Refinement of AlphaFold-Multimer structures with single sequence input

Toshiyuki Oda[1]

[1] XTus Inc.

Email: oda@pezy.co.jp

Abstract

AlphaFold2, introduced by DeepMind in CASP14, demonstrated outstanding performance in predicting protein monomer structures. It could model more than 90% of targets with high accuracy, and so the next step would surely be multimer predictions, since many proteins do not act by themselves but with their binding partners. After the publication of After AlphaFold2, DeepMind published AlphaFold-Multimer, which showed excellent performance in predicting multimeric structures. However, its accuracy still has room for improvement compared to that of monomer predictions by AlphaFold2. In this paper, we introduce a fine-tuned version of AlphaFold-Multimer, named AFM-Refine-G, which uses structures predicted by AlphaFold-Multimer as inputs and produces more refined structures without the helps of multiple sequence alignments or templates. The performance of AFM-Refine-G was assessed using two datasets, Ghani_et_al_Benchmark2 and Yin_et_al_Hard, adapted from previous studies by Ghani et al. and Yin et al., respectively. The Ghani_et_al_Benchmark2 dataset consists of 17 recently published heteromers and the Yin_et_al_Hard dataset consists of 133 multimers, including immune-related complexes and repebody-antigen complexes, with several whose correct structure AlphaFold-Multimer could not predict. We predicted five models per target (750 models in total) and analyzed the improvement in the DockQ of each model. Of 750 models, 115 had a DockQ improvement > 0.05 after refinement, demonstrating that our model is useful for the refinement of multimer structures. However, 14 structures had a degraded DockQ < -0.05 after refinement, and the overall prediction quality for targets in Yin_et_al_Hard was quite low; 97 out of 133 were classified as 'Incorrect' with CAPRI criteria, revealing that there is still room for improving multimer predictions.

Introduction

Predicting the protein structure has long been a major challenge in biology. AlphaFold2, presented by

DeepMind in CASP14, demonstrated outstanding performance, and could model > 90% (GDT_TS > 70 for 87 out of 92 domains) of the CASP14 targets with high accuracy (Jumper, et al., 2021; Jumper, et al., 2021; Pereira, et al., 2021). After the competition, its derivative: AlphaFold-Multimer, was published (Evans, et al., 2022), and also showed excellent performance in multimer prediction (Evans, et al., 2022; Yin, et al., 2022).

However, unlike the performance of AlphaFold2 when predicting monomer structure, room for improvement remains for multimer predictions. The success (including not only 'High' but also lower quality classes in CAPRI (Lensink, et al., 2007) criteria) rate is approximately 85% for general heterodimer complexes and 25% for immune-related complexes or repebody-antigen complexes (Yin, et al., 2022). Predicting the multimeric structure of proteins is as important as that of the monomeric structure because many proteins do not act alone and require their binding partners.

We assumed that there are two types of problems associated with the difficulty in predicting multimer structures: 1) Many modern protein structure prediction methods (which also seems to be true for AlphaFold2 and AlphaFold-Multimer) rely on co-evolution information between residues (Anishchenko, et al., 2021; Zheng, et al., 2021). However, to extract co-evolution information between residues in different proteins, it is necessary to find the correct sequence pairs in multiple sequence alignment (MSA). The pairing is easy when no paralogs are present or if other information (e.g., they are coded in the same operon) can be used to identify the interacting partner. Unfortunately, pairing is usually difficult because many proteins have paralogs and the number of known operons is limited. To tackle this problem, several techniques have been used (Feinauer, et al., 2016; Humphreys, et al., 2021; Ovchinnikov, et al., 2014) but the procedure cannot be expected to be perfect, and inaccurate pairings add noise to the prediction. (We should note that, at this point, although the number of cases is limited, Gao et al. showed that AlphaFold-Multimer without pairing improved the rate of prediction success (Gao, et al., 2022).) 2) Amino acid profiles constructed from evolutionarily related sequences in MSAs are useful for structure prediction (Rost and Sander, 1995). This is because evolutionarily related sequences, often referred to as family or superfamily, have similar structures in most cases (Chandonia, et al., 2022; Sillitoe, et al., 2021). Therefore, hydrophilic residues found in the same regions of evolutionarily related sequences would support the prediction that these regions are solvent accessible, and vice versa. However, sequence-specific regions, such as Complementarity Determining Regions (CDRs) of antibodies, differ from their evolutionarily related sequences in terms of both amino acid sequence and structure. Thus, information from MSA would add noise in this regard.

Therefore, further improvements should be possible by not using MSA to predict the structure. As a result, prediction without MSA has reduced noise; however, it also means that there is no support from evolutionarily related sequences. It has been reported that AF2 retrained using a single sequence as input was inaccurate (Wu, et al., 2022). Thus, we used predicted structures which,

although not perfect, were functional for input. Consequently, we trained our model to refine these predicted structures in a physically preferable situation. In selecting the training samples, we avoided segments that did not interact well with other segments and the training samples were penalized based on the number of residue-residue contacts. This procedure was based on the assumption that if a structure is not globular, if it lacked a sufficient internal interaction network, then the structure is flexible and machine learning methods cannot adequately model the rules of structures. In this regard, our model may be heavily biased toward globular structures but can still provide a useful baseline for predictions.

