OPUS-Fold3: a gradient-based protein all-atom folding and docking framework on TensorFlow

Gang Xu[1,2,3], Yiqiu Zhang[3], Qinghua Wang[4] & Jianpeng Ma[1,2,3,*]

[1] *Multiscale Research Institute of Complex Systems,*
Fudan University,
Shanghai, 200433, China

[2] *Zhangjiang Fudan International Innovation Center,*
Fudan University,
Shanghai, 201210, China

[3] *Shanghai AI Laboratory,*
Shanghai, 200030, China

[4] *Center for Biomolecular Innovation,*
Harcam Biomedicines,
Shanghai, 200131, China

September 1, 2022

Keywords: protein folding framework, protein docking framework.

[*]Correspondence: jpma@fudan.edu.cn.

Abstract

Protein folding and docking framework is crucial for computational structural biology. It can deliver a corresponding 3D structure using given constrains (e.g. distance and orientation distribution constraints proposed by trRosetta). In this paper, we propose OPUS-Fold3, a gradient-based all-atom folding and docking framework. OPUS-Fold3 is capable of modeling protein backbone and side chains either separately or simultaneously using given constrains. As a docking framework, OPUS-Fold3 is also capable of dealing with the constrains between the receptor and ligand. In addition, if a constrain (or potential function) can be represented as a function of heavy atoms' position, it can be easily introduced into OPUS-Fold3 to further improve the folding and docking accuracy. OPUS-Fold3 is written in Python and TensorFlow2.4, which is user-friendly to any source-code level modification, and can be conveniently incorporated with other TensorFlow-based models.

Introduction

Generalizing a corresponding protein 3D structure with given constrains is an important task in computational structural biology. In the past few decades, many methods have been proposed to tackle this issue [1-4]. In protein structure prediction, the contact map-based methods, such as RaptorX-Contact [5], used Crystallography and NMR System (CNS) [1] to optimize against their predicted binary distance contact constraints to obtain the final 3D prediction. There are also some methods, such as CONFOLD [6] and pyconsFold [7], that used CNS suite [1] as their underlying folding schemes and introduced some other constrains to achieve better results. Recently, depending on the pyRosetta folding scheme [2, 3], the distance and orientation distribution constraints have been proposed in trRosetta [8], which become one of the most common constraints in the field.

In our previous study, we developed OPUS-Fold2 to respectively model the backbone [9] and side chains [10] using the distance and orientation distribution constraints proposed by trRosetta [8]. However, both folding backbone exclusively [9] and folding side chains with fixed backbone [10] have limited usages. Therefore, in OPUS-Fold3, we refine and refactor the code in OPUS-Fold2 to form an all-atom folding scheme so that backbone and side chains can be adjusted simultaneously during the folding process, which is beneficial to the tasks that require all-atom simulation. In addition, OPUS-Fold3 is also capable of dealing with the constrains between the receptor and ligand, therefore it can be used as an underlying docking scheme.

Methods

Datasets

For evaluating the performance on monomer target, we use CAMEO60 [11] that contains 60 monomer hard targets released between January 2020 and July 2020 from the CAMEO website [12]. For evaluating the performance on oligomer target, we construct an oligomer dataset CAMEO75o that contains 75 targets with two peptide chains and < 1000 residues in length from CAMEO-Homo, CAMEO-Hetero, and CAMEO93o [13]. The first peptide chain in the PDB file is defined as "receptor", and the last peptide chain in the PDB file is defined as "ligand".

OPUS-Fold3

OPUS-Fold3 is a gradient-based all-atom folding and docking framework. The folding related variables in OPUS-Fold3 include backbone torsion angles (Φ, Ψ and Ω) and side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4) of all residues. The docking related variables in OPUS-Fold3 include 6 parameters in rotation matrix and 3 parameters in translation matrix. Here, we use the distance and orientation distribution constraints proposed by trRosetta [8]. Other constrains can be easily introduced into OPUS-Fold3 provided they can be represented as functions of heavy atoms' position.

