Supporting Information

Table of Contents

Supporting Texts

Supporting Figures

Supporting Tables

References

Supporting Texts

Text S1. MSA generation and spatial restraints prediction.

DeepMSA2 for MSA generation. Multiple sequence alignment in LOMETS3 is generated by DeepMSA2 (1), which is an extended pipeline based on the DeepMSA methodology (2) by adding pipelines to search larger metagenome sequence databases and including a contact-based MSA selection method based on deep learning predicted contact maps (**Figure S1**). Two individual MSA generation pipelines, dMSA and qMSA, in DeepMSA2, are used in LOMETS3. The dMSA (**Figure S1A**) pipeline is identical with the previous DeepMSA method. First, the query sequence is searched through Uniclust30 (version 2017_04) by HHblits2 to create dMSA-1. Next, the sequences identified by Jackhmmer and HMMsearch are used to construct a custom HHblits database, against which HHblits2 is run starting from the MSA generated in the previous stage to generate dMSA-2 and dMSA-3, respectively. qMSA pipeline ("Quadruple MSA" in **Figure S1B**) is used to produce four more MSAs. First, HHblits2 is used to search against the Uniclust30 database (version 2020_01) to create qMSA-1. Next, the sequences detected by Jackhmmer, HHblits3, and HMMsearch through the UniRef90, BFD, and Mgnify databases are used to construct custom HHblits-style databases, against which HHblits2 is used to search starting from the MSA generated by the previous stage to create qMSA-2, qMSA-3, and qMSA-4, respectively. The 7 MSAs (dMSA-1, dMSA-2, dMSA-3, qMSA-1, qMSA-2, qMSA-3 and qMSA-4) will be fed into DeepPotential to get 7 predicted contact maps. The final MSA of DeepMSA2 is the MSA that has the highest cumulative probability for the top 10L predicted contacts (**Figure S1C**).

DeepPotential for spatial restraints prediction. Three types of spatial restraints are used in LOMETS3 and can be predicted by DeepPotential (1). The spatial restraints include (i) contact maps, (ii) distance maps, and (iii) hydrogen bond networks (3). In the DeepPotential pipeline, a set of co-evolutionary features are extracted from the final MSA generated by DeepMSA2. These co-evolutionary features, which are inherently two-dimensional, include the raw coupling parameters from the pseudo likelihood maximized (PLM) 22-state Potts model and the raw mutual information (MI) matrix. The PLM and MI matrices are extracted from the query-specific co-evolutionary information in the given MSA. The Potts model field parameters, Hidden Markov Model (HMM) features, and the self-mutual information are the major one-dimensional inputs, along with the one-hot representation of the MSA and other descriptors, such as the number of sequences in the MSA. These two-dimensional and one-dimensional features are fed into deep convolutional residual neural networks separately, where each of them is passed through a set of one-dimensional and two-dimensional residual blocks, respectively, and are subsequently tiled together. The tiled feature representations are considered as the inputs of another fully residual neural network containing 40 2-D residual blocks which output several inter-residue interaction terms, including the Cα-Cα (Cβ-Cβ) contact map, Cα-Cα (Cβ-Cβ) distance map, and hydrogen bond networks (aa, bb and cc). Here, the predicted distance restraints and hydrogen bond restraints are represented using various bins that correspond to specific probability values. For example, for the Cα and Cβ distances, the predictions are divided into 38 bins, where the first bin represents the probability that the distance is <2Å and the final bin represents the probability that the distance is ≥20Å. The remaining 36 bins represent the probability that the distance falls in the range [2Å, 20Å), where each bin has a width of 0.5 Å. For the 3 types of hydrogen bonds, the probability is predicted using a bin width of 10° (0-180°) with an additional bin to indicate whether there is no hydrogen bond between the two residues (i.e., Cα-Cα distance ≥10Å).

Text S2. Domain partition and assembly.

A new domain partition and assembly module has been added to LOMETS3 for multi-domain protein template detection and modeling. The domain partition module combines two domain boundary prediction algorithms, ThreaDom and FUpred. Each predicted individual domain target will again be input to LOMETS3 for template detection and modeling. The final templates and models for domains will be assembled to "full-length" by DEMO2 (4) guided by the knowledge-based potential, template-based potential and spatial restraints from DeepPotential.

Domain partition. LOMETS3 combines ThreaDom (5) and FUpred (6) to predict whether a target is a single- or multi-domain protein and where the domain boundaries are. ThreaDom is a template-based method, which is mainly used for Easy targets, and FUpred predicts domains based on DeepPotential contact map, thus is mainly used for Hard targets. ThreaDom predicts domain boundary depending on LOMETS3 threading alignment coverage. A domain conservation score (DCS), which combines information from the template domain structures, terminal and internal gaps, and insertions, is calculated for each residue. The domain boundary information is derived from the DCS profile distribution. FUpred utilizes a recursive strategy to detect domain boundaries based on predicted contact maps and secondary structure information. The core idea of the algorithm is to retrieve domain boundary locations by maximizing the number of intra-domain contacts, while minimizing the number of inter-domain contacts from the contact maps. In benchmarks, FUpred achieved exceptional performance on domain boundary detection, especially for discontinuous domains (6).

Domain assembly. "Full-length" templates/models of multi-domain proteins are constructed by assembling independently predicted domain templates/models through DEMO2, which is an improved version of DEMO (4) that integrates inter-domain spatial restraints predicted by DeepPotential associated with the knowledge-based potential and template-based potentials. In the DEMO2, ten global templates that cover as many domains as possible are first identified from a non-redundant multi-domain protein library by aligning each domain model to the template using TM-align (7). Starting from each initial global template, an L-BFGS algorithm is performed to detect each domain's optimal translation vectors and rotation angles for domain templates/models. The optimization is guided by a comprehensive energy function including inter-domain steric clashes, inter-domain distance profiles deduced from the top templates, domain boundary connectivity, inter-domain distances and domain-domain interfaces (distance <20Å) predicted by DeepPotential, and a generic inter-domain contact potential (8). The translation vectors and rotation angles with the lowest energy are selected to construct the final "full-length" template/model. Finally, the model with the lowest energy is selected for side-chain reconstruction and refinement by FASPR (9) and FG-MD (10).

Text S3. Extended COFACTOR pipeline for function prediction.

COFACTOR (11) is a structure, sequence, and protein-protein interaction (PPI) based method for protein function predictions, providing predicted annotations including Gene Ontology (GO), Enzyme Commission (EC), and ligand binding sites. In LOMETS3, COFACTOR is integrated and extended by integrating the LOMETS3 threading templates associated with structural analogues to search through the BioLiP protein function database (12) for identifying functional sites. Here, BioLiP is a semi-manually curated structure-function database containing known associations of experimentally solved structures and biological functions of proteins in terms of GO terms, EC numbers, and ligand binding sites.

GO term prediction. Three types of GO terms, Molecular Function (MF), Biological Process (BP), and Cellular Component (CC) GO terms, are predicted by COFACTOR (11), which detects functional homologs through three pipelines: 1) local and global structure alignments, 2) sequence and sequence profile comparison, and 3) partner-homology based protein-protein interaction mapping.

Structure-based pipeline. The query structure is compared to a non-redundant set of known proteins in the BioLiP (12) through two sets of local and global structural alignments based on TM-align (7), for functional homology detections. Currently, the BioLiP library contains 234,574 entries annotated with GO terms. In addition, the LOMETS3 threading templates and structure analogues are also be added into the structural alignments list in this pipeline.

The overall confidence score for a particular GO term is defined by

$$Cscore^{structure} = 1 - \prod_{i=1}^{N}(1 - FCscore_i) \qquad (S1)$$

$$FCscore = \frac{2}{1 + \exp\left(-(0.25 * L_{sim} * SS_{bs} + \mathrm{TM} + 2.5 * \mathrm{ID})\right)} - 1 \qquad (S2)$$

where N is the number of templates associated with the GO term, and $FCscore_i$ is the confidence score of the i-th template hit associated with the GO term. TM is the global structure similarity in terms of TM-score between query and template, ID is the sequence identity between query and template in the aligned region, $\mathrm{SS}_{\mathrm{bs}}$ is the sequence identity at the binding site, and L_{sim} is the local structure similarity between query and template.

Sequence-based pipeline. In the sequence-based pipeline, a query is searched against the UniProt-GOA by BLAST with an E-value cutoff 0.01 to identify sequence homologs, where unreviewed annotations with Inferred from Electronic Annotation (IEA) or No biological Data available (ND) evidence codes are excluded. Similarly, a three-iteration PSI-BLAST search is performed for the query through the UniRef90 (13) database to create a sequence profile, which is used to jump-start a one-iteration PSI-BLAST search through the UniProt-GOA.

The overall confidence score of the sequence-based pipeline is calculated as

$$Cscore^{sequence} = w * GOfreq_{blast} + (1 - w) * GOfreq_{psiblast} \qquad (S3)$$

$$GOfreq = \frac{\sum_{k=1}^{N'} s'_k}{\sum_{k=1}^{N} s_k} \qquad (S4)$$

where w equals the maximum sequence identity of the query to all the template hits. $\mathrm{GOfreq}_{\mathrm{blast}}$ and $\mathrm{GOfreq}_{\mathrm{psiblast}}$ are the confidence scores for a particular GO term resulting from a BLAST or PSI-BLAST search. N is the number of templates identified, s_k is the sequence identity between the query and the k-th template and N' and s'_k are those associated with a specific GO term.

PPI-based pipeline. In this pipeline, the query is first mapped to the STRING (14) PPI database by BLAST; only the BLAST hit with the most significant E-value is subsequently considered. GO terms of the interaction partners, as annotated in the STRING database, are then collected and assigned to the query protein. The underlying assumption is that interacting protein partners tend to participate in the same biological pathway at the same sub-cellular location and therefore may have similar GO terms.

The confidence score for a GO term mapped by PPI is calculated by

$$Cscore^{PPI} = S_q * \frac{\sum_{k=1}^{N'} str'_k}{\sum_{k=1}^{N} str_k} \qquad (S5)$$

where N is the number of interacting partners, str_k is the confidence score of interaction between query and the k-th interaction partner as assigned by the STRING database and N' and str'_k are those associated with a specific GO term. S_q is the sequence identity in the first step of BLAST alignment between the query sequence and the STRING entry it is mapped to.

