Received: 22 December 2018 | Revised: 20 February 2019 | Accepted: 13 March 2019

DOI: 10.1002/prot.25684

PROTEINS WILEY

RESEARCH ARTICLE

Does inclusion of residue-residue contact information boost protein threading?

Sutanu Bhattacharya | Debswapna Bhattacharya

Department of Computer Science and Software Engineering, Auburn University, Auburn, Alabama

Correspondence
Debswapna Bhattacharya, Department of Computer Science and Software Engineering, Auburn University, 3104 Shelby Center, Auburn, AL 36849.
Email: bhattacharyad@auburn.edu

Abstract

Template-based modeling is considered as one of the most successful approaches for protein structure prediction. However, reliably and accurately selecting optimal template proteins from a library of known protein structures having similar folds as the target protein and making correct alignments between the target sequence and the template structures, a template-based modeling technique known as threading, remains challenging, particularly for non- or distantly-homologous protein targets. With the recent advancement in protein residue-residue contact map prediction powered by sequence co-evolution and machine learning, here we systematically analyze the effect of inclusion of residue-residue contact information in improving the accuracy and reliability of protein threading. We develop a new threading algorithm by incorporating various sequential and structural features, and subsequently integrate residue-residue contact information as an additional scoring term for threading template selection. We show that the inclusion of contact information attains statistically significantly better threading performance compared to a baseline threading algorithm that does not utilize contact information when everything else remains the same. Experimental results demonstrate that our contact based threading approach outperforms popular threading method MUSTER, contact-assisted ab initio folding method CONFOLD2, and recent state-of-the-art contact-assisted protein threading methods EigenTHREADER and map_align on several benchmarks. Our study illustrates that the inclusion of contact maps is a promising avenue in protein threading to ultimately help to improve the accuracy of protein structure prediction.

KEYWORDS

contact assisted threading, protein structure prediction, protein threading, residue-residue contact, template based modeling

1 | INTRODUCTION

Computational prediction of protein three-dimensional (3D) structure from its sequence is still an open problem.[1–4] Depending on whether relevant proteins are available and identifiable in the Protein Data Bank,[5] protein structure prediction methods are broadly classified in two categories: (a) ab initio folding, and (b) template-based modeling (TBM). Methods based on ab initio folding predict protein 3D structures from scratch by using the sequence information alone. But these methods achieve very limited success,[4] particularly for medium to large proteins. On the other hand, with the increase of both sequence and structure databases, TBM, including homology modeling and protein threading, is by far the most accurate approach for protein 3D structure prediction.[6–8] The success of homology modeling depends on the availability of homologous protein with known structure, whereas, threading or fold recognition is an advanced template finding strategy to identify similar templates from the template library if no close homologs are available.[3,9] The key idea behind threading approach is: there exists a limited number of basic folds in nature and many proteins share alike folds even if there is a divergence in their sequences.[10,11] There are several components of threading that includes (a) template library containing known protein structures, (b) scoring function to evaluate each query-template pair, (c) searching method, and (d) selection method for selecting the best-fit template based on given alignments. The key purpose of threading is to find the best alignment of the query sequence to the template. Consequently, the success of this approach lies in finding a similar fold from the template library that contains similar structural features such as secondary structure, solvent accessibility, and so forth. For each template fold, such features are calculated and matched against the corresponding features predicted from the query sequence. The scoring function used in threading plays a vital role in matching the query-template pair with a direct impact on threading accuracy.[12,13] Existing threading methods are based on various approaches, such as the sequence profile-profile alignment, structural profile alignment, multi-source features alignment, hidden Markov models, and machine learning.[6,7,14–30] Some of these approaches[6,7,16] employ a scoring function based on sequential features whereas others[28,31] make use of sequential and structural features for calculating the scoring function. State-of-the-art threading methods include MUSTER,[28] SEGMER,[31] HHsearch,[14] GenTHREADER,[23] SPARKS-X,[6] CNFpred,[7,16] PROSPECT,[32] RAPTOR,[12] and MRFalign.[33] Despite steady progress in threading over the past decade, there is still room for improvement particularly with the rapid growth in both sequence and structure databases.[34]

With the recent progress in protein residue-residue contact prediction powered by sequence co-evolution and machine learning,[35–47] contact map may become a valuable additional structural feature that can assist protein threading. While there exists pure contact driven ab initio folding methods such as EVFold,[48] CONFOLD,[49] CoinFold,[39] CONFOLD2[50] based on distance geometry; cutting-edge threading methods including EigenTHREADER,[51] map_align,[52] and DeepThreader[53] are increasingly integrating contact or distance information along with other features to boost threading accuracies. For example, EigenTHREADER integrates predicted contact map from MetaPSICOV[35] and sequential information whereas map_align uses a pure contact driven threading approach by maximizing the overlap of the co-evolutionary predicted contact map of the target protein to the template's true contact map. Very recently, DeepThreader proposes to select top templates by combining sequential features and predicted inter-residue distances for generating a query-template alignment.

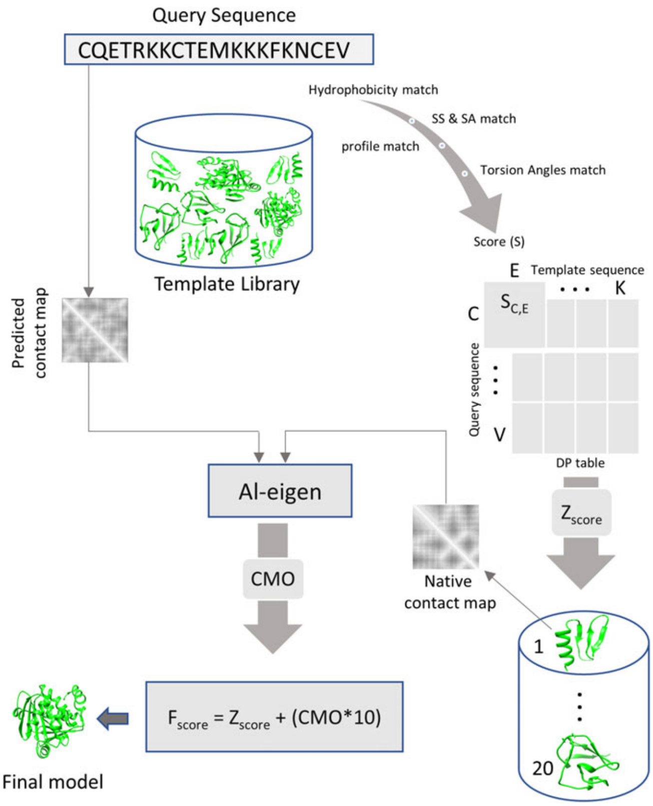

FIGURE 1 Flowchart of our work. The query sequence is threaded against the template library by calculating the similarity score using sequential and structural features. Subsequently, contact map overlap (CMO) score between the contact map of each template and the predicted contact map of the query is calculated, and integrated to the threading-based similarity score in a weighted manner to select the best-fit template for predicting the 3D structure of the query [Color figure can be viewed at wileyonlinelibrary.com]

Here, we analyze whether the addition of residue-residue contact information helps increase the accuracy of protein threading. We develop a new threading approach that integrates sequence and structural information with residue-residue contacts in order to examine how much accuracy gain can be obtained by incorporating contact information. First, we investigate whether incorporating contact information into a threading-based approach helps in predicting the top ranked models with better accuracy than a pure threading-based approach. We further explore whether residue-residue contact information along with a threading-based approach is more promising than purely contact driven ab initio folding methods. Finally, we compare the performance of our work with the state-of-the-art contact-assisted protein structure prediction methods in a blind manner using protein targets from the recently concluded 13th Critical Assessment of protein Structure Prediction (CASP13) experiment.

