Computational and Structural Biotechnology Journal 19 (2021) 6282–6290

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

MUfoldQA_G: High-accuracy protein model QA via retraining and transformation

Wenbo Wang [a], Junlin Wang [a], Zhaoyu Li [a], Dong Xu [a,b], Yi Shang [a,*]

[a] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA
[b] Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211, USA

ARTICLE INFO

Article history:
Received 7 July 2021
Received in revised form 10 November 2021
Accepted 14 November 2021
Available online 23 November 2021

Keywords:
Protein structure prediction
Protein model quality assessment
Multi-model QA methods

ABSTRACT

Protein tertiary structure prediction is an active research area and has attracted significant attention recently due to the success of AlphaFold from DeepMind. Methods capable of accurately evaluating the quality of predicted models are of great importance. In the past, although many model quality assessment (QA) methods have been developed, their accuracies are not consistently high across different QA performance metrics for diverse target proteins. In this paper, we propose MUfoldQA_G, a new multi-model QA method that aims at simultaneously optimizing Pearson correlation and average GDT-TS difference, two commonly used QA performance metrics. This method is based on two new algorithms MUfoldQA_Gp and MUfoldQA_Gr. MUfoldQA_Gp uses a new technique to combine information from protein templates and reference protein models to maximize the Pearson correlation QA metric. MUfoldQA_Gr employs a new machine learning technique that resamples training data and retrains adaptively to learn a consensus model that is better than naïve consensus while minimizing average GDT-TS difference. MUfoldQA_G uses a new method to combine the results of MUfoldQA_Gr and MUfoldQA_Gp so that the final QA prediction results achieve low average GDT-TS difference that is close to the results from MUfoldQA_Gr, while maintaining high Pearson correlation that is the same as the results from MUfoldQA_Gp. In CASP14 QA categories, MUfoldQA_G ranked No. 1 in Pearson correlation and No. 2 in average GDT-TS difference.

1. Introduction

Proteins are macromolecules playing vital roles in most biological processes [1]. Understanding their functionality is crucial in life science. The functionality of a protein largely depends on its unique 3D structure [2]. For example, antibody proteins take advantage of their structures to latch onto foreign proteins and tag them [3]. Unfortunately, determining the 3D structure of a protein from its primary amino acid sequence is difficult [4]. While protein sequence information has been acquired at an ever-growing rate, experimental methods, including electron microscopy, protein crystallography, and nuclear magnetic resonance, for determining protein structures are very expensive and time consuming [5]. With the continuous growing discrepancy between well-established sequence information on millions of proteins and the lack of understanding of their corresponding tertiary structures, computational protein structure prediction methods have become increasingly important [6]. A major event in this field is the Critical Assessment of Techniques for Protein Structure Prediction (CASP) experiment, a biennial event since 1994 [7]. It serves as a platform to provide the blind testing of the cutting-edge protein structure prediction methods designed by researchers from all over the world [8]. In 2020, 215 unique groups participated in CASP14 and 67,976 predictions were submitted [9].

During the past few decades, as reflected on the CASP results, steady progress has been made in generating high-quality 3D protein models via computational structure prediction methods [10], especially since the participation and success of AlphaFold and AlphaFold 2 by the Google/DeepMind team [11,12]. More and more organizations are investing substantial amounts of resources into this area. In the meantime, ever growing number of candidate models of various quality is making it more and more challenging to accurately assess the quality of the predicted models. A better way to predict the quality of a large pool of models that could keep up with this growth on the structure prediction side is in urgent need.

* Corresponding author at: 201 Naka Hall, EECS Department, University of Missouri, Columbia, MO 65201, USA.
E-mail address: shangy@missouri.edu (Y. Shang).

https://doi.org/10.1016/j.csbj.2021.11.021

1.1. QA problem formulation

The quality assessment problem of a predicted protein model (3D structure) can be defined as follows. Given the amino acid sequence of a target protein and a predicted model, return a (predicted) quality assessment (QA) score that approximates the similarity between the model and the native structure of the target protein. One widely used similarity measure between two 3D protein structures is GDT-TS (Global Distance Test Total Score), which is calculated as $(P_1 + P_2 + P_4 + P_8)/4$, where P_n represents the percentage of C-alpha atoms within the threshold of nÅ (n = 1, 2, 4, 8) after superimposing one structure over the other structure [13,14]. The GDT-TS value ranges from 0 to 1, where 1 means the two structures are identical.

The performance metrics for evaluating different QA methods is based on their predicted QA scores of a set of predicted models for a target protein and the corresponding GDT-TS values between the predicted models and the native structure of the target protein. Two commonly used performance metrics are 1) Average GDT-TS Difference (Abbreviated as AGD in this paper), and 2) Pearson Correlation Coefficient (PCC). Specifically, let the predicted QA scores of a set of N predicted models for a target protein be $X_i \in [0,1]$ $(i = 1, \ldots, N)$ and the corresponding GDT-TS values between the N predicted models and the native structure of the target protein (i.e., ground truth) be $Y_i \in [0,1]$ $(i = 1, \ldots, N)$. Then, AGD = $\frac{1}{N}\sum_i |X_i - Y_i|$. PCC is the Pearson correlation coefficient between X_i and Y_i , for $i = 1, \ldots, N$. A low AGD means the predicted QA scores are good approximation of the true qualities of the models, while a high PCC means that models selected based on higher predicted QA scores are likely to be the real high-quality ones.

When different QA methods are evaluated based on multiple performance metrics, such as AGD and PCC, one method may perform better than another method on one metric, but worse on another metric. As illustrated in Fig. 1, the performances of the three methods are plotted in the 2-D space of AGD and PCC. 1-PCC is shown on the Y-axis, so that on both axes, the smaller the value, the better the method. In this example, Method C is the best and dominates the other two methods because it is better than or equal to the other two methods in both AGD and PCC. Methods A and B are non-dominating since A is better than B in PCC, but worse in AGD.

1.2. Existing QA methods

Accurately assessing the quality of a predicted model is an important part of protein structure prediction [15]. Ever since its inclusion in CASP7, the model quality assessment (QA) category has always attracted many participants [16]. Based on their input, existing QA methods can be divided into two major categories: single-model and multi-model. In general, the former does not require additional models, can provide a stable score for a given predicted protein model but the accuracy is inferior. The latter requires reference models. And the results may vary for a given protein depending on the accompanying reference models. But the accuracy is usually superior.

