JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 21, Number 7, 2014
© Mary Ann Liebert, Inc.
Pp. 477–491
DOI: 10.1089/cmb.2013.0163

Simultaneous Alignment and Folding of Protein Sequences

JÉRÔME WALDISPÜHL,[1,2,3,*] CHARLES W. O'DONNELL,[3,4,*] SEBASTIAN WILL,[3,5,6,*]
SRINIVAS DEVADAS,[3,4] ROLF BACKOFEN,[5] and BONNIE BERGER[2,3]

ABSTRACT

Accurate comparative analysis tools for low-homology proteins remains a difficult challenge in computational biology, especially sequence alignment and consensus folding problems. 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* exploits sparsity in the set of super-secondary structure pairings and alignment candidates 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 few known three-dimensional structures. Testing against structurally derived sequence alignments, *partiFold-Align* significantly outperforms state-of-the-art pairwise and multiple sequence alignment tools in the most difficult low-sequence homology case. It also improves secondary structure prediction where current approaches fail. Importantly, *partiFold-Align* requires no prior training. These general techniques are widely applicable to many more protein families (*partiFold-Align* is available at http://partifold.csail.mit.edu/).

1. INTRODUCTION

The consensus fold of two proteins is their common minimum energy structure, given a sequence alignment, and 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 (Shakhnovich et al., 2005); in protein structure prediction studies, consensus-fold predictions can guide tertiary structure predictors; and in sequence alignment algorithms (Edgar and Batzoglou, 2006), 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.

[1] School of Computer Science, McGill University, Montreal, Canada.
[2] Departments of Mathematics, [3] Computer Science and AI Lab, and [4] Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts.
[5] Institut für Informatik, Albert-Ludwigs-Universität, Freiburg, Germany.
[6] Department of Computer Science and IZBI, University of Leipzig, Leipzig, Germany.
*These authors contributed equally to this manuscript.

The specific problem we address is predicting consensus folds of proteins from their unaligned sequences. This definition of consensus fold should not to be confused with the agreed structure between unrelated predictors (Selbig et al., 1999). Our approach succeeds by *simultaneously* aligning and folding protein sequences. By concurrently optimizing unaligned protein sequences for both sequence homology and structural conservation, both higher fidelity sequence alignment and higher fidelity 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) (Forrest et al., 2006). For structure prediction, this harnesses powerful evolutionary corollaries between structure.

While this class of problems has received much attention in the RNA community (Sankoff, 1985; Do et al., 2008; Hofacker et al., 2004; Mathews and Turner, 2002; Havgaard et al., 2007; Backofen and Will, 2004), 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 standard RNA structure model, requiring our initial step of constructing an abstract template for the structure. Moreover, for proteins, there is no clear chemical basis for compensatory mutations (Fariselli et al., 2001), the energy models that define 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 (Xu et al., 2003) 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 (Bradley et al., 2001; Waldispuhl et al., 2006), but do not generate consensus folds. Further, a new class of ''ensemble'' methods, for example partiFold TMB (Waldispuhl et al., 2008; O'Donnell et al., 2011), ''threads'' a profile onto an abstract template, yet does not incorporate sequence alignment information nor does it generate consensus folds.

In this article, 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) (Sutormin et al., 2003; Henikoff and Henikoff, 1992; Rice et al., 2000). To overcome the intractability of this problem, we exploit sparsity in the set of likely amino acid pairings and aligned residues (inspired from the LocARNA algorithm) (Will et al., 2007). *partiFold-Align* is thus able to achieve effectively cubic time and quadratic space in the length of its input sequences.

Then, we expand our techniques and show how our novel pairwise sequence alignment algorithm can be used to efficiently calculate reliable multiple sequence alignments. More specifically, we compute all pairwise alignments with *partiFold-Align* and apply a progressive alignment strategy to merge them. This approach enables us to integrate the information stemming from long-range structural correlations between residues into classical multiple sequence alignment methodologies.

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 (Waldispuhl et al., 2006, 2008). In tests on sequence alignments derived from structure alignments, we obtain significantly better pairwise and multiple 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 article specifically target the difficult case of alignment of low homology sequences and aim to improve the accuracy of such alignments. To complete our benchmark, we apply the multiple sequence alignment version of *partiFold-Align* to the outer membrane protein A transmembrane domain protein family. Our results show that *partiFold-Align* outperforms classical approaches on this difficult class of proteins.