Here, we present a fine-tuned version of AlphaFold-Multimer, named AFM-Refine-G, which uses multimer structures of proteins as input and outputs refined structures. Roney and Ovchinnikov demonstrated that the official parameters of AlphaFold2 can be used to improve the structures without MSA features (Roney and Ovchinnikov, 2022), and our in-house benchmarks showed that fine-tuning the official parameters of AlphaFold-Multimer could be effective (data not shown). Therefore, we did not train the model *de novo* but, instead, fine-tuned the official parameter param_multimer_model_1_v2. The performance of AFM-Refine-G was tested using two datasets, Ghani_et_al_Benchmark2 and Yin_et_al_Hard, which were adapted from studies by Ghuani et al. (Ghani, et al., 2022) and Yin et al. (Yin, et al., 2022). The Ghani_et_al_Benchmark2 dataset consists of 17 recently published heteromers and the Yin_et_al_Hard dataset consists of 133 multimers, including immune-related complexes and repebody-antigen complexes, with many of complexes that AlphaFold-Multimer (copied on 2021-11-2) could not predict correctly. First, the protein complexes in the test datasets were predicted using AlphaFold-Multimer's official pipeline, and the predicted structures were fed into the AFM-Refine-G. Then, the input and refined predicted structures were assessed in terms of DockQ and TM-score. For Ghani_et_al_Benchmark2, our model improved the quality of the predicted structures in general, whereas for Yin_et_al_Hard, some structures decreased in quality. In addition, 97 out of 133 structures in Yin_et_al_Hard were classified as “Incorrect” by CAPRI criteria, indicating that there is still room for improvement for multimer predictions.

Materials and Methods

Training Script

The codes of AlphaFold-Multimer for the loss calculation and model architecture were copied from the inference code of DeepMind's official GitHub repository downloaded on 2022-11-05. v2.2 code was cherry-picked and introduced. The codes were then modified to allow mixed precision. A bug for residues with the same residue index was resolved and "NotImplementedError" in structure module loss was commented out. Multi-chain permutation was considered when providing the ground truth.

Other codes required for training, such as processing training samples or updating parameters, were written in-house. Scripts were executed as maximally deterministic (setting random seeds and environment variables, such as XLA_FLAGS=--xla_gpu_deterministic_reductions and TF_CUDNN_DETERMINISTIC=1). The chain center-of-mass loss term and clash loss term were not implemented.

Contact-Based Spatial Cropping

To use physically preferred structures as the ground truth, we performed customized spatial cropping (Algorithm 1). We intended to obtain segments that 1) had a sufficient number of residues contacting (distance between two CA < 8.0 Å) with other segments and 2) were longer than the predefined length. The algorithm could generate samples that did not meet these criteria and these were filtered after the multi-chain permutated ground-truth process (described below).

Algorithm 1 Contact-Based Spatial Cropping

```
Input: n, the number of residues in the given structure
Input: R, a list of the residues in the given structure (i = 1 to n)

crop_size = 300
min_contact = 2
min_length = 10

if n <= crop_size then
   return R
end if

# Select a random residue
c ← RandomUniform(1 to n)

# Create a list of the residue groups
G ← Create a list of groups comprises of consecutive residues in R

T ← Take the group containing Rc

# While the number of residues in T is less than min_length,
# select a new random residue and update T.
while number of residues in T < min_length

   c ← RandomUniform(1 to n)

   T ← Group containing Rc

end

l ← crop_size

D ← A list of size n

Q ← An empty list
```

```
# Calculate the CA-CA distance between each residue and Rc
# and add a small constant to account to break ties.
for i = 1 to n do

    Di ← CA-CA distance between Ri and Rc

    Di ← Di + (absolute value of (i-c))*1.0e-8

end for

R ← Sort R according to D

# To reduce the number of iterations, assess larger number of amino acids (+20) at the first step.
for k in [20, 1] do

    while True

        S ← Take the first l residues in R

        F ← Create a list of groups comprises of consecutive residues in S

        F ← Extract groups from F with more than or equal to min_length residues

        F ← Merge groups in F that have more than min_contact contact residues with each other

        Q ← Take the group containing Rc

        if (number of residues in Q) >= crop_size then
            break
        end if

        if l >= n then
            break
        end if
```

```
        l ← l + k

    end

    if k is equal to 20 then
        l ← l - 20
    end if

end for

# If Q contains more residues than crop_size, the residues with the greatest
# distance apart in the sequence from the center residue are removed.
if (number of residues in Q) > crop_size then

    m ← Number of residues in Q

    E ← A list of size m

    for i = 1 to m do
        Ei ← absolute value of (residue index of Qi subtracted by c)
    end for

    Q ← Sort Q according to E

    Q ← Take the first crop_size residues in Q

end if

return Q
```

Multi-chain Permutated Ground Truth

After spatial cropping, we generated ground-truth candidates by considering all multi-chain permutations. We filtered out ground truth candidates where any of their segments had residues contacting less than two other segments or consisting of less than 10 amino acids. At the training stage, predicted structures were superposed using the SVDSuperimposer from Biopython (Cock, et al., 2009) , and a ground truth candidate with the smallest rms value was used for loss calculation.

Non-Globular Penalty

In addition to the square root penalty for short sequences introduced in the original training procedure of AlphaFold2 (Jumper, et al., 2021) , we introduced a penalty for nonglobular samples. This was intended to penalize entries that should be complexes but registered as monomers or incomplete complexes. These samples should have fewer contacts compared with entries that registered as complete complexes with the same number of amino acids.

If protein monomers or complexes are globular (spherical), the number of contacts becomes proportional to $N - N^{2/3}$. Therefore, we calculated the normalized number of contacts as follows:

$$C_{normalized} = C/\left(N - N^{2/3}\right)$$

where C is the number of contacts, and N is the number of amino acids.