The final function predictions are obtained from a combination of the three pipeline results through logistic regression, with confidence score calculated by

$$Cscore^{GO} = 1 - \prod_{m} (1 - Cscore^{m})^{w_m} \qquad (S6)$$

where $m \in \{\text{structure, sequence, PPI}\}$. w_m is the relative weight for each of the three methods, with $w_{sequence} = w_{PPI} = 1$ and $w_{structure} = 1 - w$, where w equals to the maximum sequence identity among identified function templates.

EC number prediction. The pipeline for EC number prediction is similar to the structure-homology based method used in GO prediction. Enzymatic homologs are collected from LOMETS3 threading templates and structure analogs, then mapped to a library containing 8,392 enzyme structures from the BioLiP library, with the active site residues obtained from the Catalytic Site Atlas database (15).

The confidence score for each predicted EC number is estimated by

$$Cscore^{EC} = \frac{2}{1 + \exp\left(-(0.25 * L_{sim} * SS_{as} + \mathrm{TM} + 2.5 * \mathrm{ID})\right)} - 1 \qquad (S7)$$

where TM is the global structure similarity in terms of TM-score between query and template, ID is the sequence identity between query and template in the aligned region, SS_{as} is the sequence identity at the active site, and L_{sim} is the local structure similarity between query and template.

Ligand binding site prediction. Ligand binding site prediction in COFACTOR consists of three steps.

First, functional homologues are collected from the LOMETS3 templates and structure analogs, and then matched through the BioLiP library, which currently contains 115,951 structure templates harboring in total 569,071 ligand-binding sites for interaction between receptor proteins and small molecule compounds, short peptides, and nucleic acids. The initial binding sites are then mapped to the query from the individual templates based on the structural alignments.

Second, the ligands from each individual template (LOMETS3 threading templates or structure analogs) are superposed to the predicted binding sites on the query structure using superposition matrices from a local alignment of the query and template binding sites. To resolve atomic clashes, the ligand poses are refined by a short Metropolis Monte Carlo simulation under rigid-body rotation and translation.

Finally, the consensus binding sites are obtained by clustering of all ligands that are superposed to the query structure, based on distances of the centers of mass of the ligands using a cutoff of 8Å. Different ligands within the same binding pocket are further grouped by the average linkage clustering with chemical similarity, using the Tanimoto coefficient (16) with a cutoff of 0.7. The model with the highest ligand-binding confidence score among all the clusters is selected.

The confidence score is defined by

$$Cscore^{LB} = \frac{2}{1 + \exp\left(-\frac{M}{M_{tot}}\left(0.25 * L_{sim} + \mathrm{TM} + 2.5 * \mathrm{ID} + \frac{2}{1 + D}\right)\right)} - 1 \qquad (S8)$$

where M is the number of ligands in the ligand cluster, M_{tot} is the total number of ligands collected from all homologous templates, TM is the global structure similarity in terms of TM-score between query and template, ID is the sequence identity between query and template in the aligned region, L_{sim} is the local structure similarity between query and template, and D is the average distance between ligands within the cluster.

Text S4. Summary of the 11 component threading methods.

The LOMETS3 server integrates predictions from 11 locally installed threading programs, which can be classified as two groups, contact-based threading method and profile-based threading method. The contact-based threading group includes 5 programs: CEthreader (17), DisCovER (18), EigenThreader (19), Hybrid-CEthreader (17), MapAlign

(20). The profile-based threading group includes 6 methods: FFAS3D (21), HHpred (22), HHsearch (23), MRFsearch (24), MUSTER (25), and SparksX (26). The predicted contact map used in contact-based threading methods is the Cβ-Cβ contact map predicted by DeepPotential, and the sequence-profile or profile-HMM is derived from the final MSA generated by DeepMSA2. All the programs run with a unified non-redundant template library that contains around 90,000 items and is updated every week. For a protein chain that consists of multiple domains, both the whole chain and individual domain structures are included in the library. Below we summarize each of the component threading methods; full details can be found in the references cited above.

CEthreader is an in-house threading program that uses predicted contact detecting templates. The input contact map predicted from DeepPotential will be decomposed as single body 7-dimensional eigenvectors by Eigen decomposition, and then be used as part of a dynamic programming scoring scheme. In addition to residue-residue contacts, the sequence-profile and secondary structure terms are also used in the scoring scheme with an affine gap by Needleman-Wunsch dynamic programming algorithm. The contact map overlapping score, CMO (see **Eq. S13** of **Text S3**), is used to rank the top threading templates. To speed up the threading approach, we first select the top 1,000 templates identified by HHsearch, and then re-align and re-rank the templates by the CEthreader.

DisCovER is a contact-based threading method utilizing an iterative double dynamic programming algorithm. In the first stage, an alignment score incorporates sequence-profile, predicted secondary structures, structure derived profiles, solvent accessibility, backbone dihedral angles, hydrophobicity, and the neighborhood effect. This score is calculated for every query-template alignment with the Needleman-Wunsch dynamic programming algorithm. In the second stage, a similarity score incorporating predicted distance term is used for re-aligning the top 50 ranked alignments in the first stage searching. A two-step iterative double dynamic programming algorithm is used to find the new optimal alignment that maximizes the similarity score when re-aligning those 50 templates.

EigenThreader is a contact-based threading method which is extended from the Al-Eigen (27) method. Similar to CEthreader, the input contact map also will be decomposed as single body eigenvectors by Eigen decomposition, but with higher dimension of 20. Instead of evaluating all eigenvector signs in contact map alignment by Needleman-Wunsch dynamic programming, an iterative search algorithm using sign inversion of eigenvectors is used. The final templates are ranked by a scoring function combined with a CMO term and secondary structure match term.

Hybrid-CEthreader is a hybrid approach based on the CEthreader pipeline, where HHsearch is replaced by a fast version of CEthreader for the first-round database scan. The fast version of CEthreader only utilizes 2-dimensional eigenvectors for contact map alignment. Then, the top 1,000 templates detected by the fast version of CEthreader will be re-aligned and re-ranked by the full version of CEthreader with 7-dimensional eigenvectors.

MapAlign is a contact-based threading method that uses a double dynamic programming algorithm to maximize the CMO between the query contact map and template contact map. The first step calculates a score between each row of the query contact map and each row of the template contact map. The score is the sum of Gaussian functions based on sequence separation. Dynamic programming finds the alignment of these two rows that results in the maximization of the calculated score. The maximized scores are then used in a second matrix, where the second dynamic programming step utilizing the Smith-Waterman algorithm returns the optimal contact alignment. This alignment is used to update the second step similarity matrix, and the second dynamic programming step is carried out again, for a 20-times iterative process.

FFAS3D is a profile-based threading method utilizing the Smith-Waterman dynamic programming algorithm to detect templates. The scoring function used in dynamic programming incorporates sequence profiles, predicted secondary structures, solvent accessibility, and residue depth. The raw alignment score after dynamic programming is calibrated as a Z-score with the mean and standard deviation of a distribution of scores. Templates are then re-ranked based on a neural network with the Z-score, sequence identities, secondary structures, Pearson's correlation coefficients, solvent accessibility, and residue depth as inputs, to generate a new alignment score.

HHsearch is based on profile hidden Markov models (HMMs), an extension of sequence profiles which also contain information on position-specific probabilities of amino acid matches, insertions, and deletions as well as the frequencies of emitting different amino acid types. A profile HMM is calculated for the query sequence from its MSA and searched against a pre-built template database of profile HMMs. Along with secondary structure information, dynamic programming is used to provide an HMM-HMM alignment that maximizes the log-sum-of-odds score, a generalization of the log-odds of sequence emission from an HMM to an HMM-HMM alignment.

HHpred is an extension of HHsearch with the addition of a new template re-ranking scheme. A simple neural network with four alignment features as inputs is used to predict the template TM-score (28) based on a query-template alignment. From the top 100 HHsearch templates, a set of accepted templates is built with the highest predicted TM-score template as the first member of the set. An iteratively updated score rewarding high coverage and penalizing worse alignment quality than pre-selected templates is used to re-rank remaining templates. Templates are added to the new set given there are positive scores.

MRFsearch is a profile-based threading method that aligns Markov random fields (MRFs) to detect templates, where each node of the MRF represents one residue and each edge specifies mutation statistics between two corresponding residues. The scoring function for an MRF-MRF alignment incorporates node alignment potentials and edge alignment potentials with equal weights. A suboptimal alignment is obtained via alternating direction method of multiples (ADMM), an algorithm that reduces the MRF-MRF alignment problem to two iteratively solved sub-problems.

MUSTER is an in-house profile-based threading method built on Needleman-Wunsch dynamic programming. The scoring function contains sequence profiles, secondary structure, structure fragment profiles, solvent accessibility, backbone torsional angles, and hydrophobic scoring matrices.

SparksX detects templates by Smith-Waterman dynamic programming. The alignment score includes profile-profile comparison, and estimated probabilities between query and template 1D structural properties that includes predicted secondary structures, solvent accessibility, and backbone dihedral angles.

Text S5. Templates ranking and target type definition.

