

Structural bioinformatics

Full-length *de novo* protein structure determination from cryo-EM maps using deep learning

Jiahua He and Sheng-You Huang  *

School of Physics, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on December 22, 2020; revised on April 3, 2021; editorial decision on May 2, 2021; accepted on May 8, 2021

Abstract

Motivation: Advances in microscopy instruments and image processing algorithms have led to an increasing number of Cryo-electron microscopy (cryo-EM) maps. However, building accurate models for the EM maps at 3–5 Å resolution remains a challenging and time-consuming process. With the rapid growth of deposited EM maps, there is an increasing gap between the maps and reconstructed/ modeled three-dimensional (3D) structures. Therefore, automatic reconstruction of atomic-accuracy full-atom structures from EM maps is pressingly needed.

Results: We present a semi-automatic *de novo* structure determination method using a deep learning-based framework, named as DeepMM, which builds atomic-accuracy all-atom models from cryo-EM maps at near-atomic resolution. In our method, the main-chain and C α positions as well as their amino acid and secondary structure types are predicted in the EM map using Densely Connected Convolutional Networks. DeepMM was extensively validated on 40 simulated maps at 5 Å resolution and 30 experimental maps at 2.6–4.8 Å resolution as well as an Electron Microscopy Data Bank-wide dataset of 2931 experimental maps at 2.6–4.9 Å resolution, and compared with state-of-the-art algorithms including RosettaES, MAINMAST and Phenix. Overall, our DeepMM algorithm obtained a significant improvement over existing methods in terms of both accuracy and coverage in building full-length protein structures on all test sets, demonstrating the efficacy and general applicability of DeepMM.

Availability and implementation: <http://huanglab.phys.hust.edu.cn/DeepMM>.

Contact: huangsy@hust.edu.cn

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

1 Introduction

Cryo-electron microscopy (cryo-EM) has now become a widely used technique for structure determination of macromolecular structures in the recent decade (Cheng, 2018; Frank, 2017; Nogales, 2016; Raunser, 2017). Advances in microscopy instruments and image processing algorithms have led to the rapid increase in the number of solved EM maps (Cheng, 2018; Frank, 2017; Nogales, 2016). The ‘resolution revolution’ in cryo-EM has paved a way for the determination of high-resolution structures of previously intractable biological systems (Adams *et al.*, 2010; Joseph *et al.*, 2020; Kim *et al.*, 2020; Li *et al.*, 2013; Luque and Castón, 2020; Punjani *et al.*, 2017; Safdari *et al.*, 2018; Scheres, 2012; Xie *et al.*, 2020; Yang *et al.*, 2018; Yin *et al.*, 2019; Zhang *et al.*, 2020). According to the statistics of the Electron Microscopy Data Bank (EMDB) (Patwardhan, 2017), there were 2435 maps deposited in 2019, which are almost four times the 640 maps released in 2015.

With the rapid growth of deposited EM maps, there is an increasing gap between the maps and reconstructed/ modeled three-dimensional (3D) structures. As of April 1, 2020, there were 10 560

EMDB maps, but only 4805 associated structures were deposited in the Protein Data Bank (PDB) (Berman *et al.*, 2000). For those maps determined at near-atomic resolution (3.0–5.0 Å), it is difficult to build high-resolution models with conventional software designed for X-ray crystallography. In view of the fact that near-atomic resolution maps take up the majority of current and henceforth released maps (Patwardhan, 2017), tools, which can reconstruct structures *de novo* from EM maps without using known structures as templates (Alnabati and Kihara, 2019), are pressingly needed. As such, some algorithms like EM-fold (Lindert *et al.*, 2009), Gorgon (Baker *et al.*, 2011), Rosetta (Frenz *et al.*, 2017; Wang *et al.*, 2015), Pathwalking (Baker *et al.*, 2012; Chen *et al.*, 2016; Chen and Baker, 2018), Phenix (Afonine *et al.*, 2018a; Terwilliger *et al.*, 2018, 2020) and MAINMAST (Terashi *et al.*, 2020; Terashi and Kihara, 2018) have been recently presented for constructing and/or assembling structure fragments from cryo-EM maps.

Despite the present progress in *de novo* structure building for cryo-EM maps, there are various limitations in current approaches. They can either only build structural fragments (Baker *et al.*, 2011; Lindert *et al.*, 2009; Terwilliger *et al.*, 2020) or have limited

accuracy in terms of coverage and/or sequence reproduction (Baker *et al.*, 2012; Frenz *et al.*, 2017; Terashi and Kihara, 2018). It remains challenging to automatically build an accurate structure from the EM maps at near-atomic resolution. Recently, machine learning has been actively applied in structure determination for EM maps, such as single-particle picking (Tegunov and Cramer, 2019), tomogram annotation (Chen *et al.*, 2017), secondary structure prediction (He and Huang, 2021; Maddhuri Venkata Subramaniya *et al.*, 2019; Mostosi *et al.*, 2020) and backbone tracing (Pfab *et al.*, 2021; Si *et al.*, 2020). However, applying deep learning to build full-length protein structures for near-atomic resolution EM maps remains a challenging work.

Here, we have developed a semi-automatic *de novo* atomic-accuracy structure reconstruction method for EM maps at near-atomic resolution through Densely Connected Convolutional Networks (DenseNets) using a deep learning-based framework, named DeepMM. Instead of tracing the protein main-chain on the raw EM density map, DeepMM first predicts the probability of main-chain atoms (N, C and C α) and C α positions near each grid point using one DenseNet (Huang *et al.*, 2017). Then, the method traces the main-chain according to the predicted main-chain probability map. The amino acid and secondary structure types are predicted by a second DenseNet. Finally, the protein sequence is aligned to the main-chain according to the predicted C α probabilities, amino acid types and secondary structure types for all-atom structure building.

2 Materials and methods

2.1 Workflow of DeepMM

The workflow of DeepMM is illustrated in Figure 1a. Specifically, starting from a cryo-EM map and the target protein sequence, DeepMM first standardizes the order of axis, and interpolates grid interval to 1.0 Å. Then, DeepMM cuts the entire map into small boxes of size 11 Å \times 11 Å \times 11 Å around each voxel. Afterwards, one

DenseNet (say DenseNet A) is used to predict the main-chain and C α probability on each of the voxels. All the predicted probability values form a 3D probability map. Next, possible main-chain paths are generated in the predicted main-chain probability map using a main-chain tracing algorithm (Terashi and Kihara, 2018). Since the main-chain points are not always on the integer grid after mean shift in main-chain tracing, the C α probability values of main-chain points will be interpolated from the predicted 3D C α probability map. Afterwards, the amino acid and secondary structure types are predicted for each main-chain point through the second DenseNet (say DenseNet B). It is worth mentioning that in order to take advantage of multi-task learning and improve prediction accuracy, C α probabilities of main-chain points are interpolated from the C α probability map that is predicted with the main-chain probability map, instead of being directly predicted on main-chain points. With the predicted C α probability, amino acid type and secondary structure type for each main-chain point, the target protein sequence is then aligned to the main-chain paths using the Smith–Waterman dynamic programming (DP) algorithm (Smith and Waterman, 1981). The resulted multiple C α models are ranked by their alignment scores. Finally, the all-atom structures are constructed from the top C α models using the *ctrip* program in the Jackal modeling package (Petrey *et al.*, 2003; Xiang and Honig, 2001) and refined by an energy minimization using Amber (Case *et al.*, 2005).

2.2 Training the DenseNets of DeepMM

Two DenseNets are embedded into our DeepMM algorithm. Figure 1b illustrates the architecture of the networks. DenseNet is a feed-forward multi-layer network which uses additional paths between earlier and later layers in a dense block. DenseNets have several compelling advantages. They alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse and substantially reduce the number of parameters (Huang *et al.*, 2017). DeepMM also employs a hard parameter-sharing multi-task learning method, which can greatly reduce the risk of overfitting

