Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model

Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang and Jinbo Xu*
Toyota Technological Institute at Chicago
jinboxu@gmail.com

*: corresponding author

The first two authors contribute equally

Abstract

Motivation: Protein contacts contain key information for the understanding of protein structure and function and thus, contact prediction from sequence is an important problem. Recently exciting progress has been made on this problem, but the predicted contacts for proteins without many sequence homologs is still of low quality and not extremely useful for de novo structure prediction.

Method: This paper presents a new deep learning method for contact prediction that predicts contacts by integrating both evolutionary coupling (EC) information and sequence conservation information through an ultra-deep neural network consisting of two deep residual neural networks. The first residual network conducts a series of 1-dimensional convolutional transformation of sequential features; the second residual network conducts a series of 2-dimensional convolutional transformation of pairwise information including output of the first residual network, EC information and pairwise potential. This neural network allows us to model very complex relationship between sequence and contact map as well as long-range interdependency between contacts and thus, obtain high-quality contact prediction.

Results: Our method greatly outperforms existing contact prediction methods and leads to much more accurate contact-assisted protein folding. For example, on the 105 CASP11 test proteins, the L/10 long-range accuracy obtained by our method is 83.3% while that by CCMpred and MetaPSICOV (the CASP11 winner) is 43.4% and 60.2%, respectively. On the 398 membrane proteins, the L/10 long-range accuracy obtained by our method is 79.6% while that by CCMpred and MetaPSICOV is 51.8% and 61.2%, respectively. Ab initio folding guided by our predicted contacts can yield correct folds (i.e., TMscore>0.6) for 224 of the 579 test proteins, while that by MetaPSICOV- and CCMpred-predicted contacts can do so for only 79 and 62 of them, respectively. Further, our contact-assisted models also have much better quality (especially for membrane proteins) than template-based models.

Introduction

De novo protein structure prediction from sequence alone is one of most challenging problems in computational biology. Recent progress has indicated that some correctly-predicted long-range contacts may allow accurate topology-level structure modeling [1] and that direct evolutionary coupling (EC) analysis of multiple sequence alignment (MSA) [2] may reveal some long-range native contacts for proteins with a large number of sequence homologs [3]. Therefore, contact prediction and contact-assisted protein folding has recently gained much attention in the community. However, for many proteins especially those without many sequence homologs, the predicted contacts by the state-of-the-art predictors such as CCMpred [4], PSICOV [5], Evfold [6], MetaPSICOV [7] and CoinDCA [8] are still of low quality and insufficient for accurate contact-assisted protein folding [9]. This motivates us to develop a better contact prediction method, especially for proteins without a large number of sequence homologs. In this paper we say two residues form a contact if they are spatially proximal in the native structure, i.e., the Euclidean distance of their C_β atoms less than 8Å [10].

Existing contact prediction methods roughly belong to two categories: (i) unsupervised evolutionary coupling (EC) analysis that predicts contacts by identifying co-evolved residues in an MSA, such as EVfold [6], PSICOV [5], CCMpred [4], Gremlin [11], and others [12-14]; and (ii) supervised machine learning methods that predict contacts from a variety of evolutionary and co-evolutionary information, e.g., SVMSEQ [15], CMAPpro [10], PconsC2 [16], MetaPSICOV [7], PhyCMAP [17] and CoinDCA-NN [3]. Meanwhile, PconsC2 uses a 5-layer supervised learning architecture [16] and CoinDCA-NN and MetaPSICOV employ a 2-layer neural network [7]. CMAPpro uses a neural network with many more layers, but it is reported that its performance saturates at about 10 layers. Evolutionary coupling (EC) analysis needs a large number of sequence homologs to be effective [16][3]. Some supervised methods such as MetaPSICOV and CoinDCA-NN outperform unsupervised EC analysis on proteins without many sequence homologs, but their performance is still limited by their shallow architectures.

To further improve supervised learning methods for contact prediction, we borrow ideas from very recent breakthrough in computer vision. We have greatly improved contact prediction by developing a brand-new deep learning model called residual neural network [18] for contact prediction. Deep learning is a powerful machine learning technique and has revolutionized image classification [19, 20] and speech recognition [21]. In 2015, ultra-deep residual neural networks [22] demonstrated state-of-the-art performance in several computer vision challenges (similar to CASP) such as image classification [23] and object recognition [24]. If we treat a protein contact map as an image, then protein contact prediction is kind of similar to (but not exactly same as) pixel-level image labeling, so some techniques effective for image labeling may also work for contact prediction. However, it is not straightforward to apply image labeling techniques to contact prediction due to the following difference between contact prediction and image labeling. First, in computer vision community, image-level labeling (i.e., classification of a single image) has been extensively studied, but there are very fewer studies on pixel-level image labeling (i.e., classification of an individual pixel). Second, in many image classification scenarios, image size is usually resized to a fixed value, but we cannot resize a contact map since we need to do prediction for each residue pair (equivalent to an image pixel). Third, contact prediction has much more complex input features (including both sequential and pairwise features) than image labeling. Fourth, the ratio of contacts in a protein is very small (<10%). That is, the number of positive and negative labels in contact prediction is extremely unbalanced.

In this paper we present a very deep residual neural network for contact prediction. Such a network can capture very complex sequence-contact relationship and long-range interdependency between contacts of a protein. We train this deep neural network using a subset of proteins with solved structures and then test its performance on public data including the CASP [25, 26] and CAMEO [27] test proteins as well as membrane proteins. Our experimental results show that our method obtains much better prediction accuracy than existing methods and also result in much more accurate contact-assisted 3D structure modeling. The deep learning method described in this manuscript will also be useful for the prediction of protein-protein and protein-RNA interfacial contacts.

