DenseCPD: Improving the Accuracy of Neural-Network-Based Computational Protein Sequence Design with DenseNet

Yifei Qi[1,2*] and John Z.H. Zhang[1-3]

[1]*Shanghai Engineering Research Center of Molecular Therapeutics & New Drug Development, School of Chemistry and Molecular Engineering, East China Normal University, Shanghai, 200062, China*
[2]*NYU-ECNU Center for Computational Chemistry at NYU Shanghai, Shanghai 200062, China*
[3]*Department of Chemistry, New York University, NY, NY 10003, USA*

[*]Correspondence to: yfqi@chem.ecnu.edu.cn

Abstract

Computational protein design remains a challenging task despite its remarkable success in the past few decades. With the rapid progress of deep-learning techniques and the accumulation of three-dimensional protein structures, using deep neural networks to learn the relationship between protein sequences and structures and then automatically design a protein sequence for a given protein backbone structure is becoming increasingly feasible. In this study, we developed a deep neural network named DenseCPD that considers the three-dimensional density distribution of protein backbone atoms and predicts the probability of 20 natural amino acids for each residue in a protein. The accuracy of DenseCPD was 51.56±0.20% in a 5-fold cross validation on the training set and 54.45% and 50.06% on two independent test sets, which is more than 10% higher than those of previous state-of-the-art methods. Two approaches for using DenseCPD predictions in computational protein design were analyzed. The approach using the cutoff of accumulative probability had a smaller sequence search space compared to that of the approach that simply uses the top-k predictions and therefore enables higher sequence identity in redesigning three proteins with Rosetta. The network and the data sets are available on a web server at http://protein.org.cn/densecpd.html. The results of this study may benefit the further development of computational protein design methods.

Introduction

Computational protein design (CPD) aims to design a protein sequence that folds into a given backbone structure, and has numerous applications in biology and chemistry. Over the past three decades, CPD has been used in a wide range of design tasks, and remarkable successes have been achieved, including the design of novel folds,[1] novel enzymes,[2,3] vaccines,[4-6] antibodies,[7,8] novel protein assemblies,[9-13] ligand/protein-binding proteins,[14-17] and membrane proteins.[18-20] More detailed descriptions of the successful designs are provided in recent reviews.[21-24] Nonetheless, accurate design of a protein structure and function is still a highly challenging task.

Along with applications, the method that drives the design is also evolving.[25] In many methods, a scoring function is used to select the low-energy amino acid sequence that fits into the desired structure.[26,27] The scoring function usually contains physics-based terms, such as van der Waals and electrostatic energy, and knowledge-based terms. For example, the *ref2015* scoring function in Rosetta[28] includes attractive and repulsive Lennard-Jones energy, solvation energy, electrostatic energy, hydrogen-bond energy, and knowledge-based terms, such as Ramachandran preferences and sidechain rotamer preference.[29] Other flavors of recently developed scoring functions include EvoDesign, which combines the evolutionary profile and physical energy,[30] and two statistical potentials ABACUS[31-33] and SEEF.[34]

Recent years have witnessed a rapid increase of deep-learning methods in computational chemistry and biology.[35-37] Particularly in CPD, a number of studies have used deep learning to tackle the sequence design problem. Zhou and coworkers developed the SPIN method to predict the sequence profile of a protein given its backbone structure.[38] SPIN was later improved on a larger dataset with a neural network that consisted of several fully connected layers.[39] We developed a neural network to predict the probability of 20 amino acids for a given residue using the geometric features of residue pairs in the input structure.[40] Using the output of this neural network as residue-type restraints in Rosetta[28] improves the average sequence identity in the redesign of three natural proteins. Chen and coworkers developed the SPROF method, which uses the two-dimensional map of the pairwise residue distance as the input, and reached an accuracy of 39.8%, representing a 5.2% improvement over that of SPIN2.[41] Yu et al. used an interesting approach that translates amino acid sequences into musical compositions and trained a recurrent neural network to generate protein sequences.[42] Greener et al. used a variational autoencoder to generate protein sequences conditioned on protein structures.[43] The autoencoder was used to generate metal-binding sites and design a novel protein that was stable in molecular dynamics simulations. In addition to these two generative models, a number of groups have also developed generative models for various purposes in protein design and engineering.[44-47] Recently, Zhang and coworkers proposed a convolutional neural network for protein sequence design and reached a state-of-the-art accuracy of 42.2% on a test set that shared less than 30% sequence identity with the training set.[48]

In this study, we aim to further improve the accuracy of CPD using deep-learning methods. To this end, we used a sophisticated neural network architecture named DensetNet that adds short paths between layers in a network and achieves a decent accuracy in image classification tasks.[49] The network, named DenseCPD, was adapted to recognize three-dimensional data that were constructed from the distribution of backbone atoms around the target residue to be predicted. The accuracy of DenseCPD exceeds those of previous methods, and strategies for using the predictions in conventional scoring function-based CPD are analyzed.

