nature portfolio

Peer Review File

DPFunc: Accurately predicting protein function via deep learning with domain-guided structure information

Corresponding Author: Professor Min Li

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Due to a disagreement not affecting the scientific content of the manuscript, the comments of reviewer #1 have been redacted by the editors with the reviewer's consent.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

[redacted]

Reviewer #2

(Remarks to the Author)
The authors present DeepDoguest, a deep learning model featuring domain-guided structural information for protein function prediction. Through a clever combination of embeddings, they successfully integrate InterPro domains, residue level features calculated using ESM-1b, and structural information from PDB or AlphaFold 2. The authors present the model clearly and soundly, and the intuition behind architectural design are provided very elegantly. Also, the available code is of good quality, and it is easy to run for experienced programmers.

Major Comments

1. Selection of SOTA models for the comparison

Based on the CAFA challenge for the past several years, the
best performing method from the challenge is missing from this comparison. NetGO 3.0 [Wang et al., 2023: https://doi.org/10.1016/j.gpb.2023.04.001] is the current state of the art, far surpassing competitors included in the comparison.

Therefore, NetGO 3.0 needs to be included and compared against the same dataset to put DeepDoguest in a more complete context in terms of the state of the art methods.

2. Details on how to get the active sites prediction

In section 2.6, the authors present a very exciting capability of DeepDoguest:The identification of individual residues that contribute to specific functional annotations, specifially for enzyme reactions. It seems that the importance of the specific residues comes from the attention mechanism, but it is not clear. The authors need to explain this in more detail.

3. Controlling for homology in the PDB dataset

The PDB dataset used to test the performance seems to be the same one used by Gligorijević et al. in the DeepFRI paper. In that paper, they show a remarkable portion of the performance is associated with elements in the test set with high similarity to elements in the training set. Could the authors also control for this similarity?

Minor comments

The authors provide the code, as well as a good methods section to explain the
model training and hyperparameters, but some details are missing:
• how many epochs are required to train the model?
• The authors specified the hardware used in training, but it would be informative to understand the training time and memory load.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
In “Accurately predicting protein function via deep learning with domain-guided structure information”, Wang et al. describe a

new deep-learning-based method for prediction of protein function. The approach, named DeepDoguest, incorporates several current ideas in protein modelling, and performs on par in benchmark settings.

It does, however, appear that many of the improvements are marginal rather than substantial. The authors should carry out bootstrapping or a similar subsampling-based approach to determine whether their findings actually are not only significant but also robust.

From the description it is a little unclear whether structure is required in all parts of the model, or just some. Domain annotation is well possible based on sequence. For those parts of a protein that are not well-structured - according to AlphaFold - will they be penalised for this lack of structure? Is pLDDT taken into account, or other metrics of prediction reliability?

In Fig. 3, the most striking change is for CC and BP, while all others are perhaps statistically significant but virtually unchanged. What is different about the CC and BP sets or cases here that leads to this striking improvement when adding domain annotations?

Some of the described observations fit well to the paradigm of function being more conserved than structure, and structure being more conserved than function, e.g. in Fig. 4. The illustration of such a case in itself is not convincing though - the authors should identify “negative controls” that have similar levels of sequence identity where DeepDoguest clearly predicts different functions.

The test case Bacillus subtilis is surprising - this organism is well-annotated and was sequenced in 1997 (Kunst et al. 1997) - could the authors elaborate which specific strain they consider newly sequenced?

For the prediction of functional or active sites, the authors need to use established benchmark sets and compare to SOTA in the field, e.g. (Cagiada et al. 2023). As a devil’s advocate looking at the current Fig. 5, one might suggest it simply predicts all histidines to be catalytically active…

Given the broad audience of the journal, it would be good if the authors described exactly what “prediction of protein function” can do, and perhaps also outline what it cannot do yet. In other words, how practically useful are GO annotations to researchers interested in individual proteins? For those performing large-scale analyses and classifications? Ideally the authors will also consider to what extent it even makes sense to predict the function of a protein in isolation, given that they have evolved for function in their cellular context and that many proteins have “moonlighting” functions that can sometimes be drastically different from the canonical, annotate functon(Jeffery 2018)

To summarise, the paper presents several interesting ideas that are in line with longstanding paradigms - the testing and illustration of successful cases needs to be more thoroughly checked though.

Minor comments:
P2 “medical text”, unusual term, perhaps this refers to literature curation?
P7 “It can be obtained that DeepDoguest gets significant improvements than other methods,”, rephrase

Cagiada, Matteo, Sandro Bottaro, Søren Lindemose, Signe M. Schenstrøm, Amelie Stein, Rasmus Hartmann-Petersen, and Kresten Lindorff-Larsen. 2023. “Discovering Functionally Important Sites in Proteins.” Nature Communications 14 (1): 4175.
Jeffery, Constance J. 2018. “Protein Moonlighting: What Is It, and Why Is It Important?” Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 373 (1738). https://doi.org/10.1098/rstb.2016.0523.
Kunst, F., N. Ogasawara, I. Moszer, A. M. Albertini, G. Alloni, V. Azevedo, M. G. Bertero, et al. 1997. “The Complete Genome Sequence of the Gram-Positive Bacterium Bacillus Subtilis.” Nature 390 (6657): 249–56.

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #2

(Remarks to the Author)
The authors have address all my comments very well. The paper is now considerably better than the initial submission. I have no more concerns.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
The authors put a lot of effort into responding to all reviewers' comments and improve the manuscript.
I have only 2 remaining minor comments:
- the difference between the lighter and darker colours in the new Fig. 4H is difficult to see on some screens. I would encourage the authors to increase the contrast, or perhaps use gray for the non-aligned positions.
- it's odd to call a method by the author's first name, Matteo, unless it were called that in the original paper

(Remarks on code availability)

Response letter

On behalf of all the contributing authors, we would like to express our sincere appreciation for reviewers' constructive comments and helpful suggestions regarding our article entitled "*Accurately predicting protein function via deep learning with domain-guided structure information*". These comments have significantly improved the presentation of our article. In response to the feedback from reviewers, we have made substantial revisions to our manuscript and have provided more experimental results and cases to support the robust of our method. All changes can be seen in the revised manuscript with yellow highlighting. We hope that this detailed response resolves the concerns of reviewers.

Following the Reviewer's suggestion, we have renamed our method from DeepDoguest to DPFunc. In the following pages, we give our detailed point-by-point responses to the review comments. We hope that our revised version meets with your approval.

Reviewer #1 (Remarks to the Author):

[redacted]

Response. We appreciate the summary from the Reviewer of our work and thank the Reviewer for the positive comments. We provide the point-by-point responses below.

[redacted]

Response. Thanks for the suggestions. We have cited and tried to compare our method with I-TASSER-MTD[1]. Notably, I-TASSER-MTD is an effective but time-consuming tool for modeling the structures of multi-domain proteins. In I-TASSER-MTD, the structure-based method COFACTOR[2,3] is integrated to predict protein functions. We tested I-TASSER-MTD for one protein with 230 residues on the web-server and locally. This process took around ten days on the web-server and about three days locally, separately. For longer proteins, it will take even more time. Given our dataset comprising over 1000 proteins, evaluate the fully automated pipeline of I-TASSER-MTD on the dataset posed challenges. Consequently, we use the protein structure predicted by AlphaFold2 (also as the input of our model) to replace the modeling structures in I-TASSER-MTD, and then uses COFACTOR in I-TASSER-MTD to annotate protein functions. The whole process costs around a month. The comparison results are shown in Table R1.

Table R1 (Table S1). Predictive performance of DPFunc compared with two web-servers on the large-scale dataset

	MF		CC		BP	
	Fmax	AUPR	Fmax	AUPR	Fmax	AUPR
COFACTOR $_{AF2}$	0.3768	0.2601	0.4394	0.2451	0.3012	0.1513
NetGO 3.0 $_{server}$	0.6308	0.5837	0.6363	0.6024	**0.4875**	0.4188
DPFunc $_{retrain}$	**0.6346**	**0.6537**	**0.6547**	**0.6887**	0.4534	**0.4290**

*$_{AF2}$ indicates that COFACTOR predict protein functions based on the structure predicted by AlphaFold2.

*$_{server}$ indicates that the results are from the web server of NetGO 3.0

*$_{retrain}$ indicates that our model is retrained on the same dataset with NetGO 3.0.

From Table R1, it can be obtained that our method outperforms COFACTOR consistently in protein function prediction on MF, CC, and BP. It needs to be mentioned that I-TASSER-MTD is proposed mainly for modeling multi-domain protein structures rather than predicting protein functions. In the revised manuscript, we have added the following paragraph (in Section 2.2) to summary the results:

"Moreover, we choose two additional web-servers as competitors, NetGO3.0[4] and COFACTOR[2,3], where NetGO3.0 is the current state-of-the-art method in the CAFA[5] challenge and COFACTOR is an effective structure-based tool for predicting protein functions as a component of I-TASSER-MTD[1] in the CASP[6] challenge."

"Similar conclusions can be drawn from Table S1. DPFunc surpasses the other two web-servers, NetGO3.0 and COFACTOR, in the vast majority of cases, except for Fmax in BP. These comparison results further prove the ability of DPFunc in protein function prediction."