Single-model methods only use one predicted model as input to calculate its quality score. Some of these methods use physics or knowledge based potential functions or predictive models built by machine learning methods [17]. Examples are as follows.

- Ornate [18] features a 3D-CNN deep learning predictive model with the input of density maps.
- SBROD [19] is a heuristic scoring function composed of four terms related to different structural features: residue–residue orientations, contacts between backbone atoms, hydrogen bonding, and solvent–solute interactions. It features a smooth function with respect to atomic coordinates and thus is applicable to continuous gradient-based optimization of protein conformations.
- VoroMQA [20] calculates statistical potentials based on the frequencies of observed interatomic contacts.
- OPUS-Cα [21] is a potential function based on seven representative molecular interactions in proteins: distance-dependent pairwise energy with orientational preference, hydrogen bonding energy, short-range energy, packing energy, tripeptide packing energy, three-body energy, and solvation energy.
- RWplus [22] is a pairwise distance-dependent atomic statistical potential function using a random-walk chain as a reference state.
- GOAP [23] is an orientation-dependent potential that only considers representative atoms, or blocks of side-chain or polar atoms, decomposed into distance and angle dependent terms.

Many recent single-model QA methods are built on top of previous QA methods. They typically use machine-learning methods to combine the results from multiple existing QA methods or feature generation tools. A well-known example is the series of ProQ methods that achieved good results in CASPs [24–28].

- ProQ [24] uses a neural network predictor with atom–atom contacts, residual-residual contacts, secondary structure, and solvent accessibility features as input.
- ProQ2 [25] uses a support vector machine (SVM) predictor with structural and predicted features, re-weighted residue-residue contact, surface area features, and position-specific scoring matrix (PSSM) as input.
- ProQ3 [26] combine the results of the Rosetta software and ProQ2 using SVM.
- ProQ3D [27] combine the results of the Rosetta software and ProQ2 using a multilayer perceptron.
- ProQ4 [28] uses a pretrained 1D-CNN that is fine-tuned using a set of descriptors.

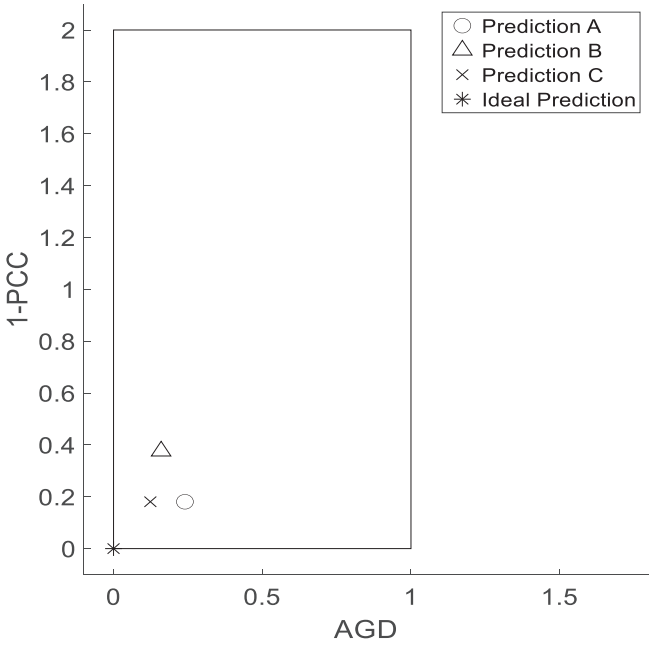

Fig. 1. An illustration of multi-criteria performance comparison of scores generated by three different QA methods.

- QAcon [29] uses a two-layer neural network with 12 features, including structural features, physicochemical properties, and residue contact predictions.
- SMOQ [30] uses SVM with protein sequence and structural features.
- DeepPTQA [31] features an inception network.

Multi-model QA methods require a set of models as input. They use these models collectively to predict the quality score for each of these models. Examples are as follows.

- The series of MULTICOM methods use different machine learning and deep learning methods to build predictors using a large number of features or descriptors as input [32,33].
- MUfoldQA_C [34,35] is a consensus-based method using information from both templates and reference models.
- Wallner [36] combines ProQ2 and Pcons using a linear formulation.

The methods proposed in this paper use two existing methods, MUfoldQA_S [34,35] and MQAPRank [37,38]. MUfoldQA_S is a single-model QA method we tested in CASP12. In the method, each input model is first compared with a set of selected templates and the corresponding GDT-TS values are calculated. Then, for each C-alpha position, use the amino acid in the target protein sequence and those in the templates to retrieve the corresponding values from the BLOSUM45 table. These values are used to calculate weights through a heuristic formula. The final MUfoldQA_S local score for each C-alpha position is the average GDT-TS values between the predicted model and all templates weighted by the corresponding heuristic weights.

MQAPRank is a multi-model QA method that first sorts the set of input models using an SVM-based single-model QA method. Then it takes the first five models as references to predict the qualities of each input model in a consensus approach, i.e., averaging the GDT-TS values between each input model and the 5 reference models.

1.3. Our contribution

In this paper, we present MUfoldQA_G, a new multi-model QA algorithm that uses information from native structures of similar proteins, as well as the whole set of candidate models to evaluate the quality of a large pool of predicted protein models. Several key innovations have contributed to its success in CASP14:

1) **MUfoldQA_Gr** is a new algorithm that consists of an iterative machine-learning process. It first uses a pretrained consensus model to make an initial prediction of the QA scores of the candidate models. Then, it utilizes an adaptive sampling and training technique to build specialized machine-learning models with increased prediction accuracy by adapting to the distribution of the reference models. Empirically, this algorithm achieved good results in the average GDT-TS difference QA metric.
2) **MUfoldQA_Gp** is a new algorithm that takes advantage of information from both protein templates and reference models. It first finds a pool of suitable reference models and calculate GDT-TS values between each candidate model and reference models. Then it utilizes MUfoldQA_S to assign weights to each reference model. The final output is the weighted average of GDT-TS values. Empirically, this algorithm achieved good results in the Pearson correlation QA metric.
3) **MUfoldQA_G** is a new algorithm that combines two predicted QA scores for a protein model, generated by MUfoldQA_Gr and MUfoldQA_Gp, into one QA score in a way to optimize both QA performance metrics (Pearson correlation and average GDT-TS difference) simultaneously. Its results achieve low average GDT-TS difference that is close to results from MUfoldQA_Gr, while maintaining high Pearson correlation that is the same as results from MUfoldQA_Gp.