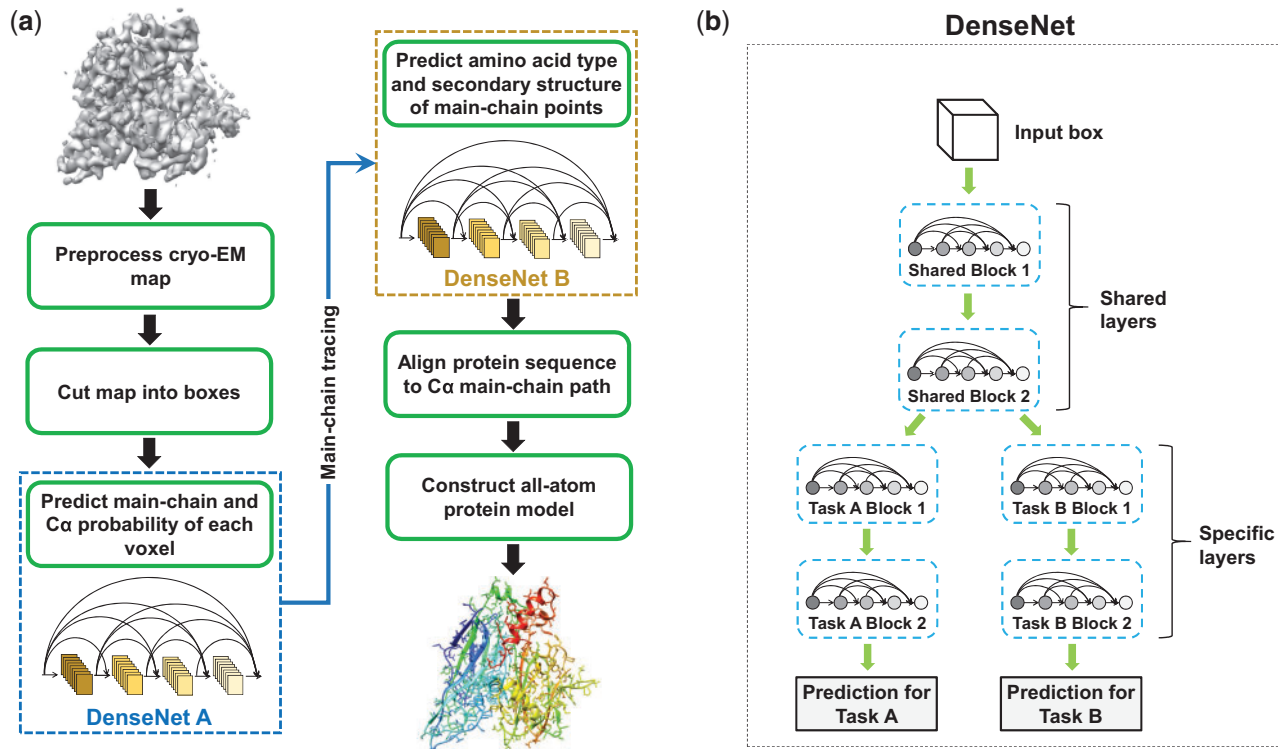


Fig. 1. Workflow of our DeepMM method. (a) The flowchart of DeepMM. DeepMM first predicts the main-chain and C α probability of each density voxel using a Densely Connected Convolutional Network (DenseNet), and then traces the protein's main-chain path on the predicted main-chain probability map. Next, the amino acid and secondary structure types for each main-chain point are predicted by a second DenseNet. The C α models are generated by aligning the target sequence to the main-chain paths. Finally, the all-atom structures are constructed from the C α models using the *ctrip* program and refined by an Amber energy minimization. (b) The multi-task deep DenseNet architecture used in DeepMM. Starting from the input EM density box around a voxel, two dense blocks are shared by both tasks in DenseNet A, while only one dense block is shared by both tasks in DenseNet B. Each prediction task employs two task-specific dense blocks and gives the final prediction

(Ruder, 2017). The first network (i.e. DenseNet A) is used to simultaneously predict the main-chain probability and C α probability of a grid point. The second network (i.e. DenseNet B) is used to predict the amino acid type and secondary structure type of a main-chain local dense point (LDP). The input for the DenseNet A are the boxes of size 11 Å × 11 Å × 11 Å. The second network (DenseNet B) takes the boxes of size 10 Å × 10 Å × 10 Å as input because main-chain points are not always on the integer grid after mean shift. For each box, the density values are normalized to the range of [0, 1] according to the maximum and minimum density values in the box. 3D convolutions and 3D pooling layers are used instead of their 2D counterparts used in traditional image processing because the density maps have three dimensions. Several dense blocks are used in both networks, each of which consists of eight densely connected layers. For DenseNet A, the first two dense blocks are shared by both tasks, whereas for DenseNet B, only one shared block is adopted. After the shared blocks, each task employs two task-specific blocks and gives the final prediction. The details of network architecture are provided in [Supplementary Table S1](#).

For DenseNet A, all the grid points above a density value D_0 were used for training, where D_0 was set to 1.0 for simulated maps at 5.0 Å resolution. For an experimental map, D_0 was set to 1/2 of its recommended contour level. The labels (main-chain probability and C α probability) of a grid point \vec{a} were calculated as follows:

$$P_{\vec{a}}^X = \min \left\{ e^{-\frac{\|\vec{a}-\vec{X}\|^2}{r_0^2}}, \forall \vec{X} \in \|\vec{a}-\vec{X}\| < r_{\text{cut}} \right\}, \quad (1)$$

where X stands for the N, C or C α atoms. The r_0 is the radius at which the probability drop to $1/e$. If no atom is within r_{cut} of a grid point, the corresponding probability is set to 0. The values of r_0 and r_{cut} are set to 1.0 Å and 2.0 Å, respectively. A total of 512 boxes were trained in one batch and 30 epochs were trained for the whole dataset. Adam optimizer with an initial learning rate of 0.001 was used to minimize the mean absolute error. Learning rate decay was adopted, where the learning rate was reduced to 1/10 of the current value after every 10 epochs. To avoid overfitting, the $\text{weight}_{\text{decay}}$ parameter of Adam optimizer was set to $1e-6$ as the L2 regularization.

For DenseNet B, one point was randomly sampled within 1.0 Å for every main-chain atom in the training set. The corresponding amino acid type and second structure type marked by STRIDE (Heinig and Frishman, 2004) were assigned to each point. Twenty types of amino acids were grouped into four classes according to their sizes, shapes and distributions in their EM density maps (Ho *et al.*, 2020), as illustrated in [Figure 2d](#). Specifically, GLY, ALA, SER, CYS, VAL, THR, ILE and PRO are grouped as Class I. LEU, ASP, ASN, GLU, GLN and MET are grouped as Class II. LYS and ARG are grouped as Class III. HIS, PHE, TYR and TRP are grouped as Class IV. Residues that have structure codes of H, G or I by STRIDE were labeled as ‘Helix’, those with codes of B/b or E were labeled as ‘Sheet’ and the other residues were labeled as ‘Coil’. All the training parameters were identical to those for DenseNet A except for using CrossEntropyLoss as loss function.

It should be noted that the experimental EM map can be quite different from the simulated map for the same structure in terms of quality and pattern. Compared with simulated maps, experimental maps often contain a lot of noises. In addition, the electron density signal in experimental maps may be totally different from that simulated from a known structure, because the simulated map assumes that all atoms interact with the electron beam as if they were uncharged and unbound. Therefore, the model trained for simulated maps cannot be directly transferrable to experimental maps. As such, two different models have been trained for simulated EM maps and experimental EM maps, respectively. All the training parameters and procedure used for simulated EM maps are essentially the same as those used for experimental EM maps unless otherwise specified.

2.3 Tracing the main-chain path

The main-chain tracing algorithm in MAINMAST (Terashi and Kihara, 2018) was used to trace the main-chain path in our

predicted main-chain probability map. In brief, LDPs are first identified using the mean shift algorithm, which iteratively shifts the initial grid points toward the local highest probability by computing the weighted average of probability values. Then, the shifted points that are within a threshold distance of 0.5 Å are clustered, and the point with the highest probability in the cluster is chosen as the representative, called LDP. The next step is to connect LDPs into a minimum spanning tree (MST) and iteratively refine the tree structure with a Tabu search method. After multiple steps of Tabu search, the longest path of the refined tree is traced as the main-chain path. The details of the algorithm can be found in the MAINMAST study (Terashi and Kihara, 2018).

2.4 Aligning target sequence to main-chain path

The Smith–Waterman DP algorithm (Smith and Waterman, 1981) is used to align the target sequence to the predicted main-chain path. The predicted C α probability value, amino acid type and secondary structure type are assigned to each point of the main-chain. Instead of using 20 amino acid types, amino acids are grouped into four classes according to their sizes, shapes and distributions in EM density maps ([Fig. 2d](#)). Secondary structures are categorized into three types of Helix, Sheet and Coil. The match between the target sequence and main-chain path is evaluated by two scoring matrices for amino acid and secondary structure, respectively ([Fig. 2b](#)). Namely, a target residue is more likely to be aligned to a main-chain point with the same amino acid type, the same secondary structure type and a higher C α probability and vice versa. The detailed alignment protocol is shown in [Figure 2a–c](#). The n residues $\{A_i(i=1, \dots, n)\}$ in the protein are aligned to m LDPs $\{L_j(j=1, \dots, m)\}$ in the main-chain path. The matching score $M(i, j)$ for a pair of A_i and L_j is computed as follows:

$$M(i, j) = w_{AA}M_{AA}(T_{AA}(A_i), T_{AA}(L_j)) + w_{SS}M_{SS}(T_{SS}(A_i), T_{SS}(L_j)), \quad (2)$$

where M_{AA} and M_{SS} are the scoring matrices for amino acid and secondary structure matching (Ho *et al.*, 2020; Wen *et al.*, 2020), respectively. For a residue A_i , the amino acid type is one of the four amino acid classes ($T_{AA}(A_i) = 1, 2, 3, 4$). The predicted amino acid type for an LDP L_j is also one of the four amino acid classes ($T_{AA}(L_j) = 1, 2, 3, 4$). Similarly, the secondary structure matching score is calculated using the secondary structure type predicted from the sequence ($T_{SS}(A_i) = 1, 2, 3$) by SPIDER2 (Heffernan *et al.*, 2016) and secondary structure type predicted on LDPs ($T_{SS}(L_j) = 1, 2, 3$). The scoring matrices M_{AA} and M_{SS} used in the alignment are shown in [Figure 2b](#). The w_{AA} and w_{SS} are the weights for corresponding matching scores and are set to 1.0 and 0.5, respectively. With the calculated matching score $M(i, j)$, an alignment is calculated with the follow rule to form a DP matrix, F , as follows:

$$F(i, j) = \max \begin{cases} F(i-1, j) + \text{gap} \\ F(i-1, j-1) - w_{C\alpha-C\alpha}|d_{\text{std}} - d| + w_{C\alpha}P_{C\alpha}(j) + M(i, j) \\ F(i, j-1), \end{cases} \quad (3)$$

where gap is the gap penalty for unaligned residues in the protein sequence. To ensure a full-length structure reconstruction, gap is set to $-10\,000.0$ so as to forbid skipped residues. The $|d_{\text{std}} - d|$ is the penalty score for C α –C α distance, where d_{std} is the standard C α –C α distance and d is the distance between LDP L_j and the last aligned LDP. The $P_{C\alpha}(j)$ is the predicted C α probability for LDP L_j . The $w_{C\alpha-C\alpha}$ and $w_{C\alpha}$ are the weights for the corresponding scores. Here, $w_{C\alpha}$ is set to 1.6, and $w_{C\alpha-C\alpha}$ is set to 1.0, 0.7 and 0.8 for ‘Helix’, ‘Sheet’ and ‘Coil’, respectively. For each combination of parameters in the main-chain tracing procedure, 160 C α models are generated. Finally, all the generated C α models are ranked by their alignment scores.

2.5 Parameter settings of DeepMM

The parameters of mean shift, MST construction and Tabu search are set to be the same to those in MAINMAST (Terashi and Kihara,

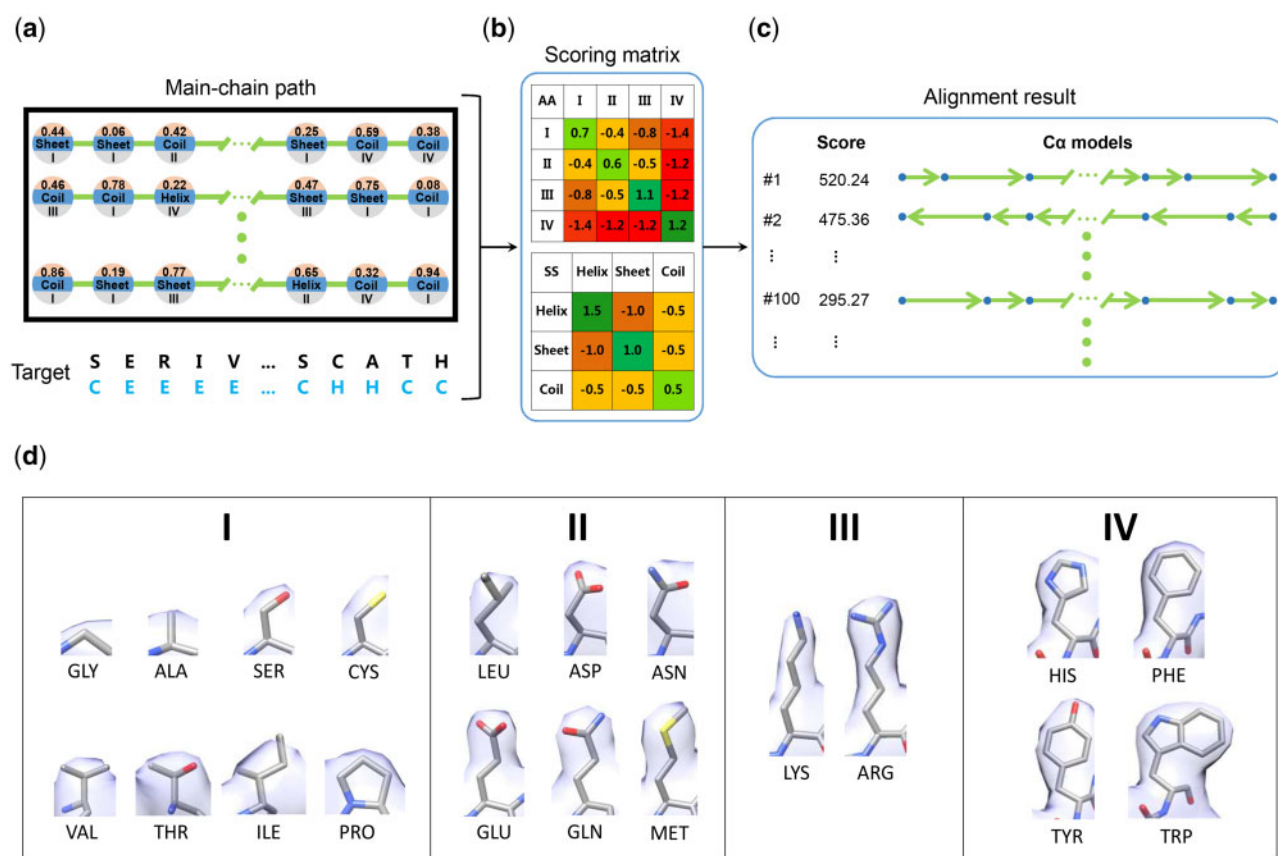


Fig. 2. Alignment protocol between the target sequence and the predicted main-chain for DeepMM. (a) DeepMM runs alignments of the target sequence of the EM map against each candidate main-chain path. Each sphere represents a predicted local dense point (LDP) on the main-chain path. Predicted information including the Cz probability (on the top), secondary structure (in the middle) and amino acid class (at the bottom) of LDPs is utilized during alignment. For the target sequence, its secondary structure is predicted by the SPIDER2 program, as illustrated in the sequence colored in azure under the amino acid sequence. (b) Scoring matrices for amino acid type matching and secondary structure matching. (c) The generated Cz models are ranked by their alignment scores. (d) Twenty amino acids are grouped into four classes according to the similarity of their side-chain EM densities

2018), unless otherwise specified. DeepMM employs several parameter combinations to generate multiple C α models for one EM map. For each combination of parameters, 10 trajectories of Tabu search are carried out, yielding 10 main-chain paths. Since DeepMM starts from the main-chain probability map, fewer parameter combinations are needed to reconstruct reliable 3D structures. For both simulated and experimental maps, the thresholds of probability (Φ_{thr}) and normalized probability (θ_{thr}) are both set to 0. For the 40 simulated maps, only one parameter combination is adopted. Specifically, the maximum number of Tabu search steps (N_{round}) is set to 100, the sphere radius of local MST (r_{local}) is set to 5.0 Å and the constraint for the length (d_{keep}) is set to 0.5 Å. For the 30 experimental maps, we employ the following 27 combinations of parameters: the sphere radius of local MST ($r_{\text{local}} = 5.0, 7.5, 10.0$ Å), the edge weight threshold ($d_{\text{keep}} = 0.5, 1.0, 1.5$ Å), and the maximum number of the Tabu search steps ($N_{\text{round}} = 2500, 5000, 7500$). For the extended EMDB-wide test set of 2931 maps, we employ fewer combinations of parameters so as to save computational cost: the edge weight threshold ($d_{\text{keep}} = 0.5, 1.0$ Å) and the maximum number of the Tabu search steps ($N_{\text{round}} = 2500, 5000$). The sphere radius of local MST (r_{local}) is set to 10 Å. For each of the generated main-chain path, 16 C α models are generated using eight different standard C α –C α distances ($d_{\text{std}} = 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8$ Å) on two sequence directions. Namely, 160 models (16 models for each of the 10 trajectories) are constructed for each parameter combination. The C α models are ranked by their alignment scores and then an RMSD cutoff of 5 Å is used to remove the one with lower alignment score in two similar structures. Finally, the top 10 scored protein C α models are selected to build the all-atom structures.

2.6 Datasets used

Our datasets consist of the protein structures from the PDB (Berman *et al.*, 2000) and their corresponding cryo-EM maps that were simulated using the EMAN2 software (Tang *et al.*, 2007) or downloaded from the EMDB (Patwardhan, 2017). For the training sets, we only used those protein chains from the PDB structure containing proteins only. For the test sets, the protein chains can be from pure-protein structures or protein–nucleic acid complexes in the PDB so as to test the general applicability of our DeepMM method.

2.6.1 Training sets

Two datasets, simulated EM map set and experimental EM map set, were used to train our DeepMM method for simulated maps and experimental maps, respectively.

For simulated EM maps, 2000 representative structures for different superfamilies in the SCOPe database (Fox *et al.*, 2014) were taken from Emap2sec (Maddhuri Venkata Subramaniya *et al.*, 2019) as training set. Those structures were removed from the training set if they have a TM-score (Zhang and Skolnick, 2005) of over 0.5 with any structure in the test set. To save the computational cost, only 100 randomly selected structures from the training set were retained. Next, we used the *e2pdb2mrc.py* program from the EMAN2 package (version 2.11) (Tang *et al.*, 2007) to generate the simulated EM maps at 5.0 Å resolution and 1.0 Å grid interval for each structure in training and test set. The training SCOPe entries used in this study were listed in Supplementary Table S5.