Template re-ranking for profile-based threading. For the six profile-based threading methods, including FFAS3D, HHpred, HHsearch, MRFsearch, MUSTER, and SparksX, a template re-ranking algorithm is utilized based on the scoring function, $Z-score(i,j)$, which combines the original profile-based alignment score, contact map overlapping score, mean absolute distance error, and hydrogen bond satisfaction. The $Z-score(i,j)$ where *i* represents *i*-th template and *j* represents *j*-th threading program, is defined as follows:

$$Z-score(i,j) = w_1\, Z-score^{MAE}(i,j) + w_2\, Z-score^{CMO}(i,j) \\ +w_3\, Z-score^{HB}(i,j) + w_4\, Z-score^{prof}(i,j) \qquad (S9)$$

where $Z-score^{MAE}(i,j)$ is the Z-score of the mean absolute error (MAE) based on the predicted distance map, $Z-score^{CMO}(i,j)$ is the Z-score of the number of overlapping contacts based on the predicted contact map (CMO), $Z-score^{HB}(i,j)$ is the Z-score based on the predicted hydrogen bond geometry (HB), and $Z-score^{prof}(i,j)$ is a score which is based on the original profile threading scores. The formulas of these four Z-scores are as follows:

$$Z-score^{MAE}(i,j) = \frac{-MAE(i,j) - \langle -MAE(j)\rangle}{\sigma(-MAE(j))} \qquad (S10)$$

$$MAE(i,j) = \frac{\sum_{m,n}^{ali}[\delta(m,n)|d_{m,n}^{query} - d_{m,n}^{template}| + (1-\delta(m,n))]}{\sum_{m,n}^{ali}\delta(m,n)} \qquad (S11)$$

where $d_{m,n}^{query}$ is the predicted distance between residue *m* and *n* in the query structure, $d_{m,n}^{template}$ is the predicted distance between residue *m* and *n* in the template structure, ali means the length of alignment, and $\delta(m,n) = \begin{cases} 1, m\ and\ n\ are\ not\ gap \\ 0, otherwise \end{cases}$. $\langle -MAE(j)\rangle$ and $\sigma(-MAE(j))$ are the average and standard deviation of the MAE scores across all templates for the *j*-th program, respectively.

$$Z-score^{CMO}(i,j) = \frac{CMO(i,j) - \langle CMO(j)\rangle}{\sigma(CMO(j))} \qquad (S12)$$

$$CMO(i,j) = \frac{N(Overlap(CM^{query}, CM^{template}))}{N(CM^{query})} \qquad (S13)$$

where $N(Overlap(CM^{query}, CM^{template}))$ is the number of overlapping contacts between the predicted contact map and the contact map derived from the aligned template, and $N(CM^{query})$ is the number of predicted contacts. $\langle CMO(j)\rangle$ and $\sigma(CMO(j))$ are the mean and standard deviation of the contact overlap scores across all templates for the *j*-th program, respectively.

$$Z-score^{HB}(i,j) = \frac{HBscore(i,j) - \langle HBscore(j)\rangle}{\sigma(HBscore(j))} \qquad (S14)$$

$$HBscore(i,j) = \sum\nolimits_{m,n}^{ali} \frac{1}{1+\left(\frac{|min(|\theta_{m,n}^{query} - \theta_{m,n}^{template}|, \pi - |\theta_{m,n}^{query} - \theta_{m,n}^{template}|)|}{\theta}\right)^2} \qquad (S15)$$

where $\theta_{m,n}^{query}$ is the predicted hydrogen bond angle between residue *m* and *n* in the query structure, $\theta_{m,n}^{template}$ is the predicted hydrogen bond angle between residue *m* and *n* in the template structure, and $\theta = 15^\circ$. $\langle HBscore(j)\rangle$ and $\sigma(HBscore(j))$ are the average and standard deviation of the alignment scores across all templates for the *j*-th program, respectively.

$$Z-score^{prof}(i,j) = \frac{S(i,j) - \langle S(j)\rangle}{\sigma(S(j))} \qquad (S16)$$

where $S(i,j)$ is the alignment score of the *i*-th template for the *j*-th program, and $\langle S(j)\rangle$ and $\sigma(S(j))$ are the average and standard deviation of the alignment scores across all templates for the *j*-th program, respectively.

Final template ranking in LOMETS3. For a given target, 220 templates (=11×20) are collected, where each threading method generates 20 top templates that are sorted by their Z-scores. The top 10 templates are finally selected from the 220 templates pool based on the following scoring function that integrates Z-score and sequence identity between the identified templates and query sequence:

$$score(i,j) = conf(j) * \frac{Z-score(i,j)}{Z_0(j)} + seqid(i,j) \qquad (S17)$$

where $seqid(i,j)$ is the sequence identity between the query and the *i*-th template from the *j*-th program, $conf(j)$ is the confidence score for the *j*-th program, which was calculated by determining the average TM-scores over the first templates to the native structures on the benchmark dataset, and $Z_0(j)$ is the Z-score cut-off for defining good/bad templates for the *j*-th program, which was determined by maximizing the Matthews correlation coefficient (MCC) for distinguishing a good template (with a TM-score ≥0.5) from a bad template (TM-score <0.5) on the benchmark Dataset-1 that contains 614 single-domain proteins. As a result, the parameters $Z_0(j)$ (and $conf(j)$) are 6.1 (0.495), 7.8 (0.478), 6.0 (0.472), 22.0 (0.471), 3.8 (0.471), 8.5 (0.461), 6.0 (0.456), 6.9 (0.445), 46.0 (0.440), 6.0 (0.437), and 83.0 (0.389) for Hybrid-CEthreader, SparksX, CEthreader, HHsearch, MapAlign, MUSTER, MRFsearch, DisCovER, FFAS3D, EigenThreader, and HHpred, respectively.

Target type definition. Based on the quality and number of threading alignments from LOMETS3, protein targets can be classified as “Easy” and “Hard”, where “Easy” target mainly represents homologous target and “Hard” target represents non-homologous protein. The target type definition used in LOMETS3 considers the threading template quality, i.e., Z-score, and the structural similarity between templates from different threading methods.

For each protein target, the first ranked template for each of the 11 threading methods in LOMETS3 are selected. Based on the 11 templates, Z_a, the average normalized Z-score (divided by Z_0), is calculated for the 11 threading methods. We further calculate the pairwise TM-scores among the 11 templates. There are 55 ($= C_{11}^2 = 11 \times 10/2$) distinct template-template pairs and corresponding TM-scores. We define TM1, TM2, TM3, and TM4 as the average TM-scores over the top 1/4, 2/4, 3/4 and 4/4 template-template pairs ranked by their TM-scores. Thus, we get a set of 9 scores, i.e., *S* = {Z_a, TM1, TM2, TM3, TM4, Z_a*TM1, Z_a*TM2, Z_a*TM3, Z_a*TM4}. Based on the set *S*, the target can be classified by the following rule,

$$\text{Target is classified as } \begin{cases} Easy, & if\ |\{s \in S | s > cut(s)\}| \geq 7 \\ Hard, & otherwise \end{cases} \qquad (S18)$$

where *cut*(*S*)={1.052, 0.508, 0.396, 0.350, 0.339, 0.353, 0.279, 0.239, 0.209}. Here, $|\{\ldots\}|$ means the number of items in the set $\{\ldots\}$.

Text S6. Model generation by an L-BFGS system.

For each pair of residues, the probabilities of structural geometry terms will be converted into smooth potentials for the gradient-descent based protein structure prediction. For each term, the negative log of the raw probability histogram will be interpolated by cubic spline as potentials. The folding potential can be written as

$$E_{fold} = w_1 E_{cbdis} + w_2 E_{cadis} + w_3 (E_{aa} + E_{bb} + E_{cc}) \qquad (S19)$$

where E_{cbdis} and E_{cadis} are distance potentials (E_{cbdis} and E_{cadis}), and E_{aa}, E_{bb} and E_{cc} are hydrogen potentials. The weights of corresponding terms are set empirically to $w_1 = 5.0$, $w_2 = 0.1$, and $w_3 = 0.5$.

Here, the hydrogen bond probability histogram is directly predicted from DeepPotential, while for distance probability histogram, the probability, $P(i,j)_{dis}$, used to calculate E_{cbdis} and E_{cadis}, is a fusion probability combining the raw probability $P(i,j)_{dis}^{dp}$ predicted from DeepPotential and statistical probability $P(i,j)_{dis}^{tem}$ derived from LOMETS3 top *N* ranked templates with alignment coverage > 0.5 for Easy targets (coverage >0.6 for Hard targets). Here, *N* is 50 for Easy targets, and 30 for Hard targets. The fusion probability $P(i,j)_{dis}$ can be calculated as

$$P(i,j)_{dis} = wP(i,j)_{dis}^{dp} + (1-w)P(i,j)_{dis}^{tem} \qquad (S20)$$

where *i* and *j* are a residue pair, and *w* is a weight, which equals to 0.8.

The backbone structure of the protein is specified by the ϕ/ψ backbone torsion angle of each residue along the query sequence, while the backbone torsion angle ω is set to 180°. Given a set of (ϕ_i, ψ_i) parameters with $i = 1, \cdots, L$, the coordinates of backbone atoms (including C_β atoms) can be recovered, thus the energy of such decoy conformation, defined in **Eq. S19**, can be computed according to the interpolated potential curves. With the help of the automatic differentiation in PyTorch (29), the gradient with respect to the parameters could be readily obtained.

We implemented the L-BFGS algorithm to iteratively update the protein structure conformations. The whole backbone folding process is performed 10 times with different initial structures built from random backbone torsion angle samplings. The final conformation with the lowest total energy is returned. Once the optimal backbone conformation is obtained, the FASPR program (9) is used to construct and repack the sidechain atoms, and FG-MD (10) is used to refine the final model by removing any clashes.

