Published in final edited form as:
Proteins. 2011 June ; 79(6): 1930–1939. doi:10.1002/prot.23016.

A multiple-template approach to protein threading

Jian Peng and **Jinbo Xu**[1]
Toyota Technological Institute at Chicago

Abstract

Most threading methods predict the structure of a protein using only a single template. Due to the increasing number of solved structures, a protein without solved structure is very likely to have more than one similar template structures. Therefore, a natural question to ask is if we can improve modeling accuracy using multiple templates. This paper describes a new multiple-template threading method to answer this question. At the heart of this multiple-template threading method is a novel probabilistic-consistency algorithm that can accurately align a single protein sequence simultaneously to multiple templates. Experimental results indicate that our multiple-template method can improve pairwise sequence-template alignment accuracy and generate models with better quality than single-template models even if they are built from the best single templates (P-value<10^{-6}) while many popular multiple sequence/structure alignment tools fail to do so. The underlying reason is that our probabilistic-consistency algorithm can generate accurate multiple sequence/template alignments. In another word, without an accurate multiple sequence/template alignment the modeling accuracy cannot be improved by simply using multiple templates to increase alignment coverage. Blindly tested on the CASP9 targets with more than one good template structures, our method outperforms all other CASP9 servers except two (Zhang-Server and QUARK of the same group). Our probabilistic-consistency algorithm can possibly be extended to align multiple protein/RNA sequences and structures.

Keywords

protein modeling; multiple-template threading; probabilistic alignment matrix; probabilistic-consistency algorithm; multiple sequence/template alignment

Introduction

Traditional protein threading method builds the 3D structure of a target protein sequence using a single template protein [1-6]. Along with many more solved protein structures deposited to the Protein Data Bank (PDB), it is more likely that a target protein without solved structure has more than one good template structures. Therefore, to make full use of the solved structures in PDB, we need to extend the classical protein threading method so that a target protein sequence can be threaded onto multiple templates simultaneously and thus, its 3D model can be built from multiple template structures.

Template-based modeling may be improved using multiple templates in several aspects. First, it is very challenging to choose the best single template for a target protein when it has several similar templates in PDB. We can circumvent this challenging problem by using multiple similar templates to build a 3D model for the target. Second, we can increase alignment coverage for the target protein using multiple templates [7,8]. That is, we can

[1]Please address all correspondence to Dr. Jinbo Xu at the Toyota Technological Institute at Chicago. Phone: 773 834 2511, Fax: 773 834 9881, j3xu@ttic.edu.

potentially align more residues in a target protein to the multiple templates than to a single template so that more target residues can be modeled by templates. In addition, multiple templates may be complementary to one another in terms of their similarity to the target protein. That is, the target protein may be similar to one template in one region and to another template in another region. Therefore, we can improve modeling accuracy by copying structure information from the most similar template regions [9]. Finally, we can also improve protein alignment accuracy by exploiting structural similarity among multiple templates. Alignment accuracy directly determines the quality of a template-based 3D model, so it is critical to generate an accurate sequence-template alignment. Previous multiple-template modeling experiments [7-11] demonstrate that the advantage of multiple-template models over the best single-template models mainly comes from increased alignment coverage or the incorporation of the best single template in model building. By contrast, this paper will show that the major advantage comes from more accurate alignment. Without an accurate multiple sequence/template alignment it is challenging to generate a multiple-template model better than the best single-template model by simply increasing alignment coverage through multiple templates.

This paper describes a novel probabilistic-consistency method that can align a single protein sequence simultaneously to multiple templates. We develop this method by extending our single-template threading method BoostThreader [3,12], which is a method based upon the combination of two machine learning techniques: regression trees and conditional random fields (CRFs). BoostThreader not only generates an accurate sequence-template alignment, but also estimates its probability. BoostThreader can also efficiently calculate the (marginal) probability of one sequence residue being aligned to one template residue. We can use a probabilistic alignment matrix to represent the alignment space of a pair of sequence and template. Each entry in the matrix is the (marginal) alignment probability of two residues calculated from BoostThreader. Our multiple-template method generates the multiple sequence/template alignment by maximizing its probabilistic consistency with all the probabilistic alignment matrices. That is, two residues aligned in the multiple sequence/template alignment should have a high probability in their alignment matrix.

The probabilistic-consistency method has been used by ProbCons [13] for multiple sequence alignment. However, ProbCons cannot be directly used for multiple-template threading when proteins under consideration are distantly-related because 1) ProbCons does not use much structure information in generating a probabilistic alignment matrix; and 2) ProbCons ignores gap penalty since it is very expensive to estimate the probability of a gap. It is fine to ignore gap penalty when proteins to be aligned are close homologs. However, ignoring gap penalty deteriorates alignment accuracy when proteins under considerations are distantly-related. By contrast, our probabilistic-consistency method takes into consideration gap penalty so that we can handle distantly-related proteins. We achieve this by developing a novel approximation method that can accurately estimate the probability of a gap efficiently.

We have tested our multiple-template threading method on recent CASP (Critical Assessment of Structure Prediction) targets. We find out that our multiple-template threading can generate models better than the best single-template models because we can build an accurate multiple sequence/template alignment. Our results also show that without an accurate alignment it cannot improve modeling accuracy much by simply using multiple templates to increase alignment coverage or to ensure that the best template is used to model the target. Our method can also generate more accurate alignments than many popular multiple sequence/structure alignment tools including MAFFT [14], ProbCons [13], T-Coffee [15,16], M-Coffee [17], MUSCLE [18] and PROMALS3D [19]. The models generated by these tools are even worse or slightly better than the best single-template models. Blindly tested on the

CASP9 targets with more than one good template structures, our method outperforms all other CASP9 servers except two (Zhang-Server and QUARK of the same group).

Results and Discussion

To evaluate our multiple-template threading method, we use a subset of 51 CASP8 targets and 48 CASP9 targets, all of which have at least two reliable templates. For each target, we determine its templates using our single-template threading program BoostThreader [3,12]. See supplementary materials for the list of targets and their templates. Note that our results on the 48 CASP9 targets are directly taken from our RaptorX/RaptorX-MSA server submissions to CASP9. That is, these results were blindly generated without knowing the native structures.