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 in the RNA community, 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 (1985) 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 (Waldispuhl et al., 2006). This is inspired by the recent LocARNA (Will et al., 2007) 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* (Caprara et al., 2004), although in such problems, contact maps implicitly signify the biochemical strength of a contact in a *solved* structure 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 (Waldispuhl et al., 2006). Importantly, β-strand contacts below a parameterizable threshold are excluded to allow for 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 Fig. 2). Coil alignments can be performed independently at each position; however, β-sheet alignments must respect residue pairs. Finally, to decompose the problem (Fig. 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

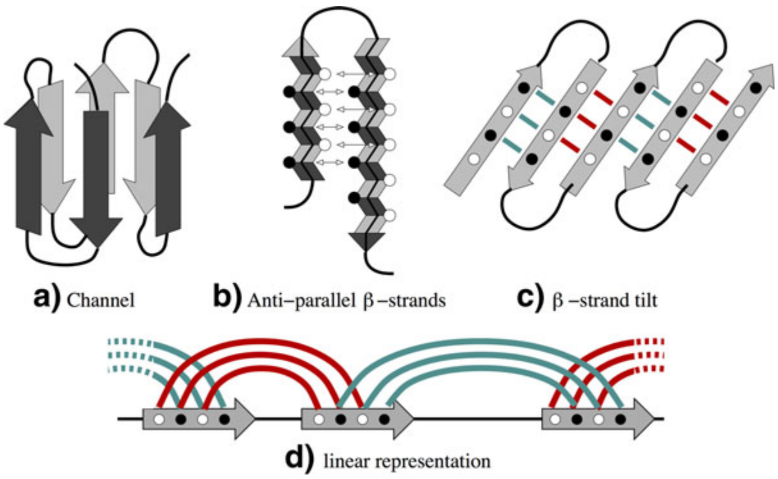

FIG. 1. Different structural elements of transmembrane β-barrels.

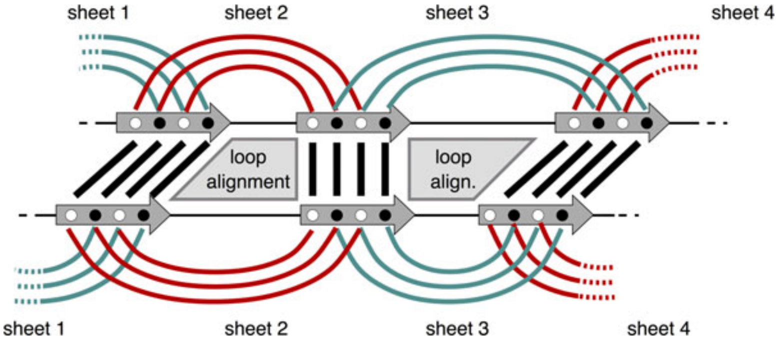

FIG. 2. Elements of TMB alignment. Differently colored amino acids in the sheet denote exposure to the membrane and to the channel, respectively. In a valid sheet alignment, only amino acids of the same type can be matched, whereas no further constraint (except length restriction) are applied to the loop alignment.

the coil extends into the extracellular 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.

2.1. The TMB alignment problem

Formally, we define an alignment $\mathcal{A}$ of two sequences a, b as a set of pairs $\{(p_1, p_2)|p_1 \in [1..|a|] \cup \{-\} \wedge p_2 \in [1..|b|] \cup \{-\}\}$ such that (i) for all (i, j), $(i', j') \in (\mathcal{A} \cap [1..|a|] \times [1..|b|])$, we have $i < i' \Rightarrow j < j'$ (noncrossing), and (ii) there is no $i \in [1..|a|]$ (resp. $j \in [1..|b|]$) where there are two different p, p' with (i, p), $(i, p') \in [1..|a|] \times [1..|b|] \cup \{-\}$ (resp. (p, j), $(p', j) \in [1..|a|] \cup \{-\} \times [1..|b|]$). 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 sequence 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.

With this, the result of 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 Figure 1, we have two different side-chain orientations, namely facing the channel (C) and facing the membrane (M). Since contacts can form only if both amino acids share the same orientation, a *TMB probabilistic contact map P of any TMB a* is a matrix $P = (P(i, i', x))_{1 \leq i < = i' \leq |a|, x \in \{C, M\}}$ where $P(i, i', x) = P(i', i, x)$ and $\forall x \in \{C, M\} : \sum_i P(i, i', x) \leq 1$. 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 matrix P that have a likelihood above a parameterizable threshold.

We weight the alignments with a scoring function that sums 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 onto the two sequences. To allow a convex optimization of this function, we introduce a parameter α distributing the weights of the two terms. Thus, 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:

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

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

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

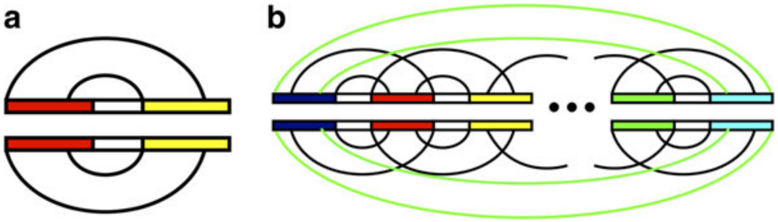

FIG. 3. Problem decomposition: **(a)** alignment of a single sheet including the enclosed loop with positive shear and **(b)** chaining of a single sheet alignment to form a β-barrel. Green arcs indicate the closing sheet connecting the beginning and end.