For experimental EM maps, all the EM density maps at 2–5 Å resolution that have associated PDB models were downloaded from

the EMDB. As of December 26, 2019, 2546 EM maps were collected. The PDB structure and its corresponding EM map that met the following criteria were removed: (i) including nucleic acid chains, (ii) missing side-chain atoms, (iii) including 'HETATM' residues, (iv) including 'UNK' residues, (v) including more than 1 subunit (MODEL) and (vi) containing <50 residues. In addition, we have also removed those protein chains with more than 300 residues to save time in the training process. Then, 1588 chains from the remaining 361 experimental EM maps were clustered with 50% sequence identity using CD-HIT (Fu *et al.*, 2012), yielding a total of 1340 chains. To ensure a valid evaluation, chains were removed from training set if they have over 30% sequence identity with any chain in the test set. Each protein chain was segmented out from the whole map using a distance of 4.0 Å (Terashi and Kihara, 2018). For good quality maps, protein chain and its associated map should have sufficient structural agreement. The cross-correlation between the experimental map and the simulated map density at the same resolution with the experimental map generated from the structure was calculated using the UCSF Chimera (Pettersen *et al.*, 2004). Only the chains with a cross-correlation of over 0.65 were kept (Maddhuri Venkata Subramaniya *et al.*, 2019). The final training set consists of 100 non-redundant protein chains. The grid intervals for experimental maps were unified to 1.0 Å using trilinear interpolation. The training EM maps and their corresponding PDB chains used in this study are listed in Supplementary Table S6.

2.6.2 Test sets

Three test sets were used to evaluate our DeepMM approach for its accuracy and general applicability, including one simulated map set and two experimental maps.

The simulated map set was taken from the test set of 40 simulated maps used by MAINMAST (Terashi and Kihara, 2018). The maps were generated at 5.0 Å resolution with a grid spacing of 1.0 Å using the *e2pdb2mrc.py* program in the EMAN2 package (Tang *et al.*, 2007).

The first experimental test set is the benchmark of 30 EM maps at 2.6–4.8 Å resolution, which has been used to evaluate MAINMAST (Terashi and Kihara, 2018). The corresponding EM maps were downloaded from the EMDB. For each EM map, a single subunit was segmented out from the whole density map at a distance cutoff of 4.0 Å.

In addition, to evaluate the accuracy and general applicability of DeepMM, we have also constructed a large test set of EMDB-wide experimental maps. The generation procedure of this set was similar to that for the experimental training set. Specifically, for each chain of the EM PDB structure at 2.5–5.0 Å resolution and no more than one subunit (MODEL) from the EMDB, a single density patch was segmented out from the whole density map at a distance cutoff of 4.0 Å. The protein chain and its corresponding EM map patch that met the following criteria were removed: (i) missing side-chain atoms, (ii) including 'HETATM' residues, (iii) including 'UNK' residues, (iv) containing <50 residues and (v) having over 30% sequence identity to any chain in the training set. We have also removed those protein chains with more than 300 residues to save time in the evaluation on this test set, though our DeepMM model is applicable to protein structures of any lengths. The cross-correlation between the experimental map and the simulated density map at the same resolution generated from the structure should be over 0.65 (Maddhuri Venkata Subramaniya *et al.*, 2019). Each protein chain was segmented out from the whole map using a distance of 4.0 Å (Terashi and Kihara, 2018). The final test set consists of 2931 protein chains, which are listed in Supplementary Table S4.

3 Results and discussion

3.1 Model reconstruction for simulated EM maps

We first evaluated the performance of our DeepMM algorithm on the test set of 40 simulated density maps at 5 Å resolution. DeepMM traced the main-chain of protein on the predicted main-chain probability map rather than the raw EM density map. Thus, the

generated C α models by our DeepMM are closer to the deposited structures with fewer search trajectories and steps compared to MAINMAST. For each of the 40 maps, DeepMM built 160 C α models, which were ranked by their alignment scores. The top-ranked model was selected as the predicted structure.

Figure 3 shows a comparison of the predicted C α models for the protein chains of different lengths by DeepMM and MAINMAST. The detailed results are provided in Supplementary Table S2. It can be seen from the figure that our DeepMM method obtained a much better performance than MAINMAST. As shown in Figure 3a, DeepMM built significantly more accurate C α models, and achieved an average C α RMSD of 0.54 Å when the top scored model was considered, compared to 1.79 Å for MAINMAST. DeepMM also generated high-quality models with <1.0 Å C α RMSD for all of the 40 maps, compared with only one such model by MAINMAST. Moreover, DeepMM achieved the high-accuracy models with <0.5 Å RMSD for 22 of 40 maps, whereas MAINMAST failed to generate any model with <0.5 Å RMSD (Fig. 3a). The program CLICK (Nguyen *et al.*, 2011) was also used to evaluate the accuracy of the C α models built by DeepMM and MAINMAST. The corresponding results are shown in Figure 3b. Similar to the results of C α RMSD comparison, DeepMM generated many more high-quality models according to the CLICK RMSD criterion and achieved an average CLICK RMSD of 0.53 Å when the top model was considered, compared to 2.18 Å for MAINMAST. In addition, DeepMM also achieved a significantly higher structure overlap than MAINMAST (Fig. 3c). Except for two top scored models with 99.75% and 99.44% structure overlap, the rest 38 top models generated by DeepMM all have a 100% structure overlap. On average, DeepMM obtained a high structure overlap of 99.98%, compare to 81.88% for MAINMAST. Figure 3 also reveals that DeepMM generated consistently high-accuracy models for all the proteins of different lengths, whereas MAINMAST tended to perform worse with the increasing number of residues in the protein, suggesting the higher robustness of DeepMM than MAINMAST.

3.2 Model reconstruction for experimental EM maps

Our DeepMM method was further tested on the benchmark of 30 experimental density maps at 2.6–4.8 Å resolution. For each of the 30 experimental density maps, DeepMM built 4320 protein C α models, which were then ranked by their alignment scores.

Figure 4a shows a comparison of the C α RMSDs for the models built by DeepMM and MAINMAST. The corresponding data are provided in Supplementary Table S3. It can be seen from the figure that DeepMM generated significantly more accurate models than MAINMAST. On average, DeepMM obtained a C α RMSD of 10.7 Å for the top scored models, which is much better than 22.4 Å by MAINMAST. Moreover, DeepMM predicted a model of <10 Å for 18 out of 30 top scored models, of which 14 models are within 5.0 Å C α RMSD. In contrast, only seven and four models are within 10.0 Å and 5.0 Å for MAINMAST, respectively. Figure 4b shows a comparison of the results for the models predicted by DeepMM and RosettaES. It can be seen from the figure that DeepMM performed much better and generated many more accurate models than RosettaES. Compared to 18 models within 10 Å RMSD by DeepMM, only 6 models were predicted within 10.0 Å RMSD by RosettaES for the top predictions. On average, Rosetta obtained an average C α RMSD of 27.0 Å, which is much higher than 10.7 Å for DeepMM.

Further examination of the predicted results also reveals that the model accuracy depends more on the quality than on the resolution of a map. Namely, compared to maps with relatively higher resolution but lower quality like EMD-3246A/B (2.8 Å) and EMD-5495 (3.5 Å), maps with relatively lower resolution but higher quality like EMD-2867 (4.3 Å) and EMD-3073 (4.1 Å) are more likely to be successful in reconstructing a correct model (Supplementary Table S3). This phenomenon can be attributed to the fact that resolution is a global estimation and resolvability is not necessarily uniform throughout the whole map (Pintilie *et al.*, 2020).

Figure 5 gives two examples of successfully reconstructed structures by DeepMM. One example, EMD-2867, which is a

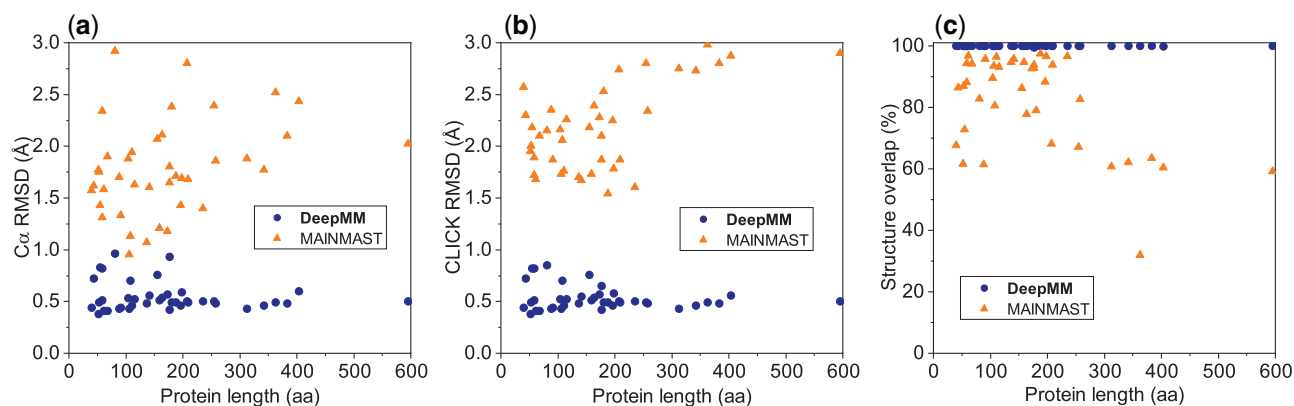


Fig. 3. Comparison of the results by DeepMM and MAINMAST for the protein chains with different lengths on the test set of 40 simulated maps. (a) The C α RMSDs of the top predicted models. (b) The RMSDs of matched C α atoms within 3.5 Å by the structure alignment tool CLICK. (c) The structure overlap calculated by CLICK, which is defined as the fraction of matched C α atoms