Results and Discussions

Input data and network architecture

The input of the CPD problem is the backbone structure of a protein. In this study, similar to previous approaches, we treat each residue separately and perform CPD by predicting the probability of 20 amino acids on each residue given its neighbor residues. A nonredundant set of protein structures with a sequence identity of 30% were prepared with PISCES[50] from X-ray structures in the protein data bank (PDB).[51] Five hundred structures from the PISCES output were randomly selected as an independent test set (T500). Additionally, the smaller test set TS50, which contains 50 protein chains and has been used in a number of previous studies,[38,40,48] was also used. The target and neighbor residues are considered by calculating the distribution of the backbone (N, C, C_α, O) and C_β atoms in a three-dimensional grid box, where the densities of different atom types are stored in different boxes or channels (See the Methods for details). This type of data representation avoids the requirement of feature engineering and has shown good performance in combination with convolutional neural networks. Prior to calculating the density distribution, the target residue is translated and oriented to a standard position, where its C_α atom is located at the origin and its C_β atom is on the positive z-axis. To determine the size and location of the grid box, we tested different sizes from 10 to 25 Å and varied the center of the box on the z-axis from -6 to 6 Å. We found that using a box size of 20 Å and center at z=2 Å covered 99.5% of the C_α atoms in residues that form contacts with the target residue, which is a good compromise between data size and coverage (**Table S1**). We therefore used a box size of 20 Å with a grid size of 1 Å to calculate the atom distribution in this study (**Fig. 1**).

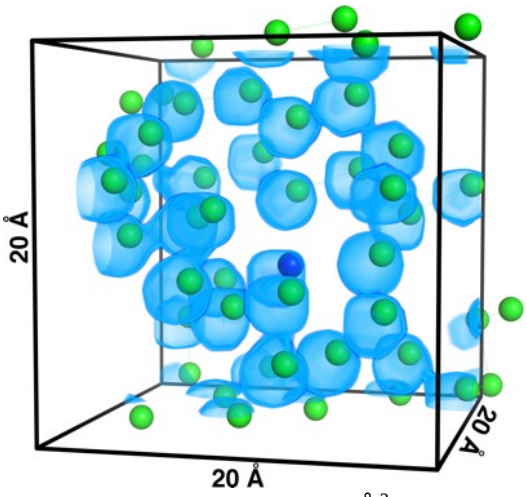

Figure 1. The 20×20×20 $Å^3$ grid box for the C_α atom density that was used as one of the channels in the input of DenseCPD. The densities of the C_α atoms around the target residue are shown as volumes. C_α atoms are shown in spheres, and the C_α atom of the target residue is colored blue. The grid lines are omitted for clarity.

The density data were then learned using DenseCPD, which adopts the network structure of DenseNet and contains a number of dense blocks that are connected by a transition block (**Fig. 2**). Each dense block consists of a number of convolution blocks, and the output of a convolution block is connected to the input of all subsequent convolution blocks in the same dense block. The convolution block contains a bottleneck operation, followed by batch normalization, ReLU

activation, and (3×3×3) convolution. The bottleneck operation, which consists of batch normalization, ReLU activation and (1×1×1) convolution, is introduced to reduce the number of input feature maps and improve the computational efficiency. The transition block is used between the dense block to perform a pooling operation and reduce the number of feature maps. Prior to pooling, a compression operation with a compression rate of 0.5 is included in the transition block. The depth of the DenseNet can be tuned by the number of dense blocks and convolution blocks, and the number of feature maps is determined by the growth rate, which is the size of feature map in the output of a convolution block. In this study, we used 3 dense blocks and 6 convolution blocks in each dense block and tested three growth rates of 15, 25, and 35. The output of DenseCPD is 20 numbers that sum to one and can be interpreted as the probabilities of 20 amino acids at the target residue.

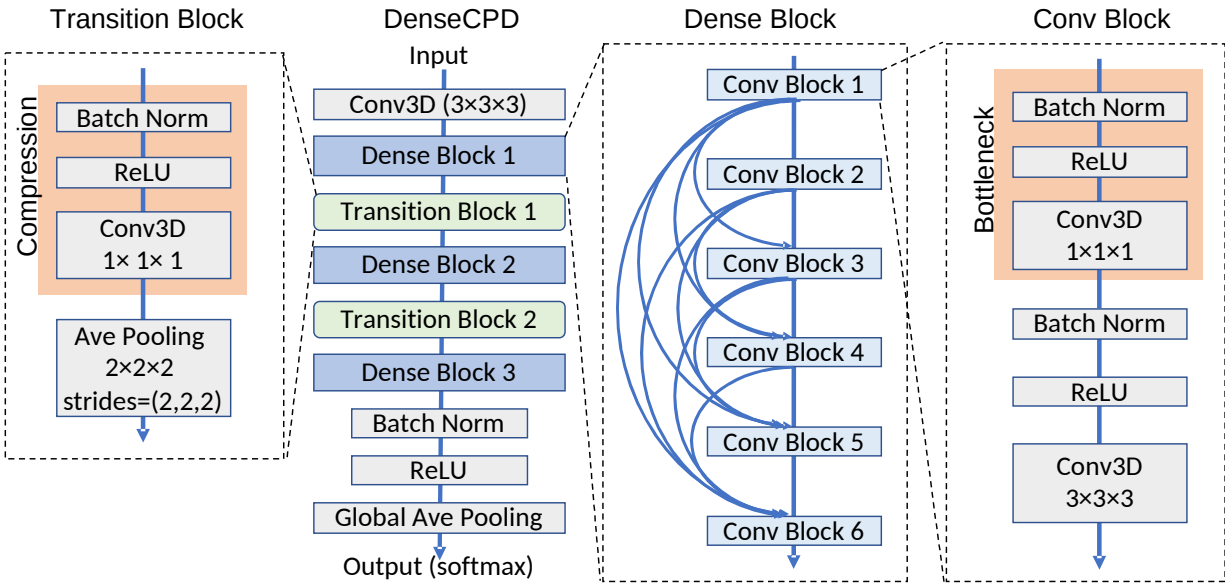

Figure 2. Network structure of DenseCPD.