2 | MATERIALS AND METHODS

2.1 | Alignment scoring function for threading

We develop a threading approach (Figure 1), which uses sequential and structural features to search a library of templates. Specifically, we use structure profile, native secondary structure, native solvent accessibility, and native torsion angles as features for the template. For the query protein, sequence profile, predicted secondary structure, predicted solvent accessibility and predicted torsion angles are used as features. We use DSSP[54] for extracting native secondary structure, native solvent accessibility and native torsion angles for each template whereas SPIDER3[55] is used to predict the corresponding features from the sequence of the query protein.

The alignment score for aligning the query position i with the template position j is:

$$\begin{aligned} Score(i,j) &= S_{seq\_prof} + S_{ss} + S_{struc\_prof} + S_{sa} + S_{psi} + S_{phi} + S_{hydro} + S_{shift} \\ &= \sum_{k=1}^{20} \frac{(Pc_q(i,k) + Pd_q(i,k))L_t(j,k)}{2} + w_1 \delta(s_q(i), s_t(j)) \\ &+ w_2 \sum_{k=1}^{20} Ps_t(j,k) L_q(i,k) + w_3 (1 - 2|SA_q(i) - SA_t(j)|) \\ &+ w_4 (1 - 2|\varphi_q(i) - \varphi_t(j)|) + w_5 (1 - 2|\Phi_q(i) - \Phi_t(j)|) \\ &+ w_6 M(AA_q(i), AA_t(j)) + w_7 \end{aligned} \tag{1}$$

where “q” stands for the query and “t” stands for the template protein. The first term in Equation (1) is the sequence profile-profile alignment between the query and the template. The frequency of the kth residue at the ith position of the multiple sequence alignment (MSA), Pc_q(i,k), and Pd_q(i,k), are obtained by two iterations of PSIBLAST[56] search against a non-redundant (nr) sequence database with an e-value cutoff of 0.001 and 1.0 to get "close" and "distant" homologues, respectively. The sequence-derived log-odds profile of the template, L_t(j,k), is obtained by PSIBLAST with an e-value cutoff of 0.001. The second term in Equation (1) compares the predicted secondary structure with the native secondary structure. The term $\delta(s_q(i), s_t(j))$ is 1 if $s_q(i) = s_t(j)$ and − 1 otherwise. Both $s_q(i)$ and $s_t(j)$ have three distinct states: helix, strand, and coil. The third term S_{struc_prof} is a structure-derived profile where Ps_t(j,k) denotes the kth amino acid's frequency at the jth position of the template, L_q(i,k) denotes the log-odds profile of the query, and both are obtained by PSIBLAST search with an e-value cutoff of 0.001. The fourth term S_{sa} computes the difference between the predicted solvent accessibility and the native solvent accessibility, where SA_q(i) is for the ith residue of the query and SA_t(j) is for the jth residue of the template. The next two terms (fifth and sixth) account for the match between the predicted torsion angles of the query ($\varphi_q(i)$ and $\Phi_q(i)$ of the ith position of the query) and the native torsion angles of the template ($\varphi_t(j)$ and $\Phi_t(j)$ of the jth position of the template). Both psi and phi angles are normalized by 360^0. The seventh term S_{hydro} matches the hydrophobic residues (V, I, L, F, Y, W, M) of the query and the template. $M(AA_q(i), AA_t(j)) = 1$ if the ith position of the query ($AA_q(i)$) and the jth position of the template ($AA_t(j)$) are both hydrophobic; $M(AA_q(i), AA_t(j)) = 0.7$ if $AA_q(i)$ and $AA_t(j)$ are identical; otherwise, $M(AA_q(i), AA_t(j)) = 0$. Finally, Needleman-Wunsch[57] dynamic programming algorithm is used to get the best query-template alignment. A position-specific gap penalty is also employed. The last term of Equation (1) is a constant, w_7, which is used to discourage the alignment of unrelated residues. Seven weight parameters and two gap penalty parameters (gap opening g_o and gap extension g_e) are used[28]: $w_1 = 0.66$, $w_2 = 0.39$, $w_3 = 1.60$, $w_4 = 0.19$, $w_5 = 0.19$, $w_6 = 0.31$, $w_7 = 0.99$, $g_o = 7.01$, $g_e = 0.55$.

Initially, the templates are ranked by the following Z_{score}

$$Z_{score} = \frac{\left(R'_{score} - < R'_{score} >\right)}{\sqrt{< R'^{2}_{score} > - < R'_{score} >^{2}}} \tag{2}$$

where R'_{score} is the score normalized by the greater one of the raw alignment score with L (full alignment length) and L′ (partial alignment length), and < … > denotes the average of all templates in the library.

2.2 | Inclusion of residue-residue contact information

A residue-residue contact map is a binary, symmetric matrix that provides a two-dimensional (2D) view of the inter-residue spatial distances in a protein 3D structure with contacts denoted as 1, and non-contacts as 0. That is, whenever the distance between any two residues in the 3D structure within some distance threshold value, typically considered between 6 and 16 Å, considering some specific atoms (mostly C_α or C_β) of the residue pairs, the corresponding residue pairs are said to in contact. In this work, we use 8 Å as the distance threshold for contact maps. Contact map overlap (CMO[52]) is used to find the similarity between two contact maps, where the higher CMO score means there is a higher likelihood of being similar. State-of-the-art method for CMO, Al-eigen[58] applies a heuristic approach to obtain a set of principal weighted eigenvectors by using eigenvalue decomposition of symmetric matrices or contact maps. Finally, the overlap score between two contact maps is obtained by calculating the optimal global alignment between two sets of weighted eigenvectors by using Needleman-Wunsch[57] global alignment algorithm. In our present work, we run Al-eigen using seven eigenvectors to get the CMO score between a pair of contact maps.

For threading template scoring, we integrate CMO score along with Z_{score} described above to calculate the final score for selecting the best-fit template. As the range of CMO is [0,1], we use the weight 10 as the weight of CMO in calculating the final score. Consequently, the final score for threading template selection is:

$$F_{score} = Z_{score} + (CMO \times 10) \tag{3}$$

After selecting the top template using Equation (3), we build the 3D model of the query protein using the query-template alignment by copying the coordinate of aligned residues from the template.

2.3 | Template libraries, benchmark data, and programs to compare

We use a representative nonredundant template library collected from https://zhanglab.ccmb.med.umich.edu/library/,[59] which contains 70,670 template structures.

We benchmark against three datasets. The first data set is the Test500[28] which contains 500 test proteins. It is a set of nonhomologous proteins with sequence identity <25% and having length from 50 to 633 residues. On Test500 dataset, we compare the performance of our work against a popular threading-based method, MUSTER.[28] MUSTER (multi-source threadER) is a threading algorithm that uses various structural and sequential single-body features to generate the query-template alignment using Needleman-Wunsch[57] dynamic programming algorithm. We also benchmark our approach against our in-house baseline threading approach that does not use the contact information but utilizes the same alignment scoring function described above. For a fair comparison, we use the same template library for all competing methods where templates with sequence identity >30% to the query protein are excluded. As the source of contact, we use both true contact maps extracted from the native structures of the query proteins and contact maps predicted from their sequences by RaptorX,[36–39] a state-of-the-art contact prediction method that integrates sequence co-evolution and deep learning. To reduce noise in RaptorX predicted contact maps, residue pairs with predicted contact probability <0.5 are excluded.