The plot of $C_{normalized}$ for the structures used in Epoch 1 is shown in Figure S1. The penalty for nonglobular samples was calculated as follows:

$$Penalty_{nongloblar} = \frac{1.0}{\sqrt{Max(|Z|,1.0)}}$$

$$Z = \frac{C_{normalized} - \mu}{\sigma}$$

where Z is the Z-score, μ is the average of the normalized number of contacts, and σ is the standard deviation of the normalized number of contacts. μ and σ were calculated using the ground-truth structure candidates (after cropping) in Epoch 1, where μ = 5.713042053300396 and σ = 0.34169036005131537. The final loss for each sample was multiplied by both the short-sequence and non-globular penalties. We also penalized samples that had a large normalized number of contacts because they could be erroneous or have extremely rare properties.

Training Dataset

The structural data in PDB (Berman, et al., 2000) were downloaded from the pdbj (Bekker, et al., 2022) ftp site (ftp.pdbj.org::ftp_data/biounit/PDB/divided/) on 2021-11-01. The .cif.gz files containing annotations were downloaded from ftp.pdbj.org::ftp_data/structures/divided/mmCIF/ on 2021-11-01.

Entries that met the following criteria were included: 1) their release dates were before 2018-04-30, 2) resolutions were better than 9.0 Å, 3) _entity_poly.type of the all polymer entities were polypeptide(L), 4) sequence lengths were less than or equal to 1024, 5) sequence lengths without 'X' or more than 4 consecutive 'H' were longer than 20, and 6) assembly data was provided in PDB format. The chains were clustered with 40% sequence identity using the MMseqs2 (cloned from https://github.com/soedinglab/MMseqs2 on 2021-11-19) easy-cluster command (Steinegger and Soding, 2017). If the members in an entry belonged to different clusters, then those clusters were merged further. We discarded assembly entries with more than 2048 total amino acids or where the number of permutations was greater than 12. A total of 0.95 of clusters were used as the training dataset and 0.05 as the validation dataset. In each epoch, one entry was randomly selected from each cluster.

Input structures for training

We generated input structures using AlphaFold-Multimer and a customized protocol that reduced computation time and resources. MSA construction was performed using HHBLITS (v 3.3.0) (Steinegger, et al., 2019) against the uniclust30 (uniclust30_2018_08)(Mirdita, et al., 2017) database. The search was repeated up to three times using a query or the output, a3m, from the previous iteration. Iterations were stopped if the number of sequences in output a3m exceeded 200 after filtering (using hhfilter with options -cov 30 -id 90). The template database was downloaded on 2021-11-01 and the maximum template date was set to 2018-04-30. kalign (Lassmann, 2019) was not used because it may unexpectedly continue to run for a long time. Structures were predicted using five official multimer weights (v2), where they were selected randomly. The number of recycling steps was randomly selected between 0-3. If a sequence contained OX or TaxID attributes in its description line then these attributes were used for pairing.

Training Procedure

The official weight param_multimer_model_1_v2 (CC-BY 4.0, Copyright 2021 DeepMind Technologies Limited, downloaded from https://storage.googleapis.com/alphafold/alphafold_params_2022-03-02.tar on 2022-03-11) was used as the source for the fine-tuned model. The atom coordinates of the predicted structure were inserted into the prev_pos of the input feature. Template embedding was not performed in this case. The single sequence data was fed to extra_msa and msa features. Mixed precision with float16 was used. Loss was scaled dynamically so that it became as large as possible but did not produce inf or nan in gradients. The Adam optimizer with warmup_exponential_decay_schedule of optax (Babuschkin, et al., 2020) was used for the parameter update. In the first step, the model was trained for one epoch. In the second step, the model was trained for 5 epochs. The other settings are listed in Table S1.

Evaluation Procedure

Predictions of Input Structures

The official AF2 repository was cloned on 2022-09-09. The downloaded data for each database are presented in Table S2. jackhmmer produced an extremely large number of hits for 6tej, so we used only swissprot instead of swissprot+tremble for MSA generation for 6tej. One structure per model was generated without relaxation. The predicted 5 unrelaxed models were used as inputs for AFM-Refine-G.

Test Dataset

We used two benchmark sets to test the performance of AFM-Refine-G. Ghani_et_al_Benchmark2 was collected from the benchmark 2 dataset introduced by Ghani et al. (Ghani, et al., 2022) , which consisted of heterodimeric complexes released after May 2018. Yin_et_al_Hard was collected from the Table S4-S8 of the paper by Yin et al. (Yin, et al., 2022). The dataset consists of immune-related complexes or repebody-antigen complexes, that AlphaFold-Multimer mostly failed to predict. Entries released before 2018-4-30 were eliminated. 6uk4 was not used because the interface between the antibody and antigen was not found in any biological assembly.

Collection of Experimental Structures

All assembly structures in the PDB were downloaded from the pdbj ftp site (ftp.pdbj.org::ftp_data/assemblies/mmCIF/divided) on 2022-09-09. When there were multiple entries (files), the file labeled "assembly1" was used. Chains were grouped according to their description, number of contacts with other chains, and visual inspections with PyMOL (Schrodinger, 2015) to find important interfaces (e.g., antibody-antigen interface). If there were multiple groups with the same types of interface, the groups that had receptor chains appearing earlier in the parsed structure by MMCIFParser of Biopython were used. Full-length sequences in sequences were used as queries.