In practice, *partiFold-Align* implements a slightly more complex stacking pair energy model as described in Waldispuhl et al. (2008). However, for pedagogical clarity, 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 sequence alignment component of the score() is given by:

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

Again, in practice, a penalty for opening gaps is added but not described here for clarity. Finally, the optimization problem our algorithm solves 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 $\mathcal{E}()$ and $\mathcal{W}()$ recursion equations require a slightly more detailed version of the scoring. An additional condition is that contacts only happen between amino acids with the same orientation, and that this orientation alternates between consecutive contacts. Hence, we introduce in τ an additional parameter *env* standing for this side-chain orientation environment 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 σ_s (i, j, env), while for loops we distinguish inner from outer loops (indicated by the loop type *lt*) with the amino acids in the loops scored using σ_l (i, j, lt). The gap 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 antiparallel strand pair as shown in Figure 3a has nested arcs and an outdegree of at most one. We introduce for this configuration a table ShA() (where ShA stands for *sheet alignment*) aligning pairs of subsequences $a_{i..i'}$ and $b_{j..j'}$. Another parameter to account for is the shear number that 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 orientations. Given an orientation environment of a contact *env*, the term $\text{next}_c(env)$ returns the orientation of the following contact. Thus, we have a table ShA(i, i'; j, j'; *env*; *lt*; s) with the following recursion:

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

where

$$\begin{aligned} ShAcontact(i, i'; j, j'; env; lt) = {} & \text{ShA}(i+1, i'-1; j+1, j'-1; \text{next}_c(env); lt; 0) \\ & + \tau(i, i'; j, j'; env) + \sigma_s(a_i, b_j, env) + \sigma_s(a_{i'}, b_{j'}, env) \end{aligned}$$

$$ShAgap(i, i'; j, j'; env; lt; s) = \max\begin{cases} \text{ShA}(i+1, i'; j, j'; env; lt; s) + g_s(a_i, env) \\ \text{ShA}(i, i'-1; j, j'; env; lt; s) + g_s(a_{i'}, env) \\ \text{ShA}(i, i'; j+1, j'; env; lt; s) + g_s(b_j, env) \\ \text{ShA}(i, i'; j, j'-1; env; lt; s) + g_s(b_{j'}, env) \end{cases}$$

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

ShAgap, ShAcontact, and *ShAshear* are introduced for better readability and will not be tabulated. The matrix LA(i, i'; j, j'; lt) represents an alignment of two loops $a_{i..i'}$ and $b_{j..j'}$, with a loop type lt. This table can be calculated using the usual sequence alignment recursion. Thus, we have

$$\mathrm{LA}(i, i'; j, j'; lt) = \begin{cases} \mathrm{LA}(i, i'-1; j, j'; lt) + g_1(a_{i'}, lt) \\ \mathrm{LA}(i, i'; j, j'-1; lt) + g_1(b_{j'}, lt) \\ \mathrm{LA}(i, i'-1; j, j'-1; lt) + \sigma_1(a_{i'}, b_{j'}, lt) \end{cases}$$

As we have already mentioned in the definition of a contact map, we use a probability threshold to reduce both space and time complexity of the alignment problem, in a similar way as is done in the LocARNA-approach (Will et al., 2007). Thus, we will tabulate only values in the ShA-matrix for those positions i, i' and j, j' where the contact probability is above a threshold in both sequences. This is handled at the granularity of strand pairs in practice to reduce complexity.

2.3. Chaining

The next problem is to chain the different single sheet alignments, as indicated by Figure 3b. To build a valid overall alignment, we have to guarantee that the subalignments agree on overlapping regions. A *strand alignment* $\mathcal{A}_s$ is just a partial alignment. The solution is to extend the matrices for sheet alignments by an additional entry for the alignment of 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 sheet alignments, the gap in one strand is then transferred to the chained sheet (by the agreement of subalignments).