The second test set is the 150 proteins in the PSICOV[40] dataset, which contains 150 single chains and single domain monomeric proteins. On this dataset, we benchmark our work against the state-of-the-art contact guided ab initio folding method CONFOLD2,[50] which builds 3D protein structures using predicted contact maps and secondary structures. It constructs a pool of models by exploring the fold space using different subsets of contacts and then selects the top five models through clustering. As the source of contact, CONFOLD2 uses contact maps predicted from MetaPSICOV,[35] another state-of-the-art contact predictor that integrates sequence co-evolution and machine learning. The published work of CONFOLD2 fails to report results for 4 targets from the PSICOV dataset. We, therefore, consider 146 targets for the current benchmarking. For a fair comparison, we use the same MetaPSICOV predicted contact maps after excluding homologous templates. To do this, we use three different increasingly stringent homology cutoffs as follows. First, we exclude all templates from the template library with sequence identity >30% to the query proteins (referred to as Cutoff-1). In addition to sequence identity cutoff, to make the template selection cutoff more stringent, we exclude templates in the same SCOP[60] (Version 1.75) family of the query proteins (referred to as Cutoff-2); and templates in both SCOP family and superfamily (referred to as Cutoff-3).

The third test set is CASP13 targets officially released with their native 3D structures in December 2018. We consider only 20 full-length targets resulting in a total of 32 domains that CASP has officially released so far. Here, we benchmark against two state-of-the-art contact-assisted methods: EigenTHREADER,[51] and map_align.[52] EigenTHREADER is a fold recognition method, which uses MetaPSICOV contact maps for searching the template library of contact maps. Similar to Al-eigen,[58] it uses eigenvector decomposition and dynamic programming to generate alignment between two sets of weighted eigenvectors. Since EigenTHREADER produces three kinds of alignment scores for each template, we use contact map overlap (CMO) to rank templates, as it gives the best result among the three. map_align uses pure co-evolutionary based contact maps to find analogous folds from the library of templates. To make a fair comparison, we use the same template library curated before CASP13 started on May 1, 2018, containing 69,041 template structures as well as the same contact maps predicted by RaptorX[36–39] for all three methods. Finally, we compare our work against DeepThreader, a recent method that utilizes predicted inter-residue distances instead of contacts for protein threading. As DeepThreader method is not yet publicly available, we use the predicted models submitted to CASP13 by RaptorX-TBM server group that employs DeepThreader method.

2.4 | Evaluation criteria

We use TM-score[61,62] to evaluate the performance of each competing method. It is calculated by

$$TM-score=\frac{1}{L}\sum_{i=1}^{L_{ali}}\frac{1}{1+\frac{d_i^2}{\left(1.34\sqrt[3]{L-15}-1.8\right)^2}} \qquad (4)$$

where d_i^2 is the distance between the ith residues of the query and the template after an optimal superimposition, L is the length of the query sequence, and L_{ali} is the length of the aligned regions. TM-score measures the similarity between two protein 3D structures and gives a score in the range [0,1], where the higher score means better similarity. TM-score >0.5 indicates the pair of proteins share the same fold, whereas a score <0.17 indicates random fold.

3 | RESULTS AND DISCUSSION

3.1 | Performance on Test500 set

As shown in Table 1, our threading method that includes residue-residue contact information (referred to as "This work") outperforms MUSTER as well as our baseline threading implementation that does not include contact (referred to as "This work$_{\text{NOCONTACT}}$") in predicting the top ranked model both when using native contact maps and RaptorX predicted contact maps. On an average, our method that includes contact information predicts the top ranked model with an average TM-score of 0.528 and 0.524 by using native contact maps and RaptorX predicted contact maps respectively. In terms of average TM-score of top ranked models, our work (with native contacts) outperforms MUSTER as well as our baseline threading method by 0.011 and 0.01 TM-score points respectively, whereas the differences are 0.007 and 0.006 TM-score points respectively when we evaluate our work (with RaptorX contacts) with MUSTER and our baseline threading method. Table 1 also shows that the performance of our baseline threading method is comparable to that of MUSTER. Our work, therefore, delivers consistently better average TM-score compared to MUSTER as well as our baseline threading method, indicating that the inclusion of the residue-residue contact information helps to boost the average accuracy of the top ranked predicted model.

To examine whether the performance boost attained by our work is statistically significant, we perform *t*-test of the TM-score improvements. On Test500 dataset, our work (with native contacts) is statistically significantly better at 95% confidence level compared to MUSTER (*P*-value = 0.001) and our baseline threading method (*P*-value = 0.002). Furthermore, the performance improvement of our work (with RaptorX contacts) is also statistically significant at 95% confidence level compared to MUSTER and our baseline threading method with *P*-values <0.05 (Table 1). Overall, the results indicate that the incorporating contact information yields statistically significantly better threading performance in terms of the top ranked model using both native and predicted contact maps.

Figures 2 and 3 show a head-to-head comparison of our work (referred to as "This work") with MUSTER and our baseline threading method (referred to as "This work$_{\text{NOCONTACT}}$") respectively. In Figures 2A,B, there are 12% and 11% more points, respectively, below the diagonal line, which indicates that our work identifies better alignment than MUSTER regardless of using native contacts or RaptorX predicted contacts. We also observe a similar trend in Figures 3A,B when we do a head-to-head comparison of this work with our baseline threading method. Figures 2C,D show the bimodal distribution of

TABLE 1 Performance comparison on Test500 dataset[a] based on the average TM-score of top ranked models (numbers in bold represent best in each category)

Contact source	MUSTER (*P*-value[b])	This work$_{\text{NOCONTACT}}$ (*P*-value[b])	This work
Native contact	0.517 (0.001)	0.518 (0.002)	**0.528**
RaptorX	0.517 (0.007)	0.518 (0.012)	**0.524**

[a] Excluding templates with sequence identity >0.30 to the query protein.
[b] One sample *t*-test's *P*-value of the TM-score difference compared to This work.