In the rest of this paper, we will first present the details of MUfoldQA_Gr, MUfoldQA_Gp, and MUfoldQA_G, and then show experimental results.

2. Methods

Formally, the input and output of the QA algorithms presented in this section are defined as follows. Given the amino acid sequence S of a target protein of length U, where U is the number of its C-alpha atoms, and a set of candidate models of the protein, M_i, $i = 1, \ldots, N$, where N is the number of models, output a quality score in range [0, 1] for each model that makes a good approximation of the GDT-TS value between M_i and the native 3D structure of the target protein.

2.1. MUfoldQA_Gp

MUfoldQA_Gp improves our previously published QA method, MUfoldQA_C [35], with a different template and reference model selection scheme. This method performs very well in terms of the PCC QA metric.

A = MUfoldQA_Gp (M, S)

Step 1. Calculate pairwise GDT-TS values between each input model and each reference model.

a) Select a set of reference models from the input set of models. Sort all input models using their MQAPRank scores [37] and choose top $Y = ceil(N*0.45)$ models as the reference model set R_y, $y = 1, \ldots, Y$, in which Y is the size of the reference model set. The constant parameter 0.45 was determined experimentally. We tested thresholds from 5% to 100% with increment 5% and selected the best one, 45%, based on experimental results.
b) Calculate the GDT-TS value G_{xy} between each input model M_x and each reference model R_y.

Step 2. Calculate local scores of reference models.

a) Create a template set using Blast [11]. Use the target protein sequence S to query a PDB database [39] with Blast to find similar proteins. If the number of similar proteins found is less than 10, add them to the template set. Otherwise, score these proteins using a heuristic formula $L = (3 - \log_{10}E) \cdot V \cdot I$, in which E represents the E-value and V is defined as template length divided by the target sequence length while I denotes the percentage of identical sequences. All these values can be either found in or calculated from the Blast report. Then, sort the similar proteins from highest L value to lowest and add protein one-by-one in the sorted order to the template set if either one of these two conditions is met: 1) The template set size is less than 10; 2) Adding this protein will cover at least one new C-alpha position on the target sequence that is not yet covered by the proteins in the template set.
b) Create a template set using HHsearch [12]. Repeat step (a) with HHsearch instead of Blast.

c) Merge the two template sets generated in (a) and (b) without removing any template. Duplicates are not checked. The rationale is that if a template has been chosen by both Blast and HHsearch, it is likely to be good. Thus, duplication gives good templates more weight.
d) For each reference model R_y, run our previously published MUfoldQA_S [34] method using templates generated in (c) to calculate the local scores, W_{yh}, for each C-alpha position h on model R_y.

Step 3. Calculate QA scores of input models.

a) For each C-alpha position *h* of an input model M_x, calculate weighted local scores based on the reference models according to this formula:

$$j_{xh} = \frac{\sum_{y=1}^{Y} G_{xy} W_{yh}}{\sum_{y=1}^{Y} W_{yh}}$$

b) For each input model M_x, calculate its QA score as the average of its local scores:

$$A_x = \frac{1}{U} \sum_{h=1}^{U} j_{xh}$$

Return QA score *A*.

In Step 3, the proposed method of combing the global GDT-TS value with weighted local score to get an updated local score might seem to be counter intuitive, because the deviations causing the lower value of the global GDT-TS value could be in completely different fragments of the structure. Here, our idea is to encode both the global and local structure quality information in the updated local scores and give the local scores in good global structures more weight. We have observed that good global structures tend to have good local structures, although not always. This idea was tested in the QA method MUfoldQA_C during CASP12 and it ranked number 2 among all QA methods.

2.2. MUfoldQA_Gr

MUfoldQA_Gr is a new multi-model QA method featuring an iterative machine-learning process. Its input and output specifications are similar to MUfoldQA_Gp except that MUfoldQA_Gr does not require target sequence S for the input. MUfoldQA_Gr performs well in terms of average GDT-TS difference.

The algorithm first learns a consensus model using training CASP datasets as follows. This learned model is referred to as the pre-trained model in the algorithm below.

MUfoldQA_Gr Pretraining:

1) For each target protein from a training CASP dataset, we sort its CASP server models by their true GDT-TS value (i.e., GDT-TS value to native structures) from high to low. Then, using a sliding window of size N, e.g., N = 150, with stride K, e.g., K = 20, to select N models to form a reference set each time.
2) Create a training set containing training examples with an input feature vector (real values in the range of [0, 1]) of size N and a single scalar output in the range of [0,1]. For each reference model set, pick one model at a time:
a. Calculate the pairwise GDT-TS value between this model and all other models in the set in the order of the naïve consensus score of the reference model.
b. The list of pairwise GDT-TS values forms the feature vector of this model, which is to be used as the input of a training example for a supervised machine-learning method. The corresponding output of the training example is the true GDT-TS value of this model.
3) Any supervised machine-learning algorithm can be applied to the training set to learn the mapping from the pairwise GDT-TS values of a model with respect to models in a reference set to its true GDT-TS value. Compared to the naïve consensus method that estimates the true GDT-TS value of a model as the average GDT-TS values between it and all models in a reference set, the learned model can represent more complex relationships and generate more accurate predictions.

In our experiments, we used CASP5 to CASP11 datasets to train machine-learning models, and CASP12 and CASP 13 dataset separately as the test set to evaluate its experimental performance. For CASP14, we used CASP5 to CASP12 datasets to train machine-learning models. We experimented with various supervised learning algorithms and found that Bagged Trees [40] worked the best.

Based on the pretrained model, MUfoldQA_Gr generates new training examples dynamically for the input model set, M, and learns new machine-learning models on demand.

B = MUfoldQA_Gr (M)

Step 1. Calculate pairwise GDT-TS value R_{xy} between each input model M_x and all input models (M_y, $y = 1, \ldots, N$).

Step 2. For each input model M_x, calculate its naïve consensus score $Q_x = \frac{1}{N} \sum_{y=1}^{N} R_{xy}$

Step 3. Sort input models (M_x, $x = 1, \ldots, N$) based on Q_x from high to low as P_x ($x = 1, \ldots, N$).

Step 4. For each input model P_x, get the pairwise GDT-TS values between it and all other input models in the *P* list (P_y, $y = 1, \ldots, N$), to form the feature vector of P_x.

Step 5. Feed the feature vector of P_x into the pretrained machine-learning model to generate its estimated QA score T_x.

Step 6. Generate a new training set from CASP datasets with model QA score distribution mimicking the distribution of T_x ($x = 1, \ldots, N$).