The loss function of trRosetta-style constrains is defined as follows:

$$loss = w_{dist}\frac{1}{N_{cons_{dist}}}\sum_{i\in cons_{dist}} score_{dist}^{i}$$

$$+w_{\omega}\frac{1}{N_{cons_{\omega}}}\sum_{i\in cons_{\omega}} score_{\omega}^{i}$$

$$+w_{\theta}\frac{1}{N_{cons_{\theta}}}\sum_{i\in cons_{\theta}} score_{\theta}^{i}$$

$$+w_{\varphi}\frac{1}{N_{cons_{\varphi}}}\sum_{i\in cons_{\varphi}} score_{\varphi}^{i}$$

Same as the definitions in trRosetta [8], $cons_{dist}$ is the collection of distance (C_β- C_β distance) constraints, in which $P_{4\leq dist<20} \geq 0.05$. $cons_{\omega}$ and $cons_{\theta}$ are the

collections of ω ($C_{\alpha1}$- $C_{\beta1}$- $C_{\beta2}$- $C_{\alpha2}$) and θ (N_1-$C_{\alpha1}$- $C_{\beta1}$- $C_{\beta2}$ and N_2-$C_{\alpha2}$- $C_{\beta2}$- $C_{\beta1}$) constraints, respectively, in which $P_{contact} \geq 0.55$. $cons_{\varphi}$ is the collection of φ ($C_{\alpha1}$- $C_{\beta1}$- $C_{\beta2}$ and $C_{\alpha2}$- $C_{\beta2}$- $C_{\beta1}$) constraints, in which $P_{contact} \geq 0.65$. w_{dist}, w_{ω}, w_{θ} and w_{φ} are the weights of each term, which are set to be 10, 8, 8 and 8, respectively. The distance and orientation distributions are converted to the energy terms by the following equations:

$$score^{i}_{dist} = -lnP^{i} + \ln\left(\left(\frac{d^{i}}{d^{N}}\right)^{\alpha} P^{N}\right)$$

$$score^{i}_{orient} = -lnP^{i} + \ln P^{N}$$

The α is set to be 1.57 [14]. Same as that in trRosetta [8], the reference state for the distance distribution is the probability of the *N*th bin [19.5, 20], and that for the orientation distribution is the probability of the last bin [165°, 180°]. Here, P^{i} refers to the probability of the *i*th bin. d^{i} refers to the distance of the *i*th bin. Cubic spline curves are generated for making the terms differentiable.

In addition, the Ramachandran scoring term [15] is used for the regulation of backbone torsion angles (*Φ* and *Ψ*). We use the radial basis function to make the probabilities differentiable. The Omega scoring term is used for the regulation of backbone torsion angle (*Ω*). The weights of the Ramachandran scoring term and the Omega scoring term are set to be 0.1 and 0.05, respectively.

In OPUS-Fold3, for backbone modeling, the original trRosetta-style constrains in trRosetta [8], the Ramachandran scoring term, and the Omega scoring term are introduced into the loss function. For side-chain modeling, the modified trRosetta-style constrains proposed by OPUS-Rota4 [10] are introduced. Specially, four sets of constrains are included, for each side-chain dihedral angle ($X_1, X_2, X_3,$ and X_4), the corresponding side-chain atoms that are required for its calculation are defined as its pseudo-C_{α} and C_{β}. The detailed pseudo-C_{α} and C_{β} for each side-chain dihedral angle can be found in Supplementary Table S1.

Within a peptide chain, the backbone of each residue is generated one by one depending on the backbone torsion angles (***Φ*, *Ψ*** and *Ω*), and the side chain is then

constructed based on the side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4). In the docking procedure, a rotation matrix that contains 6 parameters and a translation matrix that contains 3 parameters are used to model the relative position between the receptor and ligand. Therefore, the coordinates of all atoms in the ligand will be additionally transformed using the transformation matrixes mentioned above.

OPUS-Fold3 is based on TensorFlow2.4 [16], and the Adam [17] optimizer is used to optimize our loss function with an initial learning rate of 0.5.

Performance Metrics

We use TM-score [18] to measure the accuracy of the predicted backbone. Mean absolute error (MAE) of $X_1, X_2, X_3,$ and X_4 are used to measure the accuracy of the predicted side chains. In addition, ACC is used as the representation of the percentage of correct prediction with a tolerance criterion 20° for all side-chain dihedral angles (from X_1 to X_4).