We evaluate our method with others using reference-independent alignment accuracy. The reference-independent accuracy of an alignment is defined as the quality of the 3D model built from the alignment. To ensure a fair comparison between our method and other multiple alignment tools, they all use the same set of templates for a given target. Given a target, we first align its sequence to its templates using the multiple-alignment methods and then use MODELLER 9v3 (with default parameters) [20] to build 3D models from the multiple-alignments. We evaluate the quality of a 3D model, measured by TM-score [21] and GDT-TS, by comparing the model with its native structure. Both TM-score and GDT-TS are two widely-used measures for model quality. TM-score ranges from 0 to 1 while GDT-TS from 0 to 100. The higher TM-score/GDT-TS, the better quality the model has. The native structures used for evaluation are downloaded from Zhang's CASP assessment website (http://zhanglab.ccmb.med.umich.edu/casp9/).

In Table 1, BoostThreader is our single-template threading method. The difference between BoostThreader (best template) and BoostThreader lies in that the former uses the best single template to build a 3D model while the latter uses the first-ranked single template to do so. Baseline is a naïve multiple-template method which simply assembles the BoostThreader pairwise sequence-template alignments into a multiple sequence/template alignment using the target sequence as an anchor (see the central star multiple alignment approach in [22]). The baseline method may result in a multiple sequence/template alignment with a larger coverage than a single template alignment, but alignment errors from BoostThreader still persist in such a multiple template alignment and some inconsistency among the aligned templates may exist. MAFFT [14], T-Coffee [15], MUSCLE [18] and ProbCons are multiple sequence alignment methods (i.e., no structure information is used). PROMALS3D [19] is a multiple sequence/structure alignment method. Both sequence profile and structure information is employed in PROMALS3D. M-Coffee is a meta-multiple alignment tool [17], which generates a multiple-alignment by combining pairwise structure alignments generated by TMalign [23], multiple structure alignment by Matt [24] and pairwise sequence-template alignments by BoostThreader. We also developed a new program ProbCons2, which uses the same procedure as our multiple-template threading method to generate probabilistic alignment matrices between two proteins, but uses the probabilistic-consistency algorithm in ProbCons (instead of ours) to generate the final multiple-alignment. By comparing our multiple-template threading method with ProbCons2 and M-Coffee, we can demonstrate the superiority of our probabilistic-consistency algorithm.

Multiple-template threading outperforms single-template threading

As shown in Table 1, the cumulative TM-score and GDT-TS of the models (75.686 and 6585.7, respectively) generated by our multiple-template threading method are better than our single-template models (72.863 and 6265.7, respectively). A paired student t-test indicates that our multiple-template method excels BoostThreader significantly with P-

values 1.73E-13 and 9.29E-17, respectively. As shown in Figure 1, our multiple-template threading can generate better 3D models for 88 of the 99 targets than BoostThreader. Note that our multiple-template threading method is built upon BoostThreader. This indicates that using multiple templates can indeed improve modeling accuracy for most targets. Even if we use the best template (among the templates used to build multiple-template models) to build the single-template model for each target, our multiple-template method still excels the single-template method with P-values 3.32E-06 and 1.10E-08, respectively. As shown in Figure 2, our multiple-template method can generate better 3D models for 76 of the 99 targets than the best single-template method. This implies that it is still worth to use multiple-template methods instead of single-template methods even if we have a perfect template selection procedure. By contrast, previous studies indicate that it is very challenging to generate multiple-template models with (statistically significantly) better quality than the best single-template models[7,8].

A consistent and accurate multiple sequence/template alignment is critical to model quality

As shown in Table 1, our multiple-template method excels the baseline method significantly with P-values 1.72E-07 (TM-score) and 8.72E-07 (GDT-TS), respectively. In fact, the baseline method only performs marginally better than BoostThreader with P-values 0.2448 and 0.0645, respectively. The baseline method even performs slightly worse than the single-template method when the best single templates are used to build models. We further examine the performance of our method, the single-template methods and the baseline method with respect to sequence identity. Among the 99 CASP targets, there are 15, 26, 29, 13, and 16 targets with sequence identity <15%, 15%-20%, 20%-25%, 25%-30% and >30% to their best templates, respectively. The average quality of the models of the targets in each group is calculated and shown in Figure 3. Our multiple-template method is better than the single-template methods and the baseline method in the whole range of sequence identity. The baseline method is worse than the best-single-template method when the sequence identity is less than 30%. Only when the sequence identity is larger than 30%, the baseline method is slightly better than the best-single-template method. This indicates that only when the sequence identity is high and thus, there are few errors in the multiple sequence/template alignment, the increased alignment coverage from multiple templates starts to play a role in improving modeling accuracy.

The above results indicate that simply using multiple templates to increase alignment coverage does not warrant an improvement in modeling accuracy and a high-quality multiple sequence/template alignment is critical to the quality of a multiple-template model. Without an accurate multiple sequence/template alignment the benefit from increased alignment coverage and from the incorporation of the best template in model building may be offset by errors in the alignment and structural inconsistency among the aligned templates.

Our multiple-template method improves pairwise target-template alignments

In total there are 327 pairwise target-template alignments for the 99 CASP8/CASP9 test targets. The pairwise alignments extracted from our multiple-template alignments have accumulative TM-score and GDT-TS 238.30 and 20197.66, respectively, excel those built by our single-template threading method BoostThreader (236.17 and 19963.08, respectively) with P-values 7.18E-05 and 7.40E-06, respectively. The raw score difference between these two types of pairwise alignments is not very big because many of the test targets have very good templates, but a paired student's t-test indicates that the difference is significant.

