

Structural bioinformatics

# CATHER: a novel threading algorithm with predicted contacts

Zongyang Du<sup>1,†</sup>, Shuo Pan<sup>1,†</sup>, Qi Wu<sup>1</sup>, Zhenling Peng<sup>2</sup> and Jianyi Yang<sup>1,\*</sup>

<sup>1</sup>School of Mathematical Sciences, Nankai University, Tianjin 300071, China and <sup>2</sup>Center for Applied Mathematics, Tianjin University, Tianjin 300072, China

\* To whom correspondence should be addressed.

† The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Arne Elofsson

Received on April 30, 2019; revised on October 31, 2019; editorial decision on November 19, 2019; accepted on November 28, 2019

## Abstract

**Motivation:** Threading is one of the most effective methods for protein structure prediction. In recent years, the increasing accuracy in protein contact map prediction opens a new avenue to improve the performance of threading algorithms. Several preliminary studies suggest that with predicted contacts, the performance of threading algorithms can be improved greatly. There is still much room to explore to make better use of predicted contacts.

**Results:** We have developed a new contact-assisted threading algorithm named CATHER using both conventional sequential profiles and contact map predicted by a deep learning-based algorithm. Benchmark tests on an independent test set and the CASP12 targets demonstrated that CATHER made significant improvement over other methods which only use either sequential profile or predicted contact map. Our method was ranked at the Top 10 among all 39 participated server groups on the 32 free modeling targets in the blind tests of the CASP13 experiment. These data suggest that it is promising to push forward the threading algorithms by using predicted contacts.

**Availability and implementation:** <http://yanglab.nankai.edu.cn/CATHER/>.

**Contact:** yangjy@nankai.edu.cn

**Supplementary information:** [Supplementary data](#) are available at *Bioinformatics* online.

## 1 Introduction

The prediction of protein structure from its sequence is one of the most important and challenging problems in the field of computational biology (Dill and MacCallum, 2012). Many methods have been developed for protein structure prediction, which can be broadly divided into two categories: free modeling (FM, also called *ab initio* modeling) and template-based modeling (TBM).

FM methods attempt to predict the structure based on physical principle. With the assumption that the native structures have the lowest energy, they try to find conformations with low energies. Representative FM methods include Rosetta (Simons *et al.*, 1997) and QUARK (Xu and Zhang, 2012). Before the 13th Critical Assessment of protein Structure Prediction (CASP13), it was believed that FM methods only worked for short proteins (<200 amino acids) and were limited due to the lower model quality and requirement of longer running time. However, this was changed at the time of CASP13 due to the usage of deep learning-based contact and/or distance predictions as restraints to guide structure folding (Greener *et al.*, 2019; Hou *et al.*, 2019; Senior *et al.*, 2019; Xu, 2019; Xu and Wang, 2019; Zheng *et al.*, 2019). Thanks to this advance, it becomes possible for FM methods to fold protein structure with high accuracy, even with a personal computer (Xu, 2019).

TBM methods, including homology modeling and protein threading (Bowie *et al.*, 1991; Jones *et al.*, 1992), are based on the fact that proteins' tertiary structures are much more conserved than their sequences. So they aim to find homologous templates of the query sequence and predict the query structure based on the query-template alignment. Representative TBM methods include FFAS (Jaroszewski *et al.*, 2011), HHpred (Soding *et al.*, 2005), Phyre (Kelley and Sternberg, 2009), SPARKS-X (Yang *et al.*, 2011), RAPTORX (Kallberg *et al.*, 2012), I-TASSER (Yang *et al.*, 2015), etc. TBM methods have been widely used by the biologists due to their superior performance.

The accuracy of TBM methods is decided by both sequence-template alignment and the selection of template structure, which critically depend on the design of the scoring function in dynamic programming. Most threading methods such as HHpred (Soding, 2005), SPARKS-X (Yang *et al.*, 2011) and MUSTER (Wu and Zhang, 2008) use sequential information, including sequence profile, secondary structure, solvent accessibility and torsion angles, to build scoring functions. With the pioneer work by the Xu group (Wang *et al.*, 2017), the accuracy of protein contact map prediction boosted significantly, due to the increase of sequence data and advance in deep learning algorithms (Adhikari *et al.*, 2018; Kandathil *et al.*, 2019; Li *et al.*, 2019; Schaarschmidt *et al.*, 2018; Shrestha

et al., 2019; Wu et al., 2019). This makes it possible to improve threading programs by including the predicted contacts into the scoring function.

The predicted contacts have been used in a few threading methods. EigenTHREADER (Buchan and Jones, 2017) and map\_align (Ovchinnikov et al., 2017) make use of contact maps predicted by MetaPSICOV (Jones et al., 2015) and GREMLIN (Ovchinnikov et al., 2014), respectively. DeepThreader (Zhu et al., 2018) takes one step further by using predicted distances by RaptorX-Contact (Wang et al., 2017) rather than the binary contacts. We note that the predicted contacts in some of the contact-based algorithms are not used together with the sequential information. This may degrade their performance when the sequential information is good enough while the predicted contacts are not accurate.