Results

Backbone Folding

In Table 1, we compare the backbone folding performance of OPUS-Fold3 with that of the pyRosetta folding protocol in trRosetta [8] on CAMEO60 using the identical constrains as the inputs. The results show that OPUS-Fold3 achieves comparable performance to pyRosetta either using the predicted constrains from OPUS-Contact [9] or using the real constrains derived from the corresponding PDB-file. Here, OPUS-Fold3 adopts the predicted backbone torsion angles (Φ and Ψ) from OPUS-TASS2 [9] as its initial state. When using the random values as the initial backbone torsion angles (Φ and Ψ), the performance is slightly decreased (OPUS-Fold3 (random) in Table 1).

Table 1. The TM-score of each method on CAMEO60. "Predicted" denotes the results using the distance and orientation constrains predicted by OPUS-Contact. "PDB" denotes the results using the real distance and orientation constrains derived from the corresponding PDB-file.

	pyRosetta	OPUS-Fold3	OPUS-Fold3 (random)
Predicted	**0.618**	0.612	0.606
PDB	0.983	**0.987**	0.954

As examples, we show the backbone folding processes of OPUS-Fold3 and OPUS-Fold3 (random) using the real constrains derived from the corresponding PDB-file on monomer target 2020-01-04_00000019_1 in Supplementary Figure S1. The results indicate that when using the predicted backbone torsion angles (Φ and Ψ) as the initial state, OPUS-Fold3 achieves better results within less optimization epochs comparing to OPUS-Fold3 (random). For further illustration, some intermediate structures during the backbone folding process of OPUS-Fold3 are shown in Figure 1, and the folding trajectory of OPUS-Fold3 (random) is shown as a movie in Supplementary Video S1.

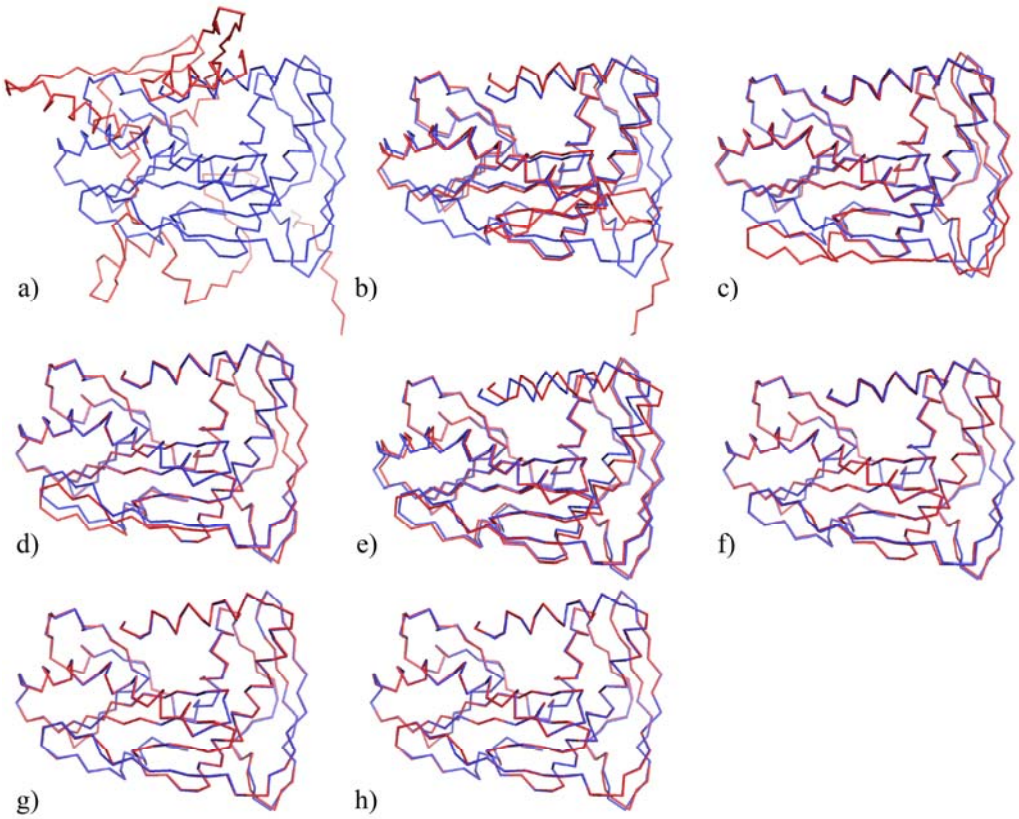

Figure 1. Some intermediate structures of target 2020-01-04_00000019_1 (with 226 residues in length) during the backbone folding process of OPUS-Fold3. The red structures are the intermediate structures and the blue structure is its native state. a)-g) show the intermediate structures at epoch 0, 200, 400, 600, 1200, 1800, and 2400, respectively. h) is the final perdition.