1) For each CASP target protein, randomly select *N* of its CASP server models so that the distribution of their GDT-TS values is similar to that of T_x ($x = 1, \ldots, N$).
2) Apply ***MUfoldQA_Gr step 1–4*** on these *N* CASP server models to generate their feature vectors and use the model's true GDT-TS as the output label of the training example.
3) Repeat (1)-(2) multiple times for each target, such as *ceil(4*(F/N))* times, to generate the training examples corresponding to one target protein, where *F* is the number of predicted models available for the current target protein.
4) Repeat (1)-(3) for all targets in the training CASP datasets and combine all training examples into a new training set.

Step 7. Apply any machine-learning algorithm, such as Bagged Trees, on this new training set to learn a new model to predict QA score.

Step 8. For each input model P_x, feed its feature vector generated in Step 4 to the new predictive model to generate its predicted QA score.

Return QA scores of all input models.

MUfoldQA_Gr contains an iterative machine learning process to build consensus-like predictors with training sets generated adaptively. Fig. 2 shows its execution time on each target against the length of the target. The average time is around 24 min. Even though the time to calculate the pairwise GDT-TS value is a function of the length of the target, the time is relatively small compared to the machine-learning part of the algorithm.

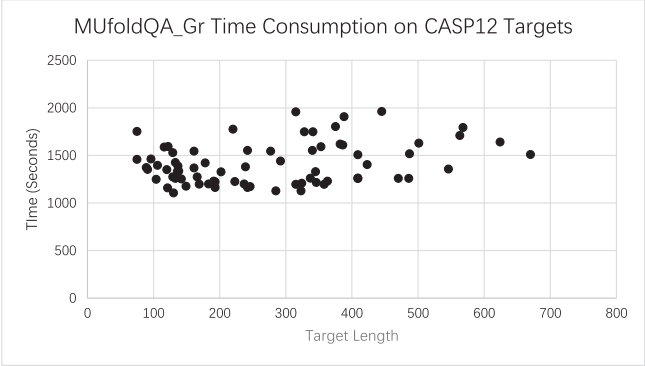

Fig. 2. MUfoldQA_Gr Time Consumption on CASP12 Targets on Intel(R) Xeon(R) Gold 6140 CPU, using MATLAB Linux R2019b.

2.3. MUfoldQA_G

MUfoldQA_G is a new multi-model QA method designed to simultaneously optimize Pearson correlation and average GDT-TS difference, two commonly used QA performance metrics. This method is based on MUfoldQA_Gp and MUfoldQA_Gr. In practice, MUfoldQA_Gp achieves high Pearson correlation, whereas MUfoldQA_Gr achieves low average GDT-TS difference. MUfoldQA_G uses a new transformation process to combine the results of the two algorithms so that it achieves good performance in both Pearson correlation and average GDT-TS difference.

The main idea of this method is as follows. Considering a set of predicted QA values ($X_n \in [0,1]$, $n = 1, \ldots, N$) and their corresponding ground truth values, the true GDT-TS values ($Y_n \in [0,1]$, $n = 1, \ldots, N$), our goal is to transform the QA values into new values such that the Pearson correlation coefficient between the QA values and the ground truth values is high and the average difference between the QA values and the ground truth values is small.

Intuitively, there are two aspects to consider: a) the relative difference between a pair of QA values should be close to their corresponding ground truth values, which loosely translates to a high Pearson correlation coefficient; b) each QA value should be as close to the corresponding ground truth value as possible, which translates to low average GDT-TS difference. In our case, MUfoldQA_Gp algorithm performs well in Pearson correlation (in other words, the relative position of two scores), while MUfoldQA_Gr is a top performer in achieving low average GDT-TS difference. We will combine the two QA scores generated by these two algorithms into one QA score to preserve their strength in both Pearson correlation and average GDT-TS difference.

Fig. 3A shows an illustration of the method. In this artificial example, there are 5 models with their ground truth values and corresponding predicted QA scores from methods A and B, as follows.

GroundTruth = [0.1,0.2,0.3,0.4,0.5].
Prediction_A = [0.2,0.4,0.6,0.8,1.0].
Prediction_B = [0.3,0.2,0.1,0.3,0.7].

When plotting predicted QA scores against the ground truth values, a perfect prediction would have all the points on the 45° diagonal line (also known as the $Y = X$ line). Given two predictions by A and B, assuming Prediction A is highly correlated with the ground truth (PCC = 1), but of high average GDT-TS difference (AGD = 0.3). While Prediction B has a low average GDT-TS difference (AGD = 0.14), but not a very high correlation with the ground truth (PCC = 0.62). Our new method could combine these two predictions into Prediction_C, where C = [0.14,0.23,0.32,0.41,0.50], achieving 1.00 PCC and 0.02 AGD.

Specifically, given any QA predictions A and B, MUfoldQA_G outputs is QA score C in [0, 1] for each model. It performs a linear mapping from A to C so that the final score C will have the same PCC value as score A. We use the following formula to calculate C, in which overbar indicates arithmetic mean:

$$b = \frac{\overline{AB} - \overline{A}\,\overline{B}}{\overline{(A)^2} - \overline{A}^2}$$

$$a = \overline{B} - b\overline{A}$$

$$C = a + bA$$

In our case, MUfoldQA_G performs a linear transformation of MUfoldQA_Gp scores. Therefore, the Pearson correlation between MUfoldQA_G scores with the ground truth is the same as that between MUfoldQA_Gp scores with the ground truth. In terms of average GDT-TS difference, Fig. 4 compares the result of MUfoldQA_Gr and MUfoldQA_G for each CASP 12 target protein. It shows that MUfoldQA_G is better on 48.61% of the targets. On 41.67% of the targets, MUfoldQA_G is slightly worse, within 0.005. On 4.17% of the target, the performance difference is between 0.005 and 0.01. On 5.56% of the targets, the performance difference is larger than 0.01. Overall, the performance of MUfoldQA_G is very close to MUfoldQA_Gr on average GDT-TS difference.

Fig. 3B demonstrates how MUfoldQA_G transforms the results from MUfoldQA_Gp and MUfoldQA_Gr on target T1019s1. The x-axis is the true GDT-TS value, and the y-axis is the predicted score using corresponding algorithms mentioned in the respective figure title. MUfoldQA_G achieves the highest PCC and lowest AGD.