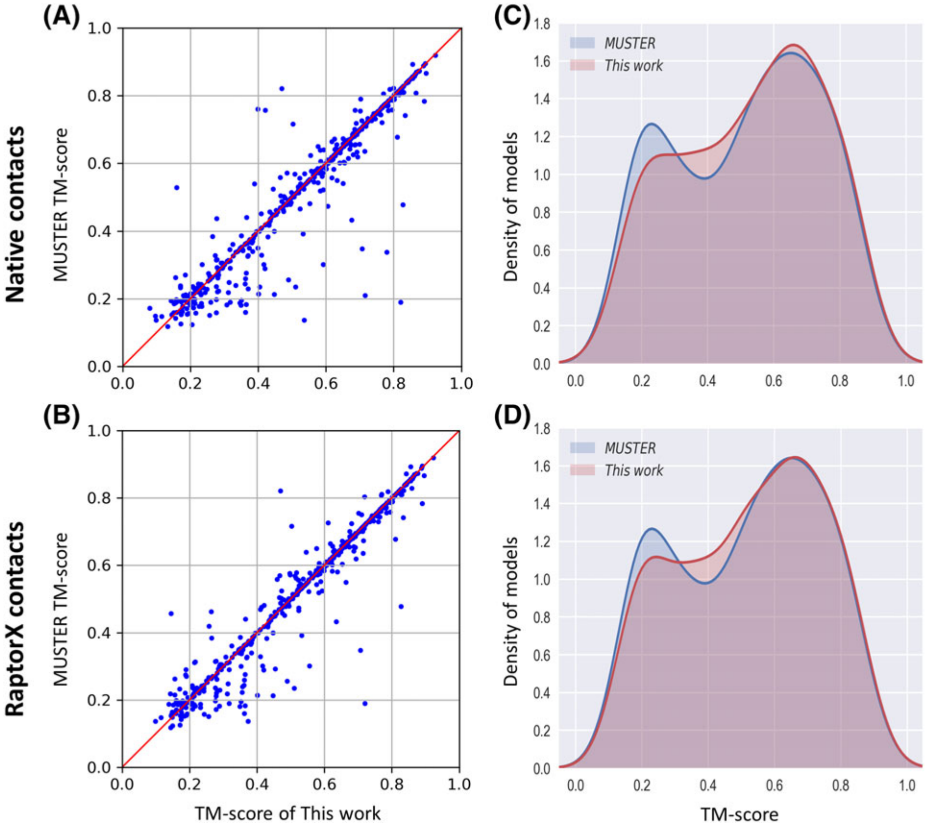

FIGURE 2 A head-to-head performance comparison of our work and MUSTER based on the accuracy of the top ranked models on Test500 dataset. A, MUSTER vs This work (with native contact maps), B, MUSTER vs This work (with RaptorX contact maps). Each point in (A) and (B) represents the TM-score of the top ranked models predicted by This work (x-axis) and MUSTER (y-axis), respectively. C, Bimodal distribution of TM-score of the top ranked models predicted by MUSTER and This work with native contact maps. D, Bimodal distribution of TM-score between MUSTER and This work with RaptorX predicted contact maps. Templates with sequence identity >30% to the query protein are excluded [Color figure can be viewed at wileyonlinelibrary.com]

TM-score of top one models predicted by our work and MUSTER. These figures illustrate bimodality due to the diversity of Test500 data set, which has roughly a balanced combination of easy targets (both methods predict the top one model with TM-score >0.5) and hard targets (both methods predict the top one model with TM-score ≤ 0.5). In Figure 2C, the highest peak of our work is slightly higher than that of MUSTER, which means the density of models of this work (with native contacts) in the TM-score range [0.6,0.8] is more than MUSTER. We also observe that the second highest peak of our work is lower than MUSTER in the TM-score range [0.1,0.3], which demonstrates a fewer number of models with low TM-score are predicted by our work compared to MUSTER. Moreover, in the TM-score range [0.3,0.5], the density of models of our work is more than MUSTER. Figure 2D shows a similar trend with the density of models of our work (with RaptorX contacts) and MUSTER being comparable for easy targets, and for a TM-score range around [0.3,0.5], our work predicts more models than MUSTER as opposed to the TM-score range [0,0.3].

A similar trend is observed in Figures 3C,D, where we plot the TM-score distribution of the top one model predicted by both of our threading approaches—one using contact information while the other does not. For easy targets and for the TM-score range [0.3,0.5], the density of models of our work (referred to as "This work") is more than that of our baseline threading method (referred to as "This work$_{\text{NOCONTACT}}$"); whereas the opposite trend is shown for TM-score range [0,0.3], which indicates that our work predicts more models with higher accuracy than that of our baseline threading method that does not include contact information. In summary, the results demonstrate that inclusion of residue-residue contact information boosts protein threading by shifting its performance distributions towards higher accuracy.

3.2 | Performance on PSICOV-150 set

Next, we compare the performance of our work with CONFOLD2, a state-of-the-art contact driven ab initio folding method, using the PSICOV-150 dataset after excluding four targets for which CONFOLD2 fails to report the performance. As shown in Table 2, our work consistently outperforms CONFOLD2. For targets with predicted contact map precision ≤50% (122/146 cases), our work (using Cutoff-1) achieves a mean TM-score of 0.628 compared to 0.573 of CONFOLD2, whereas we achieve mean TM-score of 0.627 and 0.617 using Cutoff-2 and Cutoff-3 respectively. For the remaining targets with high precision contact maps (24/146 cases), though there is an improvement in mean TM-score of both methods, our work (using Cutoff-1) outperforms CONFOLD2 by achieving a mean TM-score of 0.691, which is about 0.07 TM-score points better than that of CONFOLD2. The increase in TM-score reaches to 0.068 using Cutoff-2 or Cutoff-3. Considering all targets, our work (using Cutoff-1) predicts top ranked models with an average TM-score of 0.638 that is 0.058 TM-score points more than that of

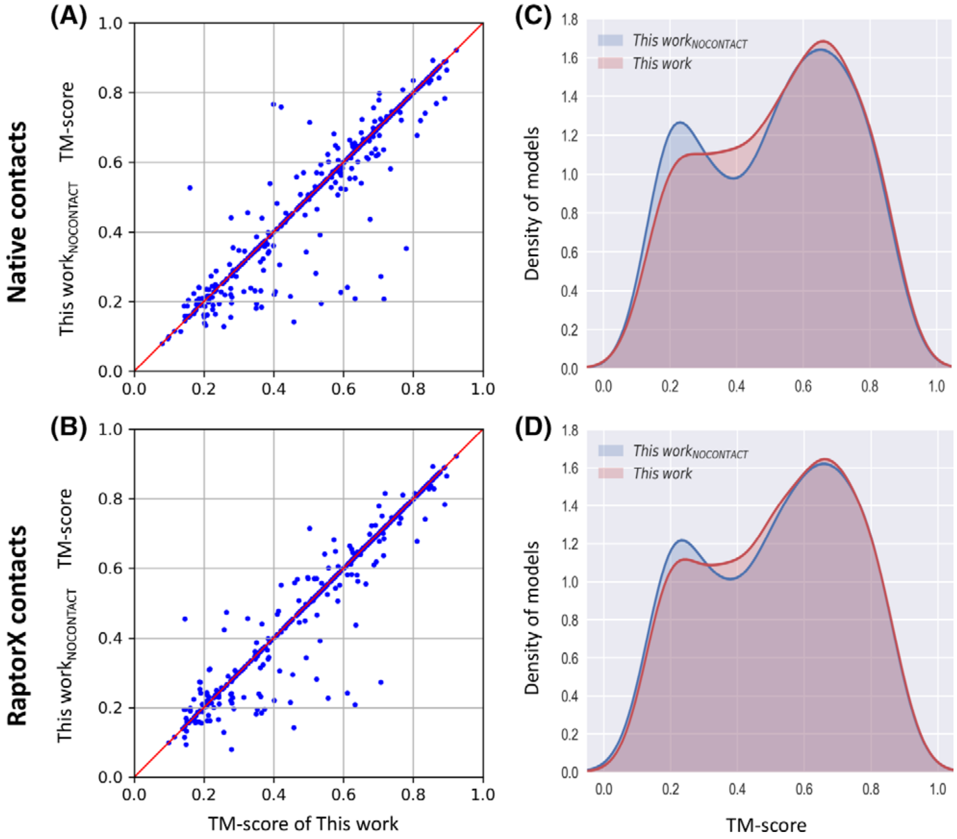

FIGURE 3 TM-score comparison between our work and our baseline threading method for the top ranked model on Test500 dataset. This work$_{NOCONTACT}$ refers to our baseline threading method, which does not use contact information; This work refers to our work. A, This work$_{NOCONTACT}$ vs This work (with native contact maps), (B) This work$_{NOCONTACT}$ vs This work (with RaptorX contact maps). Each point in (A) and (B) represents the TM-score of the top ranked models predicted by This work (x-axis) and This work$_{NOCONTACT}$ (y-axis), respectively. C, Bimodal distribution of TM score of the top ranked models predicted by This work$_{NOCONTACT}$ and This work with native contact maps, D, Bimodal distribution of TM-score between This work$_{NOCONTACT}$ and This work with RaptorX predicted contact maps [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Performance comparison of our work against CONFOLD2 on PSICOV-150 dataset[a] based on the average TM-score of top ranked predicted models (numbers in bold represent best in each category)