In this article, to make use of both the sequential information and the contact information, we developed a new threading method called CATHER (Contact-Assisted ThreadER) that combines sequential information and contact map together in the design of the scoring function in both sequence-template alignment and templates ranking. Experimental tests show that CATHER outperforms both conventional sequential information-based methods such as HHpred, MUSTER and SPARKS-X and contact-based methods EigenTHREADER and map\_align. Blind tests in the CASP13 experiments show that CATHER achieves satisfactory modeling results.

## 2 Materials and methods

### 2.1 Benchmark datasets

From our previous assessment work of contact prediction methods (Wuyun et al., 2018), we got 680 non-redundant (<25% pairwise sequence identity) single-domain globular proteins. These proteins share no sequence homology with the CASP6-CASP11 targets and were divided into three subsets: easy (344), medium (105) and hard (231). We randomly chose 40 easy targets, 60 medium targets and 100 hard targets as the training set, and the remaining 480 targets as one of our test sets (denoted by Test480). It includes 304, 45 and 131 easy, medium and hard targets, respectively. Note that to speed up the training, the size of the training set was smaller than the test set. We did try to increase the size of training set to 400, but the performance of our method did not change much. In addition, the CASP12 dataset (83 domains from 60 targets) was used as another independent test, which consists of 38 FM, 13 TBM/FM and 32 TBM domains.

The template library is the same as that used by I-TASSER (Yang and Zhang, 2015a), which can be downloaded from its official website. To fairly evaluate the performance of methods, all templates released after CASP12 are excluded. In addition, templates sharing >30% sequence identity with the query were removed during training and testing on PDB680.

### 2.2 Generation of contact map

The contact map of the query sequence is predicted by our in-house method MapPred (Wu et al., 2019), which was ranked at the Top 10 out of 46 participating methods in CASP13. MapPred is a deep learning-based contact prediction method that predicts the contacts using deep multiple sequence alignments generated from the metagenome sequence database. According to Wu et al. (2019), the training set of MapPred does not contain any CASP12 sequences or sequence homologs. The input of MapPred is the query sequence and the output includes the contact probabilities for all residue pairs in the query. We carefully chose the number of top-ranked residue pairs. Selection of more top pairs generally means higher coverage of the real map, but with lower precision. According to our training, we selected up to the top  $2L$  pairs as the input map for threading, to balance the coverage and the precision.

The contact map of each template is calculated using its native structure. The contact score for each pair of residues  $(i, j)$  is calculated using Equation (1).

$$r(i, j) = \begin{cases} 1, & \text{if } d_{ij} \leq 8\text{\AA}, \\ 1/(1 + e^{d_{ij}-8}), & \text{if } 8\text{\AA} < d_{ij} \leq 15\text{\AA}, \\ 0, & \text{if } d_{ij} > 15\text{\AA} \end{cases} \quad (1)$$

where  $d_{ij}$  is the distance between the coordinates of the residues' C- $\beta$  atoms (C- $\alpha$  for Glycine).

### 2.3 Scoring scheme for an alignment

An alignment of the query sequence and the template structure can be regarded as a mapping ( $m$ ) between  $[-1, 1, 2, 3, \dots, L_q]$  and  $[-1, 1, 2, 3, \dots, L_t]$ , where  $L_q$  and  $L_t$  are the length of the query sequence and the template structure, respectively; and  $-1$  represents gaps. The score for a sequence-template alignment in CATHER is defined as follows.

$$s = \sum_{i=1}^L [s_{\text{seq}}(i, m_i) + s_{\text{con}}(i, m_i)] + s_{\text{gap}} \quad (2)$$

where  $L$  is the length of the alignment;  $s_{\text{gap}}$  is the score of the gaps in the alignment, in which gap opening ( $g_o$ ) and extension ( $g_e$ ) penalties are applied to the inside region and are neglected for the terminal gaps; the first and the second terms are the sequential and the contact scores of the alignment, respectively, which are defined by Equations (3) and (4), respectively.

$$s_{\text{seq}}(i, m_i) = \frac{a_1}{2} \sum_{k=1}^{20} (F_c(i, k) + F_d(i, k)) \cdot L_t(m_i, k) + a_2 \cdot \delta(SS_i, SS_{m_i}) + a_3 \cdot (1 - 2|SA_i - SA_{m_i}|) \quad (3)$$

where  $F_c(i, k)$  and  $F_d(i, k)$  are the frequencies of the  $k$ th amino acid at the  $i$ th position in the multiple sequence alignment (MSA), which are generated by PSI-BLAST (Altschul et al., 1997) with an E-value cutoff of 0.001 and 1.0, respectively.  $\delta$  is a function with value 1 when the two variables are equal, and 0 otherwise. The Henikoff weighting scheme (Henikoff and Henikoff, 1994) is used in the calculation of the frequencies from the MSA.  $L_t(j, k)$  is the log-odds profile (i.e. the Position-Specific Substitution Matrix in PSI-BLAST, PSSM) of the template sequence for the  $k$ th amino acid at the  $j$ th position, obtained by PSI-BLAST with an E-value of 0.001; SS and SA are the secondary structure and the relative solvent accessibility of the corresponding residue;  $a_1$ ,  $a_2$  and  $a_3$  are weights of the corresponding components.