The first step is to extend the matrices of sheet alignments by an alignment descriptor, which is used to ensure the compatability of subsolutions used in the recursion. 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 ShA(i, i'; j, j'; env; lt; s; $\mathcal{A}_s$), where we enforce $\mathcal{A}_s$ to satisfy $\mathsf{def}(a, \mathcal{A}_s) = [i..l_1] \cup [r_1..i']$ and $\mathsf{def}(b, \mathcal{A}_s) = [j..l_2] \cup [r_2..j']$ for some $i < l_1 < r_1 < i'$ and $j < l_2 < r_1 < j'$. The new version of ShA() is

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

LA(i, i'; j, j'; lt) does receive an additional parameter since subalignment agreement in chaining is restricted to strands. For definitions *ShAgap*(), *ShAcontact*(), and *ShAshear*(), we now must check whether the associated alignment operations are compatible with $\mathcal{A}_s$. Thus, the new definition of *ShAcontact*() is

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

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

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

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 construct this, 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 two parameters; first, a variable *ct* is used to determine if 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. Second, to control the number of strand in the barrel, we add the variable *nos* storing the number of strands in the sheet.

We initialize the array *Dchain* for every i, j and any strand alignment $\mathcal{A}_s^{\text{cyc}}$ such that $\text{def}(a, \mathcal{A}_s^{\text{cyc}}) = [i..i']$ and $\text{def}(b, \mathcal{A}_s^{\text{cyc}}) = [j..j']$. This initializes the array to a nonbarrel solution. Then

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

where *lt* represents the orientation environment. Note that the strand alignment has not yet been scored. We now describe the chain rules used to build a sheet (an unclosed barrel). To account for the alignment of the first strand of this sheet (so far unscored in ShA), we introduce a function $\text{ShA}_{start}(\mathcal{A}, nos)$ returning the cost of this alignment when $nos = 2$ and returning 0 otherwise. A function *prev*() returning the previous loop type is also used to alternate loop environments between both sides of the membrane. In addition, given two alignments $\mathcal{A}_\text{s}$, $\mathcal{A}'_\text{s}$, we say that $\mathcal{A}_\text{s}$, $\mathcal{A}'_\text{s}$ agree on the strands $i..i'$ in the first sequence and $j..j'$ in the second sequence, written $agr(\mathcal{A}'_\text{s}; \mathcal{A}_\text{s}; i, i'; j, j')$). With this notation, the recursion used to build the unclosed sheets is:

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

We conclude this section by defining the recursions used to close the barrel and perform a sequence alignment of the N-terminal sequences. Since the antiparallel or parallel nature of the closing strand pair depends on the number of strands in the barrel, we introduce here a function *ShAclose*() that 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 antiparallel strand pairings if *nos* is even.

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

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

These final *Dchain*() equations assume that the strand inclinations, modeled using the shear number *s*, 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 (Fig. 1).

2.4. *Progressive alignment of multiple sequences*

We extend our pairwise alignment algorithm to align multiple proteins. To this end, we combine our pairwise alignment tool with T-Coffee (Notredame et al., 2000). Similar combinations of structure-based alignment with T-Coffee have been successful in the context of RNA (Siebert and Backofen, 2005; Otto et al., 2008).

The procedure starts with generating all-against-all pairwise alignment of the input sequences using *partiFold-Align*. Naturally, such computations are independent, which allows us to parallelize this expensive step. Then, we compile a *primary library* that lists all aligned residue pairs for each of these pairwise alignments together with primary weights. For simplicity, we assign the same weight to every residue pair and add a bonus if (according to *partiFold-Align*) both residues belong to a corresponding structure. Finally, we invoke T-Coffee given the generated library as input; this tool merges the structure-based alignments of *partiFold-Align* into a single multiple alignment.

T-Coffee follows the strategy of consistency-based progressive multiple alignment. Briefly, T-Coffee computes extended weights from the given primary weights, which reflect the consistency of each residue pair with the alignment of all input sequences. In subsequent partial multiple alignments (i.e., alignments of subsets of the input sequences), T-Coffee aligns based on these extended weights. After generating a guide tree, sequences and partial multiple alignments are progressively aligned with each other in the order given by the guide tree.

2.5. Complexity analysis of the pairwise sequence alignment algorithm

We conclude this section with a complexity analysis of the pairwise sequence alignment approach described by the recursion equations in subsections 2.2 and 2.3. Then, we 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 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 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 v, which is in $O(l_\mathrm{max}^{g_\mathrm{max}})$[1] and constant for fixed parameters l_max and g_max. As a result, there are $O(n^2m^2v)$ entries ShA$(i, i', j, j'; or; lt; s, \mathcal{A}_\mathrm{s})$ in total.

For the chaining, there are $O(nmv^2)$ entries $Dchain(i, j; \mathcal{A}_\mathrm{s}, \mathcal{A}_s^\mathrm{cyc}; ct, lt)$, each of these entries is computed by maximizing over left boundaries i' and j', orientation, loop type, shear number, and strand alignment of an entry ShA. There are $O(nmv)$ such combinations. The final cyclic closing of the chaining is computed by searching over all $O(nmv)$ alignments $\mathcal{A}_s^\mathrm{cyc}$ and pairs of positions i and j, where the last strand alignment ends.