Homology cutoff	Contact Precision[b]	CONFOLD2(*P*-value[c])	This work
Cutoff-1[d]	≤50%[e]	0.573 (0.004)	**0.628**
	>50%	0.621 (0.08[h])	**0.691**
	All	0.580 (0.0009)	**0.638**
Cutoff-2[f]	≤50%	0.573 (0.008)	**0.627**
	>50%	0.621 (0.087[h])	**0.689**
	All	0.580 (0.002)	**0.637**
Cutoff-3[g]	≤50%	0.573 (0.032)	**0.617**
	>50%	0.621 (0.087[h])	**0.689**
	All	0.580 (0.009)	**0.628**

[a] Considered 146 targets as the published work of CONFOLD2 fails to report results for four targets namely: 1atzA, 1bkrA, 1c44A, 1c52A whereas our work predicts the top ranked model with a mean TM-score >0.53, irrespective of different cutoffs.

[b] Calculated over all the contacts with probability of being in contact is at least 0.5 and showing here as a percentage.

[c] One sample *t*-test's *P*-value of the TM-score difference of our work.

[d] Excluding templates with sequence identity >0.30 to the query protein.

[e] Calculated over 122 targets.

[f] Excluding SCOP family and sequence identity >0.30 to the query protein.

[g] Excluding SCOP family and superfamily, and sequence identity >0.30 to the query protein.

[h] Calculated over only 24 targets, which might not be sufficiently large sample size to meaningfully evaluate statistical significance.

CONFOLD2. We achieve average TM-score of 0.637 and 0.628 using Cutoff-2 and Cutoff-3 respectively. It is also worth mentioning that while CONFOLD2 fails to report the performance for four targets 1atzA, 1bkrA, 1c44A, and 1c52A, our work predicts the top ranked model with an average TM-score >0.53, irrespective of different homology cutoffs.

We also perform *t*-test of the TM-score difference to examine whether the improvement attained by our work is statistically significant. As reported in Table 2, for targets with contact precision ≤50% and for all targets, our work is statistically significantly better than CONFOLD2 at 95% confidence level in all cutoffs. For targets with low precision contacts (≤ 50%), *P*-value is 0.032 in most stringent cutoff (Cutoff-3) compared to 0.004 in Cutoff-1 and 0.008 in Cutoff-2. Similarly, for all targets, *P*-value is 0.009 in Cutoff-3 compared to 0.0009 in Cutoff-1 and 0.002 in Cutoff-2. For targets with high accuracy contact maps with precision >50%, the average TM-score of our work is slightly better than CONFOLD2, but the difference is not statistically significant at 95% confidence level. It should be noted here that there are only 24 targets with high accuracy contact maps with precision >50% and a sample size of only 24 may not be large enough for a meaningful statistical significance test. It is also worth mentioning that exclusion of SCOP family, superfamily along with 30% sequence identity cutoff to the query protein is a very stringent cutoff for our work to compare it with CONFOLD2. Overall, these tests demonstrate that inclusion of

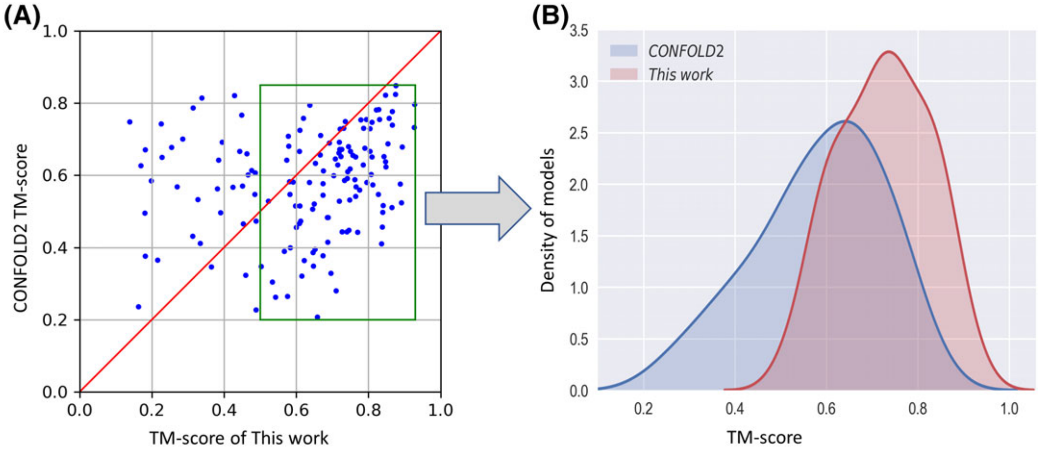

FIGURE 4 A head-to-head performance comparison between our work (using Cutoff-3) and CONFOLD2 based on TM-score of the top ranked models on PSICOV-150 dataset. A, Each point represents the TM-score of the top ranked models predicted by our work (x-axis) and CONFOLD2 (y-axis), respectively. For 109 test proteins, our work predicts top ranked models with TM-score ≥ 0.5, which is shown in quadrilateral. B, TM-score distribution of the top ranked models predicted by This work, and CONFOLD2 over the 109 test proteins [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Performance comparison over 20 full-length CASP13 targets[a] based on top ranked models by our work and two state-of-the-art contact-assisted threading methods (numbers in bold represent best in each category). TM-align results are included as a reference

Methods	Average TM-score	% time TM-score >0.5[b]
map_align[c]	0.39	18.2
EigenTHREADER	0.43	30.0
This work	**0.45**	**40.0**
TM-align[d]	0.67	85.0

[a]Officially released by CASP with native 3D structures on December 2018.
[b]Percentage of time the respective method predicts the correct fold (TM-score >0.5).
[c]Since map_align is too computationally expensive, we run it only on 11 CASP full-length targets (of length < 300 residues) out of 20 full-length targets and the results are based on those 11 targets.
[d]Using native 3D structures of the query proteins officially released by CASP.

TABLE 4 Performance comparison on CASP13 dataset over 32 domains[a] based on top ranked models by our work and two state-of-the-art contact-assisted threading methods (numbers in bold represent best in each category). TM-align results are included as a reference

Methods	Average TM-score	% time TM-score >0.5[b]
map_align[c]	0.36	14.3
EigenTHREADER[d]	0.38	25.0
This work	**0.39**	**28.1**
TM-align[e]	0.70	93.75

[a]CASP officially releases native 3D structures of 20 full-length targets in a total of 32 domains on December 2018.
[b]Percentage of time the respective method predicts the correct fold (TM-score >0.5).
[c]Considering only 14 CASP13 domains of length < 300 residues and values are based on those 14 domains.
[d]Considering only 28 domains because TM-score fails to calculate TM-score for the following domains: T0960-D4, T0960-D2, T0960-D1, and T0957-D2 as EigenTHREADER's predicted models do not have any common residues to the native domains.
[e]Using native 3D structures of the query proteins officially released by CASP.

contact information into threading yields statistically significantly better performance than contact-assisted ab initio folding.