The contact scores of an alignment are calculated as

$$s_{\text{con}}(i, m_i) = \sum_{k \in C(i)} w_{ik} \cdot p(i, k) \cdot r(m_i, m_k) \quad (4)$$

where  $C(i)$  is the set of the residues predicted to be in contact with the  $i$ th residue in the query sequence;  $p(i, k)$  is the predicted contact score of the residue pair  $(i, k)$  in the query sequence and  $r(m_i, m_k)$  is contact score for the residue pair  $(m_i, m_k)$  in the template defined by Equation (1). Residues aligned with gaps are not considered. The parameter  $w_{ik}$  is a weight function of the sequential separation of  $i$  and  $k$ , defined by Equation (5).

$$w_{ik} = \begin{cases} 0.5, & \text{if } 6 < |i - k| \leq 12, \\ 0.75, & \text{if } 12 < |i - k| \leq 23, \\ 2, & \text{if } |i - k| > 23 \end{cases} \quad (5)$$

This weight scheme is applied because the long-range contacts (sequential distance > 23) tend to have lower predicted probabilities than the short-range ( $6 < \text{sequential distance} < 12$ ) and the medium-range ( $12 < \text{sequential distance} \leq 23$ ) contacts; but the long-range contacts are more critical in deciding a protein's structure.

### 2.4 Alignment algorithm

Finding the maximum value of Equation (2) is mathematically difficult, so we use a two-step iterative approach to find an approximately optimal solution, which is shown in Figure 1.

In the beginning, initial alignment is generated by the Needleman-Wunsch dynamic programming algorithm (Needleman

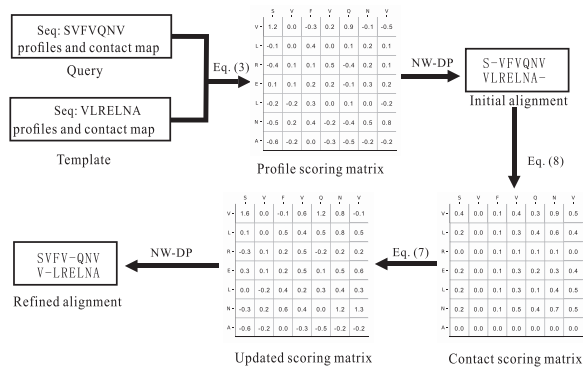


Fig. 1. The process of the alignment algorithm. The steps in the lower row are iterated until convergent or a maximum of 15 iterations are performed

and Wunsch, 1970) using the sequential profiles only. The scoring function defined by Equation (3) is used in this step. An iterative approach is then performed to refine the initial alignment using the contact map, with a similar idea used in map\_align. In the  $(k+1)$ th ( $k \geq 0$ ) iteration, the scoring function for aligning the  $i$ th residue in the query and the  $j$ th residue in the template is defined by Equation (6).

$$s^{(k+1)}(i, j) = s_{\text{seq}}(i, j) + s_{\text{con}}^{(k+1)}(i, j), \quad k \geq 0 \quad (6)$$

where the first term is defined similarly as in Equation (3) and the second term is defined by Equation (7):

$$s_{\text{con}}^{(k+1)}(i, j) = \frac{k}{k+1} s_{\text{con}}^{(k)}(i, j) + \frac{1}{k+1} s_{\text{conpot}}^{\text{aln}}(i, j), \quad k \geq 0 \quad (7)$$

where the first term is zero when  $k$  equals to zero; the second term is the *contact potential* for the  $i$ th residue in the query and the  $j$ th residue in the template, calculated based on a given alignment  $m$ :

$$s_{\text{conpot}}(i, j) = \sum_{k < i, \text{aln}(k) < j} w_{ki} \cdot p(i, k) \cdot r(j, m_k) + \sum_{k > i, \text{aln}(k) > j} w_{ki} \cdot p(i, k) \cdot r(j, m_k) \quad (8)$$

where the  $w_{ik}$  is the weight defined by Equation (5).  $p()$  and  $r()$  are the predicted and native contact score for the query and the template, respectively. The positions aligned with gaps are ignored. The idea of this equation is illustrated in Figure 2.

The iteration will stop when either the difference of the two contact score matrices is  $< 0.05$  (defined as the  $L_\infty$  norm of the difference matrix of the two matrices), or a maximum of 15 iterations have been performed. After the alignment is obtained, the profile score, the contact score and the gap score are re-calculated according to the final alignment and normalized by the square root of the aligned length, which is defined as the number of non-gap columns in the alignment.

## 2.5 Templates ranking by Z-score

There are four measurements to estimate the quality of an alignment: profile score, contact score, gap score and the aligned length. To make good use of these scores to rank templates, we propose the following way to combine these scores. We first standardize these scores by Z-score, which is defined as:

$$Z_i = \frac{R_i - \mu}{\sigma} \quad (9)$$

where  $R_i$  is the raw score of the alignment for the  $i$ th template;  $\mu$  and  $\sigma$  are mean and standard deviation of the raw scores over the pool of templates. A linear combination of the four Z-scores is calculated as the final score, which will be used to rank the alignments between the query and the templates in the library.