The resulting complexity of $O(n^2m^2 + n^2m^2v + n^2m^2v^3)$ time and $O(n^2m^2 + n^2m^2v + nmv^2)$ 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 $1/p_\mathrm{cutoff}$ 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 ShA$(i, i', j, j'; or; lt; s, \mathcal{A}_\mathrm{s})$ 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 $O(4r^2nmv)$ entries. For the chaining, this means each entry can be computed in only $O(4r^2v)$ time because of 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 alignment 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 $\mathcal{A}_\mathrm{s}$ during the chaining. The final complexities of our approach are thus $O(n^2m^2 + 4r^2nmv + 4r^2nmv) = (n^2m^2 + 4r^2nmv + 4r^2nmv) = O(n^2m^2 + 4r^2nmv)$ time and $O(n^2m^2 + 4r^2nmv + 4r^2nmv) = O(n^2m^2 + 4$ $(n^2m^2 + 4r^2nmv + 4r^2nmv) = O(n^2m^2 + 4r^2nmv)$ space.

Note that affine gap cost, as well as stacking, can be added without increasing the complexity. An example for such an extension can be again found in the area of RNA sequence-structure alignment (Will et al., 2007; Bompfunewerer et al., 2008).

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

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. To complete this section, we illustrate the impact of our technique on the multiple sequence alignment problem and show promising results on the outer membrane protein A transmembrane domain. We begin with a description of our test dataset and scoring metrics 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

3.1.1. Pairwise alignment. 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 nonhomologous 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 (Lomize et al., 2006) (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 approximately 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 Supplementary Material available online at www.liebertpub.com/cmb for an illustration of the breakdown), and for testing purposes, consider this the ''correct'' pairwise alignment. For structural alignments, the *Matt* (Menke et al., 2008) algorithm is used, which has demonstrated state-of-the-art structural alignment accuracy. During analysis, the resulting alignments are then sorted by relative sequence identity[2] (assuming the *Matt* alignment) (Doolittle, 1981; Raghava and Barton, 2006).

Our *partiFold-Align* alignments are then compared against structural alignments using the Q_{Cline} (Cline et al., 2002; Dunbrack, 2006) 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,[3] 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 useful 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) (Frishman and Argos, 1995) can then be computed; where $Q_2 = (TP + TN)/(sequence\ length)$.

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

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

3.1.2. Multiple sequence alignment. We tested the multiple sequence alignment algorithm on the seed alignment of the outer membrane protein A transmembrane domain from the Pfam database (Finn et al., 2013) (Pfam ID: PF01389). This alignment contains 13 sequences with lengths ranging from 156 to 240 amino acids and has an average sequence identity of 39%. Here, we use two metrics to measure the accuracy of the multiple sequence alignment prediction. First, we calculate the percentage of identical columns between the Pfam alignment (i.e., the reference) and the prediction. In addition, we also compute the percentage of pairs of residues found in the same columns (i.e., conserved pairs). The latter aims to give us a better estimation of the quality of the alignment even when all amino acids are not perfectly aligned in the same columns.

3.2. Model parameter selection

For our analyses, parameters must be chosen for the abstract structural template defined in section 2. In 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. (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 $E_{ct}()$ in $\varepsilon()$, effecting the structural energy score) (Waldispuhl et al., 2008), 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 article, one of three sets of structural parameters were chosen according to the 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. The substitution matrix we use is a combination of the BATMAS