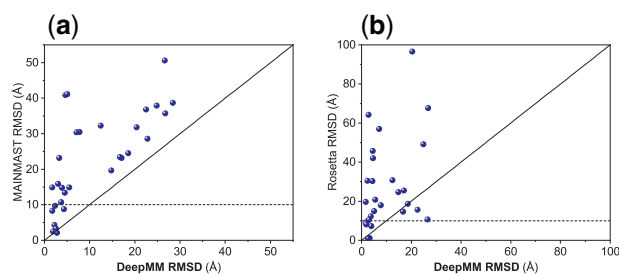


Fig. 4. Comparison of the top models for DeepMM and two other approaches on the test set of 30 experimental maps. The solid line in the figure is the plot of $y=x$, and the dashed line stands for $y=10$. (a) Comparison of the models by DeepMM and MAINMAST in terms of C α RMSD. (b) Comparison of the models by DeepMM and RosettaES in terms of C α RMSD

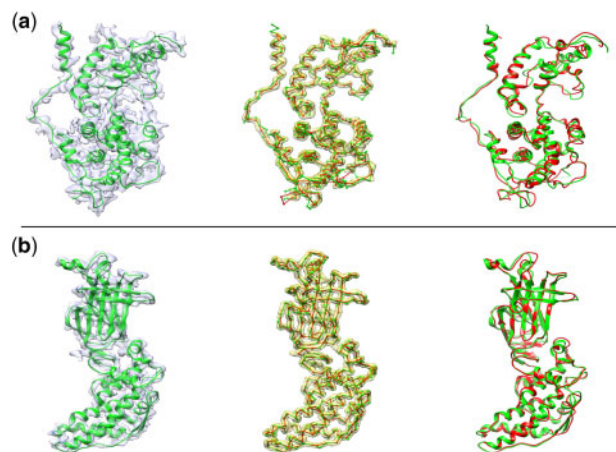


Fig. 5. Examples of the models generated by DeepMM for experimental EM maps. The EM density map (transparent grey) and its associated deposited protein structure (green) are displayed on the left side. The C α chains of the DeepMM model (red) and the deposited structure (green) are shown in ball-and-stick format on the predicted main-chain probability map (transparent yellow) in the middle. The full-atom structure generated by DeepMM (red) and the deposited protein structure (green) are displayed on the right side. (a) The nucleoprotein at 4.3 Å map resolution (EMD-2867). The top-ranked model by DeepMM has a C α RMSD of 3.1 Å. (b) The bovine rotavirus VP6 at 2.6 Å map resolution (EMD-6272). The top model by DeepMM has a C α RMSD of 1.7 Å

nucleoprotein at 4.3 Å resolution, was successfully reconstructed by DeepMM, as shown in Figure 5a. It can be seen from the figure that the predicted main-chain by DeepMM overlaps well with that of the deposited structure. Accordingly, the predicted model shows an

atomic accuracy with a C α RMSD of 3.1 Å. Figure 5b shows the results of another example, EMD-6272, which is the bovine rotavirus VP6 at 2.6 Å resolution. Because of its high resolution, DeepMM predicted a very high accurate model with a small C α RMSD of 1.7 Å. Correspondingly, the constructed full-atom model by DeepMM shows an excellent overlap with the deposited structure (Fig. 5b).

3.3 Evaluation of DeepMM on the EMDB-wide dataset

To investigate the accuracy and general applicability of our DeepMM method, we have further evaluated the performance of DeepMM on a large test set of EMDB-wide experimental maps. This large test set consists of 2931 diverse EM maps with 2.6–4.9 Å resolutions from the EMDB that have associated structures in the PDB (see Section 2). For each of the 2931 test cases, our DeepMM method was conducted to reconstruct structures using four combinations of parameters, yielding 640 models for each case. Figure 6 shows a summary of the results predicted by DeepMM. The corresponding data are provided in Supplementary Table S4. Two metrics, RMSD and TM-score, were used to evaluate the overall accuracy of predicted models. On average, DeepMM achieved a C α RMSD of 9.8 Å for the top prediction and 8.4 Å for the top 10 predictions on this test set of 2931 maps. The corresponding average TM-scores are 0.648 and 0.694 for top 1 and top 10 predictions, suggesting the high accuracy of our DeepMM approach.

Figure 6a shows the percentage of the predicted models at different C α RMSD cutoffs. It can be seen from the figure that 53.6% of the top models built by DeepMM are within 10 Å C α RMSD. For the top 10 scored predictions, 59.9% of the cases have an RMSD of <10 Å. The percentage of the models with different TM-score cutoffs is shown in Figure 6b. It can be seen from the figure that 65.6% of the top models built by DeepMM have a TM-score of >0.5. When the top 10 models were considered, the corresponding percentage increased to 73.6%. Comparing the results in Figure 6a and b also reveals that the percentages for TM-score are significantly higher than those for C α -RMSD, suggesting that the models built by DeepMM still share the same fold with deposited structure even if they have a large C α RMSD.

Figure 6c shows the percentage of correctly predicted top models (i.e. within 10 Å C α RMSD) at different resolutions. For EM maps at 2.5–3.0 Å resolution, DeepMM achieved an excellent performance in successfully reconstructing a correct model, and achieved a success rate of 95.3% and 96.2% for the top 1 and 10 scored models, respectively. The performance of DeepMM decreased with the decreasing map resolution. Specifically, for the EM maps with a resolution of 3.0–3.5 Å, 3.5–4.0 Å and 4.0–4.5 Å, DeepMM obtained a success rate of 82.5%/87.3%, 53.2%/62.1% and 22.9%/29.8% for the top 1/10 predictions, respectively. For EM maps with a resolution of 4.5 Å or worse, it is challenging for DeepMM to build correct models. On average, for the maps at 3–5 Å resolution,

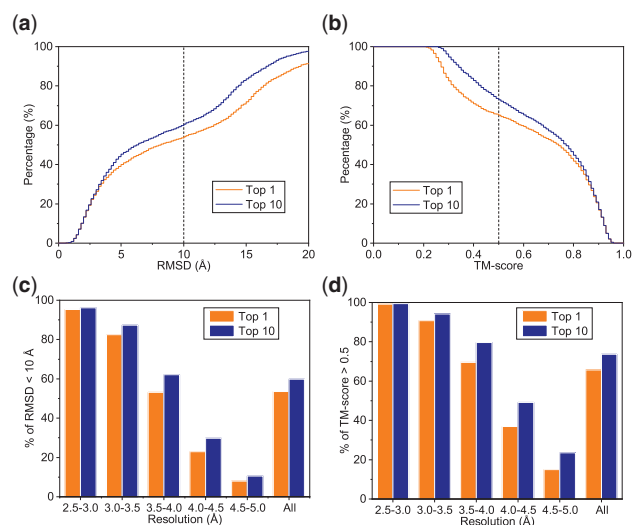


Fig. 6. Test results of DeepMM on the 2931 experimental test cases. (a) The percentage of the top scored models at different C α RMSD cutoffs. (b) The percentage of the top scored models at different TM-score cutoffs. (c) The percentages of top scored models within 10 Å RMSD in different map resolution ranges. (d) The percentages of the top scored models with a TM-score above 0.5 in different map resolution ranges

DeepMM gave a success rate of 50.3% and 57.0% in reconstructing a correct model within 10 Å C α -RMSD for the top 1 and 10 predictions, respectively. Figure 6d shows the percentage of correctly predicted top models using the criterion of TM-score > 0.5 in different resolution ranges. Similar trends in Figure 6c can be observed in Figure 6d. Specifically, for the maps with a resolution of 2.5–3.0 Å, 3.0–3.5 Å, 3.5–4.0 Å, 4.0–4.5 Å and 4.5–5.0 Å, DeepMM achieved correct models with a TM-score of > 0.5 for 99.1%/99.5%, 90.7%/94.2%, 69.5%/79.6%, 36.9%/49.1% and 15.0%/23.5% of the test cases when the top 1/10 predictions were considered, respectively. On average, for the maps at 3–5 Å resolution, DeepMM obtained a success rate of 63.0% and 71.6% in building a model with TM-score > 0.5 for the top 1 and 10 predictions, respectively.