Our multiple-template method generates better alignments than other multiple sequence/structure alignment tools

As shown in Table 1, our multiple-template method generates significantly better alignments than many popular multiple sequence alignment tools including MAFFT, T-Coffee, MUSCLE and ProbCons. This is expected since they do not use structure information and sequence profile in building alignments. Our method also outperforms several multiple sequence/structure alignment tools including PROMALS3D, M-Coffee and ProbCons2, all of which uses some "consistency" method to build a multiple sequence/structure alignment. M-Coffee, ProbCons2 and our method are built upon the same pairwise alignment procedures. That is, they all use BoostThreader to generate the (probabilistic) alignment matrix for a pair of sequence and template and TMalign/Matt to generate structure alignments among templates, but use different "consistency" algorithms to combine these initial alignments into a multiple sequence/template alignment. This experimental result indicates that our "consistency" method is better than those used in M-Coffee and ProbCons. In fact, PROMALS3D, M-Coffee and ProbCons2 are not even better than the best single-template method, which further confirms that in order to improve modeling accuracy using multiple templates it is critical to build an accurate multiple sequence/template alignment.

Our method performs especially well on distantly-related proteins. Figure 4 shows the average quality of the models of the targets in five different ranges of sequence identity. As shown in this figure, our method performs much better on the targets with low sequence identity to their best templates. When sequence identity is below 20%, PROMALS3D, M-Coffee and ProbCons2 even perform no better than BoostThreader (the first-ranked template is used to build the single-temlate model) although both M-Coffee and ProbCons2 use BoostThreader to generate pairwise alignment (matrices). When sequence identity is below 20%, PROMALS3D is even worse than our single-template method although PROMALS3D uses both structure alignment and sequence profile to build multiple alignments.

Our multiple-template threading method performs well in CASP9

This multiple-template threading method is incorporated into our CASP9 servers RaptorX/RaptorX-MSA/RaptorX-Boost for blind test. Overall, in terms of the quality of the first models RaptorX/RaptorX-MSA are only slightly second to Zhang-Server/QUARK [25,26] according to the unofficial assessment by Zhang group. On the set of 48 CASP9 targets with at least two reliable templates, RaptorX and RaptorX-MSA obtained GDT-TS 3058.50 and 3056.00, respectively. By contrast, the other leading servers Zhang-Server, QUARK, Baker-Robetta, HHpredA[27], pro-sp3-TASSER and Phyre2 [28] obtained GDT-TS 3075.5, 3088.0, 2796.5, 3029.3, 2883.9 and 2916.5, respectively. A paired student t-test shows that RaptorX-MSA excels Baker-Robetta, Phyre2 and pro-sp3-TASSER significantly ($p<0.001$) while the difference between RaptorX-MSA and Zhang-Server ($p=0.551$), QUARK ($p=0.419$) and HHpredA ($p=0.638$) is insignificant. If all the 99 CASP targets are taken into consideration, RaptorX-MSA, Zhang-Server, Baker-Robetta, HHpred/HHpredA, pro-sp3-TASSER and Phyre2 have GDT-TS 6585.71, 6540.8, 6052.31, 6288.99, 6234.11 and 6297.93, respectively. The P-values between RaptorX-MSA and Zhang-Server, Baker-Robetta, HHpred, pro-sp3-TASSER and Phyre2 are 0.272, 0.001, 0.059, 0.018 and 0.049, respectively. A similar result is also observed when TM-score is used to evaluate model quality. Among these servers, Zhang-Server, Baker-Robetta[29,30], and pro-sp3-TASSER [31] refined their post-threading models extensively using computational-expensive folding simulation techniques. Zhang-Server also uses a consensus method to choose the best templates from the outputs of ~10 threading programs and refines models using distance constraints extracted from multiple templates. By contrast, our method can generate models with better or comparable accuracy without consensus or any refinement procedure and thus, our method is much more efficient. HHpredA is also a multiple-template method derived

from HHpred [32], but not published yet. Note that in the above comparison, the performance difference among servers may come from the choice of different templates for the same target. Therefore, this experiment cannot be used to fairly benchmark alignment accuracy.

Specific examples

T0408 is a CASP8 target with two reliable templates 2af7A and 2qeuA, from which we can build two 3D models with TM-score 0.77 and 0.73, respectively, when the target is aligned to the two templates separately. Our multiple-template method can generate a 3D model with TM-score 0.86, much higher than any single-template models. In addition, even if we use only the pairwise sequence-template alignments extracted from the multiple sequence/template alignment to build a 3D model, we can generate two 3D models with TM-score 0.82 and 0.79, respectively. This indicates that our multiple-template method can also improve pairwise sequence-template alignment accuracy significantly.

T0454 is a CASP8 target with 4 templates 1pb6A, 2rasA, 2qopA and 3bhqA, among which 1pb6A is the best. The 3D model built from 1pb6A alone has TM-score 0.71. Although these 4 templates are not very similar (their TM-score is only ~0.65), our multiple-template method can generate a model with TM-score 0.76.

T0524 is a CASP9 target with 6 templates 3k25A, 3dcdA, 1lurA, 3imhA, 1so0C and 1mmzB. The 3D model built from the best template 3k25A alone has a TM-score 0.8. Our multiple-template method generates a significantly better model with TM-score 0.91. The model built from 3k25A using the pairwise alignment extracted from the multiple sequence/template alignment also has a better TM-score 0.84.

T0565 is a CASP9 target with 3 templates 3h41A, 2hbwA and 3mu1A, from which we can build three 3D models with TM-score 0.73, 0.62 and 0.67, respectively. Our multiple-template method generates a much better model with TM-score 0.82. The 3D models built from the pairwise alignments extracted from the multiple sequence/template alignment have TM-score 0.81, 0.62 and 0.74, respectively. This further confirms that our multiple-template method may significantly improve pairwise alignment accuracy.

See supplementary materials for the alignment files (in PIR format), the 3D models of these examples, and per-position RMSD of the models.

Conclusion

This paper describes a novel probabilistic-consistency algorithm for multiple-template protein threading. Along with the growth of the PDB database, multiple-template threading will become more useful for protein modeling. Our results demonstrate that our multiple-template threading can improve pairwise alignment accuracy and generate multiple-template models better than the best single-template models. Our multiple-template method can also generate better models than many existing tools, especially when the sequence identity between target and templates is low. These existing tools even cannot generate multiple-template models better than our best single-template method. Our results also show that without an accurate multiple sequence/template alignment, it is very challenging to improve modeling accuracy (over the best single-template method) by simply using multiple templates to increase alignment coverage. That is, the improvement of multiple-template threading over single-template threading mainly comes from better alignment instead of increased alignment coverage.