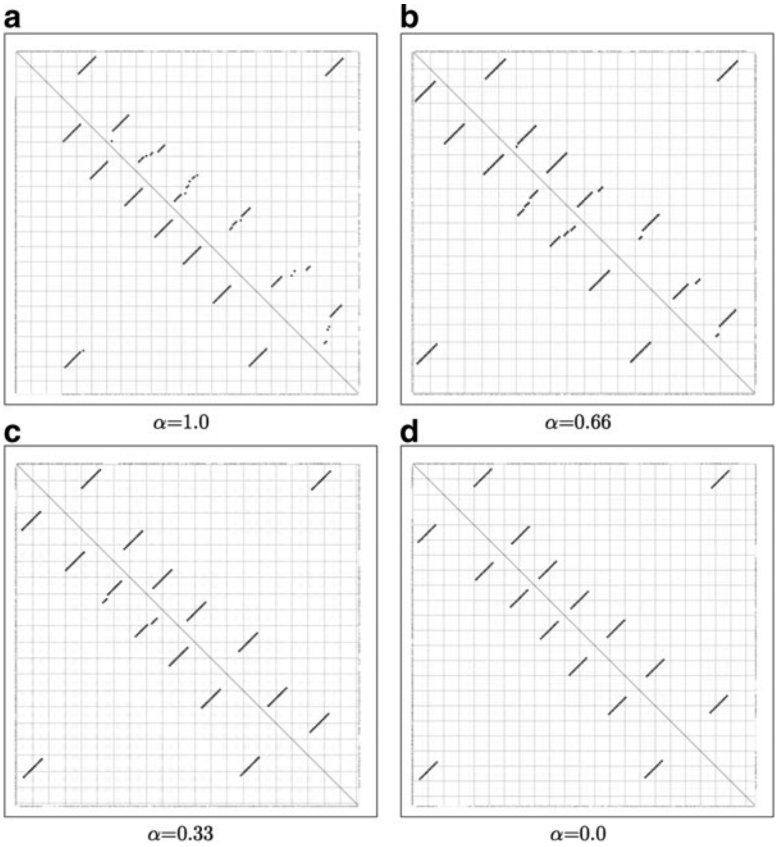

FIG. 4. Stochastic contact maps from a *partiFold-Align* run on the proteins 1BXW and 2F1V. For each of the four plots, the sequence of 1BXW and 2F1V is given on the axes (with gaps), and high probability residue–residue interactions indicated for 1BXW on the lower left half of the graph and 2F1V on the upper right half (i.e., the single-sequence probabilistic contact maps). Structural contact map alignment can be judged by how well the plot is mirrored across the diagonal. Parel **(a)** ($\alpha = 1.0$) shows an alignment that 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.

(Sutormin et al., 2003) matrix for transmembrane regions and BLOSUM (Henikoff and Henikoff, 1992) for coils. For alignments with a sequence homology below 10%, we chose a higher probability threshold value (1×10^{-5} versus 1×10^{-10}) to restrict alignments to highly likely β-strand pairs, reducing signal degradation from low-likelihood β-strand pairs with very distant sequence similarities. For these same alignments (below 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) (Rice et al., 2000), and MUSCLE (Edgar, 2004a, b). EMBOSS implements the seminal Needleman-Wunsch style global sequence alignment algorithm, while MUSCLE is widely thought as one of the most accurate of the ''fast'' alignment programs. Though it incorporates several position-specific gap penalty heuristics similar to those found in MAFFT and LAGAN (Brudno et al., 2003),[4] 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 gap penalty heuristics 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 28th 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 three 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 1 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 supplementary material). 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 individual alignment-pair statistics, as well as alternative accuracy metrics (e.g. $Q_{combined}$) can be found in supplementary material.

Across all TMBs, *partiFold-Align* alignments are more accurate than EMBOSS alignments by an average Q_{Cline} of 16.9% (4.5 ×). 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 gap penalty or other heuristics employed by MUSCLE. Our results suggest that, especially at very low percentages of sequence identity, the conservation of the structure is an important criterion to use in order to obtain accurate sequence alignments.

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

[4] We note that while EMBOSS uses only the BLOSUM substitution matrix, and *partiFold-Align* a combination of BATMAS and BLOSUM, Forrest et al. (2006) show that BATMAS-style matrices do not show improvement for EMBOSS-style algorithms.