Side-chain and All-atom Folding

In Table 2, we list the results of OPUS-Fold3 on side-chain and all-atom modeling. Here, the real constrains derived from the corresponding PDB-file are used so that we could measure the performance of OPUS-Fold3 through the differences of the predicted structure from its native counterpart. Random values are used as the initial backbone torsion angles (Φ and Ψ) and side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4).

When modeling the side chains with a fixed native backbone (OPUS-Fold3 (sc only) in Table 2), the results show that the predicted side chains are very close to their native counterparts, which indicates the effectiveness of OPUS-Fold3 on delivering the corresponding side-chain conformation with given constrains. As an example, the folding trajectory of OPUS-Fold3 (sc only) on monomer target 2020-02-15_00000234_1 (with 338 residues in length) is shown as a movie in

Supplementary Video S2.

The results also show that when modeling the side chains with a fixed native backbone at first 2400 epochs, and relaxing all atoms at last 600 epochs (OPUS-Fold3 in Table 2), the accuracy of the side-chain modeling is increased, which indicates the backbone relaxation may be helpful to the side-chain reconstruction.

When modeling the backbone and side chains simultaneously from scratch (OPUS-Fold3 (scratch) in Table 2), the backbone folding performance is better than that obtained by modeling the backbone exclusively using random initial torsion angles (OPUS-Fold3 (random) in Table 1). However, the accuracy of the side-chain modeling is decreased, which indicates that the correct backbone conformation is crucial for side-chain modeling.

Table 2. The side-chain and all-atom modeling performance of OPUS-Fold3 on CAMEO60. "OPUS-Fold3 (sc only)" denotes the procedure that models the side chains with a fixed native backbone. "OPUS-Fold3" denotes the procedure that models the side chains with a fixed native backbone at first 2400 epochs, and relaxes all atoms at last 600 epochs. "OPUS-Fold3 (scratch)" denotes the procedure that models the backbone and side chains simultaneously from scratch. Here, for each procedure, the real distance and orientation constrains derived from the corresponding PDB-file are used. Random values are set as the initial backbone torsion angles (Φ and Ψ) and side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4).

	OPUS-Fold3 (sc only)	OPUS-Fold3	OPUS-Fold3 (scratch)
TM-score	1.000	0.999	0.961
ACC	92.30%	93.21%	75.77%
MAE (X_1)	4.69	4.18	13.88
MAE (X_2)	9.07	7.97	13.86
MAE (X_3)	20.45	20.22	20.35
MAE (X_4)	29.22	26.43	25.66

Protein-protein Docking

In Table 3, we list the protein-protein docking performance of OPUS-Fold3 on CAMEO75o using the real constrains derived from the corresponding PDB-file. The results show that OPUS-Fold3 is capable of delivering the correct docking pose when the backbones of receptor and ligand are known (OPUS-Fold3 in Table 3). We show

some intermediate structures during the protein-protein docking process of OPUS-Fold3 on hetero-oligomer target 7SPP in Figure 2.

In addition, we verify the performance of OPUS-Fold3 on simultaneously modeling the backbones of receptor and ligand and the docking pose between them from scratch (OPUS-Fold3 (bbfold) in Table 3). Here, Random values are set as the initial backbone torsion angles (Φ and Ψ). As an example, the trajectory of OPUS-Fold3 (bbfold) on hetero-oligomer target 7SPP is shown as a movie in Supplementary Video S3. The results also indicate that the all-atom folding and docking procedure (OPUS-Fold3 (aafold) in Table 3) may achieve better performance than that using the constrains of backbone exclusively (OPUS-Fold3 (bbfold) in Table 3).

For further illustration, we show a folding and docking trajectory on hetero-oligomer target 7VNB as a movie in Supplementary Video S4. The backbones of receptor and ligand are known at first, and random values are set as the initial side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4). OPUS-Fold3 docks the receptor and ligand in the first 200 epochs, then models the side chains in the following 2400 epochs, and finally performs the folding and docking simultaneously on all atoms at last 600 epochs as relaxation. The results show that with given constraints, OPUS-Fold3 is capable of delivering the corresponding 3D structure either for folding or for docking usage.