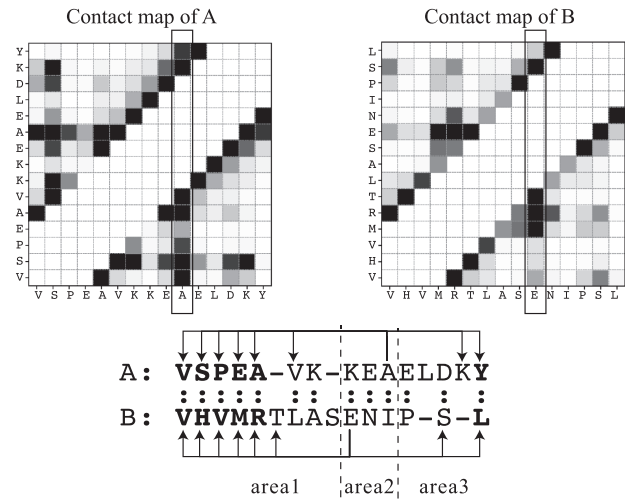


Fig. 2. An illustration of the contact potential calculation from two contact maps and an alignment. Here we aim to calculate the contact potential for aligning the residue 'A' of Sequence A and the residue 'E' of Sequence B (each marked by a rectangular in the contact map). The residue 'A' of Sequence A has contacts with eight residues (darker squares in the contact map), which are shown in bold type and marked with arrows in the alignment. It is similarly shown for Sequence B. Then the alignment is divided into three regions by the two residues in Sequences A and B, respectively. The contacts in area2 are not considered because once 'A' and 'E' are aligned, the residues before 'A' and the residues after 'E' could not be aligned to avoid cross alignments. Only the aligned contacts in area1 and area3 (shown in bold in the alignment) contribute to the contact potential

$$Z_{\text{final}} = Z_{\text{alnlen}} + z_1 \cdot Z_{\text{seq}} + z_2 \cdot Z_{\text{con}} + z_3 \cdot Z_{\text{gap}} \quad (10)$$

## 2.6 Controlled methods and performance evaluation

We compare our method with several popular threading methods including HHpred (Soding, 2005), SPARKS-X (Yang *et al.*, 2011), EigenTHREADER (Buchan and Jones, 2017), MUSTER (Wu and Zhang, 2008) and map\_align (Ovchinnikov *et al.*, 2017) on Test480 and CASP12 dataset. These programs were installed and ran locally. The contact maps used by EigenTHREADER and map\_align are the same as these used by CATHER. The top template by each threading method (including CATHER) was then submitted to MODELLER (Webb and Sali, 2014) to build a full-length model. The average TM-score (Zhang and Skolnick, 2004) over each benchmark set is used to evaluate the performance of each threading method.

## 3 Results and discussion

### 3.1 Optimization of parameters

There are five parameters in the alignment algorithm (i.e.  $a_1$ ,  $a_2$ ,  $a_3$ ,  $g_o$  and  $g_e$ ) and three parameters in the template ranking step (i.e.  $z_1$ ,  $z_2$  and  $z_3$ ), which need to be determined. The parameters in these two steps are trained separately. The parameters in the template ranking are trained after the parameters in the alignment algorithm are determined. This is meaningful because the alignment algorithm is independent of the template library, while the template ranking is not because of the Z-score calculation. When training the parameters in the alignment algorithm, we simply use the sum of the profile score, the contact score and the gap score to rank the templates.

The parameters in the alignment algorithm are trained by maximizing the average TM-score of the first model, using a grid search strategy on the training set. To train the parameters in the Z-score-based template ranking, we try to maximize the number of 'good' top models. A model is defined as good if it satisfies the following two conditions: (i) its TM-score is bigger than 0.3 unless it can be ranked at the Top 5 by the TM-score-based ranking; and (ii) the difference between its TM-score and the highest TM-score of all

**Table 1.** The average TM-scores of the top models of different methods on the test set Test480

Targets	HHP	SPX	EIG	MUS	MAP	CATHER
Easy	0.6908	0.6923	0.6819	0.728	0.6781	<b>0.7471</b>
Medium	0.376	0.4564	0.4421	0.4995	0.4176	<b>0.5426</b>
Hard	0.3268	0.3485	0.3861	0.3593	0.383	<b>0.4561</b>
Overall	0.562	0.5763	0.5787	0.6059	0.5731	<b>0.6485</b>

Note: ‘HHP’, ‘EIG’, ‘SPX’, ‘MUS’ and ‘MAP’ represent the methods HHpred, EigenTHREADER, SPARKS-X, MUSTER and map\_align, respectively. The best results are highlighted in bold type.

models is  $<0.1$ . The ranges of the parameters are  $a_1$ : [0.1, 2.5];  $a_2 \sim a_4$ : [0.1, 1];  $g_o$ : [1, 5];  $g_e$ : [0.1, 1] and  $z_1 \sim z_3$ : [1, 3]. The final parameters after optimization are:  $a_1 = 1.9$ ,  $a_2 = 0.4$ ,  $a_3 = 0.7$ ,  $g_o = 2$ ,  $g_e = 0.2$ ,  $z_1 = 1.79$ ,  $z_2 = 1.97$  and  $z_3 = 2.28$ .