Evaluation Measures

We used the DockQ (Basu and Wallner, 2016) and TM-score (Zhang and Skolnick, 2004) to evaluate the quality of the predicted models. DockQ provides a balanced assessment for multimer models, which is aimed at matching the CAPRI (Duan, et al., 2020) criteria. The ranges of DockQ < 0.23, 0.23 <= DockQ < 0.49, 0.49 <= DockQ < 0.80, and 0.80 <= DockQ correspond to Incorrect, Acceptable, Medium, and High in CAPRI criteria, respectively. As mentioned above, the list of chain groups used as "receptor(chain1)" and "ligand(chain2)" when supplied to DockQ is shown in Table S3. TM-score provides a global assessment of the predicted structures. The application is usually used for the assessment of monomers, but it can also be used for multimers with the "-ter 0" option. All cases of multi-chain permutations were considered, and the highest scores were used.

Results and Discussions

The multimer structures of two datasets, Ghani_et_al_Benchmark2 and Yin_et_al_Hard, were predicted using the official AlphaFold-Multimer pipeline. The number of multimer predictions per model was set to 1. The predicted structures were then fed to AFM-Refine-G. The number of recycling steps for AFM-Refine-G was set to 10, and intermediate structures were produced. As a result, 750 structures were produced for the input, and 8250 structures were produced as the refined output. Improvements in scores (ΔDockQ and ΔTM-score) were calculated by subtracting the values of the refined structures from those of the input structures. ΔDockQ as a function of the number of recycling steps are shown in Figure 1. As a result, the improvement saturated at approximately 2-6, and we decided to analyze only the results of the number of recycling steps = 3 hereafter.

Improvement of Accuracy

Comparisons between the input and refined structural qualities are shown in Figure 2. Multiple models improved in quality with a ΔDockQ (Refined-Input) > 0.05 with 14 out of 85 and 101 out of 665 for Ghani_et_al_Benchmark2 and Yin_et_al_Hard, respectively. Furthermore, only a small number of models decreased in quality with ΔDockQ < -0.05, there were 0 and 14 for Ghani_et_al_Benchmark2 and Yin_et_al_Hard, respectively. The numbers of acceptable or higher quality predictions in CAPRI criteria were the same or increased (Table 1, Table S4). This indicates that the refinement was generally successful. However, there was a severe degradation of the TM-score of some samples in Yin_et_al_Hard (Figure 2d). The numbers of models with ΔTM-score > 0.05 were 5 and 18 for Ghani_et_al_Benchmark2 and Yin_et_al_Hard, respectively. Although no models had ΔTM-scores < -0.05 in Ghani_et_al_Benchmark2, 36 did in Yin_et_al_Hard. Of these 36 structures, 26 out of 36 structures were TCR-peptide-MHC complexes. We conducted a structural alignment of ground truth, input, and refined structures of 26 TCR-peptide-MHC cases using the TCR chains of 6r2l as templates (Figure S2). In the input structures, the global orientations of TCR and MHC chains were accurately predicted to some extent, but most of the positions of the peptides were not. In contrast, in the refined structures, positions of peptides were incorrect, too, and many of the MHC chains were also placed wrongly; as seen in 7n1f_model_5.refined, some of them showed aggregated forms. As illustrated in the ground truth structures in Figure S2, the global orientations of TCR and MHC are consistent across different entries, indicating that they can be learned during the training of models. While the TM-score measures global similarity, DockQ is a metric that also takes into account the prediction accuracy of interfaces that are of particular interest (e.g., the interface between the TCR and the peptide). The difference in degradation between DockQ and TM-score is likely due to differences in their scoring

schemes. It is difficult to provide a clear answer to why AFM-Refine-G could not generate outputs that follow the global orientation. We should note that two technical factors may have contributed to this issue. Firstly, AFM-Refine-G employed a smaller crop_size (300) than the original AlphaFold-Multimer (384). Secondly, AFM-Refine-G was trained without incorporating chain centre-of-mass loss terms. However, it remains a matter of debate whether the global orientation of subunits should be maintained even if the prediction of interfaces that are of particular interest is inaccurate.

Predicted TM-score

AlphaFold-Multimer outputs structures and the confidence metric, weighted sum of interface predicted TM-score and predicted TM-score (iptm*0.8+ptm*0.2), to indicate the qualities of predicted structures and is used to rank them (Evans, et al., 2022). We analyzed whether this metric is still available with AFM-Refine-G; estimating the quality of the predicted structures is possible without MSAs. The scatter plot of DockQ as a function of (iptm*0.8+ptm*0.2) is shown in Figure 4. The R^2 etween the metric and the DockQ or TM-score of the input and refined structures are listed in Table 2. The R^2 for refined structures was competitive, although slightly lower than that of input structures for Ghani_et_al_Benchmark2, slightly higher than that of the input structures for Yin_et_al_Hard. Therefore, we concluded that estimating the quality of the predicted structures is possible without MSA.