References

1. Zhou X, Zheng W, Li Y, et al. I-TASSER-MTD: a deep-learning-based platform for multi-domain protein structure and function prediction[J]. Nature Protocols, 2022, 17(10): 2326-2353.
2. Zhang C, Freddolino P L, Zhang Y. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information[J]. Nucleic acids research, 2017, 45(W1): W291-W299.
3. Roy A, Yang J, Zhang Y. COFACTOR: an accurate comparative algorithm for structure-based protein function annotation[J]. Nucleic acids research, 2012, 40(W1): W471-W477.
4. Wang S, You R, Liu Y, et al. NetGO 3.0: protein language model improves large-scale functional annotations[J]. Genomics, Proteomics & Bioinformatics, 2023, 21(2): 349-358.
5. Zhou N, Jiang Y, Bergquist T R, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens[J]. Genome biology, 2019, 20: 1-23.
6. Kryshtafovych A, Schwede T, Topf M, et al. Critical assessment of methods of protein structure prediction (CASP)—Round XIV[J]. Proteins: Structure, Function,

and Bioinformatics, 2021, 89(12): 1607-1617.

[redacted]

Response. We are grateful for this important comment. In the revised manuscript, we have cited more relevant scientific literature[1-2], including PMID: 12840035 and 30733291. Additionally, we have simplified the description of existing methods and highlighted the domain architecture to emphasize the core idea of our work.

References

1. Caetano-Anollés, G. & Caetano-Anollés, D. An evolutionarily structured universe of protein architecture. Genome research 13, 1563–1571 (2003).
2. Yu, L. et al. Grammar of protein domain architectures. Proceedings of the National Academy of Sciences 116, 3636–3645 (2019).

[redacted]

Response. We thank the Reviewer for pointing out the ambiguities in the manuscript. We made this statement mainly because AlphaFold database provides broad coverage of UniProt. However, for some species or special proteins, their predicted structures are not stored in the database. Therefore, we predicted the structures of these proteins using AlphaFold2. Notably, the newer version of AlphaFold database[1] have been proposed in 2024, covering a wider range of proteins than the previous version[2] we used. To clarify, we rewrote the sentences as follows (in Section 4.1):

"For several proteins whose structures could not be downloaded directly from the database, we use AlphaFold2 to predict their structures locally."

Additionally, the numbers for Train/Valid/Test in Table 3 indicate the protein entries, and the numbers of GO terms are the unique terms across the corresponding training entries. We thank the Reviewer for pointing out the ambiguities again. To provide a clear description, we have listed the percentage of all protein entries per class in Table R2.

Table R2 (Table 3). The statistic information of two datasets

Dataset		MF		CC		BP	
PDB dataset	Train	24837	(80.2%)	11162	(70.4%)	23386	(79.5%)
	Valid	2746	(8.9%)	1296	(8.2%)	2624	(8.9%)
	Test	3399	(10.9%)	3400	(21.4%)	3400	(11.6%)
	All	30982	(100%)	15858	(100%)	29410	(100%)
CAFA dataset	Train	31463	(96.7%)	42467	(96.4%)	47333	(96.3%)
	Valid	682	(2.1%)	711	(1.6%)	767	(1.6%)
	Test	401	(1.2%)	877	(2.0%)	1039	(2.1%)
	All	32546	(100%)	44055	(100%)	49139	(100%)

In the revised manuscript, we have added the following descriptions (in Section 4.1) to clarify this point:

"The statistic information of proteins is illustrated in Table 3. Specifically, the PDB dataset contains 36,408 proteins and 2,748 GO terms, including 488 (17.8%) MF GO terms, 320 (11.6%) CC GO terms, and 1,940 (70.6%) BP GO terms."

"Finally, as shown in Table 3, there are 59,397 proteins and 28,252 GO terms, including 6,086 (21.5%) MF GO terms, 2,492 (8.8%) CC GO terms, and 19,674 (69.6%) BP GO terms."

References

1. Varadi M, Bertoni D, Magana P, et al. AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences[J]. Nucleic Acids Research, 2024, 52(D1): D368-D375.
2. Varadi M, Anyango S, Deshpande M, et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models[J]. Nucleic acids research, 2022, 50(D1): D439-D444.

[redacted]

Response. Thank you for pointing this out. For the domain information, it is domain-level encoding. Specifically, each protein sequence can be scanned and several domains can be detected, where each domain can be represented as a unique entry, denoted as $IPR \in (0,1)^{1*m}$, where m is the total number of domain entries. Then, these domain entries are fed into an embedding layer to generate domain-level features $f_{domain} \in R^{m*d}$, and then the features of these detected domain entries are summed as protein-level domain features $H \in R^{1*d}$. Subsequently, the protein-level domain features $H \in R^{1*d}$ are multiplied by the residue-level features $X^{final} \in R^{L*d}$ (learned from the contact maps) to calculate the importance of each residue. Specifically, $H \in R^{1*d}$ is multiplied by each residue feature $x^{final} \in R^{1*d}$, and a Softmax layer is used to calculate the importance of each residue. We have added the following details into the Section 2.1 to clarify this point:

"Specifically, these domain entries are fed into an embedding layer to generate domain-level dense representations that capture their unique characteristics, and then summed as protein-level domain information. To assess the importance of different residues, inspired by the transformer architecture, a novel attention mechanism is introduced to interweave the protein-level domain features and residue-level features, which detects the importance of each residue."

References

1. Jones P, Binns D, Chang H Y, et al. InterProScan 5: genome-scale protein function classification[J]. Bioinformatics, 2014, 30(9): 1236-1240.

[redacted]

Response. We agree that the comparison between our method and other SOTA methods is a bit of numerology, and all the comparisons in Figure 2 are intended to validate the robustness of our method in various respects, including difficult proteins with low sequence identities (Figure 2a), informative functions (Figure 2b-e) and functions with deeper depths (Figure 2f). Additionally, we provide more specific cases in later sections to prove the ability of our model. For instance, in Section 2.4, two cases are illustrated to show that our method effectively distinguishes between structures motifs and sequence identities. Figure 4(b-g) shows that it works well on the proteins with similar structures but dissimilar sequences. To further validate this case, we have added new cases with three proteins (5JZV-A, 3WG8-A and 5Z9R-A), as shown in Figure R1 and R2, Table R3 and R4.

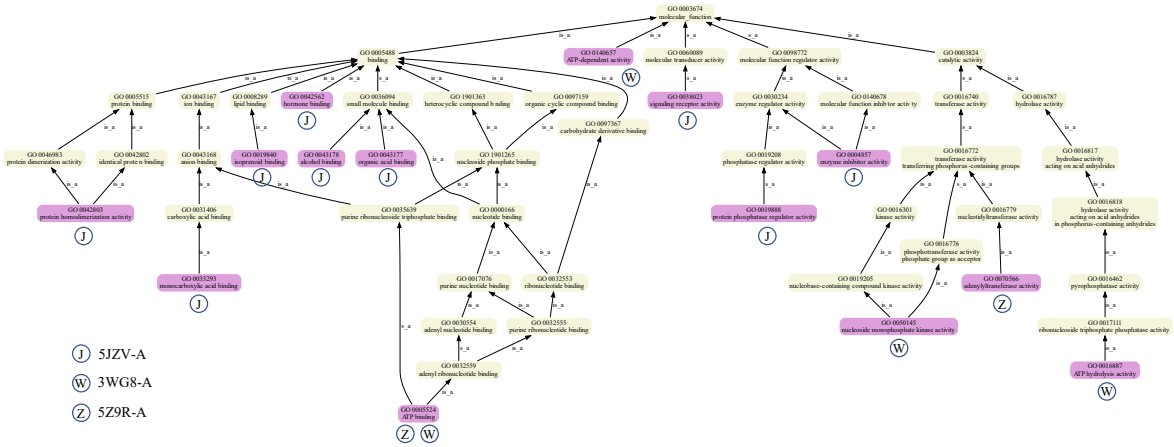

Figure R1 (Figure S4). The GO terms of three proteins (PDB ID: 5JZV-A, 3WG8-A, 5Z9R-A). The purple blocks are the deepest child GO terms, where "J", "W" and "Z" indicates 5JZV-A, 3WG8 and 5Z9R-A, respectively, and all of these GO terms are predicted by DPFunc accurately.

From Figure R1, we can see that our proposed method DPFunc accurately predicts the functions of three proteins (5JZV-A, 3WG8-A and 5Z9R-A) with high sequence identities but low TM-scores. The structure alignment results between 5JZV-A, 3WG8-

A and 5Z9R-A are shown in Figure R2. The sequence identities and common functions between these proteins calculated by BLAST are shown in Table R3 and R4.

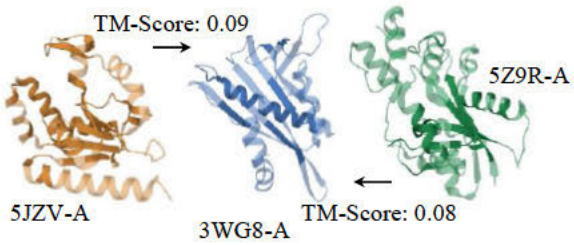

Figure R2 (Figure 4h). The structure alignment results between 5JZV-A, 3WG8-A and 5Z9R-A. Dark colors in each protein represent residues that are aligned and light colors represent residues that are not aligned.