Next, DeepMM was compared with Phenix on this test set, where the Phenix models were generated using the *phenix.map_to_model* tool (Terwilliger *et al.*, 2020) in the latest Phenix package (version 1.19.2-4158). Two metrics calculated by *phenix.chain-comparison* were used to evaluate the accuracy of a model. One is the fraction of the CA atoms in one model matching the CA atoms in another model within 3.0 Å regardless of their residue names (i.e. coverage or residue match). The other is the percentage of the sequence in the target structure reproduced by the query model (i.e. specificity of sequence match). It should be mentioned that our sequence match is conducted using 20 types of amino acids. A model with a high percentage of residue match may have a very low percentage of sequence match because of mismatching of residue types. Figure 7a and b shows the percentages of protein residues and the sequence reproduced by DeepMM and Phenix at different resolutions. Figure 7c and d gives the histograms of corresponding average values at different resolutions. It can be seen from the figure that DeepMM achieved a significantly better performance than Phenix in both residue match and sequence match, especially for those maps at low resolutions. For the maps at resolutions better than 3.0 Å, 94.2% of protein residues in the deposited structures were reproduced by our DeepMM method, compared to 85.2% by Phenix. The corresponding average sequence match is 78.0% for our DeepMM approach, which is much higher than 62.0% for Phenix. For the maps at 3–5 Å resolution, the average residue match for DeepMM is 80.7%, compared with 68.2% for Phenix. The corresponding average sequence match is 38.1% for DeepMM, which is much higher than 20.8% for Phenix. Given that the prediction of sequence match is much more challenging than that of residue match,

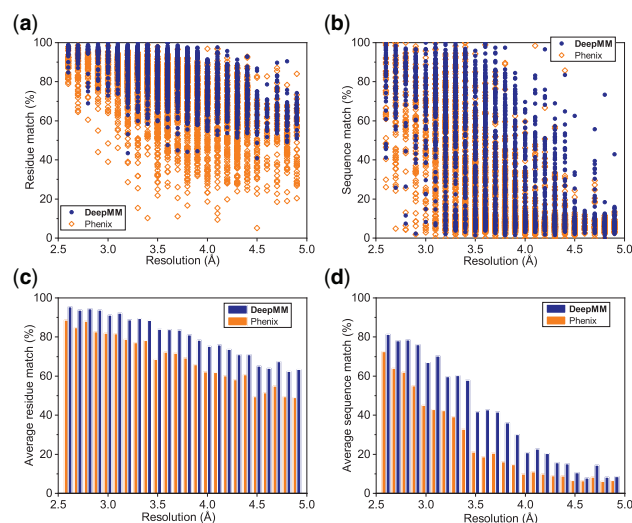


Fig. 7. Comparison of the models by DeepMM and Phenix on the large test set of 2931 experimental maps at different resolutions. The results for Phenix are colored in orange, and those for DeepMM are colored in royal blue. (a) Percentages of the protein residues in the deposited structures reproduced by DeepMM and Phenix. (b) Percentages of the sequence of the deposited structure reproduced by DeepMM and Phenix. (c) Average percentage of residue match by DeepMM and Phenix. (d) Average percentage of sequence match by DeepMM and Phenix

the much better performance of DeepMM than Phenix in sequence match demonstrated the accuracy of the model built by DeepMM.

It is worth mentioning that DeepMM can build fully connected, full-length protein models, whereas Phenix is designed to build initial models of structure fragments. Figure 8 shows the protein models built by DeepMM and Phenix for one example, Chain A of 6DW1, part of a GABAA receptor at 3.1 Å resolution. The deposited structure with its associated EM density map (EMD-8923) is displayed in Figure 8a. Figure 8b and c shows the Phenix model and its superimposition with the deposited structure, respectively. It can be seen from the figures that the model built by Phenix consists of multiple fragments without showing any secondary structures, as expected. The predicted model by Phenix for this map had a residue match of 85.7%, but gave a low sequence match of 44.4%. Therefore, although Phenix recovered most parts of the target protein structure from the EM density map, it could not assign correct residue types for the modeled fragments (Fig. 8c). In contrast, DeepMM built an excellent structure for this map, with a near-perfect residue match of 97.1% and a high sequence match of 86.8%. Therefore, the model predicted by DeepMM reproduced most of the secondary structures and had an almost identical chain trace to the deposited structure (Fig. 8d). The corresponding amino acid names were also assigned correctly by our DeepMM approach (Fig. 8e).

3.4 Quality evaluation of the DeepMM models

One of the most important issues for structure modeling of EM maps is the quality of predicted models (Lawson *et al.*, 2021). In the above sections, the performance of DeepMM has been evaluated by comparing predicted models with deposited structures. Nevertheless, the deposited structures may not be necessarily correct. Namely, some deposited structures are not optimally fitted into the EM density map. As such, we have examined the correlation between EM maps and modeled structures using the *phenix.map_model_cc* (Afonine *et al.*, 2018b) tool. *phenix.map_model_cc* calculates the correlation coefficient (CC) between the experimental map and a model map produced on the same grid. Three metrics were calculated for a given pair of map and model: CCbox, CCmask and CCpeaks. CCbox uses the entire map, CCmask uses the map values inside a mask calculated around the macromolecule and CCpeaks compares the map regions with the highest density values. *phenix-real_space_refine* (Afonine *et al.*, 2018a) was used to optimize the

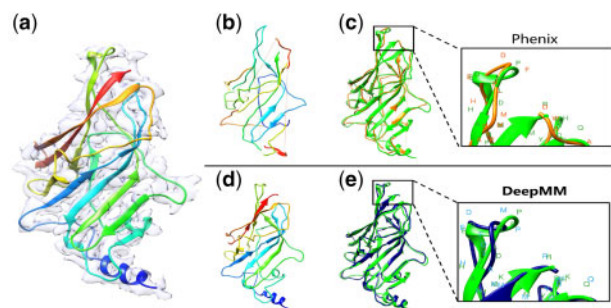


Fig. 8. Protein models reconstructed by DeepMM and Phenix for the Chain A of 6DW1 and its associated EM density map at 3.1 Å resolution (EMD-8923). (a) The deposited structure overlapped with its associated EM density map. (b) The model predicted by Phenix, which has a residue match of 85.7% and a sequence match of 44.4%. (c) The Phenix model (orange) overlapped with the deposited structure (green). The enlarged box on the right side shows that the residue names assigned by Phenix model are different from those of the deposited structure. (d) The model predicted by DeepMM, which has a residue match of 97.1% and a sequence match of 86.8%. (e) The DeepMM model (royal blue) overlapped with the deposited structure (green). The enlarged view of the top region of the protein on the right side shows that the sequence assigned by DeepMM is close to that of the deposited structure.

side-chain conformation of DeepMM models according to the EM map before calculating CC values.

It can be seen from [Supplementary Table S4](#) that most of the models built by DeepMM have lower CC values than the deposited models. This can be understood because most of the deposited models are built carefully either by manual operation or homology modeling, while DeepMM models are automatically built from EM maps without any a priori knowledge. Nevertheless, many DeepMM models can have better fitness than the deposited structure. The first example is the chain K of 5GAN, which is the Snul3 protein of yeast U4/U6.U5 tri-snRNP (Nguyen *et al.*, 2016). As shown in [Figure 9a and b](#), the DeepMM model (colored in red) has better fitness with its associated EM density map (EMD-8012) than the deposited model (colored in green), in terms of CC per residue values. Some residues on the deposited model were not properly placed, resulting in relatively low CC values. As a comparison, despite the fact that the DeepMM model has an RMSD of 2.5 Å from the deposited model, the former fits the map better than the latter on the entire chain. As a whole, the model built by DeepMM achieved the values of 0.6932, 0.6902 and 0.6908 for CCmask, CCpeaks and CCbox, respectively, which are significantly higher than 0.5520, 0.5695 and 0.5626 achieved by the deposited model. Another example is the chain I of 6DWB, a part of the PrGI isolated needle filament (Hu *et al.*, 2018). It can be seen from [Figure 9c and d](#) that a better fitness to the map (EMD-8924) is achieved by the DeepMM model. Interestingly, the DeepMM model seems to be placed at a lower position in the EM map than the deposited model, which is one of the possible reasons for its better fitness. As a whole, the model built by DeepMM achieved higher CCmask, CCpeaks and CCbox values of 0.8200, 0.755 and 0.7420, compared to 0.7688, 0.6510 and 0.6394 by the deposited model.

As shown above, the protein models built by DeepMM can not only have a correct conformation (relative to the deposited models) but also fit well to the EM map. However, DeepMM builds the full-length models of proteins from EM maps using an automatic *de novo* way. The good side of DeepMM is that it can build a full-length protein structure from an EM map. However, it may also introduce uncertainties into the built models for those maps at lower (local) resolutions. Therefore, a metric to estimate the quality of DeepMM models is needed, which is defined as follows:

$$MScore = \frac{\text{Alignment score}}{\text{Length of protein chain}}. \quad (4)$$

The *MScores* have a very good correlation with the TM-scores between the built models and corresponding deposited structures,