Atom Clashes

During the analysis, we found that there were severe atom clashes in the refined structures, which were rarely seen in the input structures. We counted the CA clashes (classified as when the distance between two CAs was less than 2.0 Å) in all predicted structures. We then analyzed the relationships between CA clashes and the structure quality. The worst input structure had four CA clashes, and the worst refined structure had 51 CA clashes (Figure 4). The refined structures with a large number of clashes had a low DockQ in terms of refined and input structures (Figure 4B and C). 26 out of 750 structures had more than 10 CA clashes, and all had a DockQ less than 0.23, placing them in the 'Incorrect' prediction in CAPRI criteria. Several factors could cause this problem: 1) Intrinsic phenomena in our model. Because AFM-Refine-G was trained with samples biased toward globular proteins, the model might try to create globular structures if other good structures are not easily found. 2) Insufficient implementation of loss terms. We did not implement the updated loss terms which was not used in the first version of AlphaFold-Multimer, for example, the chain center-of-mass loss and clash loss terms (Evans, et al., 2022). 3) Differences in optimal loss weights. The weight for the structural violation loss of AlphaFold-Multimer (0.03) was much smaller than that of AlphaFold2 (1.0) (Evans, et al., 2022) . It is possible that this value was not optimal for our model.

Conclusions

We presented AFM-Refine-G, a fine-tuned version of AlphaFold-Multimer, which refines multimeric structures without MSA. Its overall performance was good; however, approximately 70% of targets in the dataset consisting of immune-related complexes and repebody-antigen complexes were still judged as incorrect. Fine-tuning for docking, rather than refinement, may be the next promising direction. During writing this paper, AlpaFold-Multimer version 2.3.0 were published. Further assessment of the fine-tuning of updated models and the revalidation of the losses will also be a future task.

Acknowledgements

We are grateful to DeepMind Technologies Limited for providing the code and parameters of AlphaFold2 and AlphaFold-Multimer in the generous licenses, as well as the developers and maintainers of the software and databases used in the AlphaFold2 and AlphaFold-Multimer pipeline and for this benchmark.

Conflict of Interest

The author declare no competing interests.

Tables and Figures

		Input		Refined	
		T1*	**T5**	**T1**	**T5**
Ghani_et_al_Benchmark2	**Incorrect**	4	4	4	3
	Acceptable	1	1	1	2
	Medium	5	5	5	5
	High	7	7	7	7
Yin_et_al_Hard	**Incorrect**	109	101	108	97
	Acceptable	15	21	13	22
	Medium	8	10	9	11
	High	1	1	3	3

Table 1. The number of targets that predicted structures met corresponding CAPRI criteria.

* T1 indicates the result of the top 1 model which had the highest confidence metric (iptm*0.8+ ptm*0.2). T5 indicates the result of the models which had highest DockQ among all 5 models.

	AlphaFold-Multimer		AFM-Refine-G	
	DockQ	TM-score	DockQ	TM-score
Ghani_et_al_Benchmark2	0.7819	0.7323	0.7597	0.7121
Yin_et_al_Hard	0.1237	0.1963	0.2284	0.2941

Table 2. The R^2 between the two evaluation measures and the confidence metric (iptm*0.8+ ptm*0.2) produced by AlphaFold-Multimer and AFM-Refine-G.

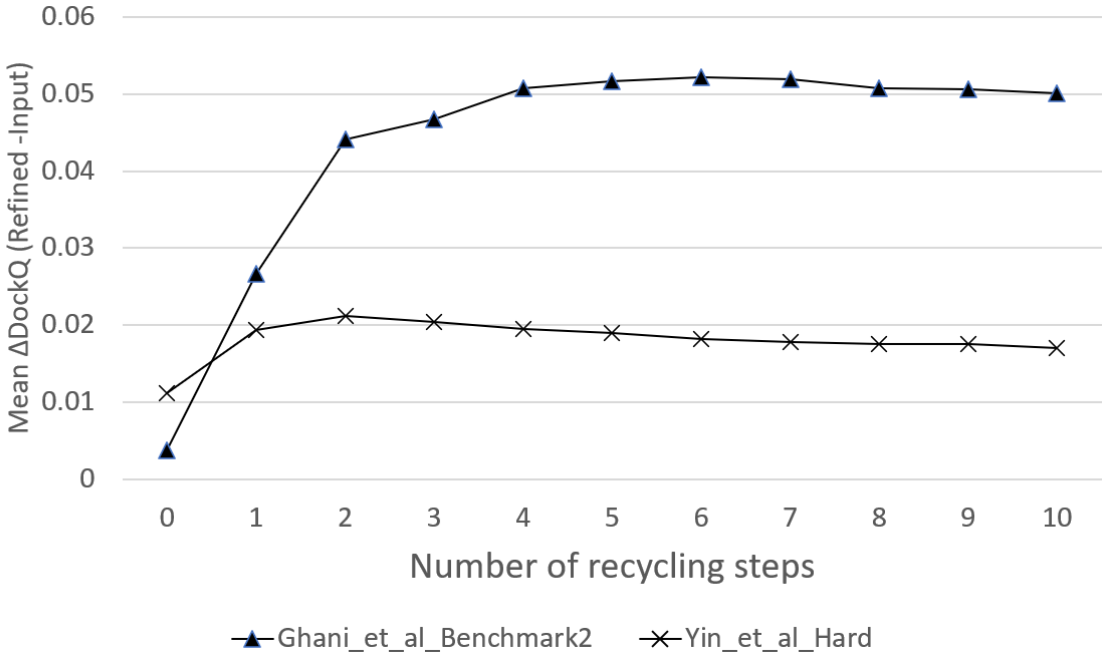


Figure 1. The mean DockQ as a function of the number of recycling steps. The ΔDockQ, DockQ of refined structures subtracted by that of input structures, calculated for all five structures per one target were averaged. A triangle indicates the result of Ghani_et_al_Benchmark2 (85 structures) and a cross indicates the result of Yin_et_al_Hard (665 structures).