Table R3 (Table S6). The sequence identities between proteins calculated by BLAST.

	5JZV-A	3WG8-A	5Z9R-A
5JZV-A	100%	87.80%	89.74%
3WG8-A	87.80%	100%	90.24%
5Z9R-A	89.74%	90.24%	100%

Table R4 (Table S7). The number of common functions between proteins.

	5JZV-A	3WG8-A	5Z9R-A
5JZV-A	31/31	5/31	20/31
3WG8-A	5/24	24/24	5/24
5Z9R-A	20/22	5/22	22/22

On the other hand, for the comments in CC, we have added more descriptions to highlight the advantages of DPFunc as follows:

"The results are shown in Figure 2(a), DPFunc consistently outperforms other methods in nearly all cases, except for the 50% threshold in BP, where it demonstrates comparable performance to ATGO+. Notably, the improvements of DPFunc are still stable as the identity threshold increases. This advantage is more pronounced in CC, where the rankings of ATGO+ and DeepGraphGO change with identities. This result persists when compared to all other SOTA methods (see Supplementary Figure S2)."

References

1. Cagiada M, Bottaro S, Lindemose S, et al. Discovering functionally important sites in proteins[J]. Nature communications, 2023, 14(1): 4175.

[redacted]

Response. Thanks for pointing this out. The review article suggested by the Reviewer seems to be missing. It is true that huge amounts of methods have been proposed to predict protein functions from sequences. We have cited several related review articles[1,2] and corrected it.

References

1. Whisstock J C, Lesk A M. Prediction of protein function from protein sequence and structure[J]. Quarterly reviews of biophysics, 2003, 36(3): 307-340.
2. Wang W, Shuai Y, Yang Q, et al. A comprehensive computational benchmark for evaluating deep learning-based protein function prediction approaches[J]. Briefings in Bioinformatics, 2024, 25(2): bbae050.

[redacted]

Response. We are very grateful for your suggestions. We find that there is some potential for enhancement of our method in identifying important active sites, especially on the predicted protein structures with low confidence (pLDDT). For example, we choose two proteins, Q9Y606 and Q9WU56. As illustrated in Figure R3, the attention mechanism of our method detects four active sites of Q9Y606 (GLY-2, THR-16, GLY-20, ASP-146) and three active sites of Q9WU56 (GLY-2, THR-18, ASP-142), where only ASP-146 in Q9Y606 and ASP-142 Q9WU56 are known as active sites. The other sites in the predicted structures are all with low confidence.

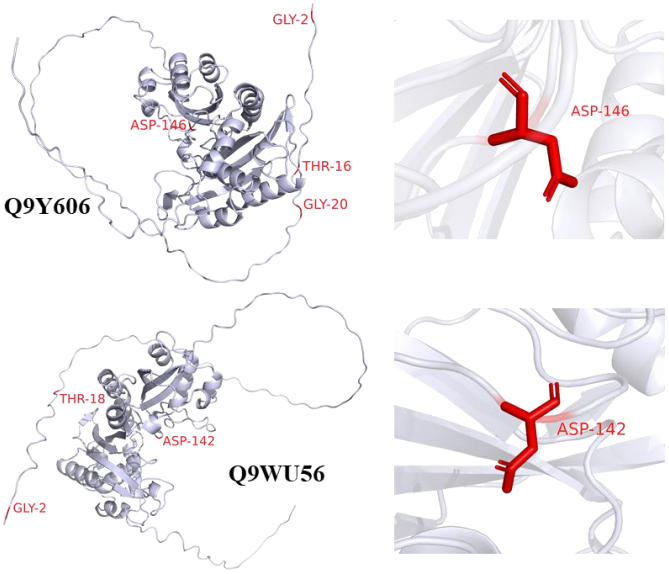

Figure R3 (Figure S7). The detected residues by DPFunc and the validated active sites of two pseudouridylate synthases. The red positions shown in the structures are the key residues detected by DPFunc. The residues in the detailed graphs are the active sites that have been validated (ASP-142 for Q9Y606, ASP-146 for Q9WU56)

As described in AlphaFold2, the low confidence (pLDDT<50) is a reasonably strong predictor of disorder, suggesting such a region is either unstructured in physiological conditions or only structured as part of a complex. Consequently, our method finally filters the residues with pLDDT<50, which reduces the number of detected residues but improve the confidence of the results.

Consequently, it is also a challenge to consider the disorder regions of proteins in future

models. To clarify this point, we have added the following paragraph (in Section 2.6):

"Notably, although DPFunc detects significant active sites effectively, finding active sites in disordered regions remains a challenge that may be further explored in future models (see Figure S7)."

[redacted]

Response. Thanks for your suggestion. We have renamed our method to DPFunc, pronounced by D-P-Func.

[redacted]

Response. Thanks for pointing this out. This is a misdescription and we have corrected it as follows:

"Further case studies on bacteria demonstrate that our model performs well on proteins with only known sequences."

"Additionally, Figure 4(j) illustrates the PR curve of these two methods, which demonstrates that DPFunc has a great improvement in terms of AUPR, proving the potential of DPFunc for annotating bacteria."

[redacted]

Response. Thanks for your suggestions. We have cited the relevant article.

[redacted]

Response. Thanks for pointing this out. We have corrected it and other typos that we found accordingly.

[redacted]

Response. We are very grateful for the Reviewer's careful reading of our manuscript. We have corrected it.

[redacted]

Response. Thanks for your suggestions. We have cited the relevant articles.

[redacted]

Response. Thanks for pointing this out. We have added the relevant articles.

[redacted]

Response. Thanks for raising this question. What we want to phrase is that our method can find several important sites in the structure that are closely related to functions with high probability. To clarify this point, we have modified our description as follows:

"In summary, DPFunc offers a more efficient way to unravel the relationships between protein structures and functions compared to existing structure-based methods. It provides researchers with important sites in the structure that may be highly relevant to functions."

[redacted]

Response. Thanks for raising this question. The PDB dataset illustrated in Table 3 is a non-redundant set, which is constructed from DeepFRI[1]. In DeepFRI, blastclust is used to cluster PDB chains at 95% sequence identity and a representative PDB chain from each cluster is selected to construct the PDB dataset. Then, cd-hit is used to split the dataset with 40% sequence identity. To clarify this point, we have added more detailed descriptions as follows:

"The first dataset is collected from DeepFRI[1], named PDB dataset, which is a non-redundant set by clustering all PDB chains at 95% sequence identity and has also been used in previous studies such as GAT-GO[2]. In this dataset, the structures of proteins are obtained from the Protein Data Bank (PDB)[3], which are all validated by experiments. The statistic information is illustrated in Table 3. Specifically, the original PDB dataset is split into training, validation and testing sets by cd-hit[4] with 40% sequence identity, contains 36,408 proteins and 2,748 GO terms, including 488 (17.8%) MF GO terms, 320 (11.6%) CC GO terms, and 1,940 (70.6%) BP GO terms."

References

1. Gligorijević V, Renfrew P D, Kosciolek T, et al. Structure-based protein function prediction using graph convolutional networks[J]. Nature communications, 2021,

12(1): 3168.
2. Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information[J]. Briefings in Bioinformatics, 2022, 23(1): bbab502.
3. Burley S K, Berman H M, Kleywegt G J, et al. Protein Data Bank (PDB): the single global macromolecular structure archive[J]. Protein crystallography: methods and protocols, 2017: 627-641.
4. Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data[J]. Bioinformatics, 2012, 28(23): 3150-3152.

[redacted]

Response. Thanks for pointing this out. ReLU is a rectified linear unit function, as shown in Equation (1). MLP represents multilayer perceptron, composed of several linear layers and ReLU. To clarify this point, we have added more detailed descriptions as follows:

$$ReLU(x) = \max(x, 0) \quad (1)$$

"ReLU is a rectified linear unit function, as shown in Equation (1)."

"where $\hat{y} \in R^{1*c}$ indicates the predicted scores of c GO terms, and MLP is the multilayer perceptron, composed of several linear layers and ReLU."

[redacted]

Response. Thank you for pointing this out. We have corrected the description. It is misdescribed, and we had intended to show that GO is loosely hierarchical as well.

[redacted]

Response. Thanks for your pointing. 'unbalanced' is the same as 'imbalanced'. To clarify this point, we have standardized the descriptions. We also have added the runtimes of our model as follows:

"On the CAFA dataset, our model took around 12 training epochs for 2 hours in MF, 5 hours in BP, and 2 hours in CC. On the smaller PDB dataset, 8 training epochs of our model in MF, BP, and CC all took no more than half an hour."

[redacted]

Response. We are very appreciative of the Reviewer's careful reading of our manuscript. We have corrected it and other typos that we found accordingly.

[redacted]

Response. Thank you for pointing this out. We have added the definition of SOTA methods in the revised manuscript.

[redacted]

Response. Thank you for pointing this out. We have rewritten it in the revised manuscript.

[redacted]

Response. We are very grateful for your suggestions. We have modified it.

[redacted]

Response. Thanks for your suggestions. We have corrected it.

[redacted]

Response. Thanks for pointing this out. It is a misrepresentation. We have rephrased it in our revision as follows:

"In this study, to further explore the performance of our methods, we re-divide the dataset, select a specific type of bacteria, Bacillus subtilis, as the test data, and remove all associated species data from the training data."