Blindly tested on the CASP9 targets with at least two reliable templates, our method outperforms almost all the CASP participating servers. More importantly, our method does

not use post-threading refinement while other CASP9 leading servers such as Zhang-Server and Baker-Rosetta refine their models extensively. A possible future direction is to combine the refinement procedures in Zhang-Server and Baker-Rosetta with our method to see if we can further advance the state-of-art protein modeling.

Our probabilistic-consistency algorithm can possibly be extended to build alignments of multiple protein sequences and structures. Such an algorithm is urgently needed since many protein families now have at least one protein with solved structures. The probabilistic-consistency algorithm implemented in ProbCons is not very good at aligning a set of distantly-related proteins because ProbCons ignores the gap penalty in order to achieve a reasonable computational efficiency. By contrast, we can accurately and efficiently estimate the probability of a gap and thus, take into consideration gap penalty in building alignments. In addition, ProbCons cannot make full use of structure information available for some proteins. Therefore, our method is more suitable for multiple sequence/structure alignment.

Our current method can only work with pairwise alignment methods that can produce probability of matches and gaps, but not those methods that produce matching scores. There are some methods such as probA [33] that can convert matching scores to probability. See http://www.tbi.univie.ac.at/~ulim/probA/ for more technical details. We can use exactly the same method as probA to convert a matching score into probability. However, we would like to emphasize that accurate alignment probability estimation is critical to the accuracy of our multiple sequence/template alignment method. Therefore, using methods other than BoostThreader to generate probability cannot warrant better accuracy than our current multiple-template method. In future we plan to combine several different pairwise alignment methods to see if we can further improve the multiple-template threading accuracy by consensus.

Materials and Methods

Overview

As shown in Figure 5, the workflow of our multiple-template threading method is as follows. Given a target sequence, we first run our single-template method BoostThreader to determine the top templates. Then we build an initial probabilistic alignment matrix for two templates from their pairwise structure alignments generated by TMalign [23] and Matt [24]. We also generate an initial probabilistic alignment matrix between the target and each template using BoostThreader. Afterwards, we run our probabilistic-consistency transformation algorithm to iteratively update all the probabilistic alignment matrices. Finally, we generate a multiple sequence/template alignment by progressive alignment and refinement and run MODELLER to build a 3D model from the alignment.

Probabilistic-consistency transformation

Given a set of proteins to be aligned, the key idea of the “consistency” method is to generate a multiple protein alignment as consistent as possible with their pairwise alignments. Instead of fixing the pairwise alignment between two proteins, the probabilistic-consistency method uses a probabilistic alignment matrix to represent all the possible alignments between two proteins; each alignment is associated with a probability. Then the probabilistic-consistency method will adjust the entries in the alignment matrices to achieve the maximum consistency among all the alignment matrices. Given two proteins x and y, let $P(x_i \bigcirc y_j)$ denote the alignment probability of two residues x_i and y_j. The probabilistic-consistency method adjusts the alignment probability between x_i and y_j through their alignments to an auxiliary protein z. If a residue z_k in z aligns to both x_i and y_j with high probability, x_i and y_j

are more likely to be aligned. We can calculate the alignment probability of x_i and y_j given z as follows.

$$P\left(x_i \circ y_j \mid z\right) = \sum_{z_k} P\left(x_i \circ y_j \circ z_k\right) + \sum_{z_{(k,k+1)}} P\left(x_i \circ y_j \circ z_{(k,k+1)}\right) \quad (1)$$

In Equation (1), $P(x_i \bigcirc y_j \bigcirc z_k)$ is the alignment probability of three residues x_i, y_j and z_k and $P(x_i \bigcirc y_j \bigcirc z_{(k,k+1)})$ is the alignment probability of two residues x_i and y_j and a gapped position $z_{(k,k+1)}$ between the k^{th} and $(k + 1)^{th}$ residues. If we assume that the alignment between x and z is independent of that between y and z, we can decompose the first item in Equation (1) into a product of $P(x_i \bigcirc z_k)$ and $P(y_j \bigcirc z_k)$. Similarly, we can also decompose the second item in Equation (1) into a product of three items: $P(x_i \bigcirc y_j)$, $P(x_i, z_{(k,k+1)})$ and $P(y_j, z_{(k,k+1)})$. It is challenging to estimate $P(x_i, z_{(k,k+1)})$ and $P(y_j, z_{(k,k+1)})$ since the probabilistic alignment matrices do not explicitly contain information relevant to gaps.

In order to estimate the second item in (1), we merge all the gapped positions in z into a single GAP state. Let $P(x_i \bigcirc z_{GAP})\left(=1 - \sum_k P\left(x_i \circ z_k\right)\right)$ denote the probability of x_i not being aligned to any residues in z. We can approximate the second item in Equation (1) as follows.

$$\begin{aligned}\sum_{z_{(k,k+1)}} P\left(x_i \circ y_j \circ z_{(k,k+1)}\right) &= \sum_{z_{(k,k+1)}} P\left(x_i \circ y_j\right) P\left(x_i \circ z_{(k,k+1)}\right) P\left(y_j \circ z_{(k,k+1)}\right) \\ &= P\left(x_i \circ y_j\right) \sum_{z_{(k,k+1)}} P\left(x_i \circ z_{(k,k+1)}\right) P\left(y_j \circ z_{(k,k+1)}\right) \\ &\approx P\left(x_i \circ y_j\right) \sum_{z_{(k,k+1)}} P\left(x_i \circ z_{(k,k+1)}\right) \sum_{z_{(l,l+1)}} P\left(y_j \circ z_{(l,l+1)}\right) \\ &= P\left(x_i \circ y_j\right) P\left(x_i \circ z_{GAP}\right) P\left(y_j \circ z_{GAP}\right)\end{aligned} \quad (2)$$

That is, the second item in Equation (1) is approximated as the product of three terms: $P(x_i \bigcirc y_j)$, $P(x_i \bigcirc z_{GAP})$ and $P(y_j \bigcirc z_{GAP})$. This approximation works well empirically. We can achieve very good alignment accuracy without incurring much more computational burden.