Accuracy and comparison with other methods

The accuracy of DenseCPD with growth rates of 35, 25, and 15 was 51.56%, 50.54%, and 48.56%, respectively, in a 5-fold cross validation on the training set, suggesting that using a higher growth rate is beneficial (**Table 1**). For the two independent test sets T500 and TS50, the highest accuracy of DenseCPD was 54.45% and 50.06%, respectively, with a growth rate of 35, which was >10% higher than that of previous methods. Comparison of the top-k accuracy of DenseCPD, ProDCoNN, and SPROF suggests that DenseCPD also has the highest accuracy when more than one prediction for each residue is allowed (**Fig. S1**). The following analysis was based on the results with a growth rate of 35.

Table 1. Accuracy of DenseCPD and other methods.

Method	Training set	Test set	
		T500	TS50
DenseCPD (growth rate=35)	51.56±0.20%	54.45%	50.06%
DenseCPD (growth rate=25)	50.54±0.37%	53.28%	49.22%

DenseCPD (growth rate=15)	48.56±0.27%	50.96%	46.61%
ProDCoNN[48]	NA	42.20%[a]	38.71%
SPROF[41]	NA	40.25%	39.16%
SPIN2[39]	NA	36.60%	33.60%
Wang's model[40]	34.00%	36.14%	33.00%
SPIN[38]	NA	30.30%[a]	30.30%

[a]Values were from a different independent test set that contains ~500 protein structures in the original references of the methods.

The distribution of the perstructure accuracy in the T500 set has a peak around 55% (**Fig. 3A**). However, two structures show much lower accuracies of 29.4% and 28.3% with PDB IDs of 2G7O and 3UMH, respectively, both of which are helical proteins. 2G7O is a tetramer under native conditions, but the structure of the X-ray asymmetric unit used for prediction is a monomer (**Fig. S2A**).[52] If the biological assembly is used, the accuracy improves to 45.6%. To further evaluate the effect of using different oligomeric structures, we calculated the accuracy for structures that have different oligomeric states in the asymmetric unit and biological assembly in the T500 set and found that the accuracy for biological assembly is 2% higher, which highlights the importance of using a biologically relevant structure for prediction.

3UMH is the human amyloid precursor protein in complex with zinc ions (**Fig. S2B**).[53] To examine the cause of the low accuracy of 3UMH, we calculated the structural and sequence similarity of 3UMH with the training set using TM-score[54] and BLAST.[55] For comparison, the same similarities were also calculated for two structures 2UVO and 3OEP, which have high accuracies of 70.4% and 69.1%, respectively, in the T500 set. Interestingly, the distribution of the TM-score suggests that there is no significant difference between 3UMH and the two high-accuracy structures (**Fig. 3B**). The BLAST search produces comparable E-values for 3UMH (0.74) and 3OEP (0.18) and finds no similar sequence for 2UVO (**Table S2**), which suggests that the sequence similarity with the training set is not likely the cause of the variable accuracy. By examining the sequences of the three structures, we observed that 3UMH has low Gly and Pro contents, whereas more than 25% of the amino acids in 2UVO and 3OEP are Gly or Pro (**Table S2**). We calculated the correlation between the perstructure accuracy and the GlyPro content and found a positive Pearson correlation of 0.56 (**Fig. 3C**), which is rationalized in the detailed analysis below.

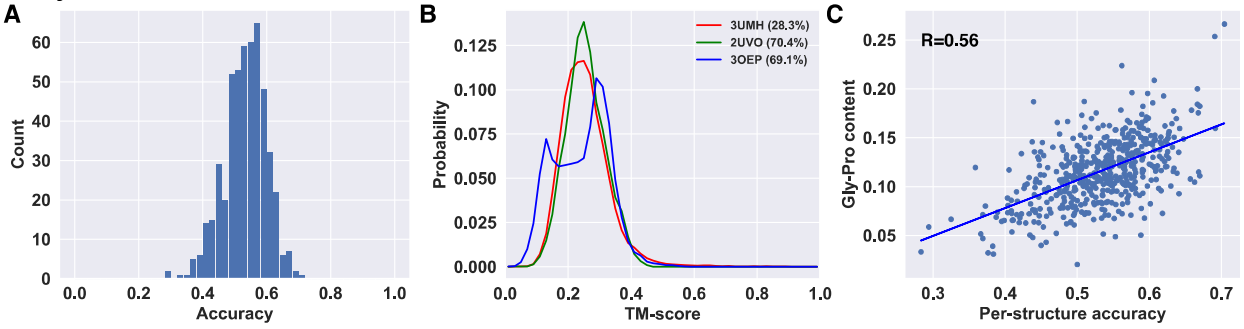

Figure 3. (A) Histogram of the perstructure accuracy of DenseCPD on the T500 test set. (B) Distribution of the TM score in the training set for 3UMH, 2UVO and 3OEP. Numbers in parentheses are the accuracies of DenseCPD for each protein. (C) Correlation between the Gly-Pro content and perstructure accuracy for the T500 set.