Results

Deep learning model for contact prediction

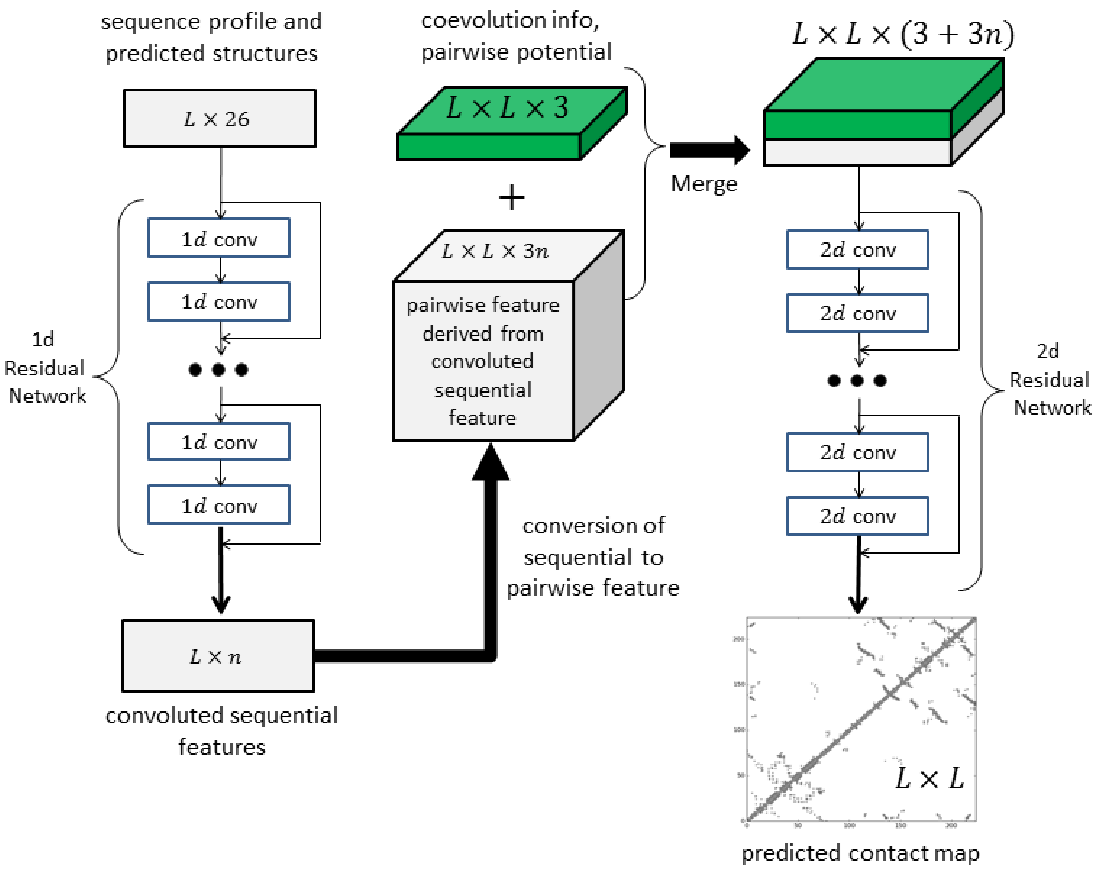
sequence profile and predicted structures
$L \times 26$
1d conv
1d conv
1d conv
1d conv
1d Residual Network
$L \times n$
convoluted sequential features
coevolution info, pairwise potential
$L \times L \times 3$
+
$L \times L \times 3n$
pairwise feature derived from convoluted sequential feature
conversion of sequential to pairwise feature
Merge
$L \times L \times (3 + 3n)$
2d conv
2d conv
2d conv
2d conv
2d Residual Network
$L \times L$
predicted contact map

Figure 1. Illustration of our deep learning model for contact prediction. Meanwhile, L is the sequence length of one protein under prediction.

Figure 1 illustrates our deep neural network model for contact prediction [28]. Different from previous supervised learning approaches for contact prediction that employ only a small number of hidden layers (i.e., a shallow architecture), our deep neural network [22] employs dozens of hidden layers. By using a very deep architecture, our model can automatically learn the complex relationship between sequence information and contacts and also implicitly model the interdependency among contacts and thus, improve contact prediction [16]. Our model consists of two major modules, each being a residual neural network. The first module conducts a series of 1-dimensional (1D) convolutional transformations of sequential features (sequence profile, predicted secondary structure and solvent accessibility). The output of this 1D convolutional network is then converted to a 2-dimensional (2D) matrix by an operation similar to outer product and fed into the 2nd module together with pairwise features (i.e., co-evolution information, pairwise contact and distance potential). The 2nd module is a 2D residual network that conducts a series of 2D convolutional transformations of its input. Finally, the output of the 2D convolutional network is fed into a logistic regression, which predicts the probability of any two residues form a contact. In addition, each convolutional layer is also preceded by a simple nonlinear transformation called rectified linear unit [29]. The output of each 1D convolutional layer has dimension $L\times m$ where L is protein sequence length and m is the number of hidden neurons at one residue. The output of a 2D convolutional layer has dimension $L\times L\times n$ where n is the number of hidden neurons for one residue pair. The number of hidden neurons may vary at each layer.

We tested our method using the 150 Pfam families described in [5], the 105 CASP11 test proteins [30], 398 membrane proteins (Supplementary Table 1) and 76 hard CAMEO test proteins released from 10/17/2015 to 04/09/2016 (Supplementary Table 2). We compare our method with some state-of-the-art methods including PSICOV [5], Evfold [6], CCMpred [4], and MetaPSICOV [7]. The former three predict contacts using direct evolutionary coupling analysis. CCMpred performs slightly better than PSICOV and Evfold. MetaPSICOV [7] is a supervised learning method and performed the best in CASP11 [30]. All the programs are run with parameters set according to their respective papers. We cannot evaluate PconsC2 [16] since we failed to obtain any results from its web server. PconsC2 did not outperform MetaPSICOV in CASP11 [30], so it may suffice to just compare our method with MetaPSICOV.

Overall Performance

We evaluate the accuracy of the top *L/k* (*k*=10, 5, 2, 1) predicted contacts where L is protein sequence length [3]. The prediction accuracy is defined as the percentage of native contacts among the top *L/k* predicted contacts. We also divide contacts into three categories according to the sequence distance of two residues in a contact. That is, a contact is short-, medium- and long-range when the sequence distance falls into [6, 11], [12, 23], and $\geqslant$24, respectively.

Table 1. Contact prediction accuracy on the 150 Pfam families.

Method	**Short**				**Medium**				**Long**			
	L/10	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**
EVfold	0.50	0.40	0.26	0.17	0.64	0.52	0.34	0.22	0.74	0.68	0.53	0.39
PSICOV	0.58	0.43	0.26	0.17	0.65	0.51	0.32	0.20	0.77	0.70	0.52	0.37
CCMpred	0.65	0.50	0.29	0.19	0.73	0.60	0.37	0.23	0.82	0.76	0.62	0.45
MetaPSICOV	0.82	0.70	0.45	0.27	0.83	0.73	0.52	0.33	0.92	0.87	0.74	0.58
Our method	0.93	0.81	0.52	0.30	0.93	0.87	0.62	0.39	0.99	0.97	0.90	0.75

Table 2. Contact prediction accuracy on 105 CASP11 test proteins.

Method	**Short**				**Medium**				**Long**			
	L/10	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**
EVfold	0.25	0.21	0.15	0.12	0.33	0.27	0.19	0.13	0.37	0.33	0.25	0.19
PSICOV	0.29	0.23	0.15	0.12	0.34	0.27	0.18	0.13	0.38	0.33	0.25	0.19
CCMpred	0.35	0.28	0.17	0.12	0.40	0.32	0.21	0.14	0.43	0.39	0.31	0.23
MetaPSICOV	0.69	0.58	0.39	0.25	0.69	0.59	0.42	0.28	0.60	0.54	0.45	0.35
Our method	0.83	0.71	0.46	0.28	0.86	0.77	0.56	0.36	0.84	0.79	0.70	0.56

Table 3. Contact prediction accuracy on 76 CAMEO test proteins.