Supporting Figures

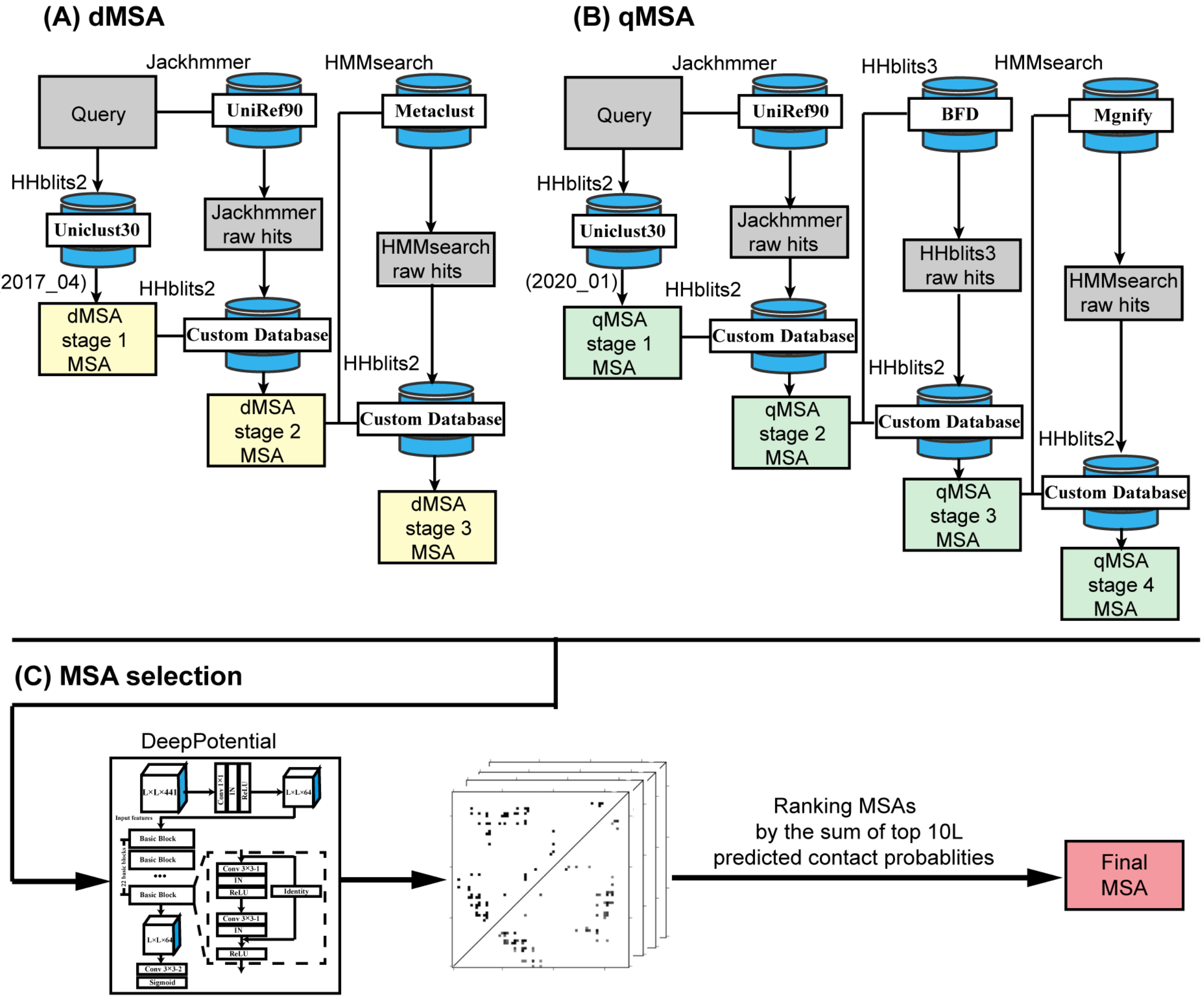


Figure S1. Flowchart of the DeepMSA2 algorithm. (A) dMSA pipeline that contains three stages of MSA generation using sequences from HHblits searching against Uniclust30, Jackhmmer searching through UniRef and HMMsearch through Metaclust. (B) qMSA pipeline which is extended from dMSA pipeline by adding a new stage for searching the BFD metagenome database. (C) MSA selection method that ranks the MSAs based on the sum of top *L* predicted contact probabilities.

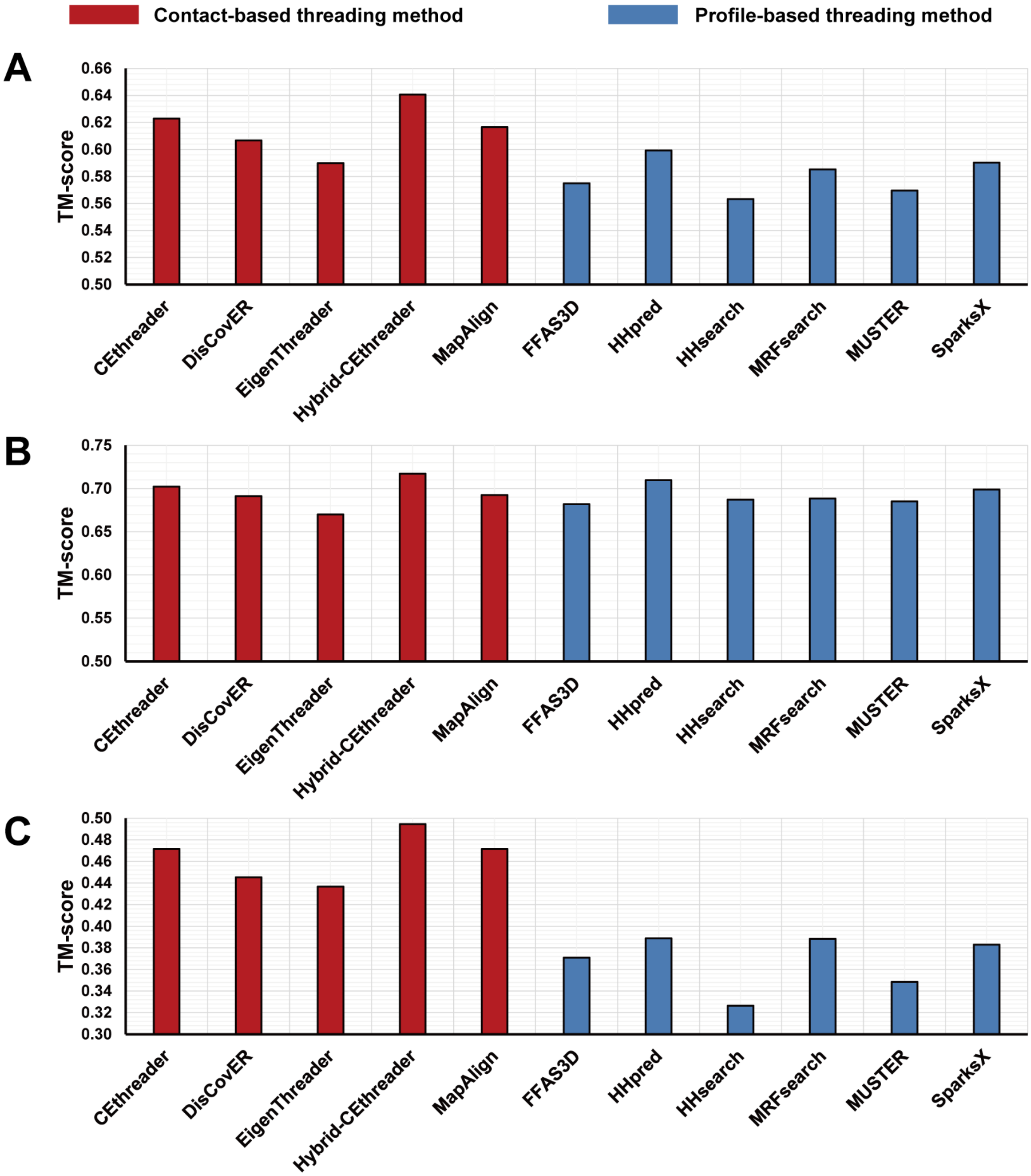


Figure S2. Average TM-scores of the first templates for 11 single threading methods in LOMETS3 on (A) All 614 single-domain targets, (B) 403 Easy targets, and (C) 211 Hard targets. The methods can be categorized as two groups, contact-based threading and profile-based threading, based on whether deep learning contact information used as input or not. Here, the original threading results (without re-ranking by deep learning predicted restraints) of the profile-based methods are shown to demonstrate the advance provided by contact-based methods.

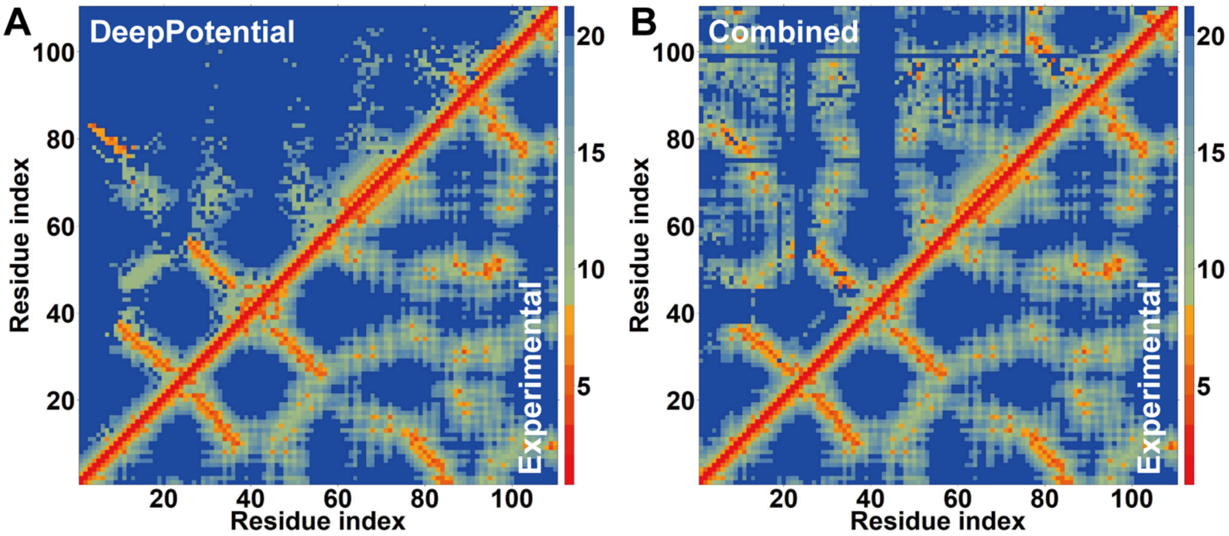


Figure S3. Inter-residue distance map from DeepPotential and LOMETS3 for PDB ID 1prtD. (A) Inter-residue distance map predicted from DeepPotential (upper triangle), and the distance map calculated from the target structure (lower triangle) for 1prtD. (B) Inter-residue distance map combined from DeepPotential and threading templates in LOMETS3 (upper triangle), and the distance map calculated from the target structure (lower triangle) for 1prtD.