[redacted]

Response. Thanks for pointing this out. We have corrected it in our revision.

[redacted]

Reviewer #2 (Remarks to the Author):

The authors present DeepDoguest, a deep learning model featuring domain-guided structural information for protein function prediction. Through a clever combination of embeddings, they successfully integrate InterPro domains, residue level features calculated using ESM-1b, and structural information from PDB or AlphaFold 2. The authors present the model clearly and soundly, and the intuition behind architectural design are provided very elegantly. Also, the available code is of good quality, and it is easy to run for experienced programmers.

Response. We appreciate the summary from the Reviewer of our work and thank the Reviewer for the positive comments. We provide the point-by-point responses below. Additionally, following the suggestion of Reviewer 1, we have renamed our method from **DeepDoguest** to **DPFunc**.

Major Comments

1. Selection of SOTA models for the comparison

Based on the CAFA challenge for the past several years, the best performing method from the challenge is missing from this comparison. NetGO 3.0 [Wang et al., 2023: https://doi.org/10.1016/j.gpb.2023.04.001] is the current state of the art, far surpassing competitors included in the comparison.
Therefore, NetGO 3.0 needs to be included and compared against the same dataset to put DeepDoguest in a more complete context in terms of the state of the art methods.

Response. Thanks for your suggestion. To compare our model with NetGO 3.0[1], we have retrained our model on the data released before 2020-01, which is the same as NetGO 3.0. The result is illustrated in Table R5.

Table R5 (Table S1). Predictive performance of DPFunc compared with two web-servers on the large-scale dataset

	MF		CC		BP	
	Fmax	AUPR	Fmax	AUPR	Fmax	AUPR
COFACTOR $_{AF2}$	0.3768	0.2601	0.4394	0.2451	0.3012	0.1513
NetGO 3.0 $_{server}$	0.6308	0.5837	0.6363	0.6024	**0.4875**	0.4188
DPFunc$_{retrain}$	**0.6346**	**0.6537**	**0.6547**	**0.6887**	0.4534	**0.4290**

*$_{AF2}$ indicates that COFACTOR predict protein functions based on the structure predicted by AlphaFold2.

*$_{server}$ indicates that the results are from the web server of NetGO 3.0

*$_{retrain}$ indicates that our model is retrained on the same dataset with NetGO 3.0.

It shows that our method outperforms NetGO 3.0 in the vast majority of cases, except for Fmax in BP, which also proves the ability of DPFunc for protein function prediction. We have added the following paragraph (in Section 2.2) to summarize the results:

"Moreover, we choose two additional web-servers as competitors, NetGO3.0[1] and COFACTOR[2,3], where NetGO3.0 is the current state-of-the-art method in the CAFA[4] challenge and COFACTOR is an effective structure-based tool for predicting protein functions as a component of I-TASSER-MTD[5] in the CASP[6] challenge."

"Similar conclusions can be drawn from Table S1. DPFunc surpasses the other two web-servers, NetGO3.0 and COFACTOR, in the vast majority of cases, except for Fmax in BP. These comparison results further proves the ability of DPFunc in protein function prediction."

References

1. Wang S, You R, Liu Y, et al. NetGO 3.0: protein language model improves large-scale functional annotations[J]. Genomics, Proteomics & Bioinformatics, 2023, 21(2): 349-358.
2. Zhang C, Freddolino P L, Zhang Y. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information[J]. Nucleic acids research, 2017, 45(W1): W291-W299.
3. Roy A, Yang J, Zhang Y. COFACTOR: an accurate comparative algorithm for structure-based protein function annotation[J]. Nucleic acids research, 2012, 40(W1): W471-W477.
4. Zhou N, Jiang Y, Bergquist T R, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens[J]. Genome biology, 2019, 20: 1-23.
5. Zhou X, Zheng W, Li Y, et al. I-TASSER-MTD: a deep-learning-based platform for multi-domain protein structure and function prediction[J]. Nature Protocols, 2022, 17(10): 2326-2353.
6. Kryshtafovych A, Schwede T, Topf M, et al. Critical assessment of methods of protein structure prediction (CASP)—Round XIV[J]. Proteins: Structure, Function, and Bioinformatics, 2021, 89(12): 1607-1617.

2. Details on how to get the active sites prediction

In section 2.6, the authors present a very exciting capability of DeepDoguest: The identification of individual residues that contribute to specific functional annotations, specifically for enzyme reactions. It seems that the importance of the specific residues comes from the attention mechanism, but it is not clear. The authors need to explain this in more detail.

Response. Thanks for pointing this out. The significant residues are obtained from the attention mechanism and the quantities of predicted structures. As illustrated in Table R6, this process can be split into two steps: (1) Firstly, we generate a candidate set composed of the residues with high attention scores. (2) Then, we consider the quantities of predicted protein structures to filter the residues with low confidence from the candidate set, where the remaining residues are regarded as the significant sites.

Table R6 (Table S5). The process of detecting significant sites

Step	Description
	# First step – generate candidate residue set
1.	$Candidate\_set = set()$
2.	For W_{att-i} in $\{W_{att-1}, W_{att-2}, \dots, W_{att-n}\}$:
3.	Sort attention scores in descending order: $S = [(r_2, s_1), (r_{10}, s_2), \dots, (r_6, s_l)]$ # $s_1 > s_2 > \cdots > s_l$ *are the sorted attention scores and* $r_2, r_{10}, \dots, r_6$ *are the corresponding residues*
4.	Calculate gaps between neighbors: $G = [g_0, g_1, \dots, g_l]$ # $g_i = s_i - s_{i+1}$
5.	Calculate cut-off value: $cutoff = mean(G)$
	# Select residues from a high score to a low score
6.	For $index$ from 1 to l:
7.	If $g_{index} < cutoff$: $Candidate\_set.add(S_{index}[0])$
8.	Else: $Candidate\_set.add(S_{index}[0])$ and Break
	# Second step – filter residues with low confidence
9.	$Significant\_set = set()$
10.	For $residue$ in $Candidate\_set$:
11.	If $pLDDT(residue) \geq 50$: $Significant\_set.add(residue)$
12.	Return $Significant\_set$

For the first step, we consider each attention head separately. For each head i, the attention scores of residues are sorted in descending order, and the gaps between neighbors are calculated. Then, the average value of these gaps is set as the cut-off and the residues are selected to compose the candidate set from higher score to lower score until the gap between residues is larger than the cut-off.

For the second step, we use pLDDT as the confidence of predicted structures. The residues with pLDDT lower than 50 are removed and the remaining residues are regarded as the significant sites.

The details are described in the revision (in Section 4.4) as follows:

"Additionally, once our model is trained, it can detect significant residues from the structures based on the attention mechanism. This process is illustrated in Table S5. Firstly, the attention scores of residues can be obtained from the trained model, denoted as W_{att-i} in Section 4.3. Then, for each head i, these attention scores are sorted in descending order and the gaps between neighbors are calculated. Furthermore, inspired by CLEAN[1], the average value of these gaps is set as the cut-off and the residues are selected to the candidate set from higher score to lower score until the gap between residues is larger than the cut-off. Moreover, considering the qualities of protein structures predicted by AlphaFold2, pLDDT is further used to filter the candidate sites, where higher pLDDT represents higher confidence[2,3]. The residues in the candidate set with pLDDT lower than 50 are removed."

References

1. Yu T, Cui H, Li J C, et al. Enzyme function prediction using contrastive learning[J].

Science, 2023, 379(6639): 1358-1363.
2. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold[J]. Nature, 2021, 596(7873): 583-589.
3. Abramson J, Adler J, Dunger J, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3[J]. Nature, 2024: 1-3.

3. Controlling for homology in the PDB dataset

The PDB dataset used to test the performance seems to be the same one used by Gligorijević et al. in the DeepFRI paper. In that paper, they show a remarkable portion of the performance is associated with elements in the test set with high similarity to elements in the training set. Could the authors also control for this similarity?

Response. We are very grateful for your suggestions. It is true that the performance of DeepFRI[1] is associated with the similarity between training and test proteins. We also used the same similarity in our test. Following DeepFRI, another structure-based method, GAT-GO[2], also shows this ability and achieves better performance than DeepFRI on proteins with the same similarities. As for our method, since the original PDB dataset is divided into training, validation and test data with a sequence identity cut-off value of 40%, we have added four additional filters at sequence identity cut-offs of 25%, 35%, 45%, 55% to generate four distinct datasets, which are used in GAT-GO. The result is shown as Figure R4.