### 3.2 Performance on the test set Test480

Table 1 shows the average TM-scores of the top models for all methods, over the easy, medium and hard targets of the test set Test480. It can be seen that CATHER outperforms all the other methods in all target groups. The overall TM-score of CATHER is 0.6485, which is 15.4%, 12.5%, 12.1%, 7% and 13.2% higher than HHpred, SPARKS-X, EigenTHREADER, MUSTER and map\_align, respectively. As expected, the improvement is more significant on hard targets than on easy targets. For example, on the easy targets, CATHER’s TM-score is 2.6% higher than the second-best method MUSTER, which increases to 26.9% on the hard targets. We note that EigenTHREADER has similar TM-score with map\_align. On the one hand, both methods have lower TM-scores than SPARKS-X and MUSTER on the easy and the medium targets, which suggests the importance of using sequential features. On the other hand, EigenTHREADER and map\_align have the second and the third highest TM-scores on the hard targets. This indicates that the contact map is useful when traditional methods fail to recognize templates. Our contact map alignment algorithm was inspired by map\_align but outperforms map\_align significantly on all groups of targets (Table 1). This is because our scoring scheme efficiently combines 1D and 2D information in the alignment.

In order to assess the statistical significance of the improvement of CATHER over other methods, statistical tests were performed as follows. For each benchmark dataset, we randomly split the dataset into two halves and then computed the average TM-score for each method on one of the subsets. This experiment was repeated 10 times to generate 10 paired results. The Anderson–Darling test was first used to test whether the data follow a normal distribution at 0.05 significance level. The paired  $t$ -test was applied for a normal distribution. Otherwise, the non-parametric Wilcoxon signed-rank test was utilized. The  $P$ -value returned from the test indicates the significance level of the difference between two compared methods. The  $P$ -values are listed in Supplementary Table S1, which shows that the improvement of CATHER over other methods is significant at 0.01 level.

Figure 3 presents the number of better targets for each method compared with CATHER on the dataset Test480 in terms of TM-score. It shows that CATHER has more targets with more accurate models than other compared methods. In addition, the advantage of CATHER over other methods is more significant on the medium and hard targets, for which CATHER generates better models for more than 60% targets.

In Figure 4 and Supplementary Figure S1, we show an example (domain ID: d2o0pa1) that CATHER outperforms other methods. After excluding templates sharing  $>30\%$  sequence identity with the query, the TM-scores between the native structure and the models generated by CATHER, HHpred, SPARKS-X, EigenTHREADER, MUSTER and map\_align are 0.7203, 0.5578, 0.2214, 0.5989, 0.5436 and 0.6339, respectively. The models were built with the templates 1TIIA, 1WFX\_A, 4IX9A, 2Q5TA2, 1WFXA2 and 1TIIA,

**Table 2.** The average TM-scores of the top models by different methods on the CASP12 targets

Targets	HHP	SPX	EIG	MUS	MAP	CATHER
TBM	0.7025	0.6468	0.6027	0.7079	0.5721	<b>0.7163</b>
TBM/FM	0.3498	0.3683	0.46	0.3674	0.3407	<b>0.5442</b>
FM	0.2426	0.2719	0.2989	0.2505	0.2638	<b>0.3426</b>
Overall	0.4367	0.4315	0.4413	0.4452	0.3947	<b>0.5183</b>

Note: The best results are highlighted in bold type.

respectively. Figure 4 shows the superimposition of the native structure and the predicted models. As we can see, the model by CATHER aligns better on the beta-sheets than other methods. Although the same template was detected by map\_align, our method generates better alignment resulting to more accurate models. The corresponding match of the native contact map and the contact map from predicted structure is presented in Supplementary Figure S1, which also illustrates that CATHER outperforms other methods.

### 3.3 Performance on CASP12 targets

We evaluated CATHER’s performance on 83 CASP12 domains (list available in Supplementary Table S5) by simulating the situation of the CASP12 experiment, i.e. excluding all templates in the library that are released after the date of the experiment (i.e. May 2, 2016). Since most CASP targets are multi-domain proteins, the domains for each CASP target were first parsed based on full-length threading alignments (Yang and Zhang, 2015b). Then, each predicted domain sequence was submitted to threading programs to predict the domain structure. The predicted domain structures for each target were then docked together based on a fast Monte Carlo simulation. It can be seen from Table 2 that on the TBM targets, HHpred and MUSTER have fairly good performance (TM-score  $> 0.7$ ) as CATHER. But when we focus on the TBM/FM targets and the FM targets, it is obvious that CATHER outperforms other methods by a considerable margin. On the 13 TBM/FM targets, the TM-score of our method is 55.6%, 47.8%, 18.3%, 48.1% and 59.7% higher than HHpred, SPARKS-X, EigenTHREADER, MUSTER and map\_align, respectively. On the 38 FM targets, the improvements are 41.2%, 26%, 14.6%, 36.8% and 29.9%, respectively, which is less significant than on the TBM/FM targets. This may be because the predicted contact map on the FM targets is less accurate than on the TBM/FM targets (the top  $L/5$  precision is 70.9% versus 82.2%).