Table 3. The protein-protein docking performance of OPUS-Fold3 on CAMEO75o. "OPUS-Fold3" denotes the procedure that models the docking pose with fixed native backbones of receptor and ligand. "OPUS-Fold3 (bbfold)" denotes the procedure that simultaneously models the backbones of receptor and ligand and the docking pose between them from scratch. "OPUS-Fold3 (aafold)" denotes the procedure that simultaneously models the all atoms of receptor and ligand and the docking pose between them from scratch. Here, for each procedure, the real distance and orientation constrains derived from the corresponding PDB-file are used. Random values are set as the initial backbone torsion angles (Φ and Ψ) and side-chain dihedral angles ($X_1, X_2, X_3,$ and X_4) for the relevant tasks.

	TM-score
OPUS-Fold3	0.994
OPUS-Fold3 (bbfold)	0.918

OPUS-Fold3 (aafold)	0.933

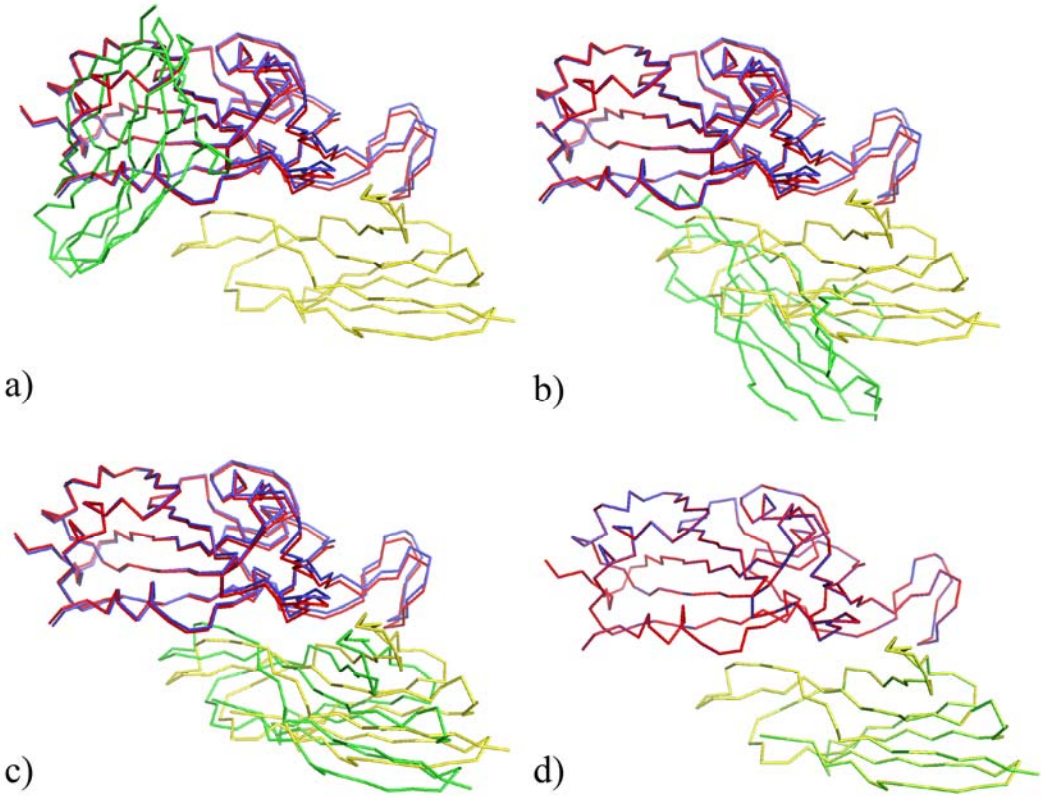

Figure 2. Some intermediate structures of hetero-oligomer target 7SPP during the protein-protein docking process of OPUS-Fold3. The blue and yellow structures are the native backbones of the receptor and ligand, respectively. The red and green structures are their corresponding intermediate structures. a)-d) show the intermediate structures at epoch 0, 40, 50, and 200, respectively.