Figure 4A shows a head-to-head comparison between both methods based on the TM-score of top ranked predicted models with our work (in Cutoff-3) significantly outperforming CONFOLD2. We chose homology Cutoff-3 for this comparison because it represents the most stringent homology cutoff. The 69.2% data points (in Figure 4A) lie below the diagonal line, which indicates that our work predicts a significantly better accurate top ranked model than CONFOLD2. In 28 cases, CONFOLD2 predicts the top ranked model with a TM-score <0.5, while our work successfully predicts the correct fold (top ranked model with TM-score >0.5). On the other hand, CONFOLD2 predicts the correct folds (top ranked model with TM-score >0.5) for 25 targets, but our work fails. Of the 146, our work predicts model from the top-ranked

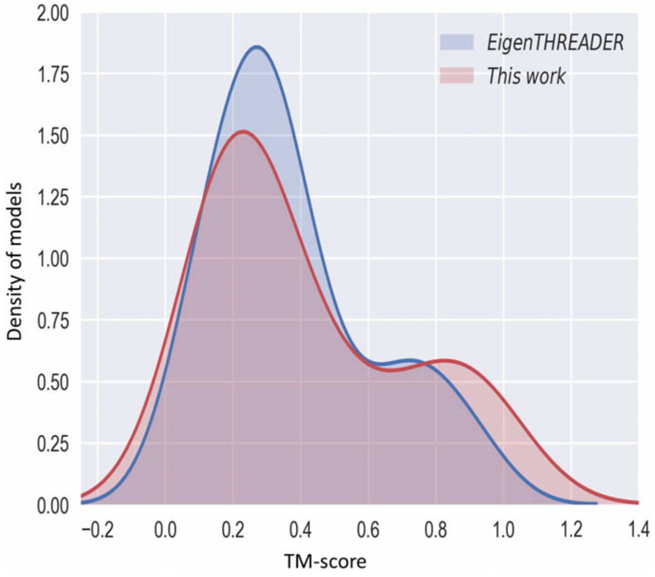

FIGURE 5 TM-score distribution of the top ranked models predicted by This work and EigenTHREADER over 28 CASP released domains. We exclude four domains for which native domains do not have any residue match with EigenTHREADER's predicted models [Color figure can be viewed at wileyonlinelibrary.com]

template with a TM-score ≥ 0.5 for 109 test proteins (marked in quadrilateral in Figure 4A). Out of these 109 test proteins, our work attains better TM-score than CONFOLD2 for 97 cases. Figure 4B shows the TM-score distribution of top ranked models predicted by our work (in Cutoff-3) and CONFOLD2. The highest peak of the distribution of our work is larger as well as skewed toward the higher accuracy (right) side compared to CONFOLD2, indicating that our work predicts more models with better accuracy than the other method

3.3 | Performance on CASP13 set

We further evaluate the performance of our work on the CASP13 dataset consisting of 20 full-length targets resulting in a total of 32 domains officially released with native 3D structures so far. For fair performance evaluation, same template library and same non-redundant (nr) sequence database are used by all competing methods and both the databases were created before CASP13 started on May 1, 2018. Furthermore, we use the same RaptorX predicted contact maps for all competing methods. We use the default settings with 65 threads to run map_align for each target. As map_align is highly computationally expensive (Refer to Table S3 [E] for target-by-target CPU hours needed by map_align), we only consider 11 full-length targets of length < 300 residues resulting in a total of 14 domains. For EigenTHREADER, we use the default setting except setting values of the parameters c (distance threshold for contact map) as 8 Å and t (number of eigenvectors) as seven. As we run Al-eigen using seven eigenvectors in our work, the same number of eigenvectors is used for EigenTHREADER to make a fair performance comparison.

As shown in Table 3, our work (referred to as "This work") outperforms map_align and EigenTHREADER over 20 full-length targets in terms of average TM-score of top ranked models and the percentage of time the top ranked model is predicted with a TM-score >0.5 (i.e. with correct fold). Our work predicts the top ranked model with an average TM-score of 0.45 compared to that of 0.43 of EigenTHREADER and 0.39 of map_align. Moreover, 40% of the time our work predicts the similar fold (top ranked model with a TM-score >0.5) which is about 21% and 10% better than map_align and EigenTHREADER respectively. It is worth mentioning that map_align's performance is analyzed over 11 full-length targets having length < 300 residues instead of 20 targets due to expensive computation. To investigate whether the performance of

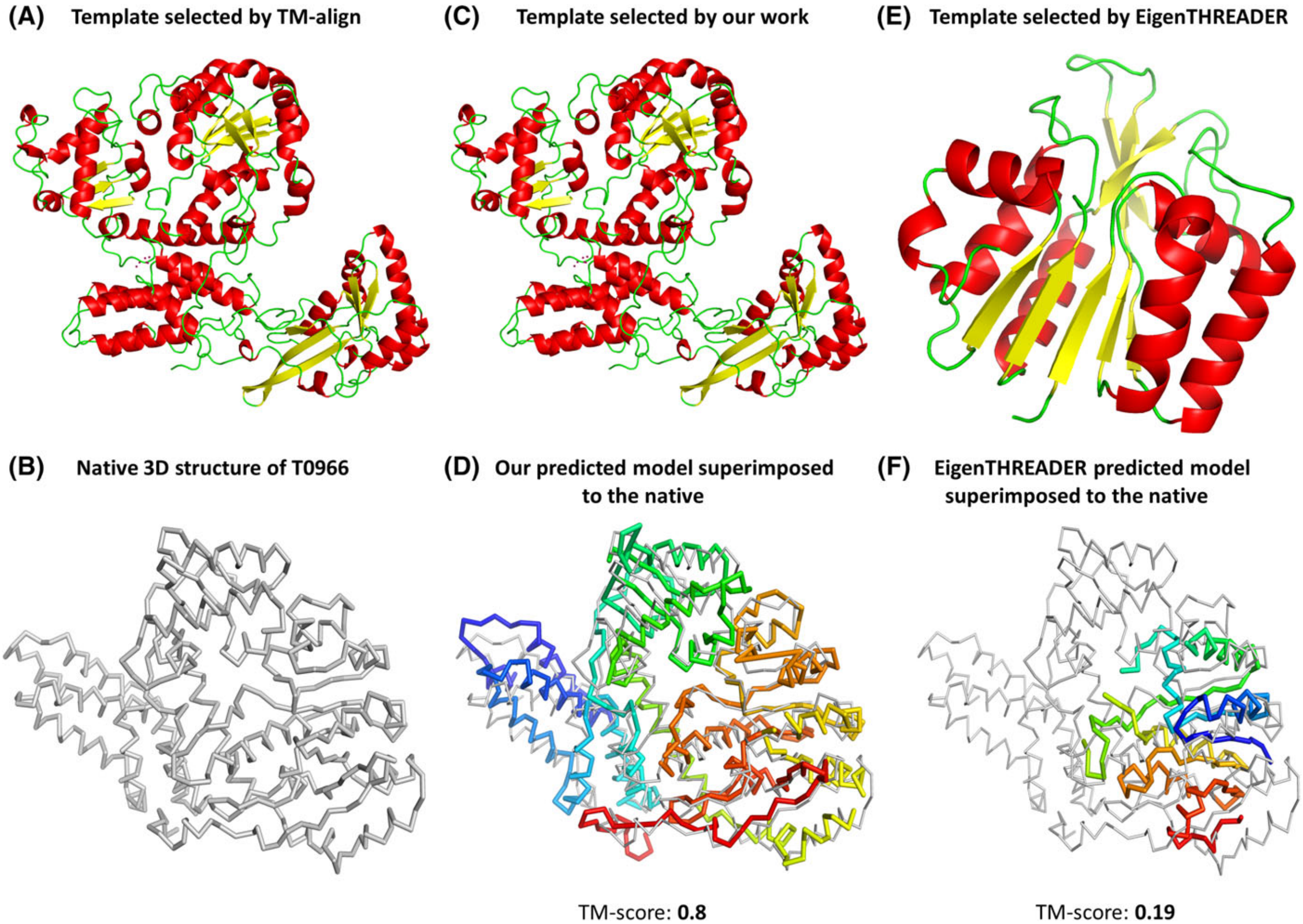
(A) Template selected by TM-align
(C) Template selected by our work
(E) Template selected by EigenTHREADER
(B) Native 3D structure of T0966
(D) Our predicted model superimposed to the native
(F) EigenTHREADER predicted model superimposed to the native
TM-score: 0.8
TM-score: 0.19