Method	**Short**				**Medium**				**Long**			
	L/10	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**

EVfold	0.17	0.13	0.11	0.09	0.23	0.19	0.13	0.10	0.25	0.22	0.17	0.13
PSICOV	0.20	0.15	0.11	0.08	0.24	0.19	0.13	0.09	0.25	0.23	0.18	0.13
CCMpred	0.22	0.16	0.11	0.09	0.27	0.22	0.14	0.10	0.30	0.26	0.20	0.15
MetaPSICOV	0.56	0.47	0.31	0.20	0.53	0.45	0.32	0.22	0.47	0.42	0.33	0.25
Our method	0.66	0.56	0.37	0.23	0.70	0.59	0.43	0.28	0.70	0.66	0.56	0.42

Table 4. Contact prediction accuracy on 398 membrane proteins.

Method	**Short**				**Medium**				**Long**			
	L/10	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**	**L/10**	**L/5**	**L/2**	**L**
EVfold	0.16	0.13	0.09	0.07	0.28	0.22	0.13	0.09	0.44	0.37	0.26	0.18
PSICOV	0.22	0.16	0.10	0.07	0.29	0.21	0.13	0.09	0.42	0.34	0.23	0.16
CCMpred	0.27	0.19	0.11	0.08	0.36	0.26	0.15	0.10	0.52	0.45	0.31	0.21
MetaPSICOV	0.45	0.35	0.22	0.14	0.49	0.40	0.27	0.18	0.61	0.55	0.42	0.30
Our method	0.61	0.47	0.28	0.16	0.67	0.55	0.35	0.22	0.80	0.75	0.64	0.49

As shown in Tables 1-4, our method outperforms CCMpred and MetaPSICOV by a very large margin on the 4 test sets regardless of how many top predicted contacts are evaluated and no matter whether the contacts are short-, medium- or long-range. The advantage of our method is the smallest on the 150 Pfam families because many of them have a pretty large number of sequence homologs. In terms of top L long-range contact accuracy, our method exceeds CCMpred and MetaPSICOV by 0.33 and 0.21, respectively, on the CASP11 set. On the CAMEO set, our method exceeds CCMpred and MetaPSICOV by 0.27 and 0.17, respectively. On the membrane protein set, our method exceeds CCMpred and MetaPSICOV by 0.28 and 0.19, respectively. Since the Pfam test set is relatively easy, in the following sections we will focus on the CASP11, CAMEO and membrane protein test sets.

Accuracy with respect to the number of sequence homologs

To examine the performance of our method with respect to the amount of homologous information available for a protein under prediction, we measure the effective number of sequence homologs in MSA by *Meff* [17] (see Method for its formula). A protein with a smaller *Meff* has fewer non-redundant sequence homologs. We divide all the test proteins into 10 bins according to *ln(Meff)* and then calculate the average accuracy of the test proteins in each bin. We merge the first 3 bins for the membrane protein set since they contain a small number of proteins.

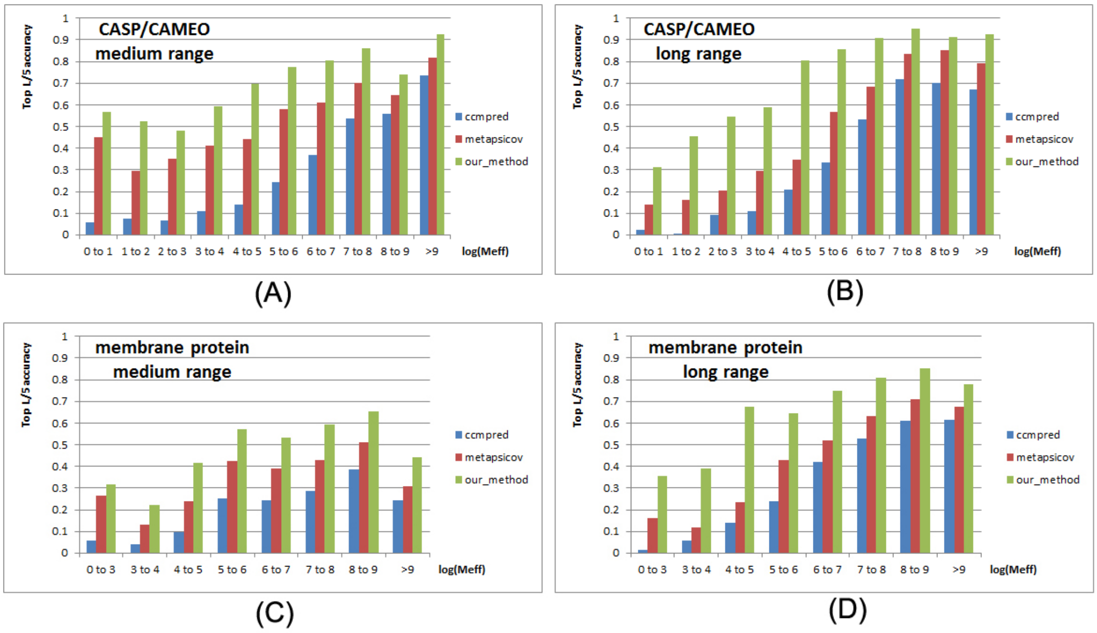

Figure 2. Top L/5 accuracy of our method (green), CCMpred (blue) and MetaPSICOV (red) with respect to the amount of homologous information measured by *ln(Meff)*. The accuracy on the combination of CASP and CAMEO is displayed in (A) medium-range and (B) long-range. The accuracy on the membrane protein set is displayed in (C) medium-range and (D) long-range.

Fig. 2 shows the top L/5 contact prediction accuracy with respect to *ln(Meff)*. Roughly speaking, the prediction accuracy increases with respect to *Meff*, i.e., the amount of homologous information. Our method outperforms both MetaPSICOV and CCMpred no matter how much homologous information is available for the protein under prediction. Our method has an even bigger advantage when *ln(Meff)≤7* (equivalently *Meff<1100*). That is, our method works much better when the protein under prediction does not have a large number of non-redundant sequence homologs. Fig. 2 also shows that no matter how many sequence homologs are available, two supervised learning methods (MetaPSICOV and our method) greatly outperform the unsupervised EC analysis method CCMpred.

Contact-assisted protein folding

One of the important goals of contact prediction is to perform contact-assisted protein folding [9]. To test if our contact prediction can lead to better 3D structure modeling than the others, we build structure models for all the test proteins using the top predicted contacts by our method, CCMpred, and MetaPSICOV, respectively. For each test protein, we feed the top predicted contacts as restraints into the CNS suite [33] to generate 3D models. We measure the quality of a 3D model by TMscore [34] , which ranges from 0 to 1, with 0 indicating the worst and 1 the best, respectively.