We next examined the amino acid-specific accuracy of DenseCPD and compared the accuracy with those of ProDCoNN and SPROF (**Fig. 4**). Two measurements were calculated for each amino acid, namely, recall and precision. Recall is the percentage of wild-type amino acids that are correctly predicted (recovered), and precision is the percentage of the prediction that is correct. Overall, all three methods perform very well for Gly and Pro due to the unique structural features of the two amino acids, which explains the positive correlation between the perstructure accuracy and the GlyPro content. DenseCPD has a comparable accuracy for Gly, Pro and Cys with those of ProDCoNN and SPROF and a better performance for all other amino acids. The improvement is especially remarkable for Gln, His, Met, Trp, and Tyr in terms of recall and Met, Trp, and Gln in terms of precision; some of these are difficult amino acids with lower than average accuracies in previous methods.

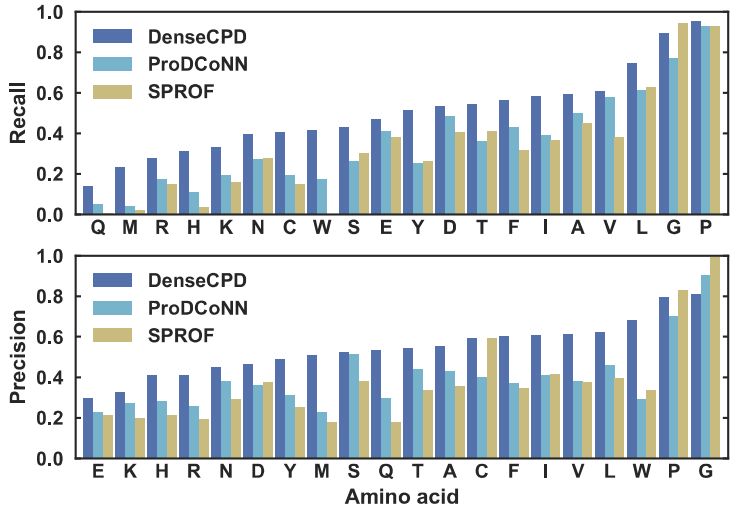

Figure 4. Recall and precision for each amino acid in DenseCPD, ProDCoNN, and SPROF. The values for ProDCoNN were taken from the original reference.

We also calculated the accuracy of residues with different secondary structures and solvent-accessible surface area (SASA, **Fig. 5**). Overall, α-helices and 3-10 helices had lower accuracies, likely due to the low abundance of Gly and Pro in helices. The Naccess program[56] was used to calculate the relative SASA value, which is defined as the absolute SASA divided by the standard SASA of each amino acid. DenseCPD performs better for buried residues, which is a desirable feature for protein design because buried residues are usually more important to the stability of a protein.

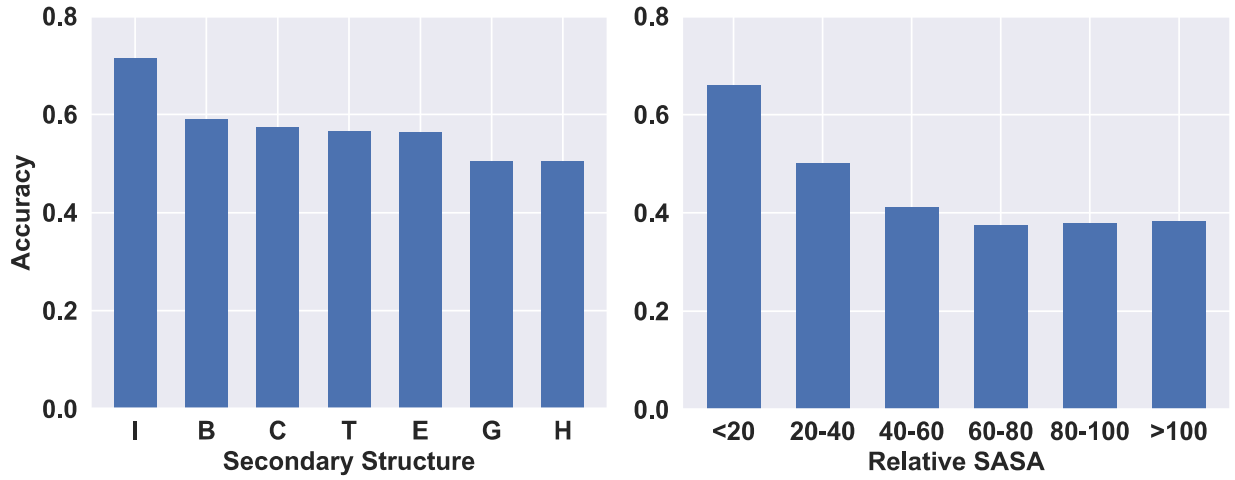

Figure 5. Accuracy of DenseCPD with respect to different secondary structures and the relative SASA. The secondary structure was assigned using Stride[57], and the code is I: π-helix, B: isolated bridge, C: coil, T: turn, E: extended conformation, H: α-helix, and G:3-10 helix.

