara, A., Carr, R., Istrail, S., Lancia, G., Walenz, B.: 1001 optimal pdb structure alignments: integer programming methods for finding the maximum contact map overlap. J Comput Biol **11**(1) (2004) 27–52

[23] Lomize, M., Lomize, A., Pogozheva, I., Mosberg, H.: OPM: Orientations of Proteins in Membranes database. Bioinformatics **22** (2006) 623–625

[24] Cowen, L., Menke, M., Berger, B.: Matt: Local Flexibility Aids Protein Multiple Structure Alignment. PLoS Comp Bio **4** (2008)

[25] Doolittle, R.: Similar amino acid sequences: chance or common ancestry? Science **214** (1981) 149–159

[26] Raghava, G., Barton, G.: Quantification of the variation in percentage identity for protein sequence alignments. BMC Bioinformatics **7** (2006) 415

[27] Dunbrack, R.L.J.: Sequence comparison and protein structure prediction. Curr Opin Struct Biol **16**(3) (2006 Jun) 374–384

[28] Cline, M., Hughey, R., Karplus, K.: Predicting reliable regions in protein sequence alignments. Bioinformatics **18**(2) (2002 Feb) 306–314

[29] Frishman, D., P., A.: Knowledge-based protein secondary structure assignment. Proteins **23** (1995) 566–579

[30] Edgar, R.C.: Muscle: multiple sequence alignment with high accuracy and high throughput. NAR **32**(5) (2004) 1792–1797

[31] Edgar, R.C.: Muscle: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics **5** (2004 Aug 19) 113

[32] Do, C.B., Woods, D.A., Batzoglou, S.: CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics **22**(14) (2006) e90–8

[33] Bompfunewerer, A.F., Backofen, R., Bernhart, S.H., Hertel, J., Hofacker, I.L., Stadler, P.F., Will, S.: Variations on RNA folding and alignment: lessons from Benasque. Journal of Mathematical Biology **56**(1-2) (2008) 129–144

APPENDIX

To elaborate upon our analysis of the partiFold-Align algorithm, we have included this Appendix which gives results for every individual pairwise alignment. Figure 6 presents all 28 TMB pairwise alignments across all 3 classes of proteins, and their corresponding Transmembrane Q_{Cline} and $Q_{combined}$ score. For this, we see that the general trends discussed in Section 3 apply to $Q_{combined}$just as much as Q_{Cline}. Figure presents all 28 TMB pairwise alignments and their corresponding whole-protein Q_{Cline} and $Q_{combined}$ scores (not restricted to transmembrane regions as discussed in Section 3.1) Again, the same trends apply. Table II gives a list of all 28 OPM database TMB pairwise alignments, their corresponding sequence identities, and their subfamily classifications. Finally, for completeness, we include here a complexity analysis of the simultaneous alignment and folding algorithm described in Section 2.