3. Results

In our experiments, we tested MUfoldQA_Gr pretraining with leave-one-out cross-validation (LOOCV). And for the complete pipeline, we tested the methods using different CASP datasets. During algorithm development, we tested the methods on CASP12 dataset. Then, we froze the code and tested it on CASP13 dataset. Finally, we participated in CASP14 and submitted our results under the group name MUfoldQA_G.

3.1. MUfoldQA_Gr pretraining leave-one-out cross-validation results

To evaluate the performance of MUfoldQA_Gr pretraining, we performed leave-one-out cross-validation in the following manner. Given datasets from CASP5 to CASP12, each time we used a single CASP dataset as the test set, while using the rest CASP datasets as the training set. The training and test errors (in terms of RMSE) from MUfoldQA_Gr pretrain pipeline are shown in the Table 1.

Table 1
MUfoldQA_Gr pretraining cross-validation results measured in RMSE.

Test set	Training error (RMSEx100)		Test error (RMSEx100)	
	Consensus	MUfoldQA_Gr Pre	Consensus	MUfoldQA_Gr Pre
CASP5	9.84	**2.56**	**14.12**	15.22
CASP6	10.92	**2.74**	9.54	**8.74**
CASP7	10.90	**2.78**	7.49	**7.06**
CASP8	10.37	**2.69**	11.95	**11.22**
CASP9	10.80	**2.77**	9.79	**8.54**
CASP10	10.74	**2.72**	9.87	**9.07**
CASP11	10.76	**2.78**	8.28	**8.27**
CASP12	10.70	**2.78**	8.98	**8.00**

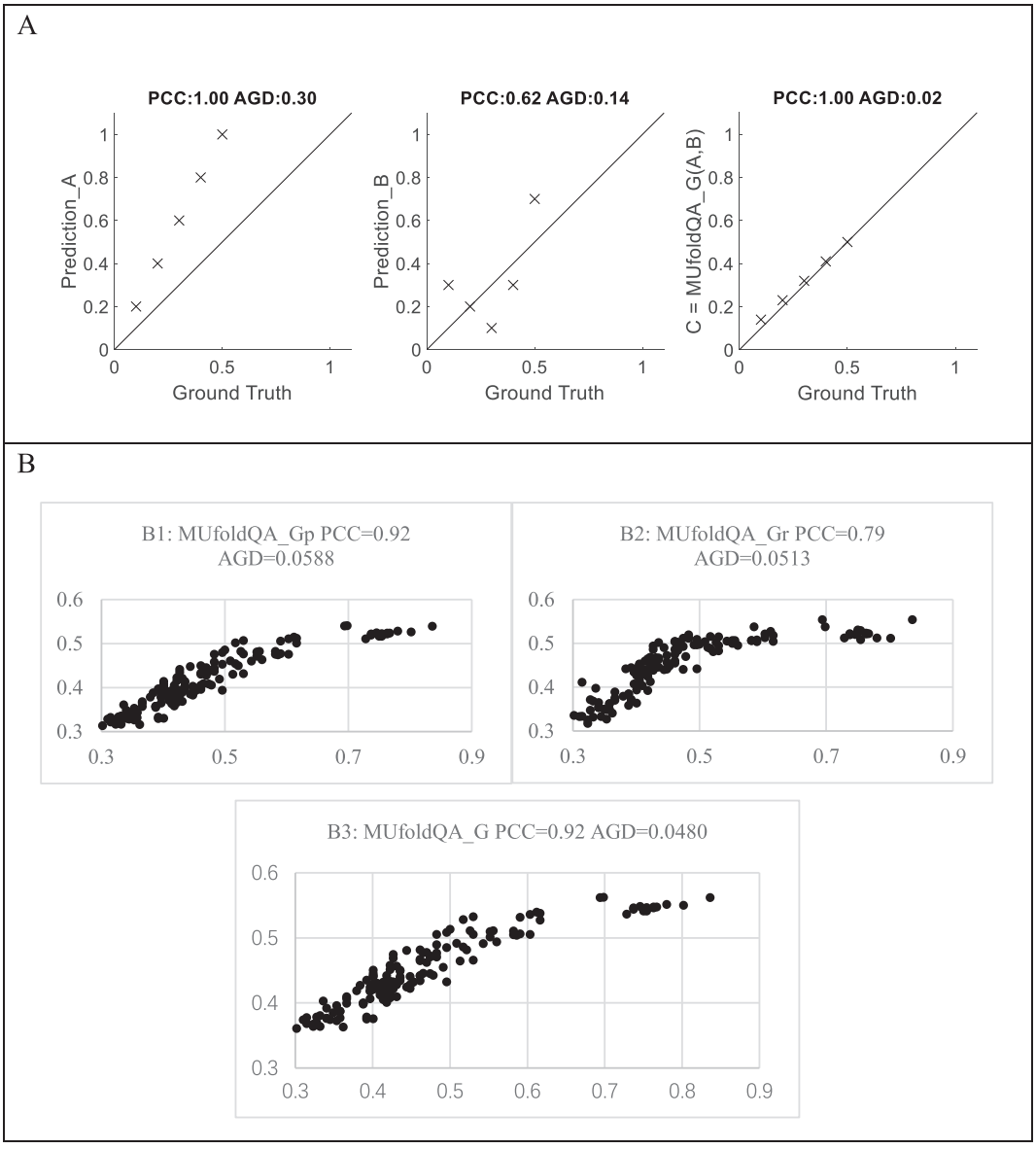

Fig. 3. An illustration of how the MUfoldQA_G process merges two sets of predictions. (A) Using smaller artificial data to intuitively show how the merging process works. (B) MUfoldQA_G transforms the results from MUfoldQA_Gp and MUfoldQA_Gr using real-word target T1019s1. The x-axis is the true GDT-TS value, and the y-axis is the predicted score. (B1) Results from MUfoldQA_Gp. (B2) Results from MUfoldQA_Gr. (B3) Results from MUfoldQA_G, which is calculated using the results from MUfoldQA_Gp and MUfoldQA_Gr.

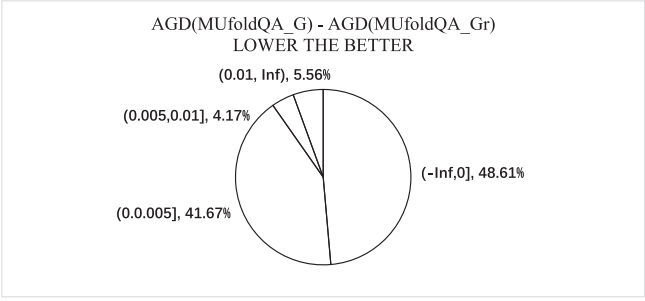

Fig. 4. Performance comparison between MUfoldQA_Gr and MUfoldQA_G in terms of average GDT-TS difference.