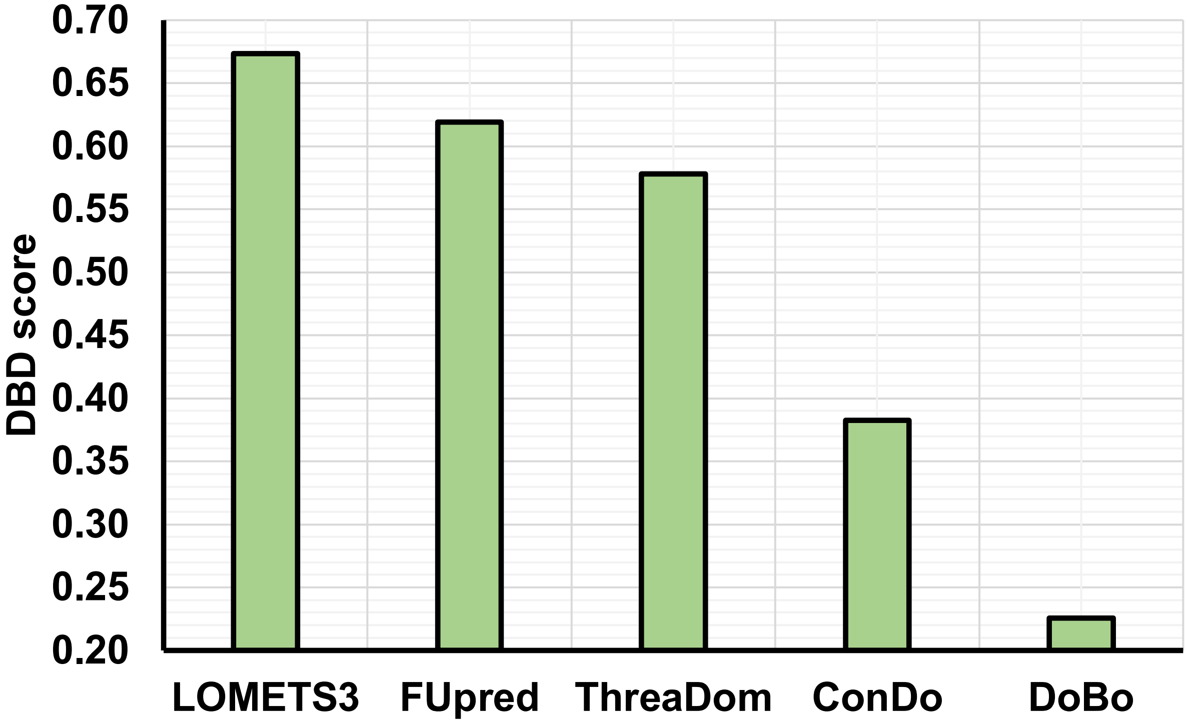


Figure S4. The DBD score for different domain partition methods on 408 multi-domain proteins.

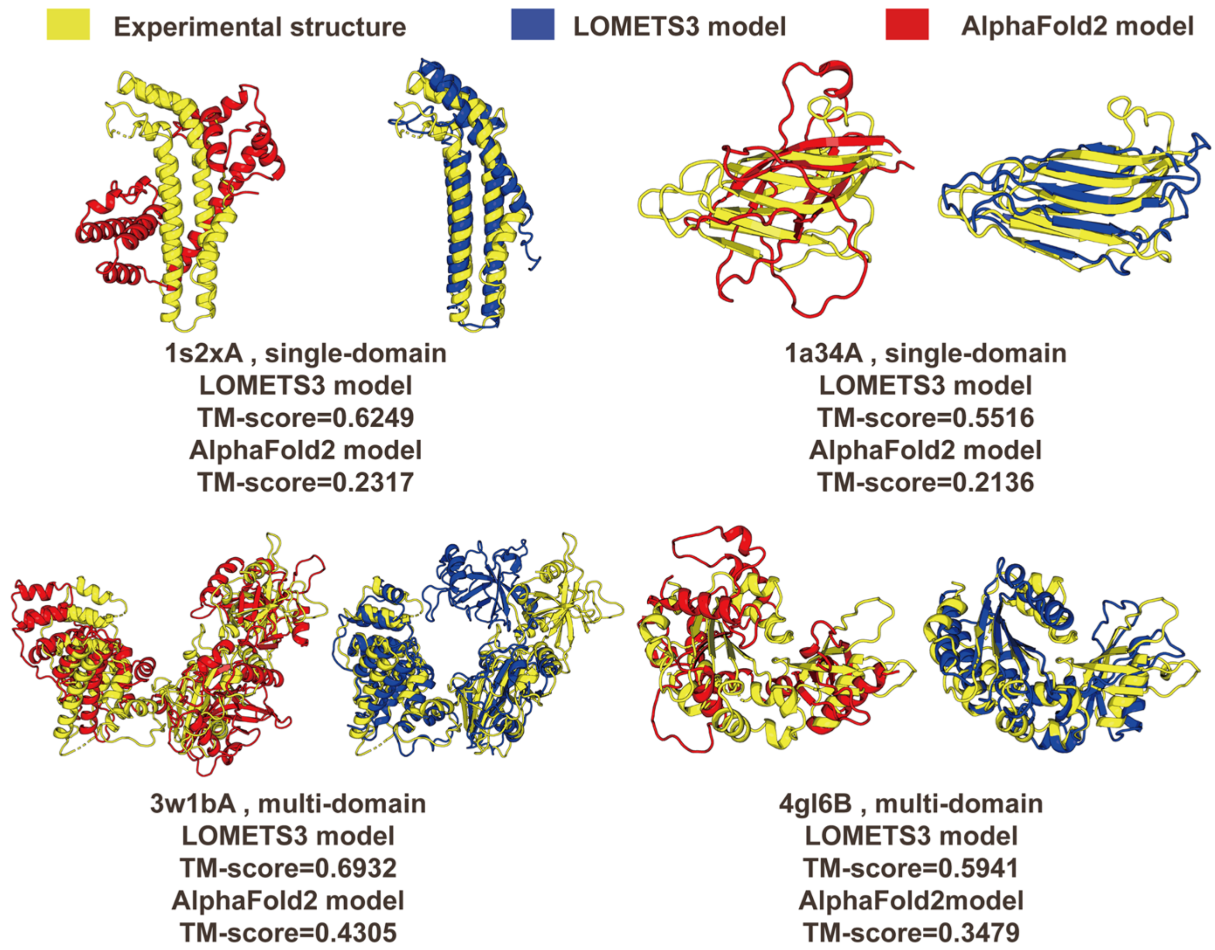


Figure S5. Case study of four targets for which LOMETS3 can generate correct models with TM-score >0.5 but AlphaFold2 cannot in benchmarking datasets, where the LOMETS3 models (blue) and the AlphaFold2 models (red) are overlaid with the experimental structures (yellow). It indicates that although very rare there is possibility that the TBM method could generate better models than the best end-to-end training approach.

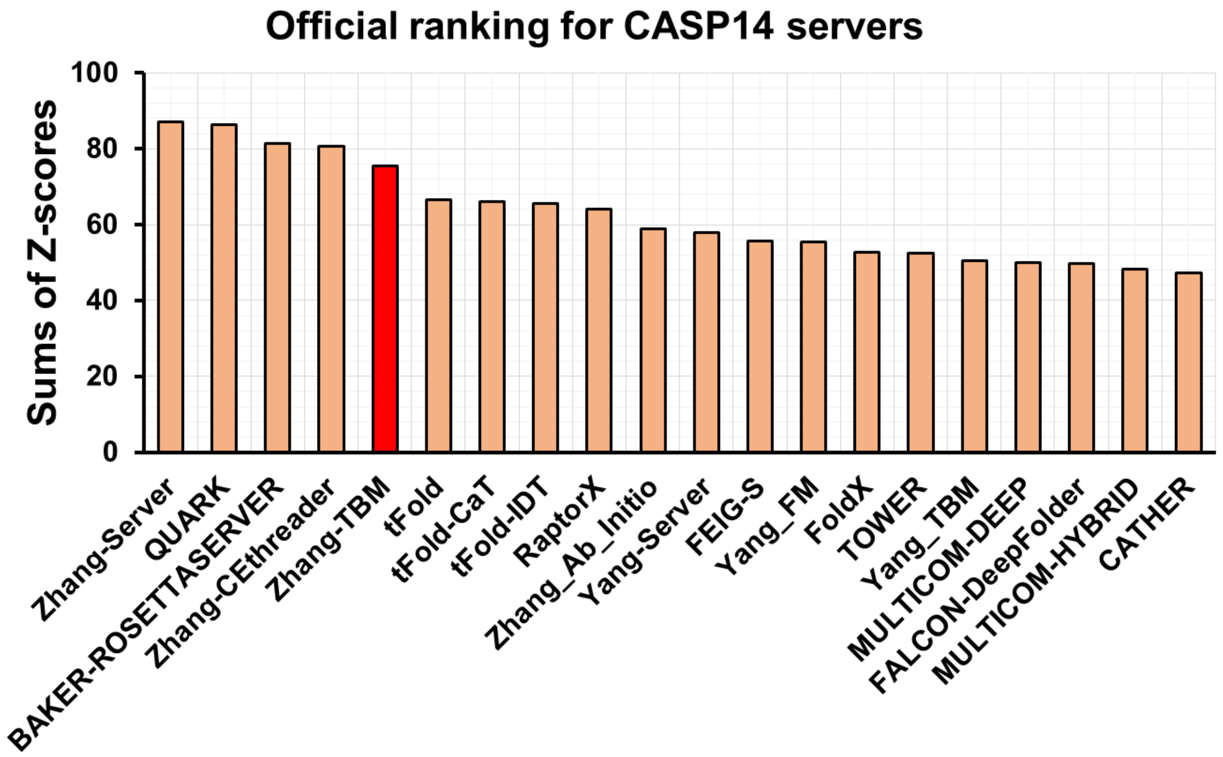


Figure S6. Summary of the modeling results of the top-20 server groups in the CASP14 experiment. The ranking of the groups is based on the analysis of Z-scores for GDT-TS, where models with Z-scores below the tolerance threshold (0.0) were removed. GDT-TS means global distance test score, which is calculated by GDT-TS = (GDT_P1 + GDT_P2 + GDT_P4 + GDT_P8)/4, where GDT_Pn indicates the percent of residues under the distance cut-off ≤nÅ. Here, LOMETS3 was registered as 'Zhang-TBM'. In addition, "Zhang-Server" and "QUARK" also utilized LOMETS3 as threading method for template detection during CASP14. Data were taken from the official CASP14 webpage at https://www.predictioncenter.org/casp14/zscores_final.cgi?gr_type=server_only.