Similarly, statistical test is conducted to assess the significance of the improvement made by CATHER over other methods, which is listed in Supplementary Table S2. It suggests that the results of CATHER and HHpred, CATHER and MUSTER are not statistically different on the TBM targets. However, on the TBM/FM and the FM targets, the improvement of CATHER over all the other methods is significant at the level of 0.01.

We also collected the results of the Top 5 servers from the website of CASP12: Zhang-Server, QUARK, BAKER-ROSETTASERVER, GOAL and RaptorX. GOAL is skipped as it did not submit models for two targets. The comparison results between CATHER and these four servers are presented in Supplementary Table S3. It shows that CATHER performs worse than these servers on the TBM targets. However, on the TBM/FM and the FM targets, CATHER has comparable TM-score to the best-performing Zhang-Server and higher TM-score than other servers, which may be due to the inclusion of predicted contact map information.

### 3.4 Performance in the blind test of CASP13

We participated in the CASP13 experiments with two servers. One is the above-mentioned method (group name CMA-align), the other is the I-TASSER Suite with CATHER as one of the threading methods to improve template selection (group name Yang-Server). According to the official results from the CASP organizers, our methods were ranked at the 13th (Yang-Server) and 16th

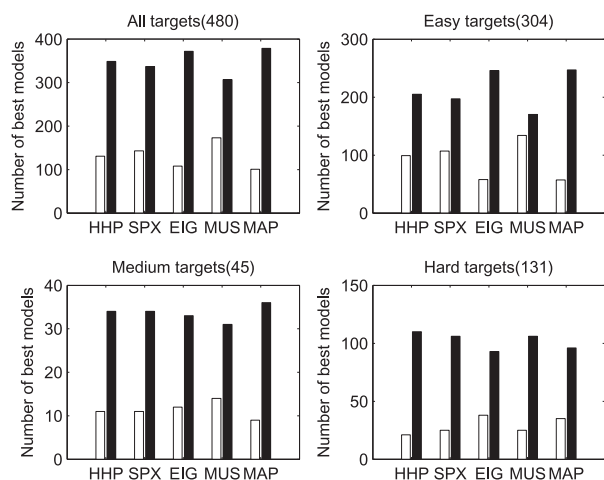


Fig. 3. The number of better targets for each method (white bars) compared with CATHER (black bars) in terms of TM-score

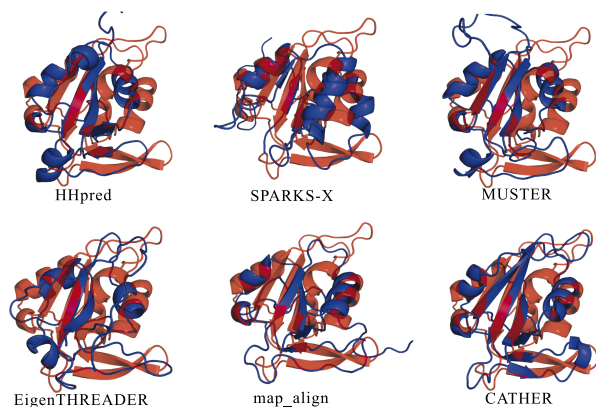


Fig. 4. The superimposition of the predicted models (in blue) against the native structure (in red) for an example target (domain ID: d2o0pa1)

(CMA-align) out of 39 participating servers. When removing the methods variants from the same lab, Yang-Server and CMA-align can be ranked at the 8th and 10th places, respectively.

We downloaded the server models from the CASP13 website for four server groups: CMA-align, Yang-Server, Zhang-CEthreader and RaptorX-TBM. According to the description in the CASP13 abstracts, Zhang-CEthreader is a contact-guided threading program and RaptorX-TBM is a distance-guided threading program based on DeepThreader (Zhu *et al.*, 2018). Since some servers failed to submit models for a few targets, only 104 common domains are assessed here (listed in Supplementary Table S6). The average TM-scores of these server models are summarized in Table 3. A few conclusions can be drawn from this table, as detailed below.

First, we can see that the multiple-TBM (Yang-Server) outperforms the single-TBM (CMA-align) for all types of targets. For TBM and TBM/FM targets, both methods have comparable accuracy. The improvement is significant for the FM targets, where the average TM-scores for Yang-Server and CMA-align are 0.3262 and 0.2875, respectively. This observation is in consistent with the conclusion of the advantage of multiple-TBM over single-TBM (Wu and Zhang, 2007; Yang *et al.*, 2015).