FIGURE 6 Performance of our work and EigenTHREADER on target T0966. A, TM-align finds 2ebfX as the best template in the template library with a TM-score (by TM-align) of 0.84, B, Experimental structure of T0966, C, Similar to TM-align, our work finds the same template (i.e. 2ebfX) as the top ranked template, D, Structural alignment between the top ranked model predicted by our work (in thick) with a TM-score of 0.8 and the native structure of T0966 (in thin). E, EigenTHREADER finds 1phpA2 as the top ranked template with a TM-score (by TM-align) of 0.36 for the same target. F, Structural alignment between the top ranked model predicted by EigenTHREADER (in thick) with a TM-score of 0.19 and the native structure of T0966 (in thin)

contact-assisted threading methods is optimal, we run TM-align[63] using CASP13 officially released native 3D structure of the query protein. TM-align performs structural superposition between the query protein and the template library in order to select the optimal template. The average TM-score (by TM-align) of the best template selected by TM-align is 0.67 and 85% of the time it finds the correct fold, revealing the gap between top templates found by the state-of-the-art contact-assisted threading methods and the best possible templates.

In Table 4, we report the results of head-to-head comparisons between all competing methods over 32 domains based on TM-score of top ranked models. For EigenTHREADER, we exclude four domains namely: T0960-D4, T0960-D2, T0960-D1, and T0957-D2 because TM-score tool fails to superimpose EigenTHREADER's predicted models with the native domains as there are no common residues. Moreover, the results of map_align are based on 14 domains of length < 300 residues. On this dataset, our work outperforms EigenTHREADER and map_align in terms of average TM-score of top ranked models and the percentage of time it finds the correct fold (top ranked model predicted with a TM-score >0.5). On an average, our work achieves a TM-score of 0.39 compared to 0.38 of EigenTHREADER and 0.36 of map_align. Moreover, 28.1% of the time our work finds the correct fold (TM-score >0.5) which is about 3% and 14% better than EigenTHREADER and map_align respectively. It is worth mentioning that our work achieves an average TM-score of 0.415 and 32% of the time it predicts the correct fold (TM-score >0.5) by considering 28 domains like EigenTHREADER. Considering 32 domains, once again 93.75% of the time TM-align finds the correct fold (TM-score > 0.5) including the average TM-score of 0.7, indicating that there is a large room for improvement. Figure 5 shows the TM-score distribution of the top ranked model predicted by our work and EigenTHREADER over 28 domains. For a low TM-score range, the density of models of our work is lower than that of EigenTHREADER as opposed to a higher TM-score range, which indicates that our work predicts more models with better accuracy compared to EigenTHREADER.

As a representative example, we present a case study on CASP13 target T0966 with 494 residues where SPIDER3 predicts the secondary structure with the Q3 accuracy of 87.6%. It is a single domain TBM-hard target as per CASP official domain classification. Figure 6B shows that TM-align detects the template (PDB ID: 2ebfX) that has a correct fold to T0966 with a TM-score (by TM-align) of 0.84 by using the native 3D structure of T0966 released by CASP. We include TM-align as a reference to see the best possible template for the target T0966 can be found in our template library. Figure 6C shows that our method detects the same template as the top template and predicts the top ranked model using that template with a TM-score of 0.8 to the native structure (Figure 6D). However, EigenTHREADER finds a different template (PDB ID: 1phpA2), which has an incorrect fold with TM-score (by TM-align) of 0.36 and predicts the top one model with a TM-score of 0.19 to the native structure (Figure 6E,F, respectively).

We also evaluate the performance of our work against RaptorX-TBM (Refer to Table S3A and S3 (B) for target-by-target performance). For the 20 full-length targets (and 32 domains), RaptorX-TBM predicts the top ranked model with a mean TM-score of 0.57 (and 0.55) that is 0.12 (and 0.16) TM-score points more than our work. Moreover, 50% of the time RaptorX-TBM finds the correct fold that is 10% more than that of our work (considering 20 full-length targets). It should be noted here that RaptorX-TBM uses inter-residue distance maps that is considered more informative than inter-residue contact maps in predicting protein 3D structure.[53] Moreover, the comparison of our work with RaptorX-TBM may not be fair since threading performance is directly dependent on the template library and RaptorX-TBM potentially is based on a different template library compared to our work.

4 | CONCLUSION

In this article, we analyze whether the inclusion of residue-residue contact information improves the performable of protein threading. We develop a new threading method by combining sequential and structural features, and subsequently incorporate residue-residue contact information in the form of contact map overlap (CMO) score. We benchmark our work on three different datasets consisting of a diverse set of protein targets of varied difficulties. Experimental results demonstrate that our work outperforms popular threading method MUSTER as well as our baseline threading approach that does not utilize contact information, state-of-the-art contact-assisted ab initio folding method CONFOLD2, and latest contact-assisted threading methods EigenTHREADER and map_align. Collectively, our study indicates that the inclusion of contact information improves protein threading. Furthermore, we show that our work attains higher accuracy compared to contact-assisted ab initio folding. Very recently, distance based threading method DeepThreader shows the usefulness of utilizing inter-residue distances instead of contacts for improving protein 3D structure prediction. Our work, therefore, can be further improved in the future by including distance information rather than binary contacts as well as combining multiple sources of contact or distance information.

ACKNOWLEDGMENT

The work is partially supported by the Auburn University new faculty start-up grant to DB. The authors would like to thank Rahmatullah Roche for helpful discussions.

ORCID

Debswapna Bhattacharya https://orcid.org/0000-0002-9630-0141

REFERENCES

1. Baker D, Sali A. Protein structure prediction and structural genomics. *Science*. 2001;294:93-96.
2. Dill KA, MacCallum JL. The protein-folding problem, 50 years on. *Science*. 2012;338:1042-1046.
3. Jones DT, Taylor WR, Thornton JM. A new approach to protein fold recognition. *Nature*. 1992;358:86-89.
4. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)--round x. *Proteins*. 2014;82(Suppl 2):1-6.
5. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res*. 2000;28:235-242.
6. Yang Y, Faraggi E, Zhao H, Zhou Y. Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*. 2011;27:2076-2082.