Approaches to apply DenseCPD prediction in protein design

A straightforward application of DenseCPD to design protein is to take the top predictions as restraints of amino acids for each residue. In general, there are two approaches to use the top predictions. The first approach is simply to use the top-*k* predictions, and the second approach is to use the top predictions whose accumulative probability *p* is above a certain threshold (referred to as acc-*p*). The rationality of the acc-*p* approach is that when the predicted probability is dominated by few amino acids, using only these amino acids instead of the top-*k* predictions would likely reduce the searching space. On the contrary, when several amino acids have equal probabilities, it is better to include all of them in the design. To compare the two approaches, we varied the *k* and *p* values and calculated the sequence coverage, which is defined as the maximal possible sequence identity for a protein given an amino-acid restraint. It is obvious that with the increase of *k* and *p*, the coverage eventually increases to one. However, for the same coverage, the number of candidates for each residue differs for the two approaches (**Fig. 6**). At low coverage, the two methods are nearly identical. When the coverage is between 0.8 and 0.95, the acc-*p* method requires fewer amino acids to reach the same level of coverage. For example, for a sequence coverage of 0.95, the top-*k* method requires ~9 amino acids per residue, whereas acc-*p* requires only ~6, which could reduce the sequence combination by 10^{17} for a protein with 100 residues. Therefore, the acc-*p* method is clearly advantageous over the top-*k* method.

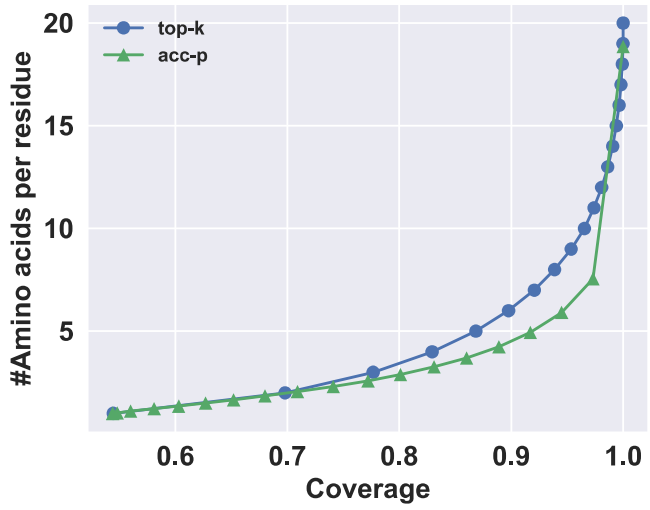

Figure 6. Average number of candidate amino acids under different sequence coverage for the top-*k* and acc-*p* approaches.

We used the top-*k* and acc-*p* approaches to design three proteins, 2IGD, 2B8I, and 1QYS, using the fixed-backbone design and the *ref2015* scoring function[58] in Rosetta.[28] The *k* and *p* values were varied to include different number of candidates for each residue and therefore different sizes of sequence search space for the protein. We compared the sequence coverage and average sequence identity of 1000 Rosetta designs for each protein as a function of the sequence space (**Fig. 7**). With the same sequence coverage, it was possible to substantially reduce the sequence search space with the acc-p approach. For example, to reach 90% sequence coverage in 2IGD, the top-*k* approach needs ~10^{51} sequence combinations, but acc-*p* only needs ~10^{33}. The average sequence identity of the Rosetta designs for the three proteins suggests that using the top-k or acc-*p* restraints improves the similarities to the native proteins compared to the restraint-free design that has the largest search space. Moreover, it is possible to obtain higher sequence identity using the acc-*p* approach.