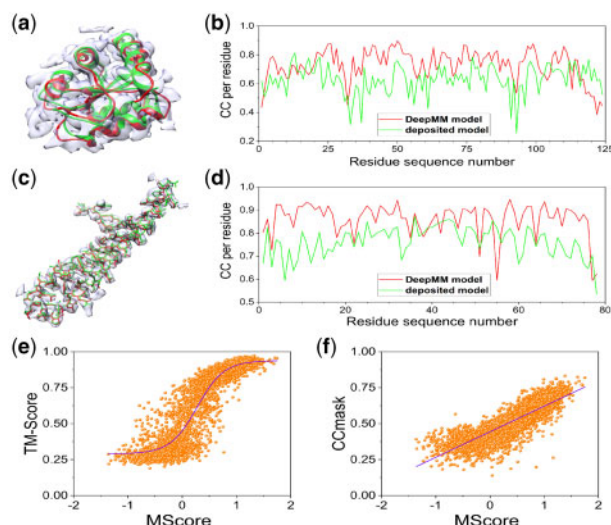


Fig. 9. Quality evaluation of the DeepMM models. (a) The DeepMM model (colored in red) of the chain K of 5GAN and the deposited structure (colored in green) overlapped with the associated EM density map (EMD-8012). (b) The CC value of each residue on the chain K of 5GAN. The residue sequence number starts from 1. (c) The DeepMM model (colored in red) of the chain I of 6DWB and the deposited structure (colored in green) overlapped with the associated EM density map (EMD-8924). (d) The CC value of each residue on the chain I of 6DWB. The residue sequence number starts from 1. (e) The TM-scores of DeepMM models (relative to the deposited structure) with different *MScores*, where the sigmoidal regression curve is in navy blue. *MScore* is calculated as the alignment score divided by the number of residues. (f) The CCmask values of DeepMM models (relative to the EM map) with different *MScores*, where the linear regression line is in navy blue.

giving a coefficient of determination (r^2) value up to 0.835. As shown in [Figure 9e](#), a sigmoidal relationship can be found between TM-scores and *MScores*. The relationship can be fitted with the following formula:

$$\text{TM-score} \approx 0.938 + \frac{-0.648}{1 + e^{4.059MScore - 1.114}}. \quad (5)$$

In terms of the fitness between map and model, *MScores* also have a very good correlation with the CCmask values. As shown in [Figure 9f](#), a linear relationship can be found between the CCmask values and *MScores*, giving a coefficient of determination (r^2) value up to 0.719. The relationship can be approximated by the following formula:

$$\text{CCmask} \approx 0.178MScore + 0.444. \quad (6)$$

The *MScores* for the DeepMM models of the above two examples, the chain K of 5GAN and the chain I of 6DWB, are 1.130 and 1.749, respectively. Based on these formulas, the model for chain K of 5GAN will get a TM-score of ~ 0.918 and a CCmask value of ~ 0.645 , which are close to the real values (see [Supplementary Table S4](#)). The same is true for the chain I of 6DWB with a predicted TM-score of 0.936 and a predicted CCmask value of 0.755. Generally speaking, according to these formulas, a DeepMM model with an *MScore* > 0.5 stands a good chance to be correct and fit the map well. A DeepMM model with a negative *MScore* is not likely to be properly modeled.

3.5 Structural modeling of multi-chain complexes

To further investigate practical applicability of DeepMM, we then applied DeepMM to several multi-chain complex examples. Since DeepMM is only applicable to individual chains, Segger (Pintilie *et al.*, 2010) was adopted to segment the original map into different regions. Afterwards, DeepMM was used to construct each chain on each region segment, which may result in different combinations of regions and chains. The final model built for the complex is the combination with the highest *MScore*. However, we found that it was

challenging for Segger to correctly segment the map automatically. As such, perfect segmentation, which was realized by segmenting the area within 4.0 Å from each chain, was added as the comparison.

The modeling results are shown in Figure 10 and listed in Table 1. It can be seen from Figure 10 that for all of the three examples, DeepMM can correctly assign chains to the perfectly segmented regions and build correct complex models. Relative to the deposited model, DeepMM achieved the C α RMSDs of 3.46 Å, 1.44 Å and 4.65 Å, respectively, for EMD-0593, EMD-0989 and EMD-22123. The TM-scores of the models built with perfect segmentation are 0.964, 0.987 and 0.912 for EMD-0593, EMD-0989 and EMD-22123, respectively. It can be seen from Figure 10 that the map segmentation results of Segger have more or less errors. Segger segmented well for EMD-0593, but confused several small fragments among different regions. The C α RMSD and TM-score between the deposited structure and DeepMM model are 12.8 and 0.742. As indicated by the negative *MScore*, DeepMM failed to build correct model for EMD-0989, which was due to the incorrect

segmentation by Segger. For EMD-22123, Segger roughly succeeded in region of chain J, while confused chain H and chain I. The C α RMSD and TM-score between the deposited structure and DeepMM model for chain J are 10.0 and 0.763. These results suggest that DeepMM can be applied to modeling of multi-chain complexes, though its performance depends on the segmentation quality.

4 Conclusion

In summary, we have developed a semi-automatic *de novo* structure determination method for near-atomic resolution cryo-EM maps using a deep learning-based framework, named as DeepMM. Our DeepMM approach can reconstruct full-length protein structures for EM maps with atomic accuracy. DeepMM was extensively validated on diverse benchmarks and compared with state-of-the-art approaches including RosettaES, MAINMAST and Phenix. DeepMM has also been evaluated on an EMDB-wide large test set of 2931 experimental maps at 2.6–4.9 Å resolution. Overall, DeepMM was able to reconstruct the protein models with TM-score >0.5 for over 60% of the test cases. DeepMM is fast and able to reconstruct an all-atom structure from an EM map within 1 h on a single-GPU machine for an average-length protein chain of 300 amino acids. Given the high computational efficiency and all-atomic accuracy, it is anticipated that DeepMM will serve as an indispensable tool for semi-automatic atomic-accuracy structure determination for near-atomic-resolution cryo-EM maps.

It is also noted that DeepMM bears some similarity to DeepTracer (Pfab *et al.*, 2021) in terms of backbone tracing, as both methods predict the backbone structures from EM maps using deep learning. However, there are significant differences between DeepMM and DeepTracer. DeepMM builds the full-length structures for individual chains using the sequence as a constraint. Therefore, DeepMM is able to give a reliable full-length protein structure for high-resolution EM maps, though it may introduce errors for low-resolution maps or low-resolution regions on the EM maps. In addition, the full-length feature in DeepMM may also result in a higher sequence reproduction due to the additional constraint of sequences. In contrast, DeepTracer only predicts the backbone segments in those regions where the map resolutions are high. It first segments a map into different regions according to the connectivity of backbone confidence map and then connect the identified C α atoms of each region into chains. The segment-and-trace fashion of DeepTracer can effectively improve its performance and robustness on multi-chain complexes, but may lose some precision

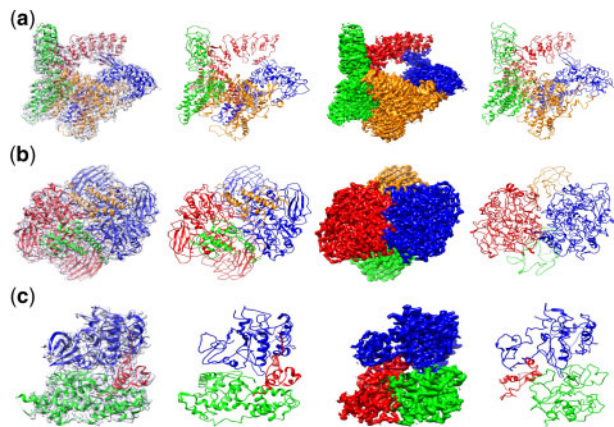


Fig. 10. Examples of the models generated by DeepMM with map segmentation for multi-chain complex structures. Different chains or regions are shown in different colors. From left to right for each row, the first figure is the deposited structure overlapping to the associated EM map, the second figure is the DeepMM model with perfect segmentation, the third figure is the Segger segmentation result and the right-most figure is the DeepMM model with Segger segmentation. (a) EMD-0593, the deposited PDB structure is 6O1N. (b) EMD-0989, the deposited PDB structure is 6LVC. (c) EMD-22123, the deposited PDB structure is 6XBZ

Table 1. The performance of DeepMM on three multi-chain complex examples through the perfect segmentation and Segger segmentation, respectively

EMDB code	Chain ID	Perfect segmentation			Segger segmentation			
		<i>MScore</i>	RMSD (Å)	TM-score	IoU	<i>MScore</i>	RMSD (Å)	TM-score
EMD-0593	A	0.924	2.64	0.950	0.608	0.633	11.63	0.731
	B	0.907	3.11	0.937	0.615	0.645	11.19	0.723
	C	0.977	3.18	0.949	0.612	0.651	11.78	0.659
	D	0.900	4.37	0.934	0.604	0.656	12.55	0.725
	All	0.927	3.46	0.964	All	0.646	12.80	0.742
EMD-0989	A	1.169	1.52	0.982	0.483	–	–	–
	B	1.429	1.64	0.932	0.177	–	–	–
	C	1.186	1.38	0.982	0.481	–	–	–
	D	1.495	0.92	0.958	0.184	–	–	–
	All	1.217	1.44	0.987	All	–	–	–
EMD-22123	H	1.195	1.18	0.889	0.277	–	–	–
	I	1.165	3.32	0.934	0.442	–	–	–
	J	0.644	5.96	0.837	0.764	0.514	9.95	0.763
	All	0.928	4.65	0.912	All	–	–	–

Here, ‘–’ stands for the case in which Segger failed to segment the corresponding region of the protein chain from the EM map, as indicated by the poor values of Intersection over Union (IoU) (Terashi *et al.*, 2020).

on individual chains. Therefore, DeepMM and DeepTracer could be complementary to each other in some way.

Acknowledgements

The authors acknowledge Professor Daisuke Kihara and his students Genki Terashi and Sai Raghavendra Maddhuri Venkata Subramaniya from Purdue University for providing their datasets.