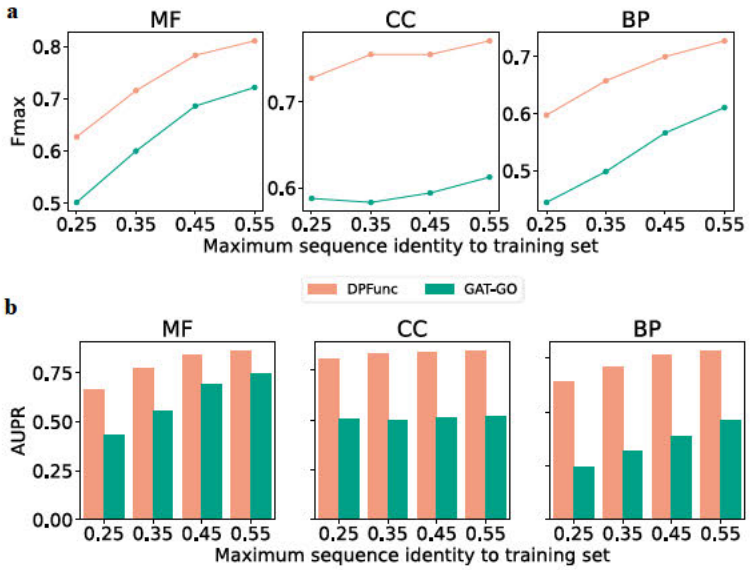

Figure R4 (Figure S1). The performance comparison of structure-based methods and DPFunc across different sequence identity cut-offs.

As shown in Figure R4, it can be obtained that both DPFunc and GAT-GO exhibit improved performance as the sequence identity cut-offs increase. Notably, DPFunc achieves better performance in all cases, which demonstrates the effective of DPFunc for protein function prediction. We have added the following paragraph to describe the results (in Section 2.2):

"We further test the effect of different sequence identities on the performance of these methods. As illustrated in Figure S1, DPFunc achieves better performance in all cases with different sequence identity cut-offs."

References

1. Gligorijević V, Renfrew P D, Kosciolek T, et al. Structure-based protein function prediction using graph convolutional networks[J]. Nature communications, 2021, 12(1): 3168.
2. Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information[J]. Briefings in Bioinformatics, 2022, 23(1): bbab502.

Minor comments

The authors provide the code, as well as a good methods section to explain the model training and hyperparameters, but some details are missing:

- how many epochs are required to train the model?

Response. Thank you for pointing this out. It took around 12 epochs to train our model on the CAFA dataset and 8 epochs on the PDB dataset. We have added detailed descriptions as follows:

"All experiments of the DPFunc are carried out using one NVIDIA Tesla V100s GPU card with 32 GB of memory. On the CAFA dataset, our model took around 12 training epochs for 2 hours in MF, 5 hours in BP, and 2 hours in CC. On the smaller PDB dataset, 8 training epochs of our model in MF, BP, and CC all took no more than half an hour."

- The authors specified the hardware used in training, but it would be informative to understand the training time and memory load.

Response. Thanks for your suggestions. We have added the training time and memory as shown above.

Reviewer #3 (Remarks to the Author):

In “Accurately predicting protein function via deep learning with domain-guided structure information”, Wang et al. describe a new deep-learning-based method for prediction of protein function. The approach, named DeepDoguest, incorporates several current ideas in protein modelling, and performs on par in benchmark settings.

Response. We thank the reviewer for these critical comments. We have added more experimental results to validate the robust of our method and more cases to show the effectiveness of our method in learning the correlation between protein sequences, structures and functions. Meanwhile, we have compared our method with another SOTA method in functional site detection to show the ability of our method in active site detection. We provide the point-by-point responses below. Additionally, following the suggestion of Reviewer 1, we have renamed our method from **DeepDoguest** to **DPFunc**.

It does, however, appear that many of the improvements are marginal rather than substantial. The authors should carry out bootstrapping or a similar subsampling-based approach to determine whether their findings actually are not only significant but also robust.

Response. Thanks for your suggestions. As bootstrapping is more suitable for the small datasets, cross-validation and repeated experiments are more widely used on large-scale datasets. To determine whether our findings actually are robust, we have added more experiments from the following aspects: (1) we first re-divide the PDB datasets with different sequence identity cut-offs to test the performance of DPFunc and a SOTA structure-based method GAT-GO[1]. (2) we randomly repeat the training and testing of all models five times on the CAFA dataset to test whether the improvements of our model are stable and significant.

For the PDB datasets, the original PDB dataset is divided into training, validation and test data with a sequence identity cut-off value of 40%, we have added four additional filters at sequence identity cut-offs of 25%, 35%, 45%, 55% to generate four distinct datasets, which are used in GAT-GO. The result is shown as Figure R5. It can be obtained that both DPFunc and GAT-GO exhibit improved performance as the sequence identity cut-offs increase. Notably, DPFunc achieves the best performance in all cases, which demonstrates the effective of DPFunc for protein function prediction. We have added the following paragraph (in Section 2.2) to describe the results:

"We further test the effect of different sequence identities on the performance of these methods. As illustrated in Figure S1, DPFunc achieves better performance in all cases with different sequence identity cut-offs."

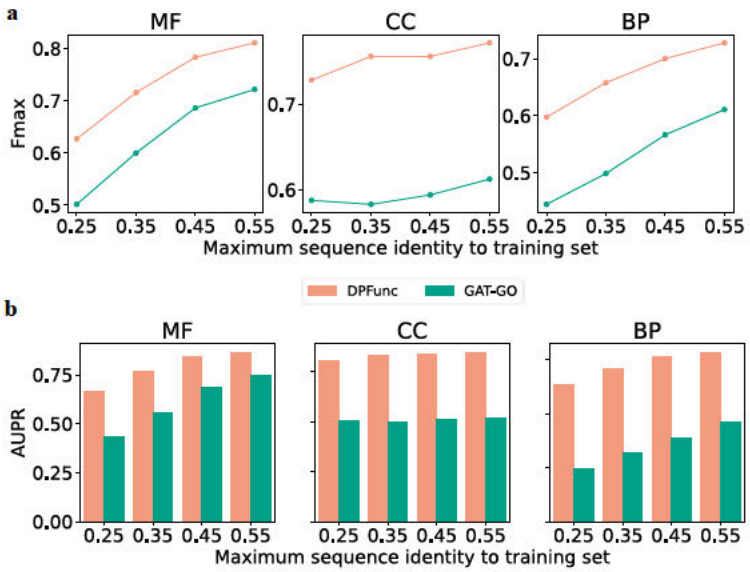

Figure R5 (Figure S1). The performance comparison of GAT-GO and DPFunc across different sequence identity cut-offs.

On the other hand, we randomly repeat the training and testing of all models five times for testing whether the improvements of our model are stable and significant. The comparisons include: (1) a common comparison in terms of Fmax and AUPR (Table R7). (2) the performance evaluation on different sequence identities (Figure R6). (3) the performance evaluation on rare GO terms with different IC values (Figure R7). (4) the performance evaluation on GO terms with deeper depths (Figure R8).

Table R7 (Table 2). Comparison on the large-scale dataset in terms of Fmax and AUPR

Ontology	Methods	Fmax	p-value	AUPR	p-value
MF	Diamond	0.592(-)	-	0.387(-)	-
	BlastKNN	0.616(-)	-	0.484(-)	-
	DeepGO	0.301(±5.47e-03)	8.40e-04	0.204(±8.21e-03)	5.65e-04
	DeepGOCNN	0.396(±5.73e-04)	3.70e-05	0.326(±4.38e-04)	4.90e-06
	TALE	0.260(±2.44e-05)	1.25e-08	0.158(±1.96e-05)	2.57e-09
	ATGO	0.454(±1.25e-05)	1.55e-07	0.442(±4.37e-06)	4.93e-08
	DeepGraphGO	0.562(±8.00e-05)	6.83e-05	0.533(±1.28e-04)	1.37e-05
	DeepGOPlus	0.589(±2.13e-06)	6.22e-06	0.548(±6.26e-05)	1.85e-05
	TALE+	0.602(±6.00e-06)	1.74e-05	0.543(±6.89e-06)	1.83e-06
	ATGO+	0.622(±6.56e-07)	2.80e-04	0.599(±3.86e-07)	1.63e-06
	DPFunc	**0.635**(±3.24e-06)	-	**0.658**(±9.22e-06)	-
CC	Diamond	0.573(-)	-	0.283(-)	-
	BlastKNN	0.596(-)	-	0.384(-)	-
	DeepGO	0.574(±4.78e-05)	5.71e-05	0.580(±6.34e-05)	2.01e-05
	DeepGOCNN	0.573(±2.45e-04)	6.33e-04	0.567(±2.26e-04)	1.45e-04
	TALE	0.548(±1.75e-05)	2.68e-06	0.510(±3.23e-04)	3.62e-05
	ATGO	0.602(±2.76e-06)	3.15e-06	0.596(±7.35e-07)	3.46e-07
	DeepGraphGO	0.634(±4.32e-07)	1.01e-04	0.590(±7.60e-06)	1.61e-06

	DeepGOPlus	0.626(±1.44e-05)	3.06e-04	0.618(±3.89e-05)	4.21e-05
	TALE+	0.608(±8.61e-07)	4.99e-06	0.591(±8.34e-05)	3.68e-05
	ATGO+	0.633(±3.06e-06)	1.12e-04	0.636(±2.13e-07)	3.79e-06
	DPFunc	**0.657**(±7.44e-06)	-	**0.695**(±9.18e-06)	-
BP	Diamond	0.429(-)	-	0.197(-)	-
	BlastKNN	0.445(-)	-	0.258(-)	-
	DeepGO	0.328(±9.89e-05)	1.05e-05	0.260(±8.05e-05)	1.99e-05
	DeepGOCNN	0.323(±3.35e-04)	1.09e-04	0.254(±3.81e-04)	5.83e-05
	TALE	0.253(±2.23e-05)	1.56e-07	0.152(±4.14e-05)	1.67e-07
	ATGO	0.396(±8.64e-07)	5.29e-07	0.341(±3.32e-07)	2.98e-07
	DeepGraphGO	0.432(±2.30e-06)	1.38e-05	0.389(±6.14e-06)	1.70e-05
	DeepGOPlus	0.438(±9.94e-06)	1.58e-04	0.365(±1.28e-05)	1.65e-05
	TALE+	0.427(±4.77e-06)	1.63e-05	0.327(±8.03e-06)	1.04e-06
	ATGO+	0.456(±4.29e-07)	2.06e-04	0.399(±2.76e-07)	9.41e-06
	DPFunc	**0.466**(±2.21e-06)	-	**0.434**(±7.17e-06)	-

* The values of Fmax and AUPR in the table are the mean and standard deviation of the results of five times repeated experiments. P-values are two-tailed Student's t-test between DPFunc and the corresponding compared methods.