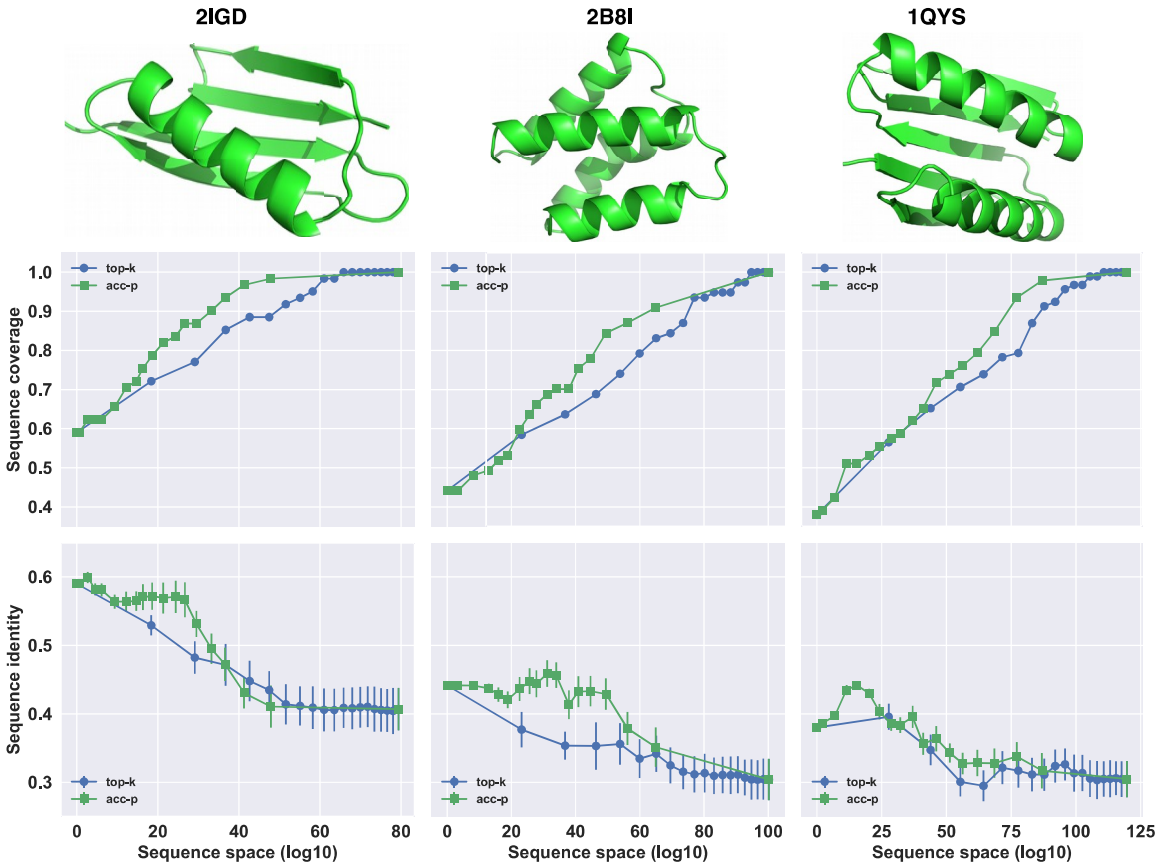

Figure 7. Comparison of the top-*k* and acc-*p* approaches for redesigning three proteins with Rosetta. Sequence coverage is the maximal possible sequence identity under one restraint. The error bars are the standard deviations of the sequence identity from 1000 Rosetta designs.

Conclusions

In this study, we developed DenseCPD to predict the amino acid probability of each residue for a given protein backbone structure. Due to the strong capability of the network architecture, the accuracy of DenseCPD exceeds 50%, which is more than 10% higher than that of previous methods. The accuracy improves for most amino acids, and the improvement was especially remarkable for amino acids that have previously been difficult to predict. We showed that it is important to use the actual biological assembly of a protein as the input of DenseCPD. Moreover, the perstructure accuracy is positively correlated with the contents of Gly and Pro due to the superior performance of DenseCPD on these two amino acids, which originates from the unique structural features of Gly and Pro. As a result of the high accuracy on Gly and Pro, DenseCPD has a lower accuracy for helices, which have lower contents of the two amino acids. Nonetheless, a potentially beneficial feature of DenseCPD is that it performs better for buried residues, which are generally more important to the stability of a protein. We further compared two approaches to utilize the DenseCPD prediction in conventional computational protein design. We found that the approach that uses an accumulative probability cutoff (acc-*p*) reduces the search space under the same sequence coverage compared to the top-*k* approach. Redesigning three proteins using Rosetta shows that with the same sequence search space, the acc-*p* approach

has a higher sequence coverage, and therefore, it is possible to achieve a higher sequence identity. We hope the results of this study will pave the way for further development of CPD methods.

Methods

Datasets and input

The X-ray structures in PDB were first culled with PISCES[50] using a 2-Å resolution cutoff, 0.3 R-value cutoff, and 30% sequence-identity cutoff, which yielded 11227 unique structures. Membrane proteins listed in the OPM database[59] and structures that share more than 30% sequence identity with the TS50 test set were removed. Five hundred structures were randomly chosen as an independent test set (T500) from the remaining entities, and the rest of the structures were randomly separated into 5 sets for cross validation. The final training set contained ~2.6 million residues, and the T500 test set contained 133,803 residues. The PDB IDs of the training and test sets are available on the web server.

For each PDB structure, all atoms except the N, C_α, C, and O atoms were removed, and the C_β atom was built using a C_α–C_β bond length of 1.55 Å, C–C_α–C_β angle of 110.5°, and N–C–C_α–C_β dihedral of 122.55°, similar to that of ProDCoNN. Each target residue along with its neighboring residues were translated and oriented so that the C_α atom of the target residue was located at (0, 0, 0), the C_β atom was on the positive z-axis, and the N atom was on the y=0 and x<0 plane. The coordinates of the atoms were converted to a density distribution on a 20×20×20 $Å^3$ grid box with a grid size of 1 Å and center at (0, 0, 2) Å. The purpose of centering the grid box at (0, 0, 2) instead of (0,0,0) was to include more neighboring residues that have contacts with the target residue. The density of an atom was distributed to its neighboring grids using a Gaussian function $\rho = \exp\left(-d^2/\left(2r^2\right)\right)$, where d is the distance between the atom and grid center, and r is the radius of the atom, which was 0.755, 0.817, 0.817, 0.821, and 0.695 Å for N, C, C_α, C_β, and O atoms. The radius was determined so that the density of each atom was 0.05 at the van der Waals radius from the CHARMM36 force field.[60] The density of each atom type was stored in a separate grid box. Therefore, the data size of one target residue was 20×20×20×5.