As shown in Fig. 3, our predicted contacts can generate much better 3D models than CCMpred and MetaPSICOV. On average, the 3D models generated by our method are better than MetaPSICOV and CCMpred by ~0.12 TMscore unit and ~0.15 unit, respectively. The average TMscore of the top 1 models generated by CCMpred, MetaPSICOV, and our method is 0.30, 0.35, and 0.47, respectively on the CASP/CAMEO dataset. On the membrane protein set, the average TMscore of the top 1 models generated by CCMpred, MetaPSICOV and our method is 0.37, 0.39, and 0.52, respectively. On the CASP/CAMEO dataset, the average TMscore of the best of top 5 models generated by CCMpred, MetaPSICOV, and our method is 0.32, 0.37, and 0.49, respectively. On the membrane protein set, the average TMscore of the best of top 5 models generated by CCMpred, MetaPSICOV, and our method is 0.40, 0.42, and 0.55, respectively. In particular, when the best of top 5 models are considered, our method can result in correct folds (i.e., TMscore>0.6) for 224 of the 579 test proteins, while MetaPSICOV and CCMpred can lead to correct folds for only 79 and 62 proteins, respectively.

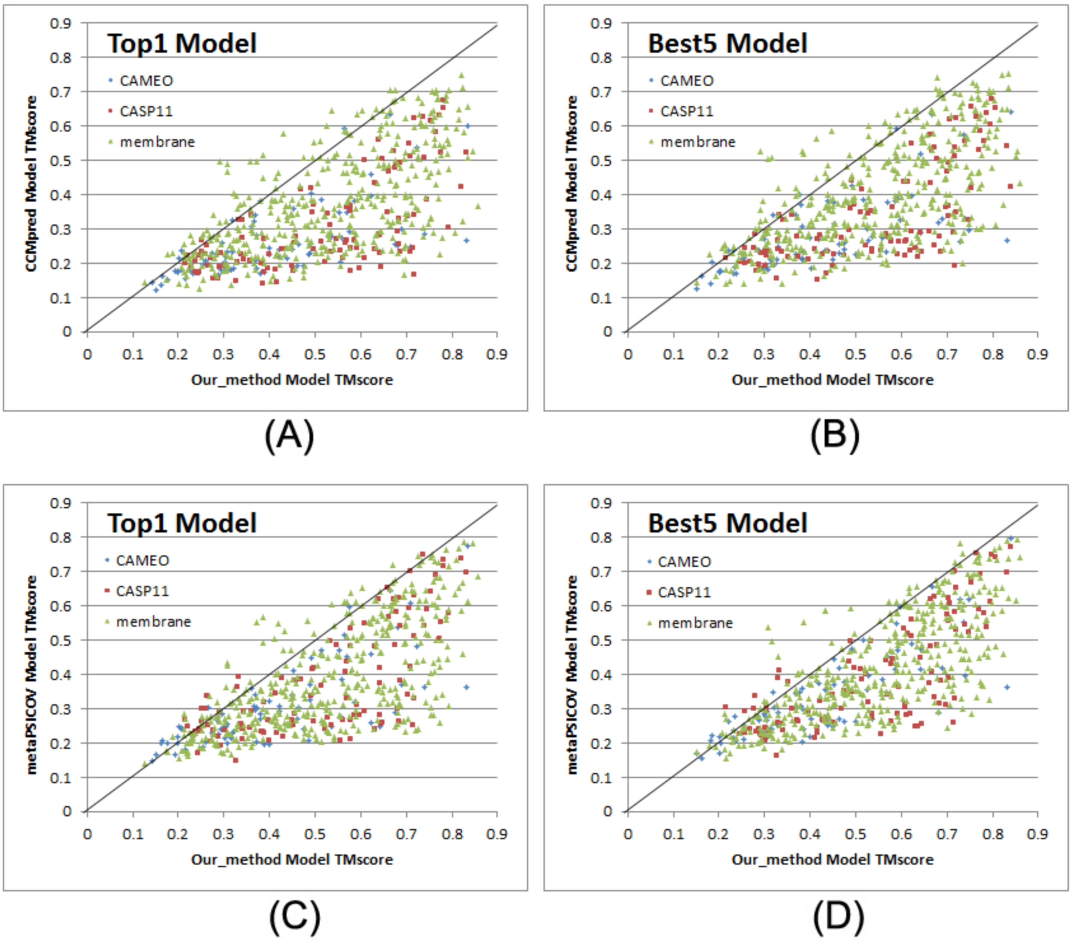

Figure 3. Head-to-head TMscore comparison of contact-assisted models generated by our method, CCMpred and MetaPSICOV on the 105 CASP11 targets (red square), 76 CAMEO targets (blue diamond) and 398 membrane protein targets (green triangle), respectively. **(A)** and **(B)**: comparison of top 1 and the best of top 5 models between our method (X-axis) and CCMpred (Y-axis). **(C)** and **(D)**: comparison of top 1 and the best of top 5 models between our method (X-axis) and MetaPSICOV (Y-axis).

Contact-assisted models vs. template-based models

To generate template-based models (TBMs) for a test protein, we first run HHblits (with the UniProt20_2016 library) to generate an HMM file for the test protein, then run HHsearch with this HMM file to search for the best templates among the 6767 training proteins, and finally run MODELLER to build a TBM from each of the top 5 templates. Fig. 4 shows the head-to-head comparison between our contact-assisted models and the TBMs on these three test sets. In summary, when only the first models are evaluated, our contact-assisted models for the 76 CAMEO test proteins have an average TMscore 0.410 while the TBMs have an average TMscore 0.317. On the 105 CASP11 test proteins, the average TMscore of our contact-assisted models is 0.516 while that of the TBMs is only 0.393. On the 398 membrane proteins, the average TMscore of our contact-assisted models is 0.524 while that of the TBMs is only 0.149. When the best of top 5 models are evaluated, on the 76 CAMEO test proteins, the average TMscore of our contact-assisted models is 0.427 while that of the TBMs is only 0.366. On the 105 CASP11 test proteins, the average TMscore of our contact-assisted models is 0.539 while that of the TBMs is only 0.441. On the 398 membrane proteins, the average TMscore of our contact-assisted models is 0.545 while that of the TBMs is only 0.187. These results indicate that when a query protein has no close templates, our contact-assisted model may have much better quality than TBM. These results imply that our deep learning model does not predict contacts by simply copying contacts from the training proteins. It also implies that contact-assisted modeling shall be very useful for membrane proteins since many of them have no close templates in PDB.