Funding

This work was supported by the National Natural Science Foundation of China [62072199 and 31670724] and the startup grant of Huazhong University of Science and Technology.

Conflict of Interest: none declared.

References

- Adams, P.D. *et al.* (2010) PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.*, **66**, 213–221.
- Afonine, P.V. *et al.* (2018a) Real-space refinement in PHENIX for cryo-EM and crystallography. *Acta Crystallogr. D Struct. Biol.*, **74**, 531–544.
- Afonine, P.V. *et al.* (2018b) New tools for the analysis and validation of cryo-EM maps and atomic models. *Acta Crystallogr. D Struct. Biol.*, **74**, 814–840.
- Alnabati, E. and Kihara, D. (2019) Advances in structure modeling methods for cryo-electron microscopy maps. *Molecules (Basel, Switzerland)*, **25**, 82.
- Baker, M.L. *et al.* (2011) Modeling protein structure at near atomic resolutions with Gorgon. *J. Struct. Biol.*, **174**, 360–373.
- Baker, M.R. *et al.* (2012) Constructing and validating initial Cx models from subnanometer resolution density maps with Pathwalking. *Structure (London, England: 1993)*, **20**, 450–463.
- Berman, H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
- Case, D.A. *et al.* (2005) The Amber biomolecular simulation programs. *J. Comput. Chem.*, **26**, 1668–1688.
- Chen, M. *et al.* (2016) De novo modeling in cryo-EM density maps with Pathwalking. *J. Struct. Biol.*, **196**, 289–298.
- Chen, M. *et al.* (2017) Convolutional neural networks for automated annotation of cellular cryo-electron tomograms. *Nat. Methods*, **14**, 983–985.
- Chen, M. and Baker, M.L. (2018) Automation and assessment of de novo modeling with pathwalking in near atomic resolution cryoEM density maps. *J. Struct. Biol.*, **204**, 555–563.
- Cheng, Y. (2018) Single-particle cryo-EM—How did it get here and where will it go. *Science (New York, N.Y.)*, **361**, 876–880.
- Fox, N.K. *et al.* (2014) SCOPe: structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.*, **42**, D304–D309.
- Frank, J. (2017) Advances in the field of single-particle cryo-electron microscopy over the last decade. *Nat. Protoc.*, **12**, 209–212.
- Frenz, B. *et al.* (2017) RosettaES: a sampling strategy enabling automated interpretation of difficult cryo-EM maps. *Nat. Methods*, **14**, 797–800.
- Fu, L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.
- He, J. and Huang, S.Y. (2021) EMNUSS: a deep learning framework for secondary structure annotation in cryo-EM maps. *Brief. Bioinformatics*, **bbab156**, doi: 10.1093/bib/bbab156.
- Heffernan, R. *et al.* (2016) Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins. *Bioinformatics*, **32**, 843–849.
- Heinig, M. and Frishman, D. (2004) STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Res.*, **32**, W500–W502.
- Ho, C.M. *et al.* (2020) Bottom-up structural proteomics: cryoEM of protein complexes enriched from the cellular milieu. *Nat. Methods*, **17**, 79–85.
- Hu, J. *et al.* (2018) Cryo-EM analysis of the T3S injectisome reveals the structure of the needle and open secretin. *Nat. Commun.*, **9**, 3840.
- Huang, G. *et al.* (2017). Densely connected convolutional networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269.
- Joseph, A.P. *et al.* (2020) Comparing cryo-EM reconstructions and validating atomic model fit using difference maps. *J. Chem. Inf. Model.*, **60**, 2552–2560.
- Kim, D.N. *et al.* (2020) Practical considerations for atomistic structure modeling with cryo-EM maps. *J. Chem. Inf. Model.*, **60**, 2436–2442.
- Lawson, C.L. *et al.* (2021) Cryo-EM model validation recommendations based on outcomes of the 2019 EMDDataResource challenge. *Nat. Methods*, **18**, 156–164.
- Li, X. *et al.* (2013) Electron counting and beam-induced motion correction enable near-atomic-resolution single-particle cryo-EM. *Nat. Methods*, **10**, 584–590.
- Lindert, S. *et al.* (2009) EM-fold: de novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps. *Structure (London, England: 1993)*, **17**, 990–1003.
- Luque, D. and Castón, J.R. (2020) Cryo-electron microscopy for the study of virus assembly. *Nat. Chem. Biol.*, **16**, 231–239.
- Maddhuri Venkata Subramaniya, S.R. *et al.* (2019) Protein secondary structure detection in intermediate-resolution cryo-EM maps using deep learning. *Nat. Methods*, **16**, 911–917.
- Mostosi, P. *et al.* (2020) Haruspex: a neural network for the automatic identification of oligonucleotides and protein secondary structure in cryo-electron microscopy maps. *Angew. Chem.*, **59**, 14788–14795.
- Nguyen, M.N. *et al.* (2011) CLICK—topology-independent comparison of biomolecular 3D structures. *Nucleic Acids Res.*, **39**, W24–W28.
- Nguyen, T. *et al.* (2016) Cryo-EM structure of the yeast U4/U6.U5 tri-snRNP at 3.7 Å resolution. *Nature*, **530**, 298–302.
- Nogales, E. (2016) The development of cryo-EM into a mainstream structural biology technique. *Nat. Methods*, **13**, 24–27.
- Patwardhan, A. (2017) Trends in the Electron Microscopy Data Bank (EMDB). *Acta Crystallogr. D Struct. Biol.*, **73**, 503–508.
- Petrey, D. *et al.* (2003) Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. *Proteins*, **53**, 430–435.
- Pettersen, E.F. *et al.* (2004) UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.*, **25**, 1605–1612.
- Pfaff, J. *et al.* (2021) DeepTracer for fast de novo cryo-EM protein structure modeling and special studies on CoV-related complexes. *Proc. Natl. Acad. Sci. USA*, **118**, e2017525118.
- Pintilie, G.D. *et al.* (2010) Quantitative analysis of cryo-EM density map segmentation by watershed and scale-space filtering, and fitting of structures by alignment to regions. *J. Struct. Biol.*, **170**, 427–438.
- Pintilie, G. *et al.* (2020) Measurement of atom resolvability in cryo-EM maps with Q-scores. *Nat. Methods*, **17**, 328–334.
- Punjani, A. *et al.* (2017) cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination. *Nat. Methods*, **14**, 290–296.
- Raunser, S. (2017) Cryo-EM revolutionizes the structure determination of biomolecules. *Angew. Chem.*, **56**, 16450–16452.
- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint, arXiv:1706.05098*.
- Safdari, H.A. *et al.* (2018) Illuminating GPCR signaling by cryo-EM. *Trends Cell Biol.*, **28**, 591–594.
- Scheres, S.H. (2012) RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J. Struct. Biol.*, **180**, 519–530.
- Si, D. *et al.* (2020) Deep learning to predict protein backbone structure from high-resolution cryo-EM density maps. *Sci. Rep.*, **10**, 4282.
- Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.
- Tang, G. *et al.* (2007) EMAN2: an extensible image processing suite for electron microscopy. *J. Struct. Biol.*, **157**, 38–46.
- Tegunov, D. and Cramer, P. (2019) Real-time cryo-electron microscopy data preprocessing with Warp. *Nat. Methods*, **16**, 1146–1152.
- Terashi, G. and Kihara, D. (2018) De novo main-chain modeling for EM maps using MAINMAST. *Nat. Commun.*, **9**, 1618.
- Terashi, G. *et al.* (2020) MAINMASTseg: automated map segmentation method for cryo-EM density maps with symmetry. *J. Chem. Inf. Model.*, **60**, 2634–2643.
- Terwilliger, T.C. *et al.* (2018) A fully automatic method yielding initial models from high-resolution cryo-electron microscopy maps. *Nat. Methods*, **15**, 905–908.
- Terwilliger, T.C. *et al.* (2020) Cryo-EM map interpretation and protein model-building using iterative map segmentation. *Protein Sci.*, **29**, 87–99.
- Wang, R.Y. *et al.* (2015) De novo protein structure determination from near-atomic-resolution cryo-EM maps. *Nat. Methods*, **12**, 335–338.
- Wen, Z. *et al.* (2020) Topology-independent and global protein structure alignment through an FFT-based algorithm. *Bioinformatics*, **36**, 478–486.

- Xiang,Z. and Honig,B. (2001) Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.*, **311**, 421–430.
- Xie,R. *et al.* (2020) SPREAD: a fully automated toolkit for single-particle cryogenic electron microscopy data 3D reconstruction with image-network-aided orientation assignment. *J. Chem. Inf. Model.*, **60**, 2614–2625.
- Yang,Y.J. *et al.* (2018) Resolution measurement from a single reconstructed cryo-EM density map with multiscale spectral analysis. *J. Chem. Inf. Model.*, **58**, 1303–1311.
- Yin,S. *et al.* (2019) Clustering enhancement of noisy cryo-electron microscopy single-particle images with a network structural similarity metric. *J. Chem. Inf. Model.*, **59**, 1658–1667.
- Zhang,B. *et al.* (2020) A new protocol for atomic-level protein structure modeling and refinement using low-to-medium resolution cryo-EM density maps. *J. Mol. Biol.*, **432**, 5365–5377.
- Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.