They are then compared with those of naïve consensus. The results show that our method has lower training errors across the board and lower test errors in all but CASP5 case.

Table 2
Performance comparison between Naïve Consensus, MUfoldQA_Gr, MUfoldQA_Gp, and MUfoldQA_G on CASP12 dataset.

Method	Average GDT-TS Difference	Pearson Correlation
Naïve Consensus	0.06222	0.7899
MUfoldQA_Gr	0.04930	0.8183
MUfoldQA_Gp	0.05520	0.8401
MUfoldQA_G	0.04948	0.8401

3.2. CASP12 results

In CASP 12, a total of 85 targets were released for QA, among which 13 targets (T0908, T0916, T0919, T0924, T0925, T0926, T0927, T0935, T0936, T0937, T0938, T0939, and T0940) were canceled, and 2 (T0865, T0929) did not show up in the official assess-

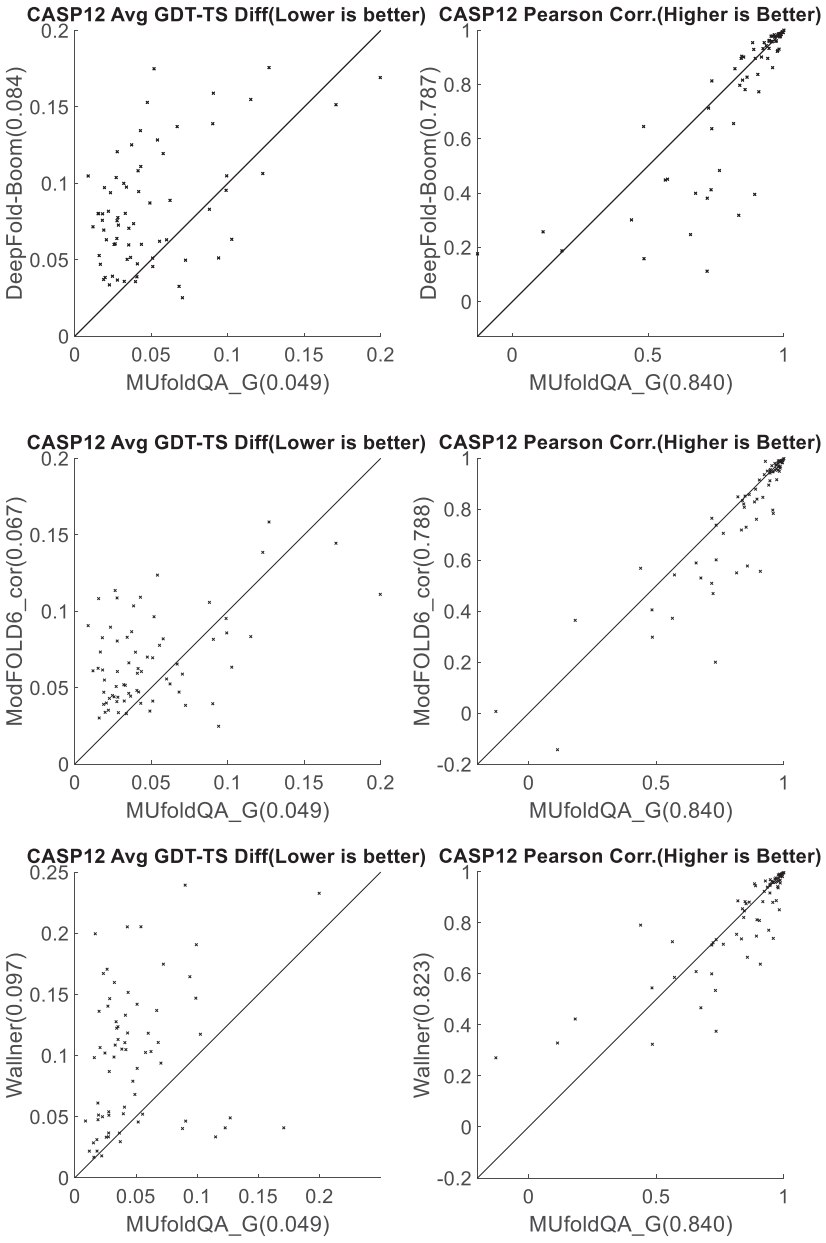

Fig. 5. Performance comparison between MUfoldQA_G and other top QA methods including DeepFold-Boom, ModFOLD6_cor, and Wallner.

Table 3
Performance comparison between Naïve Consensus, MUfoldQA_Gr, MUfoldQA_Gp, and MUfoldQA_G on CASP13 dataset.

Method	Average GDT-TS Difference	Pearson Correlation
Naïve Consensus	0.07365	0.8792
MUfoldQA_Gr	0.05677	0.8818
MUfoldQA_Gp	0.05837	0.8938
MUfoldQA_G	0.05760	0.8938

ment. We used the remaining 70 to evaluate our methods. We used the database generated in April 2016, the month before CASP 12, for Blast and HHsearch template search, and used CASP5-11 targets to train machine-learning models.

We gather the ground truth by extracting the "GDT_TS" field of https://predictioncenter.org/download_area/CASP12/results_LGA_sda/[*TargetName*].SUMMARY.lga_sda.txt and matching them back based on the group name-group code lookup table extracted from the official website. Then, we calculate the Person correlation as well as GDT-TS difference and average them across all 70 targets.

Table 2 shows that MUfoldQA_G achieves a high Pearson correlation, the same as MUfoldQA_Gp, and at the same time, low average GDT-TS difference. It outperforms Naïve consensus significantly: 20% better in terms of average GDT-TS difference and 6% better in Pearson correlation.

Fig. 5 compares MUfoldQA_G with several top QA methods in CASP12, including DeepFold-Boom, ModFold6_cor, and Wallner. We downloaded their performance scores for each target directly from the CASP website. MUfoldQA_G outperformed DeepFold-

Table 4
Pearson correlation coefficient between predicted and observed in CASP14 averaged over all targets (top 20 groups).