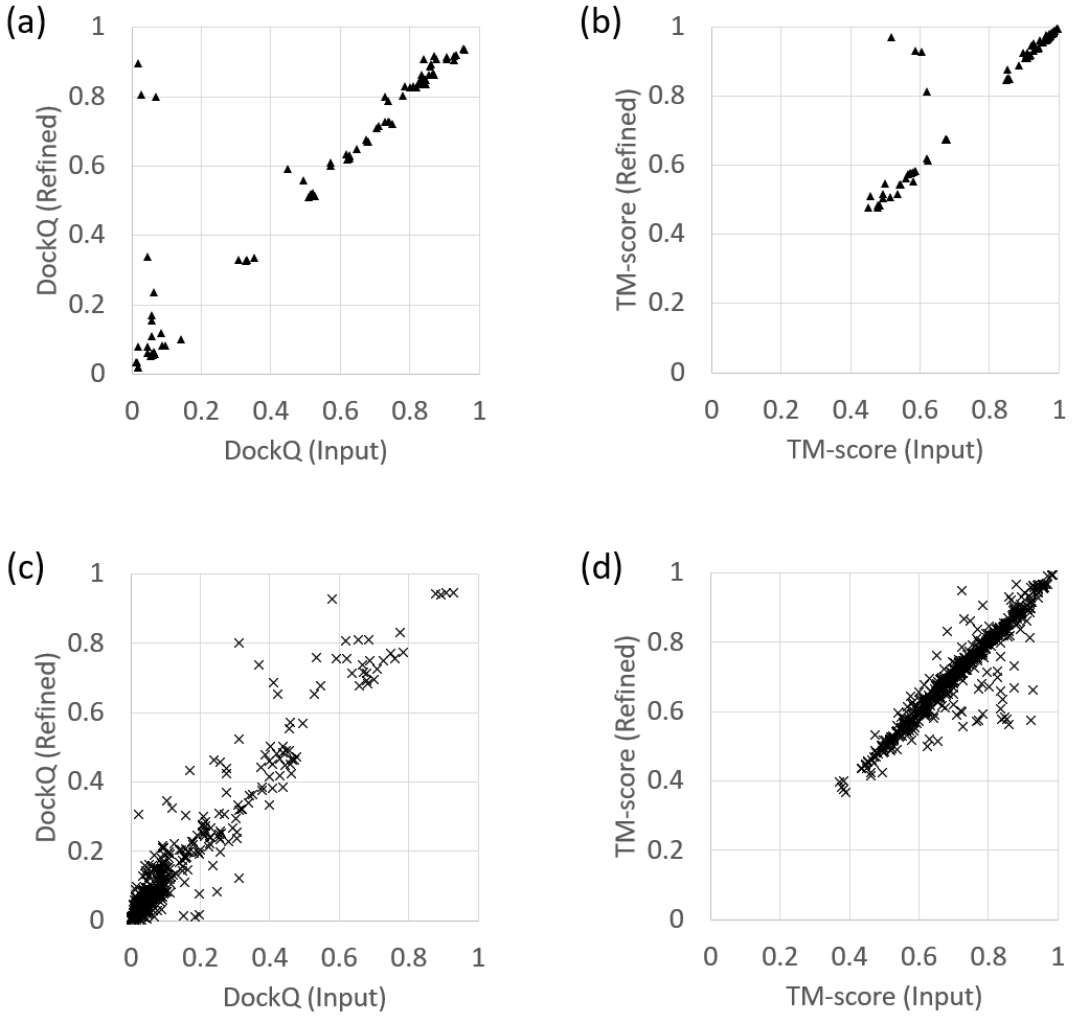


Figure 2. Comparison between qualities of input and refined structures plotting the scores of all five models for one target. Panel (a) presents a comparison of the DockQ of Ghani_et_al_Benchmark2 dataset. Panel (b) presents a comparison of the DockQ of Yin_et al_Hard dataset. Panel (c) presents a comparison of the TM-score of Ghani_et_al_Benchmark2 dataset, and panel (d) presents a comparison of the TM-score of Yin_et_al_Hard dataset.