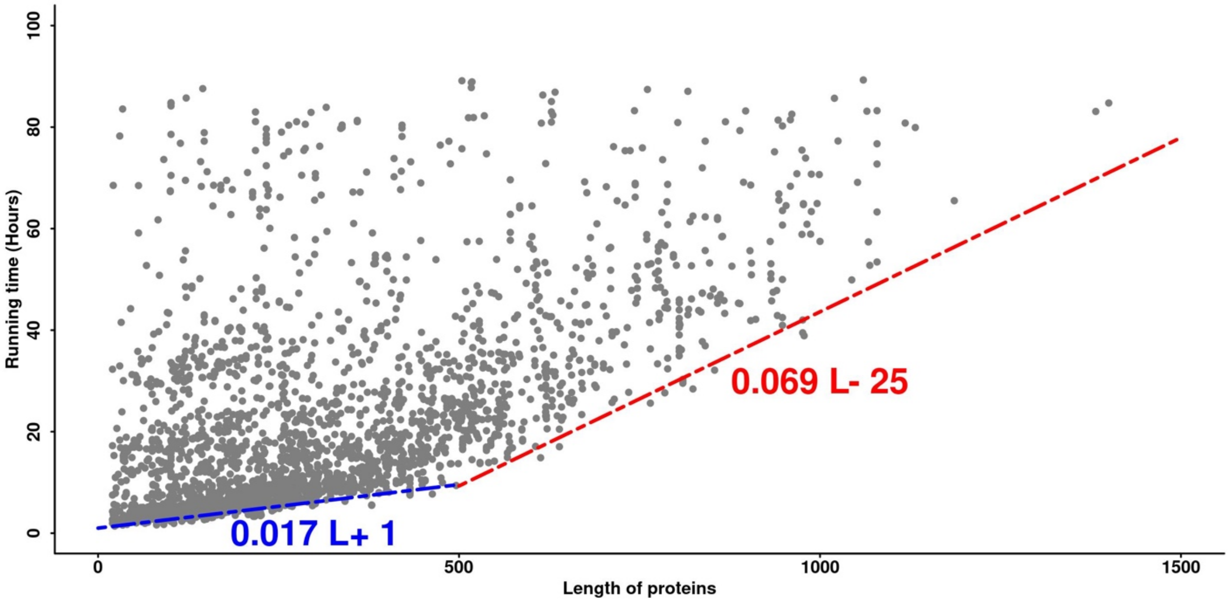
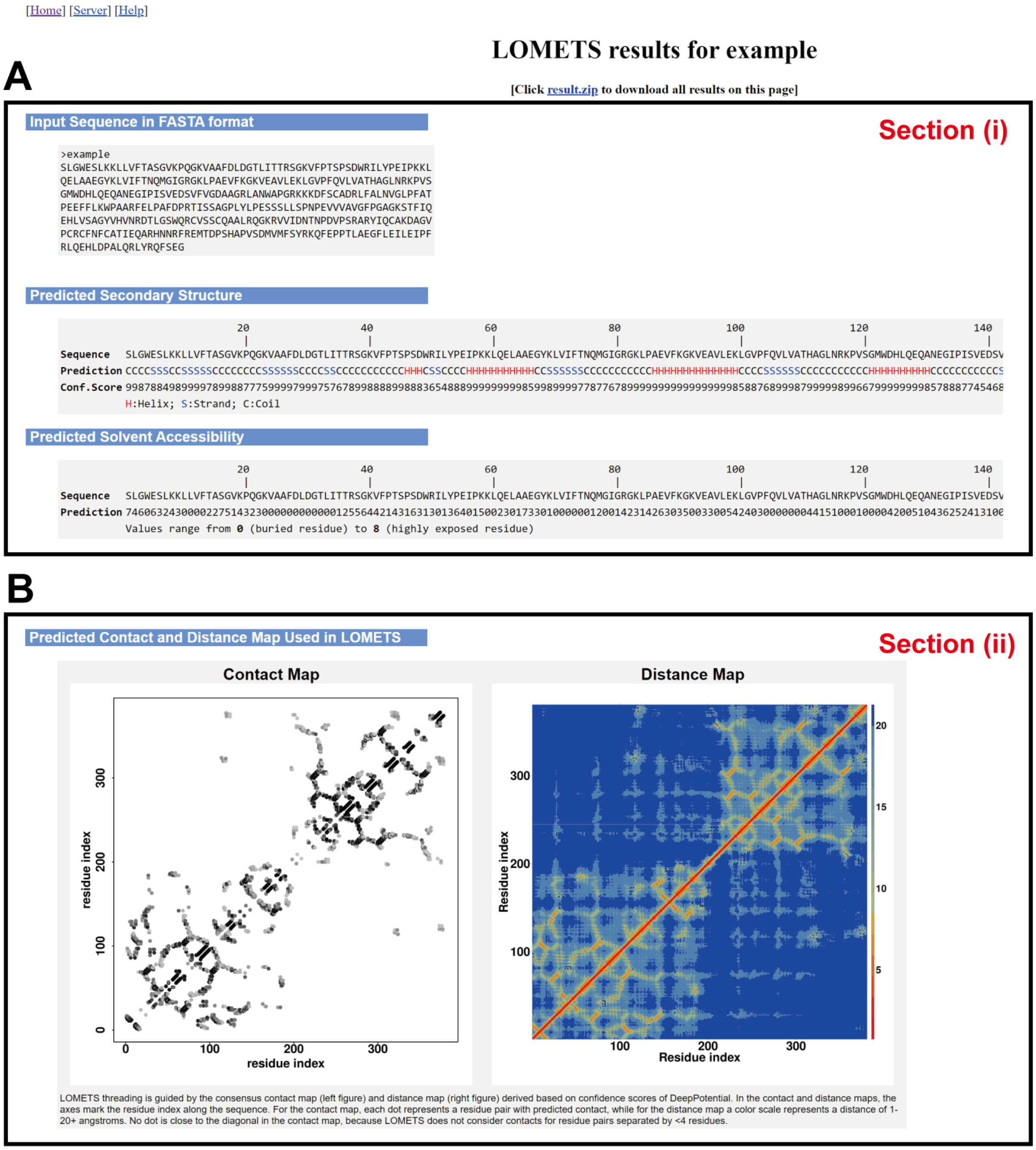


Figure S7. The response time versus protein size for the 3,779 jobs processed by the LOMETS3 server recently. In cases where many jobs are accumulated in the queue, it will take a longer time for each job to finish due to increased queue waiting time. The blue and red lines fit to the targets with the quickest response time, which should correspond to the actual running time of the LOMETS3 programs for single-domain targets when the job queue is clear. Notably, the multi-domain proteins will require longer processing time than the single-domain proteins, since they need a second round of calculation for domain-level threading and domain assembly.



[Home] [Server] [Help]

LOMETS results for example

[Click result.zip to download all results on this page]

A

Section (i)

Input Sequence in FASTA format

```
>example
SLGWESLKKLLVFTASGVKPQGKVAAFDLDGTLITTRSGKVFPTSPSDWRILYPEIPKKL
QELAAEGYKLVIFTNQMGIGRGKLPAEVFKGKVEAVLEKLGVPFQVLVATHAGLNRKPVS
GMWDHLQEQANEGIPISVEDSVFVGDAAGRLANWAPGRKKKDFSCADRLFALNVGLPFAT
PEEFFLKWPAARFELPAFDPRTISSAGPLYLPESSSLLSPNPEVVVAVGFPGAGKSTFIQ
EHLVSAGYVHVNRDTLGSWQRCVSSCQAALRQGKRVVIDNTNPDVPSRARYIQCAKDAGV
PCRCFNFCATIEQARHNNRFREMTDPSHAPVSDMVMFSYRKQFEPPTLAEGFLEILEIPF
RLQEHLDPALQRLYRQFSEG
```

Predicted Secondary Structure

```
                             20                  40                  60                  80                  100                 120                 140
                              |                   |                   |                   |                   |                   |                   |
Sequence   SLGWESLKKLLVFTASGVKPQGKVAAFDLDGTLITTRSGKVFPTSPSDWRILYPEIPKKLQELAAEGYKLVIFTNQMGIGRGKLPAEVFKGKVEAVLEKLGVPFQVLVATHAGLNRKPVSGMWDHLQEQANEGIPISVEDSV
Prediction CCCCSSSCCSSSSSCCCCCCCCSSSSSSCCCCSSCCCCCCCCCCCHHHCSSCCCCHHHHHHHHHHHCCSSSSSSCCCCCCCCCCCHHHHHHHHHHHHHHCCCCSSSSSSCCCCCCCCCCCHHHHHHHHHHCCCCCCCCCCCS
Conf.Score 9987884989999789988777599997999757678998888998883654888999999999985998999977877678999999999999999985887689998799999899667999999998578887745468
           H:Helix; S:Strand; C:Coil
```

Predicted Solvent Accessibility

```
                             20                  40                  60                  80                  100                 120                 140
                              |                   |                   |                   |                   |                   |                   |
Sequence   SLGWESLKKLLVFTASGVKPQGKVAAFDLDGTLITTRSGKVFPTSPSDWRILYPEIPKKLQELAAEGYKLVIFTNQMGIGRGKLPAEVFKGKVEAVLEKLGVPFQVLVATHAGLNRKPVSGMWDHLQEQANEGIPISVEDSV
Prediction 74606324300002275143230000000000000125564421431631301364015002301733010000001200142314263035003300542403000000044151000100004200510436252413100
           Values range from 0 (buried residue) to 8 (highly exposed residue)
```