Ranking	Group No	Group Name	Pearson	Sample Size
1	QA446	MUfoldQA_G	0.7460	67
2	QA433	DAVIS-EMAconsensus	0.7426	67
3	QA263	DAVIS-EMAconsensusAL	0.7392	67
4	QA075	MULTICOM-CLUSTER	0.7313	67
5	QA035	ModFOLDclust2	0.7310	67
6	QA214	MESHI_consensus	0.7279	66
7	QA032	MESHI	0.7276	65
8	QA216	EMAP_CHAE	0.7218	67
9	QA149	Bhattacharya-Server	0.7046	67
10	QA460	Yang_TBM	0.7029	67
11	QA198	MULTICOM-CONSTRUCT	0.6962	67
12	QA140	Yang-Server	0.6894	67
13	QA187	MULTICOM-HYBRID	0.6851	67
14	QA379	Wallner	0.6785	67
15	QA409	UOSHAN	0.6652	67
16	QA275	MULTICOM-AI	0.6557	67
17	QA167	ModFOLD8	0.6185	67
18	QA209	BAKER-ROSETTASERVER	0.6107	67
19	QA183	tFold-CaT	0.6009	67
20	QA024	DeepPotential	0.5810	66
		Many more groups omitted...		

*Seder2020 and Seder2020hard only submitted 1 prediction, making it an unfair comparison when other groups submitted at least 65 predictions. As a result, we removed these two groups from the ranking.

Table 5
GDT-TS differences between predicted and observed in CASP14, averaged over all targets (top 20 groups).

Ranking	Group No	Group Name	AGD(x100)
1	QA433_2	DAVIS-EMAconsensus	6.737
2	QA446_2	MUfoldQA_G	7.233
3	QA214_2	MESHI_consensus	7.240
4	QA032_2	MESHI	7.254
5	QA035_2	ModFOLDclust2	7.358
6	QA216_2	EMAP_CHAE	7.396
7	QA460_2	Yang_TBM	8.044
8	QA409_2	UOSHAN	8.365
9	QA140_2	Yang-Server	8.553
10	QA075_2	MULTICOM-CLUSTER	8.886
11	QA263_2	DAVIS-EMAconsensusAL	9.230
12	QA198_2	MULTICOM-CONSTRUCT	9.240
13	QA379_2	Wallner	9.993
14	QA187_2	MULTICOM-HYBRID	10.573
15	QA275_2	MULTICOM-AI	11.100
16	QA257_2	P3De	12.020
17	QA073_2	RaptorX-QA	12.060
18	QA024_2	DeepPotential	12.239
19	QA081_2	MUFOLD	12.557
20	QA209_2	BAKER-ROSETTASERVER	12.682
		Many more groups omitted...	

Boom, ModFOLD6_cor and Wallner by 41%, 27% and 49%, respectively, in terms of average GDT-TS difference and by 7%, 7% and 2%, respectively, in terms of Pearson correlation.

3.3. CASP13 results

Testing on CASP 13 dataset is challenging because only one fourth of the targets had their true structures publicly released after the event. Fortunately, the true GDT-TS values of the predicted models for some targets were publicly released. Additionally, many targets only contain a single domain. The true GDT-TS value of the predicted model on that domain is also included in the public release, which could approximate the true GDT-TS value for the whole structure for the ones that lack of such information. By using information from multiple sources, we collected a test set of 79 targets.

We tested our methods on this data set using a protein database generated in April 2018, the month before CASP 13, for Blast and HHsearch template search. We used CASP5-11 datasets for training machine-learning models. Table 3 shows that MUfoldQA_G performed the best in Pearson correlation and the second best in the average GDT-TS difference. Again, it outperformed Naïve Consensus significantly, with 21.8% better on average GDT-TS difference and 1.7% on Pearson correlation coefficient.

3.4. CASP14 results

Finally, as an ultimate blind test and comparison with other state-of-the-art methods worldwide, we participated in CASP14 in 2020. We used the May 2020 Protein Database for Blast and HHsearch template search and used CASP5-12 datasets for training machine-learning models. Table 4 shows the performance comparison of the top 20 QA groups in terms of Pearson correlation. Since the average among all targets is unavailable on CASP official website, we downloaded the per-target Pearson correlation from the official website [41] and calculated the average ourselves. For the GDT-TS difference, the averages among all targets are directly available on the official website [42], as shown in Table 5.

MUfoldQA_G performed very well in CASP14 and ranked No. 1 in Pearson correlation coefficient and No. 2 in average GDT-TS difference, respectively. It is one of the few methods that achieved high ranking on both performance metrics.

4. Conclusions

This paper presented three new QA algorithms, MUfoldQA_Gp, MUfoldQA_Gr and MUfoldQA_G. MUfoldQA_Gp effectively combines information from template and reference models. MUfoldQA_Gr employs a new two-stage prediction method and performs iterative resample-and-retrain that allows the information from the distribution of the reference models being used during training and prediction to create improved consensus-like predictors. MUfoldQA_G effectively combines the results of MUfoldQA_Gp and MUfoldQA_Gr through simultaneously optimizing two QA performance metrics, Pearson correlation and average GDT-TS difference.

We tested these methods on the CASP12 and CASP13 datasets, and eventually participated in CASP14. On CASP12 and CASP13 datasets, the methods outperformed existing state-of-the-art QA methods. In CASP 14 in 2020, MUfoldQA_G ranked No. 1 in Pearson correlation coefficient and No. 2 in the average GDT-TS difference among all QA teams.

Just like other consensus-based QA algorithms, our algorithm will not perform well when all reference models are of low quality. To reduce the impact of the reference model pool quality, further work could be done on introducing independent models generated by one or more protein structure prediction software. Using variable-sized adaptive reference model selection instead of a fixed-percentage of top models could also potentially improve the performance of the algorithm.

CRediT authorship contribution statement

Wenbo Wang: Methodology, Software, Validation, Data curation, Visualization, Investigation, Writing – original draft, Writing – review & editing. **Junlin Wang:** Data curation, Writing – original draft, Writing – review & editing. **Zhaoyu Li:** Data curation. **Dong Xu:** Conceptualization, Funding acquisition, Supervision, Project administration, Writing – review & editing. **Yi Shang:** Conceptualization, Resources, Funding acquisition, Supervision, Project administration, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is partially supported by the National Institutes of Health Grant R01-GM100701.