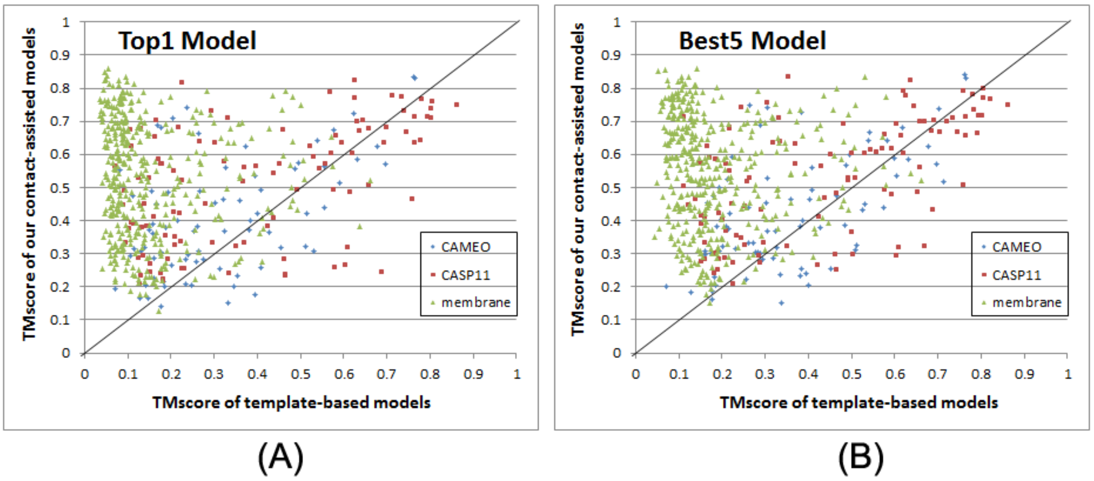

Figure 4. Comparison between our contact-assisted models of the three test sets and their template-based models built from the 6767 training proteins. (A) only the top 1 models are evaluated; (B) the best of top 5 models are evaluated.

Further, our contact-assisted models have TMscore>0.5 for 23 of the 76 CAMEO test proteins while the TBMs have TMscore>0.5 for only 18 of them. Our contact-assisted models have TMscore >0.5 for 62 of the 105 CASP11 test proteins while the TBMs have TMscore>0.5 for only 44 of them. Our contact-assisted models have TMscore>0.5 for 240 of the 398 membrane proteins while the TBMs have TMscore >0.5 for only 10 of them. Our contact-assisted models for membrane proteins are much better than their TBMs because that very few of the 6767 training proteins are good templates for the 398 test membrane proteins. When the 219 test proteins with ≤500 non-redundant sequence homologs are evaluated, the average TMscore of the TBMs is 0.254 while that of our contact-assisted models is 0.43. Among these 219 proteins, our contact-assisted models have TMscore>0.5 for 73 of them while the TBMs have TMscore>0.5 for only 17 of them.

Specific examples

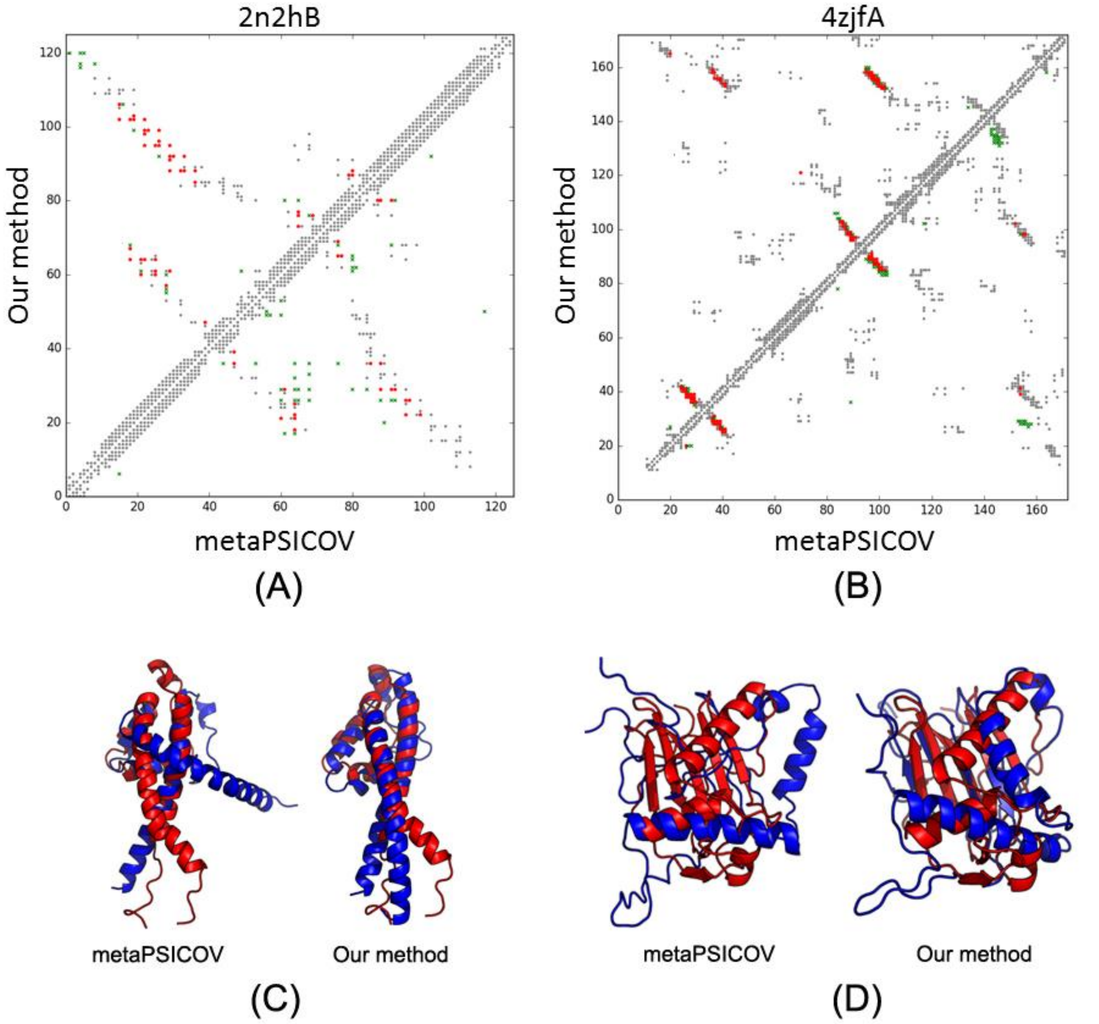

Figure 5. The predicted contact maps and contact-assisted 3D models by MetaPSICOV and our method for two proteins: 2n2hB and 4zjfA. (A) and (B) show the contact maps, in which the upper-left and lower-right triangles display the top L/2 predicted contacts by our method and MetaPSICOV, respectively. Meanwhile, grey dots represent native contacts, while red (green) represents correct (incorrect) predictions. (C) and (D) show the contact-assisted models (blue) and native structures (red).

Here we show the predicted contacts and contact-assisted models of two specific proteins Sin3a (PDB id: 2n2hB) and GP1 (PDB id: 4zjfA). Sin3a is a mainly-alpha protein consisting of two long and paired amphipathic helix [35]. The contact map predicted by our method has L/2 long-range accuracy 0.78 while that by MetaPSICOV has L/2 accuracy 0.35. As shown in the lower-right triangle of Figure 5(A), MetaPSICOV fails to predict the contacts between the paired amphipathic helices. As shown in Figure 5(C), the contact-assisted model built from MetaPSICOV-predicted contacts has TMscore only 0.359. By contrast, the model built from our predicted contacts has TMscore 0.591.