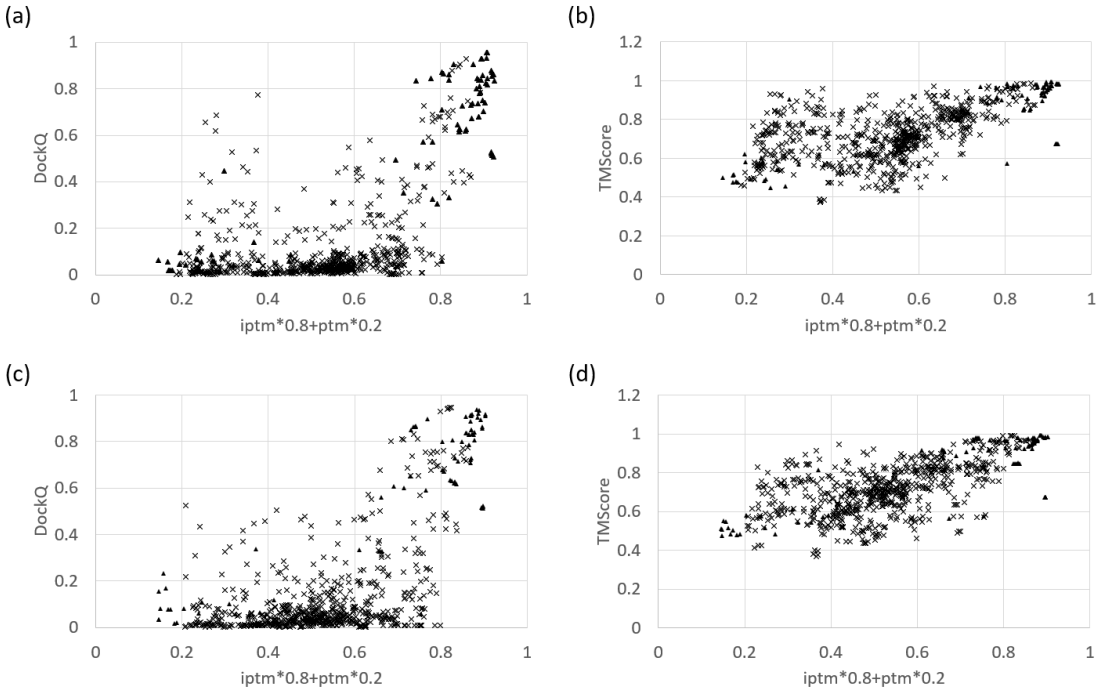
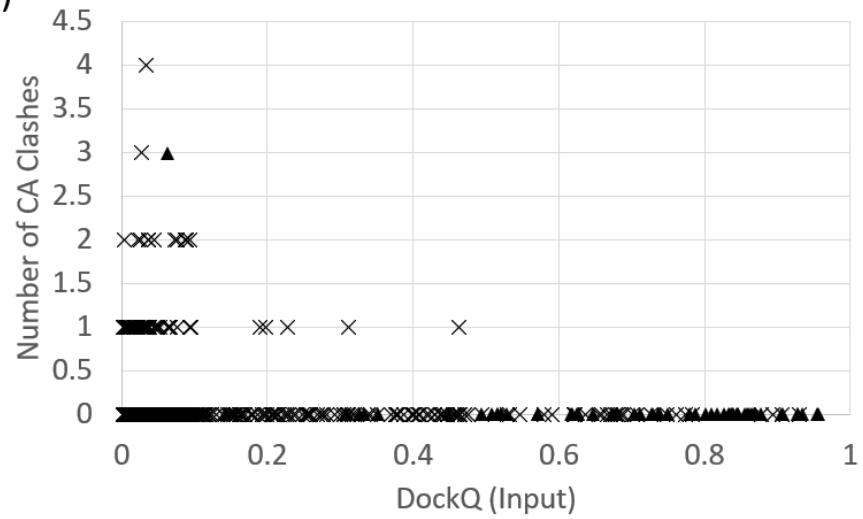
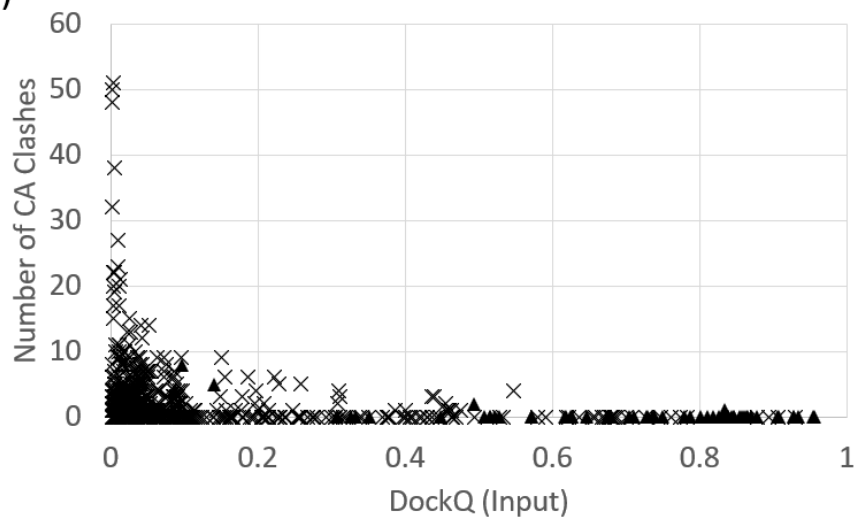


Figure 3. Correlation between the evaluation measures DockQ and TM-score and the confidence metric of AlphaFold-Multimer. Correlation between evaluation measures and confidence metric. The X-axis represents the weighted ipTM and pTM (iptm*0.8+ptm*0.2) utilized by AlphaFold-Multimer to rank its predicted models, while the Y-axis represents the DockQ or TM-score assessed using the predicted and experimental structures. Subfigure (a) shows the correlation between DockQ and iptm*0.8+ptm*0.2 of the input structures, while subfigure (b) shows the correlation between TM-score and iptm*0.8+ptm*0.2 of the input structures. Subfigure (c) displays the correlation between DockQ and iptm*0.8+ptm*0.2 of the refined structures, and subfigure (d) presents the correlation between TM-score and iptm*0.8+ptm*0.2 of the refined structures. The results for the Ghani_et_al_Benchmark2 dataset, comprising 85 structures, are indicated by triangles, while the results for the Yin_et_al_Hard dataset, comprising 665 structures, are represented by crosses. The R^2 for each plot are provided in Table 2.