Second, we compared CMA-align with Zhang-CEthreader as both methods are single-template and contact-guided threading program. Overall, CMA-align slightly outperforms Zhang-CEthreader, with average TM-scores of 0.5724 and 0.5646, respectively. On TBM targets, CMA-align has higher TM-score than Zhang-CEthreader. However, on TBM/FM and FM targets, Zhang-CEthreader outperforms CMA-align. This is probably because the

Table 3. The comparisons of four threading methods which used predicted contacts/distances in CASP13

Targets	Zhang-CEthreader	RaptorX-TBM	CMA-align	Yang-Server
TBM	0.6923	<b>0.7733</b>	0.7339	0.7357
TBM/FM	0.5094	<b>0.5451</b>	0.4649	0.4846
FM	0.3287	<b>0.3995</b>	0.2875	0.3262
Overall	0.5646	<b>0.6377</b>	0.5724	0.5871

Note: The values in the table are the average TM-scores of the top models on the CASP13 targets. The best results are highlighted in bold type.

contacts used by Zhang-CEthreader are more accurate than ours, as revealed in the official evaluation of contact predictions in CASP13 (Shrestha *et al.*, 2019).

Third, RaptorX-TBM shows significantly higher accuracy than both Zhang-CEthreader and CMA-align. The improvement is consistent through all types of targets. There are at least two possible reasons for the superior performance of RaptorX-TBM. One is the usage of distances rather than binary contacts. Apparently, the information contained in distance is much more enriched than in the binary contacts. In fact, it was shown in CASP13 that even without using templates, accurate structure models can be built by using predicted distance distribution (Senior *et al.*, 2019; Xu and Wang, 2019). The second reason is that the contacts used in RaptorX-TBM are more accurate than other methods. For example, on 31 FM targets of CASP13, the average precision for the top L/5 long-range predictions by RaptorX-Contact (used in RaptorX-TBM) is 70.054%, while 65.678% by TripletRes (used in Zhang-CEthreader) and 60.156% by Yang-Server (used in CMA-align). This suggests the direction for the future development of threading programs: usage of improved predicted distance and/or distance distribution rather than binary contacts.

Finally, we did another experiment by running the original I-TASSER Suite on the CASP13 targets. This is to compare the benefit of adding CATHER into the I-TASSER Suite (used in Yang-Server). The results are presented in Supplementary Figure S2. We can see that the TM-scores by Yang-Server are consistently higher than these by I-TASSER Suite for all groups of targets, showing the benefit of adding CATHER into I-TASSER Suite.

### 3.5 Contribution of the contact information

To discuss the contribution of contact information, we developed another threading method called PRF-align, which only uses the sequential information in both template alignment and ranking. In addition, we considered another method called PRF\_cr, with the contact information used in the stage of the template ranking only. These two methods were also tested on the benchmark datasets.

The TM-scores of these methods on the benchmark datasets are shown in Supplementary Table S4. It suggests that both CATHER and PRF\_cr outperform PRF-align, which is more significant on the hard targets than on the easy targets, showing that the contact information is very helpful for improving the threading performance. Comparison between CATHER and PRF\_cr shows that the inclusion of the contact information in both stages of template alignment and ranking in CATHER does help detecting more accurate templates than in the stage of template ranking only.

### 3.6 Analysis of the predicted contacts

As predicted contacts are used in CATHER, it is necessary to investigate how the accuracy of the predicted contacts is affected and how it can affect the threading results. One of the most key factors in contact prediction is the profile depth, measured by the number of non-redundant sequences at 90% sequence identity in the MSA (denoted by  $N_{\text{eff}}$ ). Figure 5A shows the relationship between the sequence profile depth and the accuracy of the predicted contacts. We can see that the correlation between sequence profile depth and

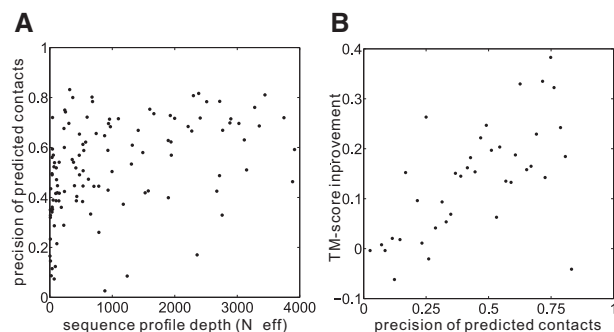


Fig. 5. (A) The relationship between the sequence profile depth and the precision of predicted contacts (top 2L). (B) The relationship between the precision of the predicted contact map and the TM-score improvement of CATHER over PRF-align

the contact precision is weak when  $N_{\text{eff}}$  is  $<1000$ . But when  $N_{\text{eff}}$  gets higher, the contact prediction becomes more accurate. Figure 5B shows the relationship between the accuracy of the predicted contact map and the TM-score improvement of CATHER over PRF-align, which is defined as the difference between the TM-scores of the top models by CATHER and PRF-align. Here, all targets with TM-scores  $>0.4$  were excluded, because they can be predicted well by using the sequential features only and the inclusion of the contact map does not improve much (such as for the easy targets). It suggests that the higher the precision is, the more improvement the contact map can bring in (Pearson's correlation coefficient is 0.63).