GP1 is the receptor binding domain of Lassa virus. It has a central β-sheet sandwiched (with 5 beta strands numbering from 1, 2, 7, 4, and 3) by the N and C termini on one side and an array of α-helices and loops on the other [36]. The key to form this fold lies in the placement of beta7 between beta3 and beta4, which are shown in the contact map around residue pairs (150, 40) and (150, 100). As shown in the upper-right triangle of Figure 5(B), our method successfully predicts these contacts and has L/2 long-range contact accuracy 0.72. The 3D model built from our contacts has TMscore 0.491, as shown in the right picture of Figure 5(D). On the contrary, MetaPSICOV predicts few contacts in these regions and its L/2 long-range accuracy is only 0.32. The 3D model built from the MetaPSICOV-predicted contacts has TMscore only 0.246, as shown in the left of Figure 5(D).

Conclusion and Discussion

In this paper we have presented a new deep (supervised) learning method for protein contact prediction. Our method distinguishes itself from previous supervised learning methods in that our model employs two deep residual neural networks to model sequence-contact relationship, one for modeling of sequential features (i.e., sequence profile, predicted secondary structure and solvent accessibility) and the other for modeling of pairwise features (e.g., coevolution information). Ultra-deep residual network is the latest breakthrough in computer vision and has demonstrated the best performance in the computer vision challenge tasks (similar to CASP) in 2015. Our method is also unique in that we model a contact map as a single image and then conduct pixel-level labeling on the whole image (by considering the relationship between two pixels), while previous supervised learning methods predict if two residues form a contact or not independent of the other residues. Our experimental results indicate that our method dramatically improves contact prediction, exceeding currently the best methods (e.g., CCMpred, Evfold, PSICOV and MetaPSICOV) by a very large margin. Our contact prediction also leads to much higher quality of contact-assisted structure modeling. Further, our experimental results also show that our contact-assisted models are much better than template-based models when no good templates are available for a protein sequence.

In current implementation, we found out that our model achieves pretty good performance when using around 60-70 convolutional layers. A natural question to ask is can we further improve prediction accuracy by using many more convolutional layers? In computer vision, it has been shown that a 1001-layer residual neural network can yield better accuracy for image-level classification than a 100-layer network (but no result on pixel-level image labeling). Currently we cannot apply more than 100 layers to our model due to insufficient memory of a GPU card (12G). We are going to circumvent the memory limitation by extending our training algorithm so that it can run on multiple GPU cards. Then we will train a model with hundreds of layers to see if we can further improve prediction accuracy or not.

Method

Deep learning model details

Residual network blocks. Our network consists of two residual neural networks, each in turn consisting of some residual blocks concatenated together. Fig. 6 shows an example of a residual block consisting of 2 convolution layers and 2 activation layers. In this figure, X_l and X_{l+1} are the input and output of the block, respectively. The activation layer conducts a simple nonlinear transformation of its input without using any parameters. Here we use the ReLU activation function [29] for such a transformation. Let $f(X_l)$ denote the result of X_l going through the two activation layers and the two convolution layers. Then, X_{l+1} is equal to $X_l + f(X_l)$. That is, X_{l+1} is a combination of X_l and its nonlinear transformation. Since $f(X_l)$ is equal to the difference between X_{l+1} and X_l, f is called residual function and this network called residual network. In the first residual network, X_l and X_{l+1} represent sequential features and have dimension $L \times n_l$ and $L \times n_{l+1}$, respectively, where L is protein sequence length and n_l (n_{l+1}) can be interpreted as the number of features or hidden neurons at each position (i.e., residue). In the 2nd residual network, X_l and X_{l+1} represent pairwise features and have dimension $L \times L \times n_l$ and $L \times L \times n_{l+1}$, respectively, where n_l (n_{l+1}) can be interpreted as the number of features or hidden neurons at one position (i.e., residue pair). Typically, we enforce $n_l \leq n_{l+1}$ since one position at a higher level is supposed to carry more information. When $n_l < n_{l+1}$, in calculating $X_l + f(X_l)$ we shall pad zeros to X_l so that it has the same dimension as X_{l+1} . To speed up training, we also add a batch normalization layer [38] before each activation layer, which normalizes its input to have mean 0 and standard deviation 1. The filter size (i.e., window size) used by a 1D convolution layer is 17 while that used by a 2D convolution layer is 3×3 or 5×5. By stacking many residual blocks together, even if at each convolution layer we use a small window size, our network can model very long-range interdependency between input features and contacts as well as the long-range interdependency between two different residue pairs. We fix the depth (i.e., the number of convolution layers) of the 1D residual network to 6, but vary the depth of the 2D residual network. Our experimental results show that with ~60 hidden neurons at each position and ~60 convolution layers for the 2nd residual network, our model can yield pretty good performance. Note that it has been shown that for image classification a convolutional neural network with a smaller window size but many more layers usually outperforms a network with a larger window size but fewer layers. Further, a 2D convolutional neural network with a smaller window size also has a smaller number of parameters than a network with a larger window size.

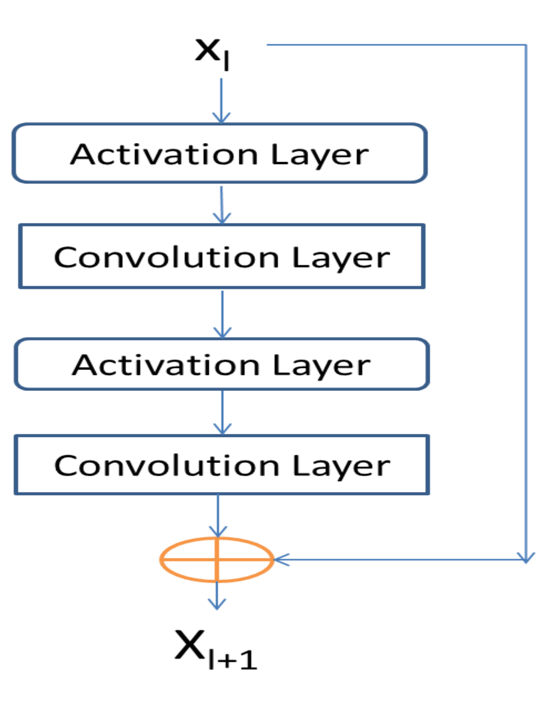

Figure 6. A building block of our residual network with X_l and X_{l+1} being input and output, respectively. Each block consists of two convolution layers and two activation layers.