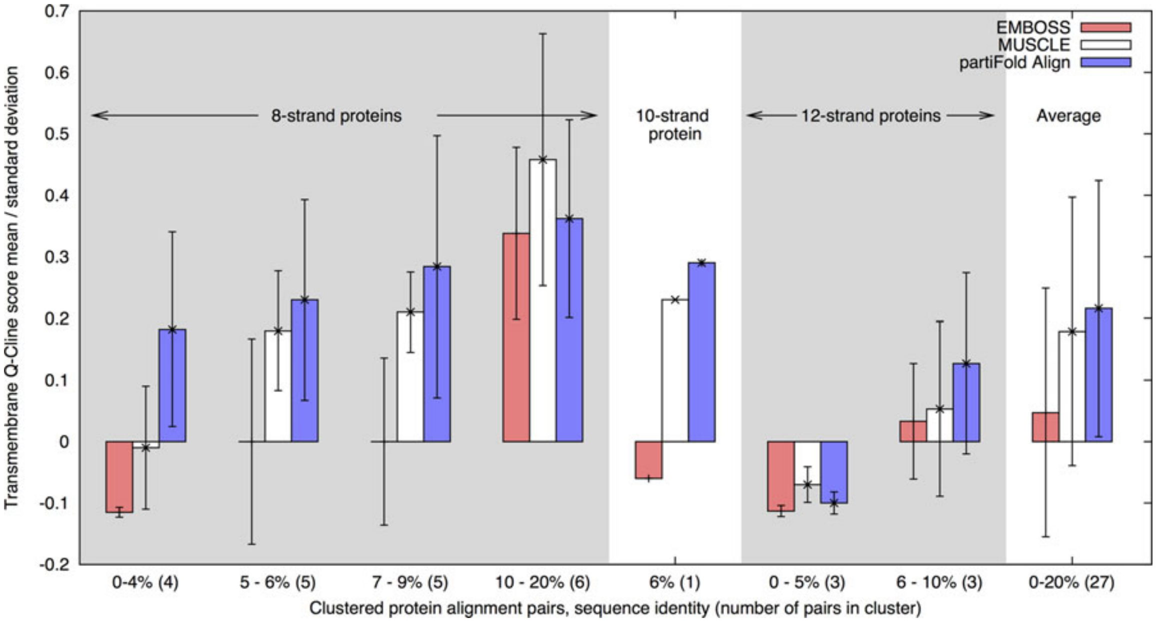

FIG. 5. Mean and standard deviation Q_{Cline} scores for 8-, 10-, and 12-stranded TMBs. Each of the three categories of proteins are clustered and ordered according to sequence identity, with the number of alignments in each cluster in parentheses. Note: By definition, Q_{Cline} scores range between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

report in Table 1 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 in Waldispuhl et al., 2006), 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

Table 1. Secondary Structure Assignment Accuracy

			Consensus	
Category	*PDB id*	*Single seq.*	*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)

Percentage Q_2 of secondary structure prediction correctly assigned residues (transmembrane and non-transmembrane regions). The 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 over all possible alignment pairs for this PDB ID.

Table 2. Accuracy of the Multiple Sequence Alignment Prediction of Outer Membrane Protein A Transmembrane Domain (Pfam ID: PF01389)

Method	*Identity*	*Similarity*	*Alignment length*
partiFold-Align	**54.1**	**92.7**	247
Needlemann-Wunsch	41.7	84.8	255
MUSCLE	53.7	87.0	248

We benchmarked *partiFold-Align* against a progressive alignment strategy based on pairwise sequence alignments calculated with the Needlemann-Wunsch algorithm and MUSCLE. For each software, we report the percentage of identical column (identity), the percentage of conserved pairs of residues (similarity), and the number of columns of the alignments.

the remaining three (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 that have poor single sequence structure predictions 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-strand category do not show any clear change in the secondary structure accuracy. However, recalling that this category covers 3 distinct superfamilies in 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 (Waldispuhl et al., 2008), 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 conclude from this benchmark that the consensus-folding approach can be used to improve the structure prediction of low homology sequences, provided 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.

3.5. Multiple sequence alignment accuracy

To complete this benchmark, we evaluated the performance of the multiple sequence version of *partiFold-Align*. As for the pairwise sequence alignment version, we benchmarked *partiFold-Align* against a classical methodology using the Needlemann-Wunsch combined with the progressive alignment strategy implemented in T-Coffee and MUSCLE. Importantly, the multiple sequence alignment version based on the Needlemann-Wunsch algorithm uses the same progressive alignment methodology as *partiFold-Align*. The results of this computation experiment are reported in Table 2.

Our results show that *partiFold-Align* outperforms other approaches in terms of percentage of identical columns and of percentage conserved pairs of residues. As expected, *partiFold-Align* outperforms a progressive alignment algorithm using the Needlemann-Wusch pairwise sequence algorithm, which does not incorporate the structural signal captured by our model. More interestingly, *partiFold-Align* also outperforms MUSCLE, which uses a more complex sequence alignment scoring system. It is worth noting that *partiFold-Align* produces more compact alignments that are closer to the size of the reference alignment (i.e., 240 columns). All these findings show that the structural information modeled in *partiFold-Align* has the potential to improve the alignment of remote homologs.

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) (Shenker et al., 2011). 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.

We also show that this approach results in significant improvements in multiple sequence alignments. In this study, we illustrated the potential of our techniques on the difficult case of the transmembrane domain A of gram-negative outer membrane proteins. Further, we believe that this methodology could enable us to perform reliable genome-wide annotations of transmembrane proteins.

Finally, 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 (Do et al., 2006, 2008). 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 known structure or alignments with which to use for training. Supplementary Material, including more detailed results, a web server, and the source code, can be found Online at www.liebertpub.com/cmb

ACKNOWLEDGMENT

B.B. was partially supported by NIH grant R01GM081871.

AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

REFERENCES

Backofen, R., and Will, S. 2004. Local sequence-structure motifs in RNA. *J. Bioinform. Comput. Biol.* 2, 681–698.

Bompfunewerer, A.F., Backofen, R., Bernhart, S.H., et al. 2008. Variations on RNA folding and alignment: lessons from Benasque. *J. Math. Bio.* 56, 129–144.