Treating all the templates of a target equally, we have the following probabilistic-consistency transformation formula,

$$P^{t+1}\left(x_i \circ y_j\right) \leftarrow \frac{1}{|M|} \sum_{z \in M} \left(\sum_k P^t\left(x_i \circ z_k\right) P^t\left(y_j \circ z_k\right) + P^t\left(x_i \circ y_j\right) P^t\left(x_i \circ z_{GAP}\right) P^t\left(y_j \circ z_{GAP}\right) \right) \quad (3)$$

where M is the set of available templates and t is the number of iterations of probability adjustment. We can efficiently calculate $P^t(x_i \bigcirc z_{GAP})$ and $P^t(y_j \bigcirc z_{GAP})$ before each round of probabilistic-consistency transformation starts so that the second item in the above equation can be efficiently calculated.

We iteratively update the probabilistic alignment matrices until convergence or 20 iterations of probability adjustment are executed. Once the probabilistic-consistency transformation is finished, we will perform progressive alignment and iterative refinement to generate a multiple protein alignment.

Note that by using the above approximation method the gap penalty will not be amplified by n times (n is the protein length) because for a given residue, its gap probability is not uniformly distributed among all the possible positions. Usually the gap probability at only several positions (no more than 3 in our test examples) dominates the rest. This is because that only reasonably good templates are used to build a multiple-template model for the

target and all the templates are mutually similar. If we align some randomly-chosen proteins, the gap penalty may be amplified larger by our approximation method.

Comparison with ProbCons

The probabilistic-consistency algorithm has been used by ProbCons [13] for multiple sequence alignment. However, ProbCons ignores the second item in the right hand side of Equation (1) since ProbCons does not have an efficient method to estimate this item. It is fine to ignore this item when the following two conditions are satisfied: 1) proteins under consideration are close homologs since in this case the second item is much smaller than the first item; and 2) only a small number of iterations are executed to update the probabilistic alignment matrices. It is not very difficult to prove that if the second item in Equation (1) is ignored, then all the probabilistic alignment matrices will approach 0 when the number of probability-consistency iterations approaches to infinity. This is because at each round of probability adjustment, we will lose some alignment probability mass due to the loss of the second item in Equation (1). In the case we need to align a set of distantly-related proteins, we can neither ignore the second item, nor can we just update the probabilistic alignment matrices for a small number of iterations. Otherwise we cannot achieve the best alignment accuracy.

Experimental results confirm our analysis. As shown in Figure 6, when the number of probabilistic-consistency iterations is small (<6), both our method and ProbCons generate alignments with almost the same accuracy. However, when more than six rounds of probability adjustments are executed, ProbCons deteriorates the alignment dramatically while our method improves the alignment a lot. Note that in this experiment, ProbCons uses BoostThreader to generate the initial probabilistic alignment matrices, so the comparison shown in this figure is fair.

ProbCons fails to generate good alignments when more iterations of probabilistic-consistency transformation are executed because the probabilistic alignment matrices (PAM) in ProbCons approach to zero too fast. There are two major reasons why PAM in ProbCons approaches to zero so fast. One is the underestimation of $\sum_{i,j} P\left(x_i \circ y_j \circ z_k\right)$ by assuming independence between x and y and the other is ignoring gap probability. Our method partially corrects the issue in ProbCons by not ignoring gap probability. To validate our analysis, we have randomly picked up some test examples (see supplementary files) and for each one we have calculated the sum of all the entries in all the probabilistic alignment matrices after each round of probabilistic-consistency transformation. Experimental results indicate that the sum in ProbCons goes to zero extremely fast. By contrast, the sum in our method decreases much more slowly, although it still decreases mainly due to the underestimation of $\sum_{i,j} P\left(x_i \circ y_j \circ z_k\right)$. This may indicate that ignoring gap probability causes a more serious issue than independence assumption. By the way, if we want to further improve alignment accuracy, we need a better estimation of $P(x_i \bigcirc y_j \bigcirc z_k)$ to further reduce or even avoid the decay of the probabilistic alignment matrices (i.e., we cannot assume x and y are totally independent), which is currently under investigation by our group.

The probabilistic alignment matrix for a pair of target and template

Given a pair of target and template (x, y), their probabilistic alignment matrix $P_{x,y}$ is computed using our single-template threading method BoostThreader as follows.

$$P\left(x_i \circ y_j\right) = P_{x,y}(i,j) = \sum_{a \in A} \mathrm{I}\left(x_i \circ y_j \in a\right) P(a \mid x, y),$$

where $P_{x,y}(i, j)$ is the (marginal) alignment probability of residues x_i and y_j; A is the set of all possible alignments between x and y; $\mathrm{I}(x_i \bigcirc y_j \in a)$ is an indicator function, which equals to 1 if x_i and y_j are paired in the alignment a, otherwise 0; $P(a \mid x, y)$ is the probability of an alignment a between x and y calculated from BoostThreader. The probabilistic alignment matrix can be efficiently computed using the forward-backward algorithm in $O(L^2)$ time where L is the average length of sequence/template. See BoostThreader [3,12] for a detailed description of the forward-backward algorithm.

The probabilistic alignment matrix for two templates

We construct a probabilistic alignment matrix between two templates using two structure alignment programs TMalign [23] and Matt [24]. From a pairwise structure alignment, we build a binary matrix by setting the entry corresponding to two aligned residues with value 1. The probabilistic alignment matrix is the average of two binary matrices.

Progressive alignment

Given all the probabilistic alignment matrices, it is still NP-hard to calculate the optimal multiple sequence/template alignment maximizing the probabilistic consistency. The computational complexity is exponential with respect to the number of proteins to be aligned. We use a heuristic method, called progressive alignment, to generate a multiple sequence/template alignment [34]. The method first builds a guide tree, which represents the hierarchical relationship among proteins, and then builds the multiple protein alignment gradually. We use the same procedure as ProbCons to build the guide tree and the final multiple sequence/template alignment.