Our deep learning method for contact prediction is unique in at least two aspects. First, our model employs two multi-layer residual neural networks, which have not been applied to contact prediction before. Residual neural networks can pass both linear and nonlinear information from end to end (i.e., from the initial input to the final output). Second, we do contact prediction on the whole contact map by treating it as an individual image. In contrast, previous supervised learning methods separate the prediction of one residue pair from the others. By doing contact prediction simultaneously for all the residue pairs of one protein sequence, we can easily model the long-range interdependency between two residue pairs and the long-range relationship between one contact and input features.

Conversion of sequential features to pairwise features. We convert the output of the first module of our model (i.e., the 1-d residual neural network) to a 2D representation using an operation similar to outer product. Simply speaking, let v={v_1, v_2, …, v_i, …, v_L} be the final output of the first module where L is protein sequence length and v_i is a feature vector storing the output information for residue *i*. For a pair of residues *i* and *j*, we concatenate v_i , $v_{(i+j)/2}$ and v_j to a single vector and use it as one input feature of this residue pair. The input features for this pair also include mutual information, the EC information calculated by CCMpred and pairwise contact potential [39, 40].

Loss function. We use maximum-likelihood method to train model parameters. That is, we maximize the occurring probability of the native contacts (and non-contacts) of the training proteins. Therefore, the loss function is defined as the negative log-likelihood averaged over all the residue pairs of the training proteins. Since the ratio of contacts among all the residue pairs is very small, to make the training algorithm converge fast, we assign a larger weight to the residue pairs forming a contact. The weight is assigned such that the total weight assigned to contacts is approximately 1/8 of the number of non-contacts in the training set.

Regularization and optimization. To prevent overfitting, we employ L_2-norm regularization to reduce the parameter space. That is, we want to find a set of parameters with a small L_2 norm to minimize the loss function, so the final objective function to be minimized is the sum of loss function and the L_2 norm of the model parameters (multiplied by a regularization factor). We use a stochastic gradient descent algorithm to minimize the objective function. It takes 20-30 epochs (each epoch scans through all the training proteins exactly once) to obtain a very good solution. The whole algorithm is implemented by Theano [41] and mainly runs on a GPU card.

Training and test data

We test our method using some public datasets, including the 150 Pfam families [5], the 105 CASP11 test proteins, 76 recently-released hard CAMEO test proteins (Supplementary Table 1) and 398 membrane proteins (Supplementary Table 2). For the CASP test proteins, we use the official domain definitions, but we do not parse a CAMEO or membrane protein into domains.

Our training set is a subset of PDB25 created in February 2015, in which any two proteins share less than 25% sequence identity. We exclude a protein from the training set if it satisfies one of the following conditions: (i) sequence length smaller than 26 or larger than 700, (ii) resolution worse than 2.5Å, (iii) has domains made up of multiple protein chains, (iv) no DSSP information, and (v) there is inconsistency between its PDB, DSSP and ASTRAL sequences [42]. Finally, we also exclude the proteins sharing >25% sequence identity or having a BLAST E-value <0.1 with any of our test proteins.

In total there are 6767 proteins in our training set, from which we have trained 7 different models. For each model, we randomly sampled ~6000 proteins from the training set to train the model and used the remaining proteins to validate the model and determine the hyper-parameters (i.e., regularization factor). The final model is the average of these 7 models.

Protein features

We use similar but fewer protein features as MetaPSICOV. In particular, the input features include protein sequence profile (i.e., position-specific scoring matrix), predicted 3-state secondary structure and 3-state solvent accessibility, direct co-evolutionary information generated by CCMpred, mutual information and pairwise potential [39, 40]. To derive most features for a protein, we need to generate its MSA (multiple sequence alignment). For a training protein, we run PSI-BLAST (with E-value 0.001 and 3 iterations) to scan through the NR (non-redundant) protein sequence database dated in October 2012 to find its sequence homologs, and then build its MSA and sequence profile and predict other features (i.e., secondary structure and solvent accessibility).

For a test protein, we generate four different MSAs by running HHblits [43] with 3 iterations and E-value set to 0.001 and 1, respectively, to search through the uniprot20 HMM library released in November 2015 and February 2016. From each individual MSA, we derive one sequence profile and employ our in-house tool RaptorX-Property [44] to predict the secondary structure and solvent accessibility accordingly. That is, for each test protein we generate 4 sets of input features and accordingly 4 different contact predictions. Then we average these 4 predictions to obtain the final contact prediction. This averaged contact prediction is about 1-2% better than that predicted from a single set of features (detailed data not shown). Although currently there are quite a few approaches such as Evfold and PSICOV that can generate direct evolutionary coupling information, we only employ CCMpred to do so because it is very fast when running on a GPU card [4].

Programs to compare and evaluation metrics

We compare our method with PSICOV [5], Evfold [6], CCMpred [4], and MetaPSICOV [7]. MetaPSICOV [7] performed the best in CASP11 [30]. All the programs are run with parameters set according to their respective papers. We evaluate the accuracy of the top L/k (k=10, 5, 2, 1) predicted contacts where L is protein sequence length [3]. The prediction accuracy is defined as the percentage of native contacts among the top L/k predicted contacts. We also divide contacts into three groups according to the sequence distance of two residues in a contact. That is, a contact is short-, medium- and long-range when its sequence distance falls into [6, 11], [12, 23], and ⩾24, respectively.

Calculation of Meff

Meff measures the amount of homologous information in an MSA (multiple sequence alignment). It can also be interpreted as the number of non-redundant sequences in an MSA. To calculate the Meff of an MSA, we first calculate the sequence identity between any two protein sequences in the MSA. Let a binary variable S_{ij} denote the similarity between two protein sequences i and j. S_{ij} is equal to 1 if and only if the sequence identity between i and j is at least 70%. For a protein i, we calculate the sum of S_{ij} over all the proteins (including itself) in the MSA and denote it as S_i. Finally, we calculate Meff as the sum of $1/S_i$ over all the protein sequences in this MSA.

3D model construction by contact-assisted folding

We use a similar approach as described in [9] to build the 3D models of a test protein by feeding predicted contacts and secondary structure to the Crystallography & NMR System (CNS) suite [33]. We predict secondary structure using our in-house tool RaptorX-Property [44] and then convert it to distance, angle and h-bond restraints using a script in the Confold package [9]. For each test protein, we choose top L predicted contacts (L is sequence length) no matter whether they are short-, medium- or long-range and then convert them to distance restraints. That is, a pair of residues predicted to form a contact is assumed to have distance between 3.5Å and 8.0 Å. Then, we generate twenty 3D structure models using CNS and select top 5 models by the NOE score yielded by CNS[33]. The NOE score mainly reflects the degree of violation of the model against the input constraints (i.e., predicted secondary structure and contacts). The lower the NOE score, the more likely the model has a higher quality. When CCMpred- and MetaPSICOV-predicted contacts are used to build 3D models, we also use the secondary structure predicted by RaptorX-Property to warrant a fair comparison.