Conclusion

In this paper, we propose OPUS-Fold3, a gradient-based protein all-atom folding and docking framework. OPUS-Fold3 is capable of accurately delivering the corresponding protein 3D folding and docking conformation using given constrains (e.g. comparable performance to pyRosetta on backbone folding task). OPUS-Fold3 is an open-source method that is written in Python and TensorFlow2.4, so that it can be conveniently incorporated with other TensorFlow-based models.

Acknowledgements

The work was supported by Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), and ZJLab. The work was also supported by National Key Research and Development Program of China (No. 2021YFF1200400).

Data and Software Availability

The source code of OPUS-Fold3 can be downloaded from http://github.com/OPUS-MaLab/opus_fold3. It is freely available for academic usage only.

Authors' contributions

Gang Xu and Jinapeng Ma designed the project. Gang Xu conducted the experiments. All authors contributed to the manuscript composing.

Conflict of Interest

None declared.

References

1. Brunger, 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.; Read, R. J.; Rice, L. M.; Simonson, T.; Warren, G. L., Crystallography & NMR system: A new software suite for macromolecular structure determination. *Acta Crystallogr D* **1998,** *54*, 905-921.
2. Chaudhury, S.; Lyskov, S.; Gray, J. J., PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **2010,** *26* (5), 689-691.
3. Rohl, C. A.; Strauss, C. E. M.; Misura, K. M. S.; Baker, D., Protein structure prediction using rosetta. *Method Enzymol* **2004,** *383*, 66-+.
4. Xu, G.; Wang, Q.; Ma, J., OPUS-Fold: An Open-Source Protein Folding Framework Based on Torsion-Angle Sampling. *J Chem Theory Comput* **2020,** *16* (6), 3970-3976.
5. Wang, S.; Sun, S. Q.; Li, Z.; Zhang, R. Y.; Xu, J. B., Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *Plos Comput Biol* **2017,** *13* (1).
6. Adhikari, B.; Bhattacharya, D.; Cao, R.; Cheng, J., CONFOLD: Residue-residue contact-guided ab initio protein folding. *Proteins* **2015,** *83* (8), 1436-49.
7. Lamb, J.; Elofsson, A., pyconsFold: a fast and easy tool for modelling and docking using distance predictions. *Bioinformatics* **2021**.
8. Yang, J. Y.; Anishchenko, I.; Park, H.; Peng, Z. L.; Ovchinnikov, S.; Baker, D., Improved protein structure prediction using predicted interresidue orientations. *P Natl Acad Sci USA* **2020,** *117* (3), 1496-1503.
9. Xu, G.; Wang, Q.; Ma, J., OPUS-X: An Open-Source Toolkit for Protein Torsion Angles, Secondary Structure, Solvent Accessibility, Contact Map Predictions, and 3D Folding. *Bioinformatics* **2021**.
10. Xu, G.; Wang, Q.; Ma, J., OPUS-Rota4: a gradient-based protein side-chain modeling framework assisted by deep learning-based predictors. *Brief Bioinform* **2022,** *23* (1).
11. Xu, G.; Wang, Q.; Ma, J., OPUS-Rota3: Improving Protein Side-Chain Modeling by Deep Neural Networks and Ensemble Methods. *J Chem Inf Model* **2020,** *60* (12), 6691-6697.
12. Haas, J.; Barbato, A.; Behringer, D.; Studer, G.; Roth, S.; Bertoni, M.; Mostaguir, K.; Gumienny, R.; Schwede, T., Continuous Automated Model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins* **2018,** *86 Suppl 1*, 387-398.
13. Xu, G.; Wang, Y.; Wang, Q.; Ma, J., Studying protein–protein interaction through side-chain modeling method OPUS-Mut. *Briefings in Bioinformatics* **2022**, bbac330.
14. Zhou, H.; Zhou, Y., Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability

prediction. *Protein Sci* **2002,** *11* (11), 2714-26.
15. Huang, X.; Pearce, R.; Zhang, Y., EvoEF2: accurate and fast energy function for computational protein design. *Bioinformatics* **2020,** *36* (4), 1135-1142.
16. Abadi, M.; Barham, P.; Chen, J. M.; Chen, Z. F.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; Zheng, X. Q., TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation* **2016**, 265-283.
17. Kingma, D. P.; Ba, J., Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations* **2015**.
18. Zhang, Y.; Skolnick, J., Scoring function for automated assessment of protein structure template quality. *Proteins* **2004,** *57* (4), 702-10.