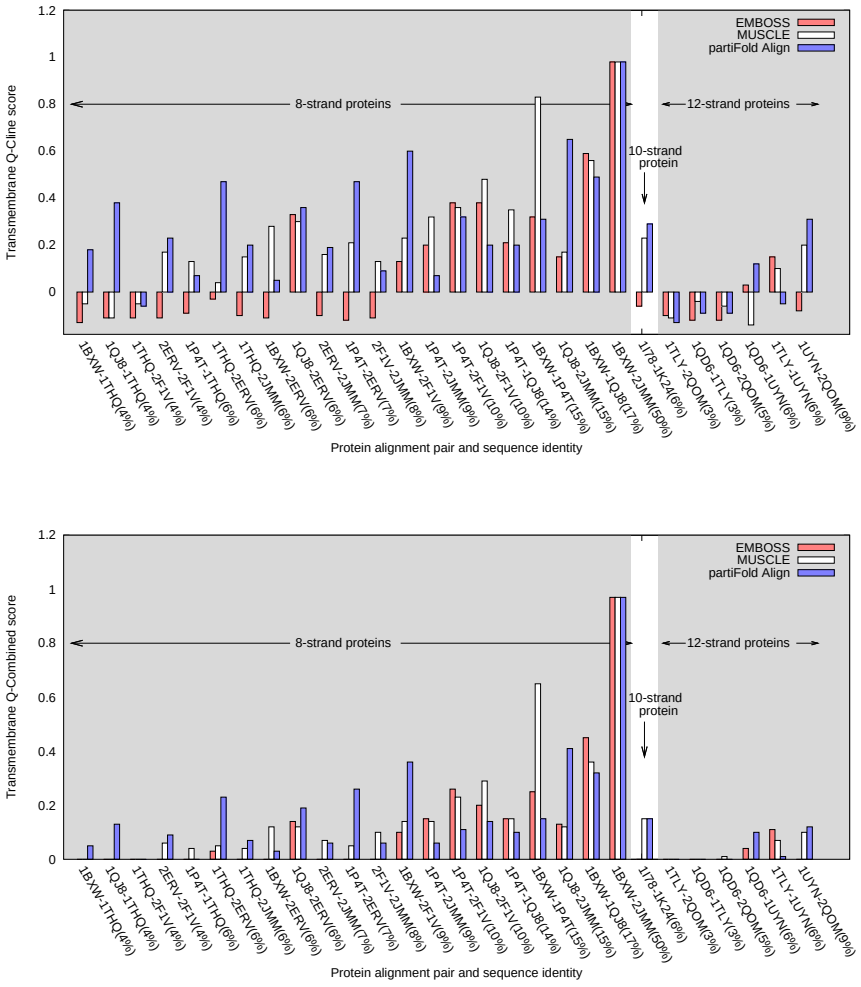


Fig. 6. Transmembrane region Q_{Cline} and $Q_{combined}$ scores in order of increasing sequence identity.

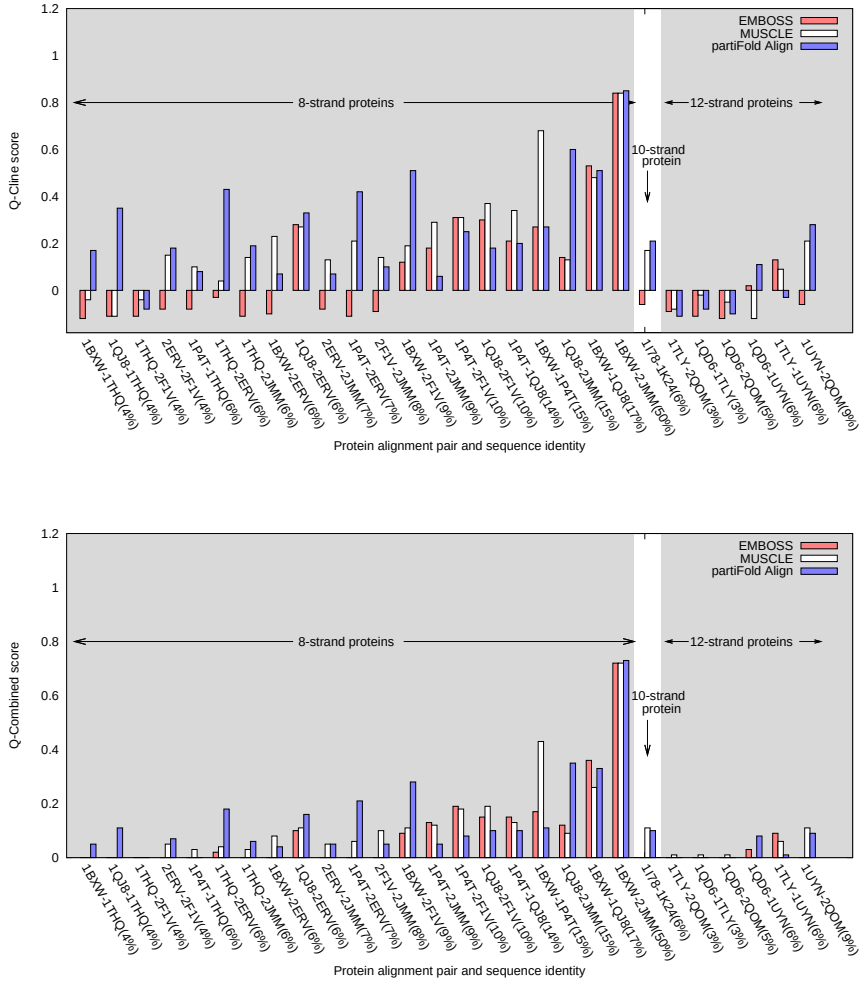


Fig. 7. Whole-protein Q_{Cline} and $Q_{combined}$ scores in order of increasing sequence identity.

Complexity Analysis

We begin with a complexity analysis of the approach described by the recursion equations in Section 2, and then further discuss refinements that were made to improve the complexity. Let n and m denote the lengths of the two sequences. For the analysis, loop type, orientation, and shear number are negligible as they are constantly bounded. First, there are $O(n^2m^2)$ entries $\mathrm{LA}(i, i', j, j')$ for loop alignments; each is computed in constant time. For a fixed strand alignment $\mathcal{A}_\mathrm{s}$, there are $O(n^2 \cdot m^2)$ many entries $\mathrm{ShA}(i, i', j, j'; or; lt; s; \mathcal{A}_\mathrm{s})$. By our recursion equations, each entry is computed in constant time. Now, for TMBs the maximal length of a strand alignment $l_{\max}$ and the maximal number of gaps $g_{\max}$ in a strand alignment can be bounded by small constants. We denote the number of such bounded alignments by ν, which is in $O(l_{\max}^{g_{\max}})$[4] and constant for fixed parameters $l_{\max}$ and $g_{\max}$. As a result, there are $O(n^2m^2\nu)$ entries $\mathrm{ShA}(i, i', j, j'; or; lt; s, \mathcal{A}_\mathrm{s})$ in total.