(a)


Number of CA Clashes
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
DockQ (Input)

(b)


Number of CA Clashes
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1
DockQ (Input)

(c)

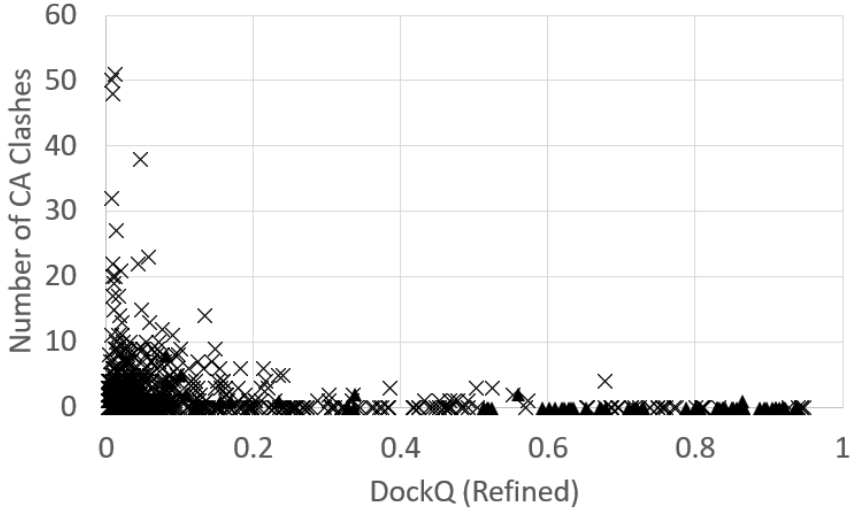

Number of CA Clashes
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1
DockQ (Refined)

Figure 4. Number of CA atom clashes in the predicted models as a function of DockQ. A CA clash was defined as an instance where the distance between two CAs was less than 2.0 Å. Subfigure (a) shows the number of clashes in the input structures as a function of the DockQ of the input structures. Subfigure (b) displays the number of clashes in the refined structures as a function of the DockQ of their corresponding input structures, and subfigure (c) presents the number of clashes in the refined structures as a function of the DockQ of the refined structures themselves.

References

Anishchenko, I., *et al.* Protein tertiary structure prediction and refinement using deep learning and Rosetta in CASP14. *Proteins* 2021;89(12):1722-1733.

Babuschkin, I., *et al.* 2020. The DeepMind JAX Ecosystem

Basu, S. and Wallner, B. DockQ: A Quality Measure for Protein-Protein Docking Models. *PLoS One* 2016;11(8):e0161879.

Bekker, G.J., *et al.* Protein Data Bank Japan: Celebrating our 20th anniversary during a global pandemic as the Asian hub of three dimensional macromolecular structural data. *Protein Sci* 2022;31(1):173-186.

Berman, H.M., *et al.* The Protein Data Bank. *Nucleic Acids Res* 2000;28(1):235-242.

Chandonia, J.M., *et al.* SCOPe: improvements to the structural classification of proteins - extended database to facilitate variant interpretation and machine learning. *Nucleic Acids Res* 2022;50(D1):D553-D559.

Cock, P.J., *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 2009;25(11):1422-1423.

Duan, R., *et al.* Performance of human and server prediction in CAPRI rounds 38-45. *Proteins* 2020;88(8):1110-1120.

Evans, R., *et al.* Protein complex prediction with AlphaFold-Multimer. *bioRxiv* 2022:2021.2010.2004.463034.

Feinauer, C., *et al.* Inter-Protein Sequence Co-Evolution Predicts Known Physical Interactions in Bacterial Ribosomes and the Trp Operon. *PLoS One* 2016;11(2):e0149166.

Gao, M., *et al.* AF2Complex predicts direct physical interactions in multimeric proteins with deep learning. *Nat Commun* 2022;13(1):1744.

Ghani, U., *et al.* Improved Docking of Protein Models by a Combination of Alphafold2 and ClusPro. *bioRxiv* 2022:2021.2009.2007.459290.

Humphreys, I.R., *et al.* Computed structures of core eukaryotic protein complexes. *Science* 2021;374(6573):eabm4805.

Jumper, J., *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;596(7873):583-589.

Jumper, J., *et al.* Applying and improving AlphaFold at CASP14. *Proteins* 2021;89(12):1711-1721.

Lassmann, T. Kalign 3: multiple sequence alignment of large data sets. *Bioinformatics* 2019.

Lensink, M.F., Mendez, R. and Wodak, S.J. Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins* 2007;69(4):704-718.

Mirdita, M., *et al.* Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res* 2017;45(D1):D170-D176.

Ovchinnikov, S., Kamisetty, H. and Baker, D. Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *Elife* 2014;3:e02030.

Pereira, J., *et al.* High-accuracy protein structure prediction in CASP14. *Proteins* 2021;89(12):1687-1699.

Roney, J.P. and Ovchinnikov, S. State-of-the-art estimation of protein model accuracy using AlphaFold. *bioRxiv* 2022:2022.2003.2011.484043.

Rost, B. and Sander, C. Progress of 1D protein structure prediction at last. *Proteins* 1995;23(3):295-300.

Schrodinger, LLC. The PyMOL Molecular Graphics System, Version 1.8. In.; 2015.

Sillitoe, I., *et al.* CATH: increased structural coverage of functional space. *Nucleic Acids Res* 2021;49(D1):D266-D273.

Steinegger, M., *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* 2019;20(1):473.

Steinegger, M. and Soding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* 2017;35(11):1026-1028.

Wu, R., *et al.* High-resolution de novo structure prediction from primary sequence. *bioRxiv* 2022:2022.2007.2021.500999.

Yin, R., *et al.* Benchmarking AlphaFold for protein complex modeling reveals accuracy determinants. *Protein Sci* 2022;31(8):e4379.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* 2004;57(4):702-710.

Zheng, W., *et al.* Protein structure prediction using deep learning distance and hydrogen-bonding restraints in CASP14. *Proteins* 2021;89(12):1734-1751.