## 4 Conclusions

The increasing accuracy in protein contact map prediction opens a new avenue to improve the performance of threading. We have developed a new contact-assisted threading algorithm named CATHER using both conventional sequential profiles and contact maps predicted by a deep learning-based algorithm. Benchmark tests on an independent test set and the CASP12 targets demonstrated that CATHER made significant improvement over other methods which only use either sequential profile or predicted contact map. Our method was ranked at the top 10 among all 39 participated server groups on 32 FM targets in the blind tests of the CASP13 experiments. Experiments show that the improvement is due to the usage of the predicted contacts together with the sequential profiles.

## Funding

The work was supported by the National Natural Science Foundation of China (NSFC 11871290 and 61873185), Fundamental Research Funds for the Central Universities, Fok Ying-Tong Education Foundation (161003), KLMDASR and the Thousand Youth Talents Plan of China.

*Conflict of Interest:* none declared.

## References

Adhikari, B. et al. (2018) DNCON2: improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*, **34**, 1466–1472.

Altschul, S.F. et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bowie, J.U. et al. (1991) A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, **253**, 164–170.

Buchan, D.W.A. and Jones, D.T. (2017) EigenTHREADER: analogous protein fold recognition by efficient contact map threading. *Bioinformatics*, **33**, 2684–2690.

Dill, K.A. and MacCallum, J.L. (2012) The protein-folding problem, 50 years on. *Science*, **338**, 1042–1046.

Greener, J.G. et al. (2019) Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints. *Nat. Commun.*, **10**, 3977.

Henikoff, S. and Henikoff, J.G. (1994) Position-based sequence weights. *J. Mol. Biol.*, **243**, 574–578.

Hou, J. et al. (2019) Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins*, **87**, 1165.

Jaroszewski, L. et al. (2011) FFAS server: novel features and applications. *Nucleic Acids Res.*, **39**, W38–W44.

Jones, D.T. et al. (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.

Jones, D.T. et al. (2015) MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, **31**, 999–1006.

Kallberg, M. et al. (2012) Template-based protein structure modeling using the RaptorX web server. *Nat. Protoc.*, **7**, 1511–1522.

Kandathil, S.M. et al. (2019) Prediction of interresidue contacts with DeepMetaPSICOV in CASP13. *Proteins*, **87**, 1092.

Kelley, L.A. and Sternberg, M.J. (2009) Protein structure prediction on the web: a case study using the Phyre server. *Nat. Protoc.*, **4**, 363–371.

Li, Y. et al. (2019) Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. *Proteins*, **87**, 1082.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Ovchinnikov, S. et al. (2014) Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *Elife*, **3**, e02030.

Ovchinnikov, S. et al. (2017) Protein structure determination using metagenome sequence data. *Science*, **355**, 294–298.

Schaarschmidt, J. et al. (2018) Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age. *Proteins*, **86** (Suppl. 1), 51–66.

Senior, A.W. et al. (2019) Protein structure prediction using multiple deep neural networks in CASP13. *Proteins*, **87**, 1141.

Shrestha, R. et al. (2019) Assessing the accuracy of contact predictions in CASP13. *Proteins*, **87**, 1058.

Simons, K.T. et al. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, **268**, 209–225.

Soding, J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**, 951–960.

Soding, J. et al. (2005) The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.*, **33**, W244–W248.

Wang, S. et al. (2017) Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput. Biol.*, **13**, e1005324.

Webb, B. and Sali, A. (2014) Protein structure modeling with MODELLER. *Methods Mol. Biol.*, **1137**, 1–15.

Wu, Q. et al. (2019) Protein contact prediction using metagenome sequence data and residual neural networks. *Bioinformatics*. doi: 10.1093/bioinformatics/btz477.

Wu, S. and Zhang, Y. (2007) LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Res.*, **35**, 3375–3382.

Wu, S.T. and Zhang, Y. (2008) MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins*, **72**, 547–556.

Wuyun, Q. et al. (2018) A large-scale comparative assessment of methods for residue-residue contact prediction. *Brief. Bioinform.*, **19**, 219–230.

Xu, D. and Zhang, Y. (2012) *Ab initio* protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins*, **80**, 1715–1735.

Xu, J. (2019) Distance-based protein folding powered by deep learning. *Proc. Natl. Acad. Sci. USA*, **116**, 16856.

Xu, J. and Wang, S. (2019) Analysis of distance-based protein structure prediction by deep learning in CASP13. *Proteins*, **87**, 1069.

Yang, J. and Zhang, Y. (2015a) I-TASSER server: new development for protein structure and function predictions. *Nucleic Acids Res.*, **43**, W174–W181.

Yang, J. and Zhang, Y. (2015b) Protein structure and function prediction using I-TASSER. *Curr. Protoc. Bioinformatics*, **52**, 5.8.1–15.

- Yang, J. *et al.* (2015) The I-TASSER Suite: protein structure and function prediction. *Nat. Methods*, **12**, 7–8.
- Yang, Y. *et al.* (2011) 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*, **27**, 2076–2082.
- Zhang, Y. and Skolnick, J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.
- Zheng, W. *et al.* (2019) Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins*, **87**, 1149.
- Zhu, J. *et al.* (2018) Protein threading using residue co-variation and deep learning. *Bioinformatics*, **34**, i263–i273.