7. Ma J, Wang S, Zhao F, Xu J. Protein threading using context-specific alignment potential. *Bioinformatics*. 2013;29:i257-i265.
8. Peng J, Xu J. Low-homology protein threading. *Bioinformatics*. 2010; 26:i294-i300.
9. Bowie JU, Lüthy R, Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*. 1991;253:164-170.
10. Kinch LN, Grishin NV. Evolution of protein structures and functions. *Curr Opin Struct Biol*. 2002;12:400-408.
11. Zhang Y, Skolnick J. The protein structure prediction problem could be solved using the current PDB library. *Proc Natl Acad Sci U S A*. 2005;102:1029-1034.
12. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. *J Bioinform Comput Biol*. 2003;1:95-117.
13. Zhou H, Zhou Y. Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins*. 2004;55:1005-1013.
14. Söding J. Protein homology detection by HMM-HMM comparison. *Bioinformatics*. 2005;21:951-960.
15. Peng J, Xu J. Boosting protein threading accuracy. *Res Comput Mol Biol*. 2009;5541:31-45.
16. Ma J, Peng J, Wang S, Xu J. A conditional neural fields model for protein threading. *Bioinformatics*. 2012;28:i59-i66.
17. Jaroszewski L, Rychlewski L, Li Z, Li W, Godzik A. FFAS03: a server for profile--profile sequence alignments. *Nucleic Acids Res*. 2005;33: W284-W288.
18. Rychlewski L, Jaroszewski L, Li W, Godzik A. Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci*. 2000;9:232-241.
19. Cheng J, Baldi P. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*. 2006;22:1456-1463.
20. Marti-Renom MA, Madhusudhan MS, Sali A. Alignment of protein sequences by their profiles. *Protein Sci*. 2004;13:1071-1087.
21. Ginalski K, Pas J, Wyrwicz LS, von Grotthuss M, Bujnicki JM, Rychlewski L. ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acids Res*. 2003; 31:3804-3807.
22. Zhou H, Zhou Y. Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins*. 2005;58:321-328.
23. Jones DT. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol*. 1999;287:797-815.
24. Wu S, Zhang Y. LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Res*. 2007;35:3375-3382.
25. Gniewek P, Kolinski A, Kloczkowski A, Gront D. BioShell-threading: versatile Monte Carlo package for protein 3D threading. *BMC Bioinformatics*. 2014;15:22.
26. Rost B, Schneider R, Sander C. Protein fold recognition by prediction-based threading. *J Mol Biol*. 1997;270:471-480.
27. Olmea O, Rost B, Valencia A. Effective use of sequence correlation and conservation in fold recognition. *J Mol Biol*. 1999;293:1221-1239.
28. Wu S, Zhang Y. MUSTER: improving protein sequence profile–profile alignments by using multiple sources of structure information. *Proteins*. 2008;72:547-556.
29. Lobley A, Sadowski MI, Jones DT. pGenTHREADER and pDomTHREADER: new methods for improved protein fold recognition and superfamily discrimination. *Bioinformatics*. 2009;25:1761-1767.
30. Peng J, Xu J. A multiple-template approach to protein threading. *Proteins*. 2011;79:1930-1939.
31. Wu S, Zhang Y. Recognizing protein substructure similarity using segmental threading. *Structure*. 2010;18:858-867.
32. Xu Y, Xu D. Protein threading using PROSPECT: design and evaluation. *Proteins*. 2000;40:343-354.
33. Ma J, Wang S, Wang Z, Xu J. MRFalign: protein homology detection through alignment of Markov random fields. *PLoS Comput Biol*. 2014; 10:e1003500.
34. Yan R, Xu D, Yang J, Walker S, Zhang Y. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Sci Rep*. 2013;3(2619).
35. Jones DT, Singh T, Kosciolek T, Tetchner S. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*. 2015;31:999-1006.
36. Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput Biol*. 2017;13:e1005324.
37. Wang S, Li Z, Yu Y, Xu J. Folding membrane proteins by deep transfer learning. *bioRxiv*. 2017;5:181628. https://doi.org/10.1101/181628.
38. Wang S, Sun S, Xu J. Analysis of deep learning methods for blind protein contact prediction in CASP12. *bioRxiv*. 2017;86(S1):181586. https://doi.org/10.1101/181586.
39. Wang S, Li W, Zhang R, Liu S, Xu J. CoinFold: a web server for protein contact prediction and contact-assisted protein folding. *Nucleic Acids Res*. 2016;44:W361-W366.
40. Jones DT, Buchan DWA, Cozzetto D, Pontil M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*. 2012;28:184-190.
41. Seemayer S, Gruber M, Söding J. CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*. 2014;30:3128-3130.
42. Kaján L, Hopf TA, Kalaš M, Marks DS, Rost B. FreeContact: fast and free software for protein contact prediction from residue co-evolution. *BMC Bioinformatics*. 2014;15:85.
43. Kamisetty H, Ovchinnikov S, Baker D. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era. *Proc Natl Acad Sci U S A*. 2013;110:15674-15679.
44. Buchan DWA, Jones DT. Improved protein contact predictions with the MetaPSICOV2 server in CASP12. *Proteins: Structure, Function, and Bioinformatics*. 2018;86:78-83.
45. Adhikari B, Hou J, Cheng J. DNCON2: improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*. 2018;34:1466-1472.
46. He B, Mortuza SM, Wang Y, Shen H-B, Zhang Y. NeBcon: protein contact map prediction using neural network training coupled with naïve Bayes classifiers. *Bioinformatics*. 2017;33:2296-2306.
47. Hanson J, Paliwal K, Litfin T, Yang Y, Zhou Y. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*. 2018;34:4039-4045.
48. Hopf TA, Colwell LJ, Sheridan R, Rost B, Sander C, Marks DS. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*. 2012;149:1607-1621.
49. Adhikari B, Bhattacharya D, Cao R, Cheng J. CONFOLD: residue-residue contact-guided ab initio protein folding. *Proteins*. 2015;83:1436-1449.
50. Adhikari B, Cheng J. CONFOLD2: improved contact-driven ab initio protein structure modeling. *BMC Bioinformatics*. 2018;19:22.
51. Buchan DWA, Jones DT. EigenTHREADER: analogous protein fold recognition by efficient contact map threading. *Bioinformatics*. 2017; 33:2684-2690.
52. Ovchinnikov S, Park H, Varghese N, et al. Protein structure determination using metagenome sequence data. *Science*. 2017;355:294-298.
53. Zhu J, Wang S, Bu D, Xu J. Protein threading using residue co-variation and deep learning. *Bioinformatics*. 2018;34:i263-i273.
54. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. 1983;22:2577-2637.
55. Heffernan R, Yang Y, Paliwal K, Zhou Y. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*. 2017;33:2842-2849.
56. Altschul SF, Madden TL, Schäffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25:3389-3402.
57. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*. 1970;48:443-453.
58. Di Lena P, Fariselli P, Margara L, Vassura M, Casadio R. Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*. 2010;26:2250-2258.

59. Yang J, Yan R, Roy A, Xu D, Poisson J, Zhang Y. The I-TASSER suite: protein structure and function prediction. *Nat Methods*. 2015;12: 7-8.
60. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*. 1995;247:536-540.
61. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*. 2004;57:702-710.
62. Xu J, Zhang Y. How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics*. 2010;26:889-895.
63. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*. 2005;33:2302-2309.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Bhattacharya S, Bhattacharya D. Does inclusion of residue-residue contact information boost protein threading? *Proteins*. 2019;1–11. https://doi.org/10.1002/prot.25684