References

[1] He Z, Ma W, Zhang J, Xu D. A New Hidden Markov Model for Protein Quality Assessment Using Compatibility Between Protein Sequence and Structure. Tsinghua Sci Technol 2015;19(6):559–67.
[2] Mulnaes D, Koenig F, Gohlke H. TopSuite Web Server: A Meta-Suite for Deep-Learning-Based Protein Structure and Quality Prediction. J Chem Inf Model 2021;61(2):548–53.
[3] Mertz L. New, At-Home Antibody Test for Detecting, Tracking COVID-19. IEEE Pulse 2020;11(5):28–31.
[4] Kuhlman B, Bradley P. Advances in protein structure prediction and design. Nat Rev Mol Cell Biol 2019;20(11):681–97.
[5] Johnson MS, Srinivasan N, Sowdhamini R, Blundell TL. Knowledge-based protein modeling. Crit Rev Biochem Mol Biol 1994;29(1):1–68.
[6] Cao R, Bhattacharya D, Hou J, Cheng J. DeepQA: improving the estimation of single protein model quality with deep belief networks. BMC Bioinf 2016;17(1):495.
[7] Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A. A study of quality measures for protein threading models. BMC Bioinf 2001;2(1):1–15.
[8] Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP) — round x. Proteins 2014;82(S2):1–6.
[9] CASP14 in numbers - CASP14, https://www.predictioncenter.org/casp14/numbers.cgi; [Accessed May 15 2021].
[10] Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (CASP)-Round XIII. Proteins 2019;87(12):1011–20.
[11] Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, et al. Improved protein structure prediction using potentials from deep learning. Nature 2020;577(7792):706–10.
[12] AlQuraishi M, Valencia A. AlphaFold at CASP13. Bioinformatics 2019;35(22):4862–5.
[13] Zemla A. LGA: A method for finding 3D similarities in protein structures. Nucleic Acids Res 2003;31(13):3370–4.
[14] Zemla A, Venclovas ?, Moult J, Fidelis K. Processing and evaluation of predictions in CASP4. Proteins 2001;45(S5):13–21.
[15] Olechnovič K, Kulberkytė E, Venclovas Č. CAD-score: a new contact area difference-based function for evaluation of protein structural models. Proteins 2013;81(1):149–62.
[16] Baldassarre F, Menéndez Hurtado D, Elofsson A, Azizpour H, Ponty Y. GraphQA: protein model quality assessment using graph convolutional networks. Bioinformatics 2021;37(3):360–6.
[17] Cossio P, Granata D, Laio A, Seno F, Trovato A. A simple and efficient statistical potential for scoring ensembles of protein structures. Sci Rep 2012;2(1):351.
[18] Pagès G, Charmettant B, Grudinin S, Valencia A. Protein model quality assessment using 3D oriented convolutional neural networks. Bioinformatics 2019;35(18):3313–9.
[19] Karasikov M, Pagès G, Grudinin S, Valencia A. Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. Bioinformatics 2019;35(16):2801–8.
[20] Olechnovič K, Venclovas Č. VoroMQA: Assessment of protein structure quality using interatomic contact areas. Proteins 2017;85(6):1131–45.
[21] Wu Y, Lu M, Chen M, Li J, Ma J. OPUS-Ca: a knowledge-based potential function requiring only Calpha positions. Protein Sci 2007;16(7):1449–63.
[22] Zhang J, Zhang Y, Fernandez-Fuentes N. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. PLoS ONE 2010;5(10):e15386. https://doi.org/10.1371/journal.pone.001538610.1371.
[23] Zhou H, Skolnick J. GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. Biophys J 2011;101(8):2043–52.
[24] Wallner B, Elofsson A. Can correct protein models be identified? Protein Sci 2003;12(5):1073–86.
[25] Ray A, Lindahl E, Wallner B. Improved model quality assessment using ProQ2. BMC Bioinf 2012;13(1):1–12.
[26] Uziela K, Shu N, Wallner B, Elofsson A. Pro Q3: Improved model quality assessments using Rosetta energy terms. Sci Rep 2016;6(1):1–10.
[27] Uziela K, Menéndez Hurtado D, Shu N, Wallner B, Elofsson A. ProQ3D: improved model quality assessments using deep learning. Bioinformatics 2017;33(10):1578–80.
[28] Hurtado DM, Uziela K, Elofsson A. Deep transfer learning in the assessment of the quality of protein models. arXiv preprint arXiv:1804.06281 2018.
[29] Cao R, Adhikari B, Bhattacharya D, Sun M, Hou J, Cheng J. QAcon: single model quality assessment using protein structural and contact information with machine learning techniques. Bioinformatics 2017;33(4):586–8.
[30] Cao R, Wang Z, Wang Y, Cheng J. SMOQ: a tool for predicting the absolute residue-specific quality of a single protein model with support vector machines. BMC Bioinf 2014;15(1):1–8.
[31] Wang J, Wang W, Shang Y. A New Approach Of Applying Deep Learning To Protein Model Quality Assessment. IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2019;2019:2387–92.
[32] Cao R, Bhattacharya D, Adhikari B, Li J, Cheng J. Large-scale model quality assessment for improving protein tertiary structure prediction. Bioinformatics 2015;31(12):i116–23.
[33] Chen X, Liu J, Guo Z, Wu T, Hou J, Cheng J. Protein model accuracy estimation empowered by deep learning and inter-residue distance prediction in CASP14. Sci Rep 2021;11(1):10943.
[34] Wang W, Wang J, Xu D, Shang Y. Two New Heuristic Methods for Protein Model Quality Assessment. IEEE/ACM Trans Comput Biol Bioinform 2020;17(4):1430–9.
[35] Wang W, Li Z, Wang J, Xu D, Shang Y. PSICA: a fast and accurate web service for protein model quality analysis. Nucleic Acids Res 2019;47(W1):W443–50.
[36] Elofsson A, Joo K, Keasar C, Lee J, Maghrabi AHA, Manavalan B, et al. Methods for estimation of model accuracy in CASP12. Proteins 2018;86:361–73.
[37] Jing X, Dong Q. MQAPRank: improved global protein model quality assessment by learning-to-rank. BMC Bioinf 2017;18(1):275.
[38] Jing X, Wang K, Lu R, Dong Q. Sorting protein decoys by machine-learning-to-rank. Sci Rep 2016;6(1):1–11.
[39] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res 2000;28(1):235–42.
[40] Quantile MN, Forests R. J. Mach. Learn. Res. 2006;7(6):983–99.
[41] Results - CASP14 (Correlation), https://www.predictioncenter.org/casp14/qa_corr.cgi; [Accessed May 15 2021].
[42] Results - CASP14 (Differences), https://www.predictioncenter.org/casp14/qa_diff_mqas.cgi; [Accessed May 15 2021].