As illustrated in Table R7, the standard deviation of Fmax and AUPR indicates that the performance of our model is stable. Additionally, the mean values of Fmax and AUPR and corresponding p-values also prove that the improvements of our model are significant.

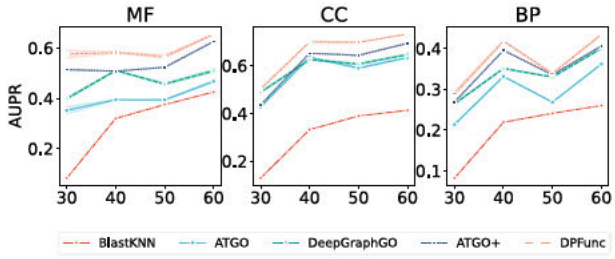

Figure R6 (Figure 2a). The performance evaluation of DPFunc and other representative methods on difficult protein sets with different sequence similarities to training proteins.

From Figure R6, it can be obtained that DPFunc consistently outperforms other methods in nearly all cases, except for the 50% threshold in BP where it demonstrates comparable performance to ATGO+. Similar conclusion can be drawn from Figure R7 and R8, DPFunc surpasses other deep learning-based methods in all conditions.

As illustrated in Figure R7, DPFunc consistently outperforms other methods when predicting GO terms with fewer samples, and the improvement remains for more specific GO terms (IC≥3).

Figure R8 shows the performance of these methods on GO terms with deeper nodes (depths≥8 in MF and BP, depths≥6 in CC as the maximum is 7 in CC), which evaluates the performance on each selected GO term and means their AUPR values as the final metric. DPFunc still achieves the best performance, except for being slightly weaker than BlastKNN in BP.

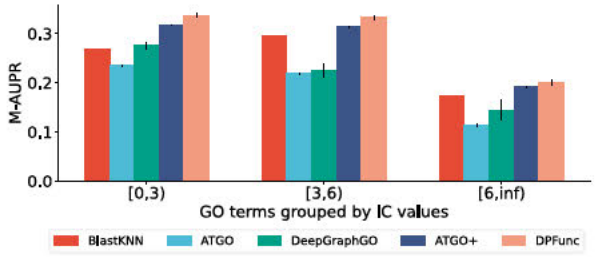

Figure R7 (Figure 2c). The performance evaluation of DPFunc and other representative methods on rare GO terms with different IC values, where GO terms with higher IC values are more informative and valuable.

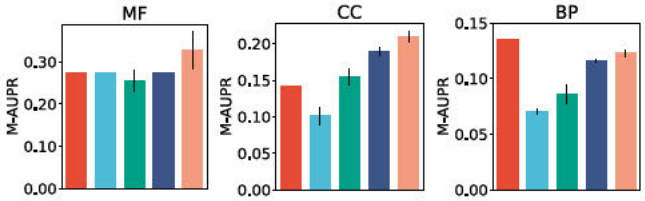

Figure R8 (Figure 2f). The performance of representative methods on GO terms with deeper depths, where the distances between GO terms and root node (MF/CC/BP) are larger than 8, 6, and 8, respectively.

References

1. Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information[J]. Briefings in Bioinformatics, 2022, 23(1): bbab502.

From the description it is a little unclear whether structure is required in all parts of the model, or just some. Domain annotation is well possible based on sequence. For those parts of a protein that are not well-structured - according to AlphaFold - will they be penalised for this lack of structure? Is pLDDT taken into account, or other metrics of prediction reliability?

Response. We are very grateful for this comment. First of all, protein structure is required in our model and is the input to the "residue feature learning module" (Figure. 1b), which is independent from the domain annotation at the beginning. Then, the learned features are fused with domain annotations as the input of the "protein feature learning module" to predict functions. Notably, the protein structures can be either native or predicted, the former corresponds to the PDB dataset and the latter corresponds to the CAFA dataset which is predicted by AlphaFold2. To clarify this point, we have added following details (in Section 2.1) as follows:

"The overall architecture of DPFunc is shown in Figure 1. It consists of three modules: (1) a residue-level feature learning module based on a pre-trained protein language model and graph neural networks for propagating features between residues through protein structures which can be the native structures from the PDB database or the predicted structures by AlphaFold2. (2) a protein-level feature learning module for

extracting the whole structure features from residue-level features guided by domain information from sequences. (3) a protein function prediction module for annotating functions to proteins based on protein-level features."

On the other hand, for those parts of a protein that are not well-structured, we use pLDDT[1] as the metric to evaluate the qualities of the predicted structures. We consider the quantities in the process of significant sites detection but not in model training. As illustrated in Table R8, after our model detects the candidate active site set (step 1-8), the residues with pLDDT lower than 50 are removed (step 9-12).

Table R8 (Table S5). The process of detecting significant sites

Step	Description
	# First step – generate candidate residue set
1.	$Candidate\_set = set()$
2.	For W_{att-i} in $\{W_{att-1}, W_{att-2}, \dots, W_{att-n}\}$:
3.	Sort attention scores in descending order: $S = [(r_2, s_1), (r_{10}, s_2), \dots, (r_6, s_l)]$ # $s_1 > s_2 > \cdots > s_l$ *are the sorted attention scores and* $r_2, r_{10}, \dots, r_6$ *are the corresponding residues*
4.	Calculate gaps between neighbors: $G = [g_0, g_1, \dots, g_l]$ # $g_i = s_i - s_{i+1}$
5.	Calculate cut-off value: $cutoff = mean(G)$
	# Select residues from a high score to a low score
6.	For $index$ from 1 to l:
7.	If $g_{index} < cutoff$: $Candidate\_set.add(S_{index}[0])$
8.	Else: $Candidate\_set.add(S_{index}[0])$ and Break
	# Second step – filter residues with low confidence
9.	$Significant\_set = set()$
10.	For $residue$ in $Candidate\_set$:
11.	If $pLDDT(residue) \geq 50$: $Significant\_set.add(residue)$
12.	Return $Significant\_set$

The details are described in the revision (in Section 4.4) as follows:

"Additionally, once our model is trained, it can detect significant residues from the structures based on the attention mechanism. This process is illustrated in Table S5. Firstly, the attention scores of residues can be obtained from the trained model, denoted as W_{att-i} in Section 4.3. Then, for each head i, these attention scores are sorted in descending order and the gaps between neighbors are calculated. Furthermore, inspired by CLEAN[2], the average value of these gaps is set as the cut-off and the residues are selected to the candidate set from higher score to lower score until the gap between residues is larger than the cut-off. Moreover, considering the qualities of these protein structures predicted by AlphaFold2, pLDDT is further used to filter the candidate sites, where higher pLDDT represents higher confidence[1,3]. The residues in the candidate set with pLDDT lower than 50 are removed."

References

1. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold[J]. Nature, 2021, 596(7873): 583-589.
2. Yu T, Cui H, Li J C, et al. Enzyme function prediction using contrastive learning[J].

Science, 2023, 379(6639): 1358-1363.
3. Abramson J, Adler J, Dunger J, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3[J]. Nature, 2024: 1-3.

In Fig. 3, the most striking change is for CC and BP, while all others are perhaps statistically significant but virtually unchanged. What is different about the CC and BP sets or cases here that leads to this striking improvement when adding domain annotations?

Response. Thank you for pointing this out. We have analyzed the improvements in MF, CC and BP in detail and concluded that domain annotations bring the same degree of enhancement in all three ontologies. As illustrated in Figure R9, Figure R9a-b evaluate the role of domain annotations in general. After adding the domain annotations, in terms of Fmax, DPFunc gets the improvements of 4.4%, 2.7% and 5.7% in MF, CC and BP, respectively. Additionally, the improvements in AUPR also achieves 4.8%, 2.7% and 4.5% in MF, CC and BP, respectively.