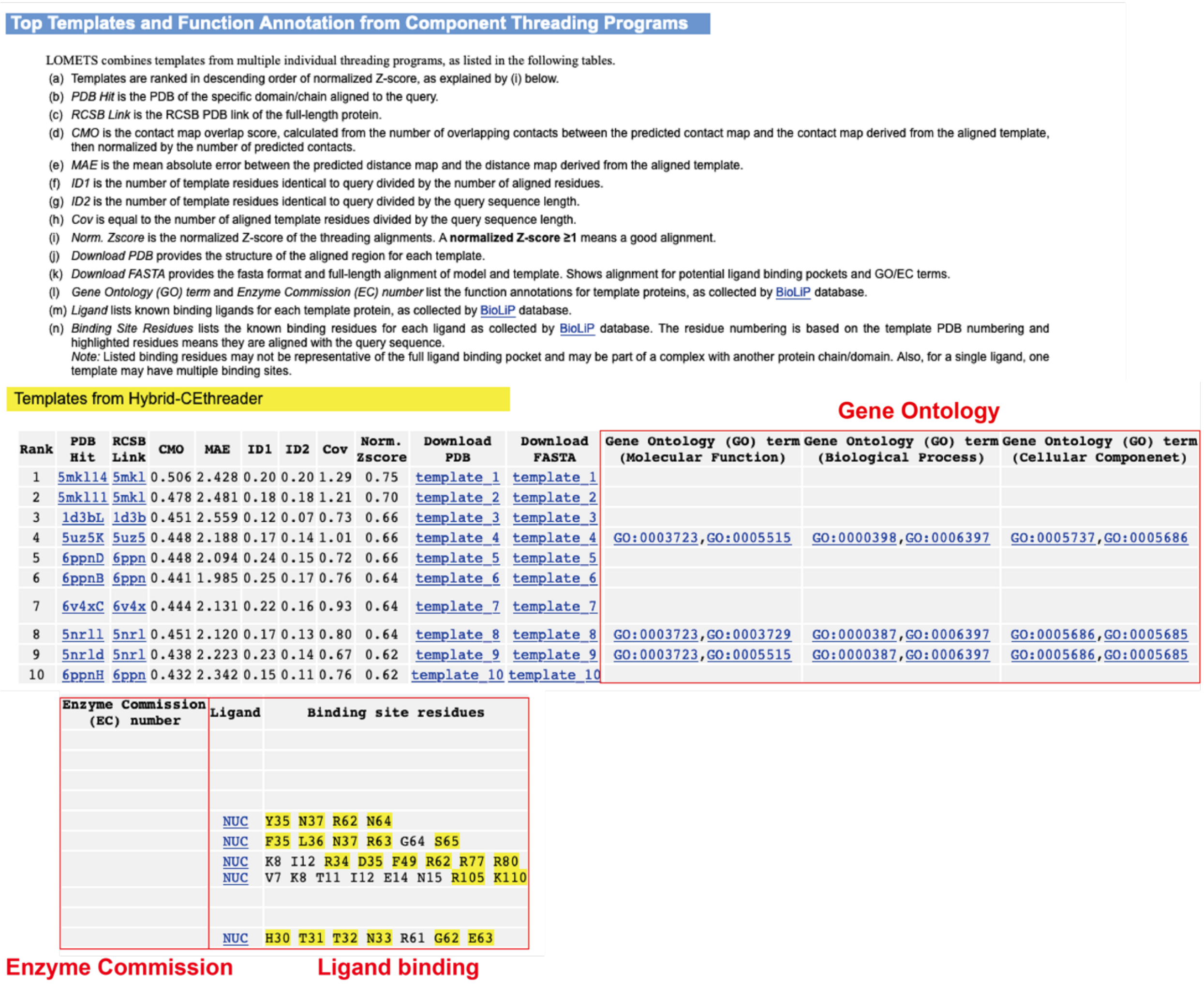
B

Section (ii)

Predicted Contact and Distance Map Used in LOMETS

LOMETS threading is guided by the consensus contact map (left figure) and distance map (right figure) derived based on confidence scores of DeepPotential. In the contact and distance maps, the axes mark the residue index along the sequence. For the contact map, each dot represents a residue pair with predicted contact, while for the distance map a color scale represents a distance of 1-20+ angstroms. No dot is close to the diagonal in the contact map, because LOMETS does not consider contacts for residue pairs separated by <4 residues.

Figure S8. Outputs of the LOMETS3 results page, (A) summary of the user input, including sequence, predicted secondary structure, and predicted solvent accessibility, (B) spatial restraints predicted by deep learning, including contact map and distance map.



Top Templates and Function Annotation from Component Threading Programs

LOMETS combines templates from multiple individual threading programs, as listed in the following tables.

(a) Templates are ranked in descending order of normalized Z-score, as explained by (i) below.
(b) *PDB Hit* is the PDB of the specific domain/chain aligned to the query.
(c) *RCSB Link* is the RCSB PDB link of the full-length protein.
(d) *CMO* is the contact map overlap score, calculated from the number of overlapping contacts between the predicted contact map and the contact map derived from the aligned template, then normalized by the number of predicted contacts.
(e) *MAE* is the mean absolute error between the predicted distance map and the distance map derived from the aligned template.
(f) *ID1* is the number of template residues identical to query divided by the number of aligned residues.
(g) *ID2* is the number of template residues identical to query divided by the query sequence length.
(h) *Cov* is equal to the number of aligned template residues divided by the query sequence length.
(i) *Norm. Zscore* is the normalized Z-score of the threading alignments. A **normalized Z-score ≥1** means a good alignment.
(j) *Download PDB* provides the structure of the aligned region for each template.
(k) *Download FASTA* provides the fasta format and full-length alignment of model and template. Shows alignment for potential ligand binding pockets and GO/EC terms.
(l) *Gene Ontology (GO) term* and *Enzyme Commission (EC) number* list the function annotations for template proteins, as collected by BioLiP database.
(m) *Ligand* lists known binding ligands for each template protein, as collected by BioLiP database.
(n) *Binding Site Residues* lists the known binding residues for each ligand as collected by BioLiP database. The residue numbering is based on the template PDB numbering and highlighted residues means they are aligned with the query sequence.
Note: Listed binding residues may not be representative of the full ligand binding pocket and may be part of a complex with another protein chain/domain. Also, for a single ligand, one template may have multiple binding sites.

Templates from Hybrid-CEthreader

Rank	PDB Hit	RCSB Link	CMO	MAE	ID1	ID2	Cov	Norm. Zscore	Download PDB	Download FASTA	Gene Ontology (GO) term (Molecular Function)	Gene Ontology (GO) term (Biological Process)	Gene Ontology (GO) term (Cellular Componenet)	Enzyme Commission (EC) number	Ligand	Binding site residues
1	5mkl14	5mkl	0.506	2.428	0.20	0.20	1.29	0.75	template_1	template_1						
2	5mkl11	5mkl	0.478	2.481	0.18	0.18	1.21	0.70	template_2	template_2						
3	1d3bL	1d3b	0.451	2.559	0.12	0.07	0.73	0.66	template_3	template_3						
4	5uz5K	5uz5	0.448	2.188	0.17	0.14	1.01	0.66	template_4	template_4	GO:0003723,GO:0005515	GO:0000398,GO:0006397	GO:0005737,GO:0005686			
5	6ppnD	6ppn	0.448	2.094	0.24	0.15	0.72	0.66	template_5	template_5					NUC	Y35 N37 R62 N64
6	6ppnB	6ppn	0.441	1.985	0.25	0.17	0.76	0.64	template_6	template_6					NUC	F35 L36 N37 R63 G64 S65
7	6v4xC	6v4x	0.444	2.131	0.22	0.16	0.93	0.64	template_7	template_7					NUC NUC	K8 I12 R34 D35 F49 R62 R77 R80 V7 K8 T11 I12 E14 N15 R105 K110
8	5nrl1	5nrl	0.451	2.120	0.17	0.13	0.80	0.64	template_8	template_8	GO:0003723,GO:0003729	GO:0000387,GO:0006397	GO:0005686,GO:0005685			
9	5nrld	5nrl	0.438	2.223	0.23	0.14	0.67	0.62	template_9	template_9	GO:0003723,GO:0005515	GO:0000387,GO:0006397	GO:0005686,GO:0005685			
10	6ppnH	6ppn	0.432	2.342	0.15	0.11	0.76	0.62	template_10	template_10					NUC	H30 T31 T32 N33 R61 G62 E63

Figure S9. Output of the LOMETS3 results page showing the top templates from the component threading programs and the associated function annotations.

Supporting Tables

Table S1. Comparison between the TM-scores of the first templates identified by LOMETS3 and LOMETS2 on 614 single-domain test proteins. P-values are calculated between the TM-scores for LOMETS3 and LOMETS2 using one-sided Student's t-tests. #{TM-score>0.5} is the number of targets with a TM-score >0.5.

Method	**Type**	**TM-score**	**P-value**	**RMSD**	**#{TM-score>0.5}**
LOMETS3	All	0.656	*	4.38	529
	Easy	0.725	*	3.48	395
	Hard	0.525	*	6.09	134
LOMETS2	All	0.624	1.7E-28	5.84	479
	Easy	0.709	6.9E-10	4.60	382
	Hard	0.462	2.3E-19	8.23	97

Table S2. Comparison between the TM-scores of the original and re-ranked first templates identified by profile-based threading methods on 614 test proteins. P-values are calculated between the TM-scores of the original and re-ranked templates for profile-based threading methods using one-sided Student's t-tests. #{TM-score>0.5} is the number of targets with a TM-score >0.5.

Method	Type	TM-score	P-value	RMSD	#{TM-score>0.5}
SparksX	All (614 targets)	0.590	*	6.98	436
SparksX^r		0.634	1.2E-35	5.25	484
MRFsearch		0.585	*	6.88	413
MRFsearch^r		0.622	5.7E-39	4.53	456
FFAS3D		0.575	*	6.80	415
FFAS3D^r		0.611	4.0E-29	5.23	453
MUSTER		0.570	*	7.47	411
MUSTER^r		0.626	5.3E-29	5.39	465
HHsearch		0.563	*	7.33	414
HHsearch^r		0.631	1.9E-45	5.18	475
SparksX	Easy (403 targets)	0.699	*	4.67	380
SparksX^r		0.715	1.0E-10	4.09	389
MRFsearch		0.689	*	4.60	363
MRFsearch^r		0.709	1.5E-13	3.58	379
FFAS3D		0.682	*	4.48	371
FFAS3D^r		0.700	2.9E-10	3.79	384
MUSTER		0.685	*	4.78	368
MUSTER^r		0.712	2.7E-09	4.16	386
HHsearch		0.687	*	5.08	365
HHsearch^r		0.714	3.9E-17	3.95	387
SparksX	Hard (211 targets)	0.383	*	11.39	56
SparksX^r		0.478	8.5E-25	7.48	95
MRFsearch		0.388	*	11.23	50
MRFsearch^r		0.456	5.9E-26	6.36	77
FFAS3D		0.371	*	11.24	44
FFAS3D^r		0.440	9.1E-21	8.00	69
MUSTER		0.349	*	12.60	43
MUSTER^r		0.461	3.3E-20	7.72	79
HHsearch		0.327	*	11.61	49
HHsearch^r		0.472	2.8E-27	7.51	88

"r" represents that the templates are re-ranked by DeepPotential predicted spatial restraints.

Table S3. Comparison between the TM-scores of the first templates identified by LOMETS3 and different component threading programs on 614 single-domain test proteins. P-values are calculated between the TM-scores for LOMETS3 and the component threading methods using one-sided Student's t-tests. #{TM-score>0.5} is the number of targets with a TM-score >0.5.