Neural-network architecture and training

DenseCPD was constructed using the Keras library (http://keras.io). Bottlenecks were included in the convolution block, and a compression rate of 0.5 was applied in the transition layer. A weight decay of 10^{-4} was used on the convolution layers. Training was performed for 20 epochs using the categorical cross entropy as the loss function and the Adam method for optimization with a learning rate of 0.001 and a batch size of 240. The training samples were weighted as: $W_i = N_{max}/N_i$, where N_{max} is the maximal number of samples of all 20 residue types, and N_i is the number of samples of residue type i. This bias would balance the uneven distribution of residue types in natural proteins and force the neural network to learn more from the residue types that are underrepresented in the training set. The output of the neural network is the probability of 20 amino acids for the target residue. The number of trainable parameters is 0.56M, 1.5M, and 3M for growth rates of 15, 25, and 35.

Data availability

The DenseCPD network and the datasets are available on the web server http://protein.org.cn/densecpd.html

Acknowledgements

This work was supported by the Natural Science Foundation of Shanghai (Grant no. 19ZR1473600), the National Natural Science Foundation of China (Grant no. 31700646, 91753103, 21433004), and the Ministry of Science and Technology of China (Grant no. 2016YFA0501700). We thank the Supercomputer Center of East China Normal University for providing us computer time.

References

1 Kuhlman, B. *et al.* Design of a novel globular protein fold with atomic-level accuracy. *Science* **302**, 1364-1368 (2003).
2 Jiang, L. *et al.* De novo computational design of retro-aldol enzymes. *Science* **319**, 1387-1391 (2008).
3 Rothlisberger, D. *et al.* Kemp elimination catalysts by computational enzyme design. *Nature* **453**, 190-195 (2008).
4 Correia, B. E. *et al.* Computational design of epitope-scaffolds allows induction of antibodies specific for a poorly immunogenic HIV vaccine epitope. *Structure* **18**, 1116-1126 (2010).
5 Marcandalli, J. *et al.* Induction of Potent Neutralizing Antibody Responses by a Designed Protein Nanoparticle Vaccine for Respiratory Syncytial Virus. *Cell* **176**, 1420-1431 e1417 (2019).
6 Correia, B. E. *et al.* Proof of principle for epitope-focused vaccine design. *Nature* **507**, 201-206 (2014).
7 Leaver-Fay, A. *et al.* Computationally Designed Bispecific Antibodies using Negative State Repertoires. *Structure* **24**, 641-651 (2016).
8 Lewis, S. M. *et al.* Generation of bispecific IgG antibodies by structure-based design of an orthogonal Fab interface. *Nat Biotechnol* **32**, 191-198 (2014).
9 Bale, J. B. *et al.* Accurate design of megadalton-scale two-component icosahedral protein complexes. *Science* **353**, 389-394 (2016).
10 Gonen, S., DiMaio, F., Gonen, T. & Baker, D. Design of ordered two-dimensional arrays mediated by noncovalent protein-protein interfaces. *Science* **348**, 1365-1368 (2015).
11 Hsia, Y. *et al.* Design of a hyperstable 60-subunit protein dodecahedron. *Nature* **535**, 136-139 (2016).
12 King, N. P. *et al.* Accurate design of co-assembling multi-component protein nanomaterials. *Nature* **510**, 103-108 (2014).
13 King, N. P. *et al.* Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science* **336**, 1171-1174 (2012).
14 Liu, S. *et al.* Nonnatural protein-protein interaction-pair design by key residues grafting. *Proc Natl Acad Sci U S A* **104**, 5330-5335 (2007).