Bradley, P., Cowen, L., Menke, M., et al. 2001. 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, 14819–14824.

Brudno, M., Do, C.B., Cooper, G.M., 2003. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Gen. Res.* 13, 721–731.

Caprara, A., Carr, R., Istrail, S., et al. 2004. 1001 optimal PDB structure alignments: integer programming methods for finding the maximum contact map overlap. *J. Comput. Biol.* 11, 27–52.

Cline, M., Hughey, R., and Karplus, K. 2002. Predicting reliable regions in protein sequence alignments. *Bioinformatics* 18, 306–314.

Do, C.B., Foo, C.-S., and Batzoglou, S. 2008. A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics* 24, i68–0i76.

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

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

Dunbrack, R.L.J. 2006. Sequence comparison and protein structure prediction. *Curr. Opin. Struct. Biol.* 16, 374–384.

Edgar, R.C. 2004a. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* 5, 113.

Edgar, R.C. 2004b. Muscle: multiple sequence alignment with high accuracy and high throughput. *NAR* 32, 1792–1797.

Edgar, R.C. and Batzoglou, S. 2006. Multiple sequence alignment. *Curr. Opin. Struct. Biol.* 16, 368–373.

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

Finn, R.D., Bateman, A., Clements, J., et al. 2013. Pfam: the protein families database. *Nucleic Acids Res.*

Forrest, L.R., Tang, C.L., and Honig, B. 2006. On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. *Biophys. J*. 91, 508–517.

Frishman, D., and Argos, P. 1995. Knowledge-based protein secondary structure assignment. *Proteins* 23, 566–579.

Havgaard, J.H., Torarinsson, E., and Gorodkin, J. 2007. Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Comput. Biol*. 3, 1896–1908.

Henikoff, S., and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. *PNAS* 89, 10915–10919.

Hofacker, I.L., Bernhart, S.H.F., and Stadler, P.F. 2004. Alignment of RNA base pairing probability matrices. *Bioinformatics* 20, 2222–2227.

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

Mathews, D.H. and Turner, D.H. 2002. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.* 317, 191–203.

Menke, M., Berger, B., and Cowen, L. 2008. Matt: local flexibility aids protein multiple structure alignment. *PLoS Comp. Bio*. 4, e10.

Notredame, C., Higgins, D.G., and Heringa, J. 2000. T-coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol*. 302, 205–217.

O'Donnell, C.W., Waldispühl, J., Lis, M., et al. 2011. A method for probing the mutational landscape of amyloid structure. *Bioinformatics* 27, i34–i42.

Otto, W., Will, S., and Backofen, R. 2008. Structural local multiple alignment of RNA. German Conference on *Bioinformatics*, 178–177.

Raghava, G., and Barton, G. 2006. Quantification of the variation in percentage identity for protein sequence alignments. BMC *Bioinformatics* 7, 415.

Rice, P., Longden, I., and Bleasby, A. 2000. Emboss: the european molecular biology open software suite. *Trends Genet*. 16, 276–277.

Sankoff, D. 1985. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Comput*. 45, 810–825.

Selbig, J., Mevissen, T., and Lengauer, T. 1999. Decision tree-based formation of consensus protein secondary structure prediction. *Bioinform*. 15, 1039–1046.

Shakhnovich, B.E., Deeds, E., Delisi, C., and Shakhnovich, E. 2005. Protein structure and evolutionary history determine sequence space topology. *Genome Res*. 15, 385–392.

Shenker, S., O'Donnell, C.W., Devadas, S., et al. 2011. Efficient traversal of beta-sheet protein folding pathways using ensemble models. *J. Comput. Biol*. 18, 1635–1647.

Siebert, S., and Backofen, R. 2005. MARNA: multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons. *Bioinformatics* 21, 3352–3359.

Sutormin, R.A., Rakhmaninova, A.B., and Gelfand, M.S. 2003. Batmas30: amino acid substitution matrix for alignment of bacterial transporters. *Proteins* 51, 85–95.

Waldispuhl, J., Berger, B., Clote, P., and Steyaert, J.-M. 2006. Predicting transmembrane beta-barrels and interstrand residue interactions from sequence. *Proteins* 65, 61–74.

Waldispuhl, J., O'Donnell, C.W., Devadas, S., et al. 2008. Modeling ensembles of transmembrane beta-barrel proteins. *Proteins* 71, 1097–1112.

Will, S., Reiche, K., Hofacker, I.L., et al. 2007. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.* 3, e65.

Xu, J., Li, M., Kim, D., and Xu, Y. 2003. RAPTOR: Optimal protein threading by linear programming. *J. Bioinform. Comput. Biol.* 1, 95–117.

Address correspondence to:
Bonnie Berger
Department of Mathematics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

E-mail: bab@mit.edu