Method	Type	TM-score	P-value	RMSD	#{TM-score>0.5}
LOMETS3	All (614 targets)	**0.656**	*	**4.38**	**529**
Hybrid-CEthreader[c]		0.641	3.8E-13	4.91	502
SparksX[p]		0.634	4.4E-30	5.25	484
HHsearch[p]		0.631	8.9E-26	5.18	475
MUSTER[p]		0.626	5.6E-38	5.39	465
CEthreader[c]		0.623	6.4E-30	6.12	485
MRFsearch[p]		0.622	7.9E-49	4.53	456
MapAlign[c]		0.617	4.7E-38	6.40	470
FFAS3D[p]		0.611	5.9E-66	5.23	453
DisCovER[c]		0.607	3.9E-59	5.04	467
HHpred[p]		0.599	7.5E-37	6.48	441
EigenThreader[c]		0.590	1.0E-73	7.10	431
LOMETS3	Easy (403 targets)	**0.725**	*	**3.48**	**395**
Hybrid-CEthreader[c]		0.717	9.8E-05	3.87	390
SparksX[p]		0.715	2.0E-10	4.09	389
HHsearch[p]		0.714	1.3E-08	3.95	387
MUSTER[p]		0.712	3.1E-12	4.16	386
CEthreader[c]		0.702	1.2E-12	4.73	382
MRFsearch[p]		0.709	2.6E-23	3.58	379
MapAlign[c]		0.693	1.1E-22	4.74	376
FFAS3D[p]		0.700	3.0E-33	3.79	384
DisCovER[c]		0.691	1.8E-33	3.90	380
HHpred[p]		0.710	3.8E-09	4.54	381
EigenThreader[c]		0.670	2.8E-46	5.52	357
LOMETS3	Hard (211 targets)	**0.525**	*	**6.09**	**134**
Hybrid-CEthreader[c]		0.495	9.2E-11	6.89	112
SparksX[p]		0.478	3.1E-19	7.48	95
HHsearch[p]		0.472	2.6E-19	7.51	88
MUSTER[p]		0.461	5.5E-29	7.72	79
CEthreader[c]		0.472	1.8E-17	8.76	103
MRFsearch[p]		0.456	1.7E-29	6.36	77
MapAlign[c]		0.472	1.2E-14	9.55	94
FFAS3D[p]		0.440	6.4E-39	8.00	69
DisCovER[c]		0.445	7.3E-27	7.20	87
HHpred[p]		0.389	2.9E-29	10.20	60
EigenThreader[c]		0.437	2.2E-29	10.10	74

"c" represents the contact-based threading method, and "p" represents the profile-based threading method. Templates of the profile-based threading methods are re-ranked by the restraints from DeepPotential.

Table S4. Comparison between the full-length models generated by LOMETS3 and the control modeling methods on 614 single-domain test proteins. The control modeling methods include L-BFGS folding system with DeepPotential restraints, and MODELLER with LOMETS3 templates. P-values are calculated between the TM-scores for the LOMETS3 and the control modeling programs using one-sided Student's t-tests. #{TM-score>0.5} is the number of targets with a TM-score >0.5.

Method	Type	TM-score	P-value	RMSD	#{TM-score>0.5}
LOMETS3	All	0.814	*	3.93	602
	Easy	0.837	*	3.59	400
	Hard	0.768	*	4.58	202
L-BFGS + DeepPotential	All	0.797	6.6E-55	4.21	595
	Easy	0.820	3.2E-39	3.82	398
	Hard	0.752	2.0E-17	4.95	197
MODELLER + LOMETS3 template	All	0.695	1.7E-94	7.18	561
	Easy	0.759	6.1E-60	5.94	398
	Hard	0.575	1.7E-35	9.56	163

Table S5. Comparison between the TM-scores of the assembled templates by LOMETS3 (with domain partition and assembly) and full-chain threading templates (without domain partition and assembly) on 408 multi-domain targets for full-length assessment and 927 domains extracted from the 408 multi-domain proteins for domain-level assessment. P-values are calculated between the TM-scores for assembled templates and full-chain threading templates using one-sided Student's t-tests. #{TM-score>0.5} is the number of targets with a TM-score >0.5.

Target	Method	Type	TM-score	P-value	#{TM-score>0.5}
Full-length (408 proteins)	**Assembled template**	All	0.548	*	234
	full-chain threading template		0.506	5.1E-15	202
	Assembled template	Easy	0.581	*	213
	full-chain threading template		0.555	6.5E-07	193
	Assembled template	Hard	0.427	*	21
	full-chain threading template		0.327	2.2E-12	9
Domain-level (927 domains)	**Assembled template**	All	0.646	*	760
	full-chain threading template		0.500	3.1E-35	559
	Assembled template	Easy	0.687	*	716
	full-chain threading template		0.539	1.6E-25	531
	Assembled template	Hard	0.412	*	44
	full-chain threading template		0.278	1.4E-08	28

Table S6. Comparison between the models generated by AlphaFold2 with LOMETS3 templates and default HHsearch templates on 614 single-domain test proteins. P-values are calculated between the TM-scores for the AlphaFold2 with LOMETS3 templates and default HHsearch templates using one-sided Student's t-tests.

Method	Type	TM-score	P-value	RMSD
AlphaFold2 (with LOMETS3 templates)	All	0.9254	*	2.20
	Easy	0.9366	*	2.00
	Hard	0.9040	*	2.57
AlphaFold2 (with default HHsearch templates)	All	0.9210	4.49E-18	2.28
	Easy	0.9327	1.47E-11	2.11
	Hard	0.8986	3.39E-08	2.62

References

1. Zheng, W., Li, Y., Zhang, C., Zhou, X., Pearce, R., Bell, E.W., Huang, X. and Zhang, Y. (2021) Protein structure prediction using deep learning distance and hydrogen-bonding restraints in CASP14. *Proteins: Structure, Function, and Bioinformatics*, **n/a**.
2. Zhang, C., Zheng, W., Mortuza, S.M., Li, Y. and Zhang, Y. (2020) DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, **36**, 2105-2112.
3. Yang, J., Yan, R., Roy, A., Xu, D., Poisson, J. and Zhang, Y. (2015) The I-TASSER Suite: protein structure and function prediction. *Nature Methods*, **12**, 7-8.
4. Zhou, X., Hu, J., Zhang, C., Zhang, G. and Zhang, Y. (2019) Assembling multidomain protein structures through analogous global structural alignments. *Proceedings of the National Academy of Sciences*, **116**, 15930.
5. Xue, Z., Xu, D., Wang, Y. and Zhang, Y. (2013) ThreaDom: extracting protein domain boundary information from multiple threading alignments. *Bioinformatics*, **29**, i247-i256.
6. Zheng, W., Zhou, X., Wuyun, Q., Pearce, R., Li, Y. and Zhang, Y. (2020) FUpred: detecting protein domains through deep-learning-based contact map prediction. *Bioinformatics*, **36**, 3749-3757.
7. Zhang, Y. and Skolnick, J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, **33**, 2302-2309.
8. Zheng, W., Zhang, C., Li, Y., Pearce, R., Bell, E.W. and Zhang, Y. (2021) Folding non-homologous proteins by coupling deep-learning contact maps with I-TASSER assembly simulations. *Cell Reports Methods*, **1**, 100014.
9. Huang, X., Pearce, R. and Zhang, Y. (2020) FASPR: an open-source tool for fast and accurate protein side-chain packing. *Bioinformatics*, **36**, 3758-3765.
10. Zhang, J., Liang, Y. and Zhang, Y. (2011) Atomic-Level Protein Structure Refinement Using Fragment-Guided Molecular Dynamics Conformation Sampling. *Structure*, **19**, 1784-1795.
11. Zhang, C., Freddolino, P.L. and Zhang, Y. (2017) COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Research*, **45**, W291-W299.
12. Yang, J., Roy, A. and Zhang, Y. (2013) BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, **41**, D1096-D1103.
13. Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C.H. and the UniProt, C. (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926-932.
14. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K.P. *et al.* (2015) STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, **43**, D447-D452.
15. Furnham, N., Holliday, G.L., de Beer, T.A.P., Jacobsen, J.O.B., Pearson, W.R. and Thornton, J.M. (2014) The Catalytic Site Atlas 2.0: cataloging catalytic sites and residues identified in enzymes. *Nucleic Acids Research*, **42**, D485-D489.
16. Rogers David, J. and Tanimoto Taffee, T. (1960) A Computer Program for Classifying Plants. *Science*, **132**, 1115-1118.
17. Zheng, W., Wuyun, Q., Li, Y., Mortuza, S.M., Zhang, C., Pearce, R., Ruan, J. and Zhang, Y. (2019) Detecting distant-homology protein structures by aligning deep neural-network based contact maps. *PLOS Computational Biology*, **15**, e1007411.
18. Bhattacharya, S., Roche, R. and Bhattacharya, D. (2020) DisCovER: distance-based covariational threading for weakly homologous proteins. *bioRxiv*, 2020.2001.2031.923409.
19. Buchan, D.W.A. and Jones, D.T. (2017) EigenTHREADER: analogous protein fold recognition by efficient contact map threading. *Bioinformatics*, **33**, 2684-2690.
20. Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G.A., Kim, D.E., Kamisetty, H., Kyrpides, N.C. and Baker, D. (2017) Protein structure determination using metagenome sequence data. *Science*, **355**, 294.
21. Xu, D., Jaroszewski, L., Li, Z. and Godzik, A. (2013) FFAS-3D: improving fold recognition by including optimized structural features and template re-ranking. *Bioinformatics*, **30**, 660-667.
22. Meier, A. and Söding, J. (2015) Automatic prediction of protein 3D structures by probabilistic multi-template homology modeling. *PLoS computational biology*, **11**, e1004343.
23. Söding, J. (2005) Protein homology detection by HMM–HMM comparison. *Bioinformatics*, **21**, 951-960.

24. Ma, J., Wang, S., Wang, Z. and Xu, J. (2014) MRFalign: Protein Homology Detection through Alignment of Markov Random Fields. *PLOS Computational Biology*, **10**, e1003500.

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

26. Zhou, H. and Zhou, Y. (2005) Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins: Structure, Function, and Bioinformatics*, **58**, 321-328.

27. Di Lena, P., Fariselli, P., Margara, L., Vassura, M. and Casadio, R. (2010) Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, **26**, 2250-2258.

28. Zhang, Y. and Skolnick, J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, **57**, 702-710.

29. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017) Automatic differentiation in pytorch.