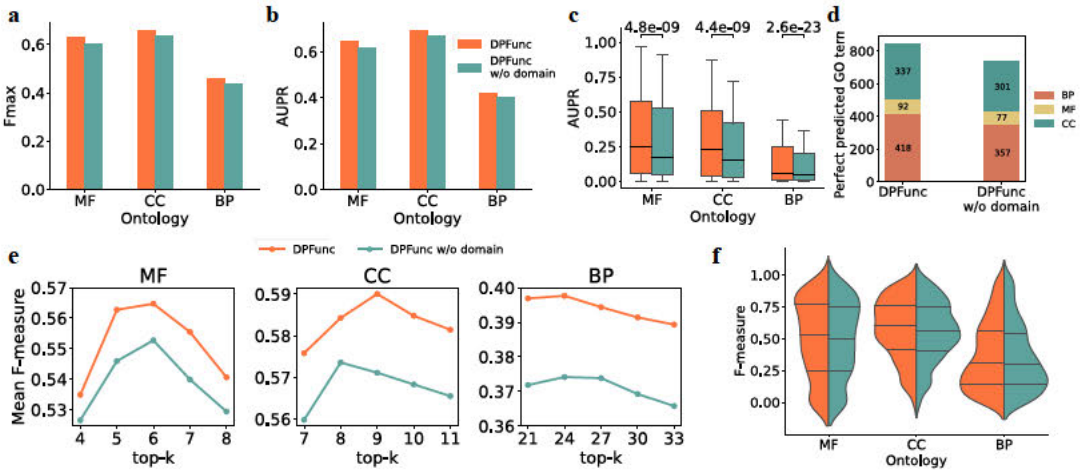

Figure R9 (Figure 3). The analyses of the role of domain information. **a-b.** The comprehensive comparison of DPFunc and DPFunc w/o domain in terms of Fmax and AUPR. **c.** The performance on each function. AUPR values are calculated separately for each GO term (remove the perfect predicted GO terms which are shown in Figure 3d), and the medians of the boxes represent the comprehensive performance of corresponding methods. Two-side paired t-tests are conducted and the resulting P values are annotated at the top of the boxes. **d.** The number of perfect predicted GO terms. **e.** The performance of top-k predicted functions of each protein. Since there are 7, 11, and 30 GO terms per protein on average in MF, BP, and CC, different ranges of k are selected (4-8 for MF, 7-11 for CC, and 21-33 for BP, respectively). **f**. The performance of top-k predicted functions of each protein, where k is exactly set as 5, 9, 24 for MF, CC, and BP, respectively.

Next, we evaluate the performance of these two models from both the GO-level (Figure R9c-d) and protein-level (Figure R9e-f). For the GO-level, each GO term is evaluated separately. Figure R9d shows the number of perfectly predicted GO terms (AUPR=1), it can be observed that DPFunc with domain annotations achieves better performance.

As for the other GO terms, Figure R9c shows that DPFunc also have a remarkable median AUPR improvement of 12.0%, 14.7%, and 16.3% for MF, CC, and BP domains, respectively, demonstrating the significant roles of incorporating domain information. For the protein-level, we focus on evaluating predictions with high confidence scores. Specifically, we assess the results with the top k prediction scores of these two models, where k is determined by the average number of GO terms per protein (approximately ~7 for MF, ~11 for CC, and ~30 for BP). As shown in Figure R9e, it can be observed that DPFunc achieves better performance after incorporating the domain information, demonstrating mean F-measure improvements exceeding 1.6%~3.1% for MF, 1.9%~3.3% for CC, and 5.5%~6.7% for BP. Similar conclusions can be drawn from Figure R9f, which shows the distribution of predictions over specific k values (5 for MF, 9 for CC, and 24 for BP).

Some of the described observations fit well to the paradigm of function being more conserved than structure, and structure being more conserved than function, e.g. in Fig. 4. The illustration of such a case in itself is not convincing though - the authors should identify "negative controls" that have similar levels of sequence identity where DeepDoguest clearly predicts different functions.

Response. Thanks for your suggestion. We have added a "negative control" case, including three proteins (PDB ID: 5JZV-A, 3WG8-A, 5Z9R-A). As shown in Table R9 and Table R10, these proteins have high sequence identities but different structures and functions (see Figure R10 and R11).

Table R9 (Table S6). The sequence identities between proteins calculated by BLAST.

	5JZV-A	3WG8-A	5Z9R-A
5JZV-A	100%	87.80%	89.74%
3WG8-A	87.80%	100%	90.24%
5Z9R-A	89.74%	90.24%	100%

Table R10 (Table S7). The number of common functions between proteins.

	5JZV-A	3WG8-A	5Z9R-A
5JZV-A	31/31	5/31	20/31
3WG8-A	5/24	24/24	5/24
5Z9R-A	20/22	5/22	22/22

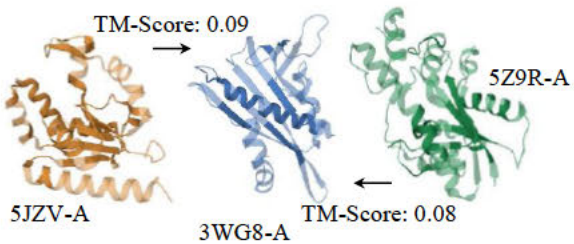

Figure R10 (Figure 4h). The structure alignment results between 5JZV-A, 3WG8-A and 5Z9R-A. Dark colors in each protein represent residues that are aligned and light

colors represent residues that are not aligned.

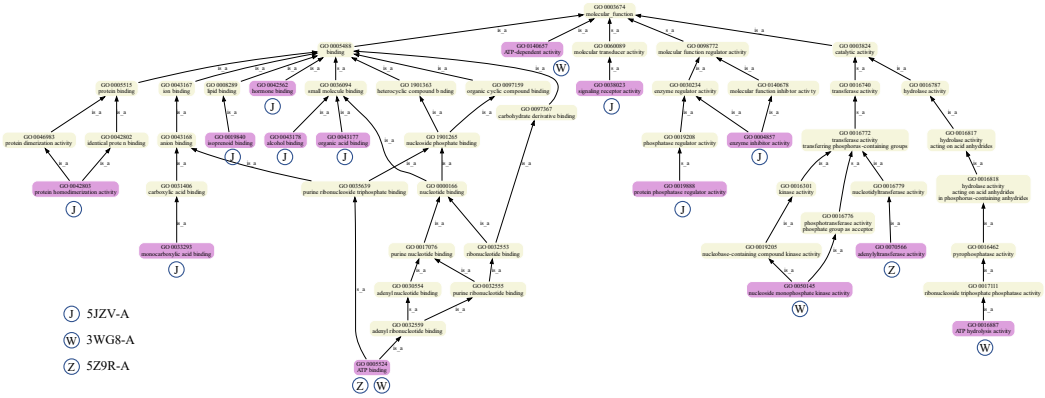

Figure R11 (Figure S4). The GO terms of three proteins (PDB ID: 5JZV-A, 3WG8-A, 5Z9R-A). The purple blocks are the deepest child GO terms, where "J", "W" and "Z" indicates 5JZV-A, 3WG8 and 5Z9R-A, respectively, and all of these GO terms are predicted by DPFunc accurately.

DPFunc predicts their functions with 100% accuracy on all three proteins, which also demonstrates the capability of our model on proteins with high sequence identities but different structures. To clarify this point, we have renamed Section 2.4 ("DPFunc effectively distinguishes structure motifs and sequence identities") and added following descriptions:

"Additionally, there also exist scenarios where proteins with high sequence identities have different structures and functions. It is necessary for models to distinct these proteins and corresponding functions. Consequently, we present three proteins here to evaluate the capability of DPFunc in this scenario (PDB ID: 5JZV-A, 3WG8-A, 5Z9R-A, see Figure 4h). As illustrated in Table S6 and Table S7, these proteins have high sequence identities but different functions. For instance, the sequence identity between 5JZV-A and 3WG8-A is 87.8% but they have only 5 common functions. For these proteins, DPFunc predicts their functions with 100% accuracy, as shown in Figure S4, which demonstrates the ability of our model on proteins with high sequence identities but distinct structures."

The test case Bacillus subtilis is surprising - this organism is well-annotated and was sequenced in 1997 (Kunst et al. 1997) - could the authors elaborate which specific strain they consider newly sequenced?

Response. Thanks for pointing this out. It is a misrepresentation. What we are trying to express is the challenge in accurately predicting the functions of newly sequenced species, a phenomenon that is particularly common in bacteria and viruses[1]. Consequently, following the previous study[1], we choose a type of bacteria as the target species and remove any associated species data from the training data to test the performance of our model. We have rephrased it in our revision (in Section 2.5) as follows:

"In this study, to further explore the performance of our methods, we re-divide the dataset, select a specific type of bacteria, *Bacillus subtilis*[2], as the test data, and remove any associated species data from the training data."

References

1. Torres M, Yang H, Romero A E, et al. Protein function prediction for newly sequenced organisms[J]. Nature Machine Intelligence, 2021, 3(12): 1050-1060.
2. Kunst F J, Ogasawara N, Moszer I, et al. The complete genome sequence of the gram-positive bacterium Bacillus subtilis[J]. Nature, 1997, 390(6657): 249-256.

For the prediction of functional or active sites, the authors need to use established benchmark sets and compare to SOTA in the field, e.g. (Cagiada et al. 2023). As a devil's advocate looking at the current Fig. 5, one might suggest it simply predicts all histidines to be catalytically active…

Response. Thanks for your suggestions. We have compared our method with the SOTA method in functional sites prediction, i.e., Cagiada Matteo's method[1]. Consistent with the benchmark in their articles (Figure 4b in their article), we select three public proteins from M-CSA that appear in both our data and their data, i.e., P82385, O32727, Q8VQN0, for performance comparison. The statistical information is illustrated as Table R11. In this comparison, for each predicted residues, its closest distance to a known active site is used to measure the accuracy, which is also used in Matteo's. The result is shown as Figure R12.