Iterative refinement

Progressive alignment cannot guarantee a globally optimal solution. Errors appearing in the early stage of the progressive alignment are likely to be propagated to the final result. We use an iterative refinement method to improve the quality of the alignment [35]. In the beginning of each refinement step, proteins under consideration are randomly partitioned into two subsets. Then a new alignment is constructed by aligning the alignments of these two subsets through maximizing the probabilistic consistency. In this work, we run 100 iterative refinement steps after progressive alignment.

Selection of top templates

Given a target protein, BoostThreader first threads it to all the templates in the database PDB95. PDB95 is a set of representative proteins with solved structures and any two proteins in this set have less than 95% sequence identity. Afterwards, BoostThreader ranks all the templates using a neural network regression model, which predicts the quality (i.e., TMscore) of a target-template alignment [3]. A template is discarded if its alignment to the target has a predicted quality less than 90% of the best predicted quality. At most 20 templates are kept for further selection. The pairwise structure similarity between any two templates, measured by TMscore, is calculated using TMalign/Matt. A template is discarded if its structure similarity with the first-ranked template is low (e.g., TM-score<0.65) or less than 90% of the best predicted sequence-template alignment quality. By this way, we make sure that the target and its top templates are mutually similar and thus, a meaningful multiple alignment can be constructed among them.

Computational complexity

The computational complexity of each round of probabilistic-consistency adjustment in our method is in the same order of magnitude as that of ProbCons. The total computational time of both ProbCons and our method is also linear with respect to the number of probability-consistency adjustment iterations. Since our method usually executes more rounds of probability adjustment to achieve the best alignment accuracy for a set of distantly-related proteins, it takes more but reasonable time for our method to terminate.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This work is financially supported by the National Institute of Health grant R01GM089753 (to JX) and the National Science Foundation grant DBI-0960390 (to JX).

Reference

1. Zhou H, Zhou Y. SPARKS 2 and SP3 servers in CASP6. Proteins. 2005; 61(Suppl 7):152–156. [PubMed: 16187357]
2. Wu S, Zhang Y. MUSTER: Improving protein sequence profile-profile alignments by using multiple sources of structure information. Proteins. 2008; 72(2):547–556. [PubMed: 18247410]
3. Peng J, Xu J. Boosting Protein Threading Accuracy. Lect Notes Comput Sci. 2009; 5541:31. [PubMed: 20169009]
4. Shi J, Blundell TL, Mizuguchi K. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J Mol Biol. 2001; 310(1):243–257. [PubMed: 11419950]
5. Xu J, Li M. Assessment of RAPTOR's linear programming approach in CAFASP3. Proteins. 2003; 53(Suppl 6):579–584. [PubMed: 14579349]
6. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. J Bioinform Comput Biol. 2003; 1(1):95–117. [PubMed: 15290783]
7. Larsson P, Wallner B, Lindahl E, Elofsson A. Using multiple templates to improve quality of homology models in automated homology modeling. Protein Science. 2008; 17(6):990–1002. [PubMed: 18441233]
8. Cheng JL. A multi-template combination algorithm for protein comparative modeling. Bmc Structural Biology. 2008; 8
9. Fernandez-Fuentes N, Madrid-Aliste CJ, Rai BK, Fajardo JE, Fiser A. M4T: a comparative protein structure modeling server. Nucleic Acids Res. 2007; 35(Web Server issue):W363–368. [PubMed: 17517764]
10. Rykunov D, Steinberger E, Madrid-Aliste CJ, Fiser A. Improved scoring function for comparative modeling using the M4T method. J Struct Funct Genomics. 2009; 10(1):95–99. [PubMed: 18985440]
11. Joo K, Lee J, Lee S, Seo JH, Lee SJ, Lee J. High accuracy template based modeling by global optimization. Proteins. 2007; 69:83–89. [PubMed: 17894332]
12. Peng J, Xu J. Low-homology protein threading. Bioinformatics. 2010; 26(12):i294–300. [PubMed: 20529920]
13. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S. ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res. 2005; 15(2):330–340. [PubMed: 15687296]
14. Katoh K, Misawa K, Kuma K, Miyata T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res. 2002; 30(14):3059–3066. [PubMed: 12136088]
15. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000; 302(1):205–217. [PubMed: 10964570]

16. Notredame C. Computing multiple sequence/structure alignments with the T-coffee package. Chapter 3:Unit 3. Curr Protoc Bioinformatics. 2010; 8:1–25.
17. Wallace IM, O'Sullivan O, Higgins DG, Notredame C. M-Coffee: combining multiple sequence alignment methods with T-Coffee. Nucleic Acids Research. 2006; 34(6):1692–1699. [PubMed: 16556910]
18. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004; 32(5):1792–1797. [PubMed: 15034147]
19. Pei JM, Kim BH, Grishin NV. PROMALS3D: a tool for multiple protein sequence and structure alignments. Nucleic Acids Research. 2008; 36(7):2295–2300. [PubMed: 18287115]
20. Sali A. Comparative protein modeling by satisfaction of spatial restraints. Mol Med Today. 1995; 1(6):270–277. [PubMed: 9415161]
21. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins. 2004; 57(4):702–710. [PubMed: 15476259]
22. Gusfield D. Efficient methods for multiple sequence alignment with guaranteed error bounds. Bull Math Biol. 1993; 55(1):141–154. [PubMed: 7680269]
23. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. 2005; 33(7):2302–2309. [PubMed: 15849316]
24. Menke M, Berger B, Cowen L. Matt: local flexibility aids protein multiple structure alignment. PLoS Comput Biol. 2008; 4(1):e10. [PubMed: 18193941]
25. Zhang Y. I-TASSER: Fully automated protein structure prediction in CASP8. Proteins. 2009; 77:100–113. [PubMed: 19768687]
26. Zhang Y. I-TASSER server for protein 3D structure prediction. BMC Bioinformatics. 2008; 9:40. [PubMed: 18215316]
27. Hildebrand A, Remmert M, Biegert A, Soding J. Fast and accurate automatic structure prediction with HHpred. Proteins. 2009; 77:128–132. [PubMed: 19626712]
28. Kelley LA, Sternberg MJE. Protein structure prediction on the Web: a case study using the Phyre server. Nature Protocols. 2009; 4(3):363–371.
29. Das R, Baker D. Macromolecular modeling with rosetta. Annu Rev Biochem. 2008; 77:363–382. [PubMed: 18410248]
30. Raman S, Vernon R, Thompson J, Tyka M, Sadreyev R, Pei J, Kim D, Kellogg E, DiMaio F, Lange O, Kinch L, Sheffler W, Kim BH, Das R, Grishin NV, Baker D. Structure prediction for CASP8 with all-atom refinement using Rosetta. Proteins. 2009; 77(Suppl 9):89–99. [PubMed: 19701941]
31. Zhou H, Skolnick J. Protein structure prediction by pro-Sp3-TASSER. Biophys J. 2009; 96(6): 2119–2127. [PubMed: 19289038]
32. Soding J. Protein homology detection by HMM-HMM comparison. Bioinformatics. 2005; 21(7): 951–960. [PubMed: 15531603]
33. Muckstein U, Hofacker IL, Stadler PF. Stochastic pairwise alignments. Bioinformatics. 2002; 18(Suppl 2):S153–160. [PubMed: 12385998]
34. Feng DF, Doolittle RF. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J Mol Evol. 1987; 25(4):351–360. [PubMed: 3118049]
35. Gotoh O. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. J Mol Biol. 1996; 264(4): 823–838. [PubMed: 8980688]