Template-based modeling (TBM) of the test proteins

To generate template-based models (TBMs) for a test protein, we first run HHblits (with the UniProt20_2016 library) to generate an HMM file for the test protein, then run HHsearch with this HMM file to search for the best templates among the 6767 training proteins of our deep learning model, and finally run MODELLER to build a TBM from each of the top 5 templates.

Author contributions

J.X. conceived the project, developed the algorithm and wrote the paper. S.W. did data analysis and wrote the paper. S.S. helped developing the algorithm. R.Z. helped with data analysis. Z.L. helped with the algorithm development.

References

1. Kim, D.E., DiMaio, F., Yu‐Ruei Wang, R., Song, Y., Baker, D.: One contact for every twelve residues allows robust and accurate topology‐level protein structure modeling. Proteins: Structure, Function, and Bioinformatics 82, 208-218 (2014)

2. de Juan, D., Pazos, F., Valencia, A.: Emerging methods in protein co-evolution. Nature Reviews Genetics 14, 249-261 (2013)

3. Ma, J., Wang, S., Wang, Z., Xu, J.: Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. Bioinformatics btv472 (2015)

4. Seemayer, S., Gruber, M., Söding, J.: CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. Bioinformatics 30, 3128-3130 (2014)

5. Jones, D.T., Buchan, D.W., Cozzetto, D., Pontil, M.: PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28, 184-190 (2012)

6. Marks, D.S., Colwell, L.J., Sheridan, R., Hopf, T.A., Pagnani, A., Zecchina, R., Sander, C.: Protein 3D structure computed from evolutionary sequence variation. PloS one 6, e28766 (2011)

7. Jones, D.T., Singh, T., Kosciolek, T., Tetchner, S.: MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics 31, 999-1006 (2015)

8. Wang, S., Li, W., Zhang, R., Liu, S., Xu, J.: CoinFold: a web server for protein contact prediction and contact-assisted protein folding. Nucleic acids research gkw307 (2016)

9. Adhikari, B., Bhattacharya, D., Cao, R., Cheng, J.: CONFOLD: residue‐residue contact‐guided ab initio protein folding. Proteins: Structure, Function, and Bioinformatics 83, 1436-1449 (2015)

10. Di Lena, P., Nagata, K., Baldi, P.: Deep architectures for protein contact map prediction.

Bioinformatics 28, 2449-2457 (2012)

11. Kamisetty, H., Ovchinnikov, S., Baker, D.: Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. Proceedings of the National Academy of Sciences 110, 15674-15679 (2013)

12. Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., Aurell, E.: Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Physical Review E 87, 012707 (2013)

13. Göbel, U., Sander, C., Schneider, R., Valencia, A.: Correlated mutations and residue contacts in proteins. Proteins: Structure, Function, and Bioinformatics 18, 309-317 (1994)

14. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D.S., Sander, C., Zecchina, R., Onuchic, J.N., Hwa, T., Weigt, M.: Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proceedings of the National Academy of Sciences 108, E1293-E1301 (2011)

15. Wu, S., Zhang, Y.: A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. Bioinformatics 24, 924-931 (2008)

16. Skwark, M.J., Raimondi, D., Michel, M., Elofsson, A.: Improved contact predictions using the recognition of protein like contact patterns. PLoS Comput Biol 10, e1003889 (2014)

17. Wang, Z., Xu, J.: Predicting protein contact map using evolutionary and physical constraints by integer programming. Bioinformatics 29, i266-i273 (2013)

18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

19. Krizhevsky, A., Hinton, G.: Convolutional deep belief networks on cifar-10. Unpublished manuscript 40, (2010)

20. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in neural information processing systems, pp. 2377-2385. (Year)

21. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine 29, 82-97 (2012)

22. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521, 436-444 (2015)

23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097-1105. (Year)

24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9. (Year)

25. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A.: Critical assessment of methods of protein structure prediction (CASP)—round x. Proteins: Structure, Function, and Bioinformatics 82, 1-6 (2014)

26. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A.: Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. Proteins: Structure, Function, and Bioinformatics (2016)

27. Haas, J., Roth, S., Arnold, K., Kiefer, F., Schmidt, T., Bordoli, L., Schwede, T.: The Protein Model Portal—a comprehensive resource for protein structure and model information. Database 2013, bat031 (2013)

28. Pinheiro, P.H., Collobert, R.: Recurrent Convolutional Neural Networks for Scene Labeling. In: ICML, pp. 82-90. (Year)

29. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807-814. (Year)

30. Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A., Kryshtafovych, A.: New encouraging developments in contact prediction: Assessment of the CASP11 results. Proteins: Structure, Function, and Bioinformatics (2015)

31. Kozma, D., Simon, I., Tusnady, G.E.: PDBTM: Protein Data Bank of transmembrane proteins after 8 years. Nucleic acids research gks1169 (2012)

32. Wang, S., Peng, J., Ma, J., Xu, J.: Protein secondary structure prediction using deep convolutional neural fields. Scientific reports 6, (2016)

33. Briinger, A.T., Adams, P.D., Clore, G.M., DeLano, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N.S.: Crystallography & NMR system: A new software suite for macromolecular structure determination. Acta Crystallogr D Biol Crystallogr 54, 905-921 (1998)

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

35. Clark, M.D., Marcum, R., Graveline, R., Chan, C.W., Xie, T., Chen, Z., Ding, Y., Zhang, Y., Mondragón, A., David, G.: Structural insights into the assembly of the histone deacetylase-associated Sin3L/Rpd3L corepressor complex. Proceedings of the National Academy of Sciences 112, E3669-E3678 (2015)

36. Cohen-Dvashi, H., Cohen, N., Israeli, H., Diskin, R.: Molecular mechanism for LAMP1 recognition by Lassa Virus. Journal of virology 89, 7584-7592 (2015)

37. Söding, J., Remmert, M., Biegert, A., Lupas, A.N.: HHsenser: exhaustive transitive profile search using HMM–HMM comparison. Nucleic acids research 34, W374-W378 (2006)

38. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of The 32nd International Conference on Machine Learning, pp. 448-456. (Year)

39. Miyazawa, S., Jernigan, R.L.: Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. Macromolecules 18, 534-552 (1985)

40. Betancourt, M.R., Thirumalai, D.: Pair potentials for protein folding: choice of reference states and sensitivity of predicted native states to variations in the interaction schemes. Protein Science 8, 361-369 (1999)

41. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: A CPU and GPU math compiler in Python. In: Proc. 9th Python in Science Conf, pp. 1-7. (Year)

42. Drozdetskiy, A., Cole, C., Procter, J., Barton, G.J.: JPred4: a protein secondary structure prediction server. Nucleic acids research gkv332 (2015)

43. Remmert, M., Biegert, A., Hauser, A., Söding, J.: HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature methods 9, 173-175 (2012)

44. Wang, S., Li, W., Liu, S., Xu, J.: RaptorX-Property: a web server for protein structure property prediction. Nucleic acids research gkw306 (2016)