For the chaining, there are $O(nm\nu^2)$ entries $Dchain(i, j; \mathcal{A}_\mathrm{s}, \mathcal{A}_s^{\mathrm{cyc}}; ct, lt)$, each of these entries is

[4]More precisely, the number of alignments of two sequences of length n with k gaps is $\frac{2}{n+k}\binom{n+k}{k}$

computed by maximizing over left boundaries i' and j', orientation, loop type, shear number and strand alignment of an entry ShA. There are $O(nm\nu)$ such combinations. The final cyclic closing of the chaining is computed by searching over all $O(nm\nu)$ alignments $\mathcal{A}_s^{\text{cyc}}$ and pairs of positions i and j, where the last strand alignment ends.

The resulting complexity of $O(n^2m^2 + n^2m^2\nu + n^2m^2\nu^3)$ time and $O(n^2m^2 + n^2m^2\nu + nm\nu^2)$ space is now reduced drastically, yielding a practicable approach. The main reduction is due to the use of a threshold p_{cutoff} for the probabilities in our probabilistic contact map. As a result, the contact degree is bounded by $1/p_{\text{cutoff}}$ and the quadratically many contacts considered for the above analysis are thus reduced to only linearly many *significant* ones. Now, as mentioned before, we only compute entries of $\text{ShA}(i, i', j, j'; or; lt; s, \mathcal{A}_\text{s})$ where all positions i, i' j and j' are within a narrow range r from a significant contact (p, p'); r is bounded by the shear number s and $g_{\max}$. Thus there remain only $O(4r^2nm\nu)$ entries. For the chaining, this means each entry can be computed in only $O(4r^2\nu)$ time due to the constant contact degree. Time and space complexity are thus reduced by a factor of $O(nm)$.

For TMB alignment, we further reduce the complexity due to the following observation. Because TMB alignments structures contain no bulges, all strand alignments have equal length and have their gaps at the same positions. This eliminates further degrees of freedom in choosing the overlapping strand alignments $\mathcal{A}_\text{s}$ during the chaining. The final complexities of our approach are thus $O(n^2m^2+4r^2nm\nu+4r^2nm\nu) = O(n^2m^2 + 4r^2nm\nu)$ time and $O(n^2m^2 + 4r^2nm\nu + 4r^2nm\nu) = O(n^2m^2 + 4r^2nm\nu)$ space.

Note that affine gap cost as well as stacking can be added without increasing the complexity. An example for such an extension of a in this way similar DP algorithm is again found in the area of RNA sequence-structure alignment. [21], [33].

Number of strands	Sequence identity range	Pair sequence identity	Protein pair	Classification
8-stranded	0-4%	4%	1BXW-1THQ	OMPA-like / OMPA-like (LAA)
		4%	1QJ8-1THQ	
		4%	1THQ-2F1V	OMPA-like (LAA) / OMPA-like
		4%	2ERV-2F1V	
	5-9%	6%	1P4T-1THQ	OMPA-like / OMPA-like (LAA)
		6%	1THQ-2ERV	OMPA-like (LAA) / OMPA-like (LAA)
		6%	1THQ-2JMM	OMPA-like (LAA) / OMPA-like
		6%	1BXW-2ERV	OMPA-like / OMPA-like (LAA)
		6%	1QJ8-2ERV	
		7%	2ERV-2JMM	OMPA-like (LAA) / OMPA-like
		7%	1P4T-2ERV	OMPA-like / OMPA-like (LAA)
		8%	2F1V-2JMM	OMPA-like / OMPA-like
		9%	1BXW-2F1V	
		9%	1P4T-2JMM	
	10-20%	10%	1P4T-2F1V	
		10%	1QJ8-2F1V	
		14%	1P4T-1QJ8	
		15%	1BXW-1P4T	
		15%	1QJ8-2JMM	
		17%	1BXW-1QJ8	
	50%	50%	1BXW-2JMM	
10-stranded	6%	6%	1I78-1K24	OMPT-like / OMPT-like
12-stranded	0-5%	3%	1TLY-2QOM	Nucleoside-specific porin / Autotransporter
		3%	1QD6-1TLY	OM phospholipase / Nucleoside-specific porin
		5%	1QD6-2QOM	OM phospholipase / Autotransporter
	6-10%	6%	1QD6-1UYN	OM phospholipase / Autotransporter
		6%	1TLY-1UYN	Nucleoside-specific porin / Autotransporter
		9%	1UYN-2QOM	Autotransporter / Autotransporter

TABLE II

BREAKDOWN OF OPM DATABASE TMB PROTEINS USED IN ANALYSIS. LAA DISTINGUISHES A FAMILY WITHIN THE OMPA-LIKE SUPERFAMILY OF PROTEINS INVOLVED WITH LIPID A ACYLATION