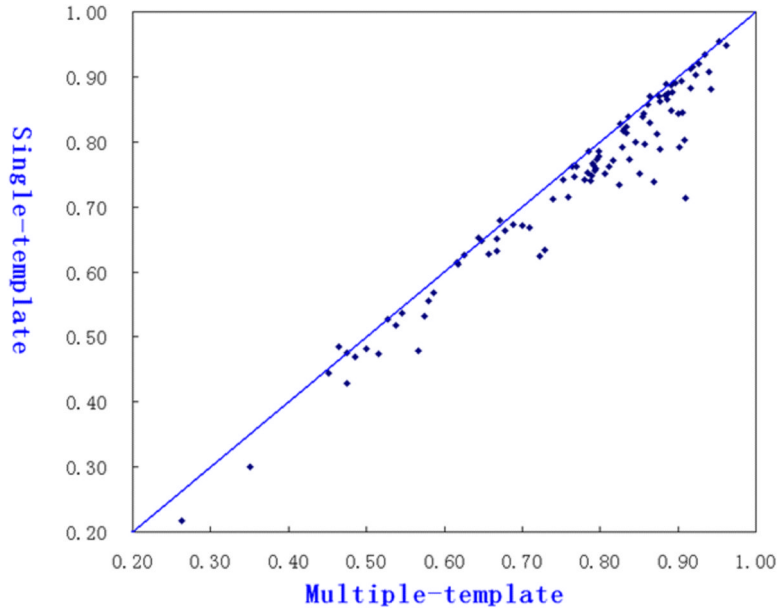

Figure 1. The quality (TM-score) of the models built by our multiple-template and single-template methods for the 99 CASP8 and CASP9 targets

For 88 out of 99 targets (points below the diagonal line), our multiple-template method yields higher TM-score than our single-template method.

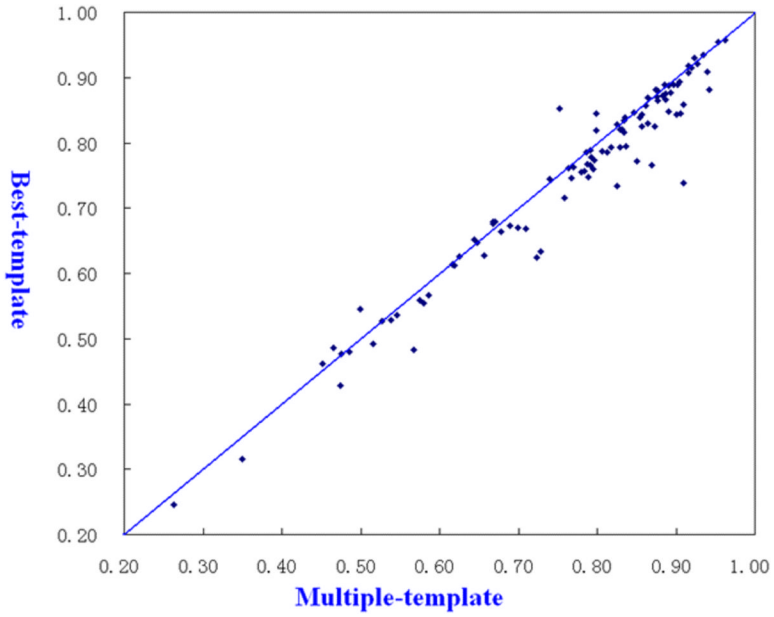

Figure 2. The quality (TM-score) of the models built by our multiple-template and best-single-template methods for the 99 CASP8 and CASP9 targets

For 76 out of 99 targets (points below the diagonal line), our multiple-template method yields higher TM-score than our best-single-template method.

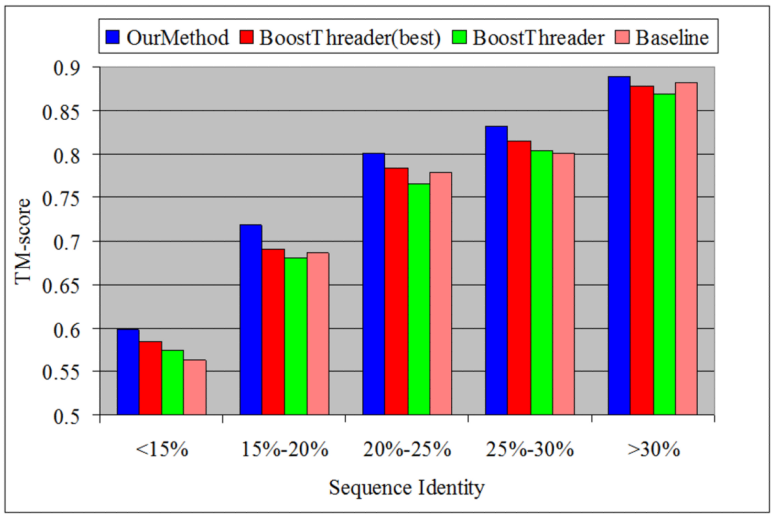

Figure 3. Average TM-score of the models built from multiple templates and single templates for the targets in a group

The targets are divided into 5 groups according to their sequence identity to their best templates. BoostThreader (best) represents the models built from the best single templates while BoostThreader represents the models built from the first-ranked templates.