15 Zhou, L. *et al.* A protein engineered to bind uranyl selectively and with femtomolar affinity. *Nat Chem* **6**, 236-241 (2014).
16 Chevalier, A. *et al.* Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74-79 (2017).
17 Silva, D. A. *et al.* De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186-191 (2019).
18 Joh, N. H. *et al.* De novo design of a transmembrane Zn(2)(+)-transporting four-helix bundle. *Science* **346**, 1520-1524 (2014).
19 Korendovych, I. V. *et al.* De novo design and molecular assembly of a transmembrane diporphyrin-binding protein complex. *J Am Chem Soc* **132**, 15516-15518 (2010).
20 Lu, P. *et al.* Accurate computational design of multipass transmembrane proteins. *Science* **359**, 1042-1046 (2018).
21 Huang, P. S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320-327 (2016).
22 Norn, C. H. & Andre, I. Computational design of protein self-assembly. *Curr Opin Struct Biol* **39**, 39-45 (2016).
23 Yang, W. & Lai, L. Computational design of ligand-binding proteins. *Curr Opin Struct Biol* **45**, 67-73 (2016).
24 Vaissier Welborn, V. & Head-Gordon, T. Computational Design of Synthetic Enzymes. *Chem Rev* **119**, 6613-6630 (2019).
25 Liu, H. & Chen, Q. Computational protein design for given backbone: recent progresses in general method-related aspects. *Curr Opin Struct Biol* **39**, 89-95 (2016).
26 Li, Z., Yang, Y., Zhan, J., Dai, L. & Zhou, Y. Energy functions in de novo protein design: current challenges and future prospects. *Annu Rev Biophys* **42**, 315-335 (2013).
27 Boas, F. E. & Harbury, P. B. Potential energy functions for protein design. *Curr Opin Struct Biol* **17**, 199-204 (2007).
28 Leaver-Fay, A. *et al.* ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* **487**, 545-574 (2011).
29 Alford, R. F. *et al.* The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput* **13**, 3031-3048 (2017).
30 Pearce, R., Huang, X., Setiawan, D. & Zhang, Y. EvoDesign: Designing Protein-Protein Binding Interactions Using Evolutionary Interface Profiles in Conjunction with an Optimized Physical Energy Function. *J Mol Biol* **431**, 2467-2476 (2019).
31 Xiong, P. *et al.* Increasing the Efficiency and Accuracy of the ABACUS Protein Sequence Design Method. *Bioinformatics* (2019).
32 Xiong, P., Chen, Q. & Liu, H. Computational Protein Design Under a Given Backbone Structure with the ABACUS Statistical Energy Function. *Methods Mol Biol* **1529**, 217-226 (2017).
33 Xiong, P. *et al.* Protein design with a comprehensive statistical energy function and boosted by experimental selection for foldability. *Nat Commun* **5**, 5330 (2014).
34 Topham, C. M., Barbe, S. & Andre, I. An Atomistic Statistically Effective Energy Function for Computational Protein Design. *J Chem Theory Comput* **12**, 4146-4168 (2016).
35 LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444 (2015).
36 Angermueller, C., Parnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Mol Syst Biol* **12**, 878 (2016).

37 Goh, G. B., Hodas, N. O. & Vishnu, A. Deep learning for computational chemistry. *J Comput Chem* **38**, 1291-1307 (2017).

38 Li, Z., Yang, Y., Faraggi, E., Zhan, J. & Zhou, Y. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins* **82**, 2565-2573 (2014).

39 O'Connell, J. *et al.* SPIN2: Predicting sequence profiles from protein structures using deep neural networks. *Proteins* **86**, 629-633 (2018).

40 Wang, J., Cao, H., Zhang, J. Z. H. & Qi, Y. Computational Protein Design with Deep Learning Neural Networks. *Sci Rep* **8**, 6349 (2018).

41 Chen, S. *et al.* To Improve Protein Sequence Profile Prediction through Image Captioning on Pairwise Residue Distance Map. *bioRxiv*, 628917 (2019).

42 Yu, C. H., Qin, Z., Martin-Martinez, F. J. & Buehler, M. J. A Self-Consistent Sonification Method to Translate Amino Acid Sequences into Musical Compositions and Application in Protein Design Using Artificial Intelligence. *ACS Nano* **13**, 7471-7482 (2019).

43 Greener, J. G., Moffat, L. & Jones, D. T. Design of metalloproteins and novel protein folds using variational autoencoders. *Sci Rep* **8**, 16189 (2018).

44 Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods* **16**, 1315-1322 (2019).

45 Karimi, M., Zhu, S., Cao, Y. & Shen, Y. De Novo Protein Design for Novel Folds using Guided Conditional Wasserstein Generative Adversarial Networks (gcWGAN). *bioRxiv*, 769919 (2019).

46 Riesselman, A. *et al.* Accelerating Protein Design Using Autoregressive Generative Models. *bioRxiv*, 757252 (2019).

47 Sabban, S. & Markovsky, M. RamaNet: Computational De Novo Protein Design using a Long Short-Term Memory Generative Adversarial Neural Network. *bioRxiv*, 671552 (2019).

48 Zhang, Y. *et al.* ProDCoNN: Protein Design using a Convolutional Neural Network. *Proteins* (2019).

49 Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely Connected Convolutional Networks. *arXiv e-prints*, arXiv:1608.06993 (2016).

50 Wang, G. & Dunbrack, R. L., Jr. PISCES: a protein sequence culling server. *Bioinformatics* **19**, 1589-1591 (2003).

51 Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res* **28**, 235-242 (2000).

52 Lu, J. *et al.* Protonation-mediated structural flexibility in the F conjugation regulatory protein, TraM. *EMBO J* **25**, 2930-2939 (2006).

53 Dahms, S. O. *et al.* Metal binding dictates conformation and function of the amyloid precursor protein (APP) E2 domain. *J Mol Biol* **416**, 438-452 (2012).

54 Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702-710 (2004).

55 Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J Mol Biol* **215**, 403-410 (1990).

56 'NACCESS', Computer Program (Department of Biochemistry and Molecular Biology, University College London., 1993).

57 Frishman, D. & Argos, P. Knowledge-based protein secondary structure assignment. *Proteins* **23**, 566-579 (1995).

58 Park, H. *et al.* Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules. *J Chem Theory Comput* **12**, 6201-6212 (2016).
59 Lomize, M. A., Lomize, A. L., Pogozheva, I. D. & Mosberg, H. I. OPM: orientations of proteins in membranes database. *Bioinformatics* **22**, 623-625 (2006).
60 Huang, J. & MacKerell, A. D., Jr. CHARMM36 all-atom additive protein force field: validation based on comparison to NMR data. *J Comput Chem* **34**, 2135-2145 (2013).