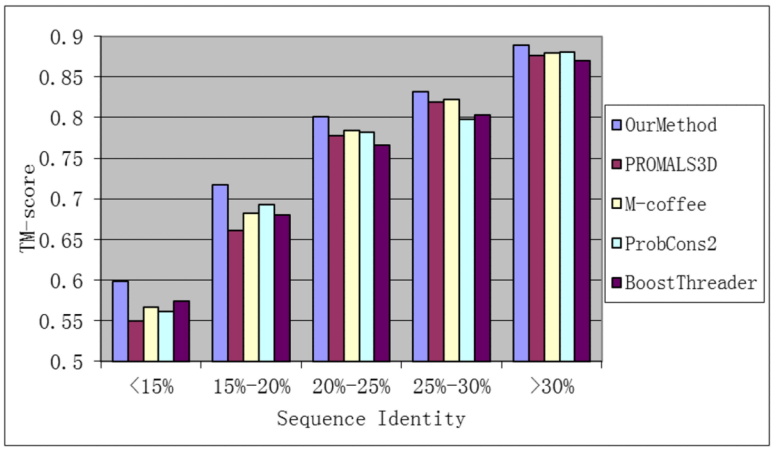

Figure 4. Average TM-score of the models for the targets in a group

The targets are divided into 5 groups according to their sequence identity to their best templates.

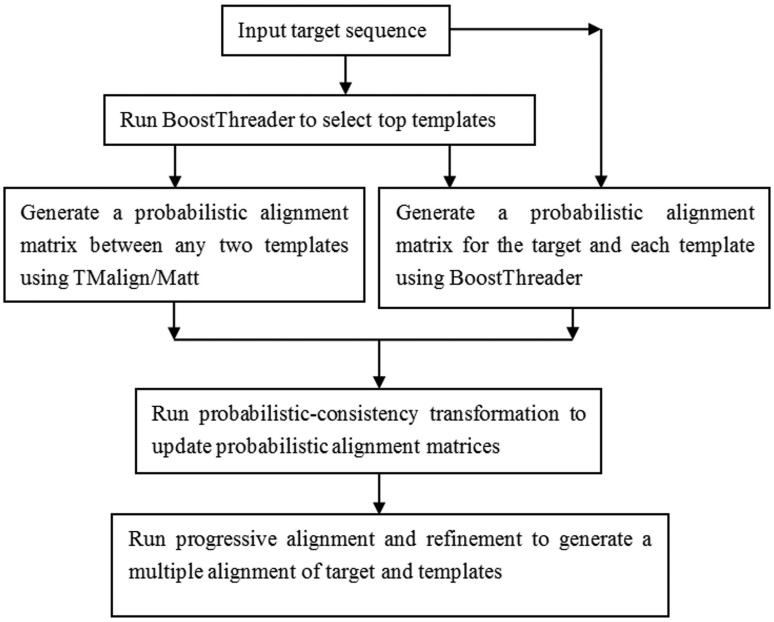

Figure 5.
The overall flowchart of our multiple-template threading method.

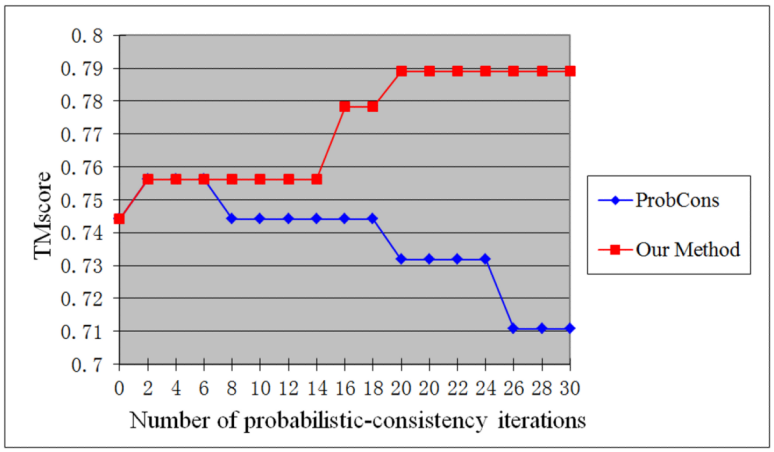

Figure 6.
The quality (measured by TM-score) of the alignment between T0592 and two templates 3ilmb and 3fnje with respect to the number of iterations for probabilistic-consistency transformation.

Table 1

Cumulative TMscore and GDT-TS of the models generated by various multiple sequence/structure alignment methods. P-values in the table are calculated from a paired student t-test between our method and others. The smaller the P-value is, the more significant the performance difference between our method and others. See text for the description of the methods in this table.

	Model Quality Score		P-value	
Methods	**TM-score**	**GDT-TS**	**TM-score**	**GDT-TS**
Our method	**75.686**	**6585.7**	-	-
Baseline	73.386	6353.4	1.72E-07	8.72E-07
BoostThreader	72.863	6265.7	1.73E-13	9.29E-17
BoostThreader (best template)	74.065	6381.5	3.32E-06	1.10E-08
MAFFT	66.368	5715.9	5.69E-10	7.10E-10
T-coffee	67.697	5852.1	1.34E-07	1.33E-07
MUSCLE	66.556	5715.3	2.35E-09	2.29E-09
ProbCons	67.193	5804.9	4.87E-08	5.51E-08
PROMALS3D	72.636	6309.2	1.62E-04	5.52E-04
ProbCons2	73.553	6390.6	1.55E-03	3.07E-03
M-coffee	73.721	6414.9	6.34E-04	3.15E-03