Table R11 (Table S8). The statistical information of protein active sites.

Protein	PDB	Chain	Active Sites
P82385	1DO6	A	7
O32727	1L0O	A	16
Q8VQN0	1JC5	A	5

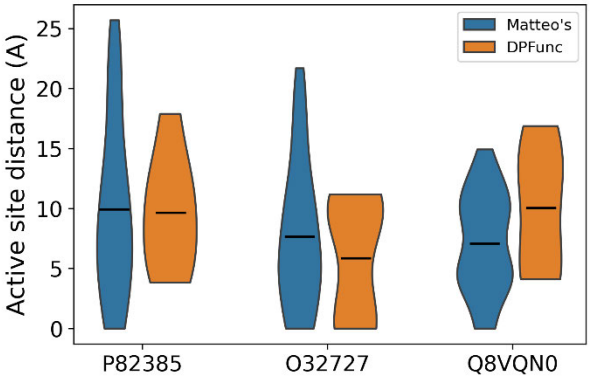

Figure R12 (Figure S6). The distribution of distances between predicted residues and the nearest known active sites.

It can be obtained that our method achieves comparable performance on P82385, better performance on O32727 and worse performance on Q8VQN0. Accordingly, we have added following descriptions (in Section 2.6) to summarize the comparison results:

"Additionally, we compare our method with another SOTA method[1] in the field of functional site prediction, denoted as "Matteo's". As illustrated in Table S8, we test the two approaches on three common proteins that appear in both our data and Matteo's data. The known active sites can be obtained from M-CSA database[2]. The results in Figure S6 show that our method achieves comparable performance with Matteo's method, further supporting the effect of DPFunc on active site detection."

On the other hand, for the four proteins in Figure 5(b), we further align their sequences using Clustal Omega[3], as shown in Figure R13.

Figure R13 (Figure S5). The sequence alignment results of WSD1, WSD6, WSD7 and WSD11.

It can be observed that the positions we choose is aligned (H147 for WSD1, H163 for WSD6, H135 for WSD7 and H144 for WSD11), which further support the co-evolutionary conservation of these residues. To clarify this point, we have added following descriptions:

"It is worth noting that each of these four proteins has a known active site, where HIS-147 for WSD1[4,5], HIS-163 for WSD6[5], HIS-135 for WSD7[5], and HIS-144 for WSD11[6,7]. Moreover, we use Clustal Omega[3] to align these sequences. As illustrated in Figure S5, the four positions are aligned as expected, which further support the co-evolutionary conservation of these residues. As for these proteins, DPFunc detects all of these active sites accurately."

References.

1. Cagiada M, Bottaro S, Lindemose S, et al. Discovering functionally important sites in proteins[J]. Nature communications, 2023, 14(1): 4175.
2. Ribeiro A J M, Holliday G L, Furnham N, et al. Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites[J]. Nucleic acids research, 2018, 46(D1): D618-D623.
3. Sievers F, Wilm A, Dineen D, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega[J]. Molecular systems biology, 2011, 7(1): 539.
4. Li F, Wu X, Lam P, et al. Identification of the wax ester synthase/acyl-coenzyme A: diacylglycerol acyltransferase WSD1 required for stem wax ester biosynthesis in Arabidopsis[J]. Plant physiology, 2008, 148(1): 97-107.
5. Patwari P, Salewski V, Gutbrod K, et al. Surface wax esters contribute to drought tolerance in Arabidopsis[J]. The Plant Journal, 2019, 98(4): 727-744.
6. Takeda S, Iwasaki A, Matsumoto N, et al. Physical interaction of floral organs controls petal morphogenesis in Arabidopsis[J]. Plant physiology, 2013, 161(3): 1242-1250.
7. Takeda S, Iwasaki A, Tatematsu K, et al. The half-size ABC transporter FOLDED

PETALS 2/ABCG13 is involved in petal elongation through narrow spaces in Arabidopsis thaliana floral buds[J]. Plants, 2014, 3(3): 348-358.

Given the broad audience of the journal, it would be good if the authors described exactly what “prediction of protein function” can do, and perhaps also outline what it cannot do yet. In other words, how practically useful are GO annotations to researchers interested in individual proteins? For those performing large-scale analyses and classifications? Ideally the authors will also consider to what extent it even makes sense to predict the function of a protein in isolation, given that they have evolved for function in their cellular context and that many proteins have “moonlighting” functions that can sometimes be drastically different from the canonical, annotate function (Jeffery 2018)

Response. We are very grateful for your suggestion. We have added more descriptions to clarify what "protein function prediction" can do and cannot do in "Introduction" and "Discussion" as follows:

"The individual proteins after mutations, for example, are necessary to verify that the specific functions are retained[1,2,3]. " (in "Introduction")

"Consequently, developing computational methods for automated protein function prediction is crucial for bridging the widening gap between the number of annotated and new protein sequences generated by high-throughput technology, which benefits biologists in discovering proteins of interest and serve as a guide for protein virtual screening and protein design[4,5]. " (in "Introduction")

"Since proteins perform functions in cellular context, their functions are dynamically transformed with the environment. How to accurately predict the dynamic functions is another challenge to be addressed in the future[6]." (in "Discussion")

References

1. Leveson-Gower R B, Mayer C, Roelfes G. The importance of catalytic promiscuity for enzyme design and evolution[J]. Nature Reviews Chemistry, 2019, 3(12): 687-705.
2. Soskine M, Tawfik D S. Mutational effects and the evolution of new protein functions[J]. Nature Reviews Genetics, 2010, 11(8): 572-582.
3. Ng P C, Henikoff S. Predicting the effects of amino acid substitutions on protein function[J]. Annu. Rev. Genomics Hum. Genet., 2006, 7(1): 61-80.
4. Sumida K H, Núñez-Franco R, Kalvet I, et al. Improving protein expression, stability, and function with ProteinMPNN[J]. Journal of the American Chemical Society, 2024, 146(3): 2054-2061.
5. Madani A, Krause B, Greene E R, et al. Large language models generate functional protein sequences across diverse families[J]. Nature Biotechnology, 2023, 41(8): 1099-1106.
6. Jeffery C J. Protein moonlighting: what is it, and why is it important?[J]. Philosophical transactions of the Royal Society B: biological sciences, 2018, 373(1738): 20160523.

To summaries, the paper presents several interesting ideas that are in line with longstanding paradigms - the testing and illustration of successful cases needs to be more thoroughly checked though.

Response. We appreciate the positive comments from the Reviewer. We have added more testing and cases as mentioned above.

Minor comments:

P2 “medical text”, unusual term, perhaps this refers to literature curation?

Response. Thanks for pointing this out. We have corrected it and other typos that we found accordingly.

P7 “It can be obtained that DeepDoguest gets significant improvements than other methods,”, rephrase

Response. Thanks for pointing this out. We have corrected it.

Response letter

On behalf of all the contributing authors, we would like to express our sincere appreciation for reviewers' constructive comments and helpful suggestions regarding our article entitled "*DPFunc: Accurately predicting protein function via deep learning with domain-guided structure information*". These comments have significantly improved the presentation of our article. In response to the feedback from reviewers, we have made substantial revisions to our manuscript and have provided more experimental results and cases to support the robust of our method. We hope that this detailed response resolves the concerns of reviewers.

Reviewer #2 (Remarks to the Author):

The authors have addressed all my comments very well. The paper is now considerably better than the initial submission. I have no more concerns.

Response. We are glad that the Reviewer is satisfied with our Revision.

Reviewer #3 (Remarks to the Author):

The authors put a lot of effort into responding to all reviewers' comments and improve the manuscript.

Response. We appreciate the positive comments on our previous revision.

I have only 2 remaining minor comments:

- the difference between the lighter and darker colours in the new Fig. 4H is difficult to see on some screens. I would encourage the authors to increase the contrast, or perhaps use gray for the non-aligned positions.

Response. Thank you for the good suggestions. For the Figure 4H, we have increased the contrast between the aligned and non-aligned positions to highlight the difference, as shown in Figure R1.

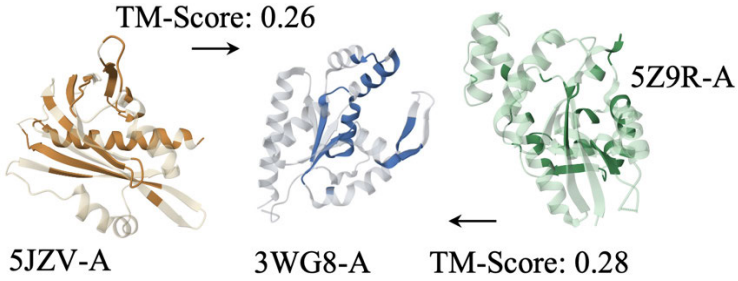

Figure R1. The structure alignment results between 5JZV-A, 3WG8-A and 5Z9R-A. Dark colors in each protein represent residues that are aligned and light colors represent residues that are not aligned.

- it's odd to call a method by the author's first name, Matteo, unless it were called that in the original paper

Response. Thanks for pointing this out. We have double-checked the original paper and corresponding code links. There is no specific name for the proposed method in the paper. Consequently, in the revision, we use the "Ref [75]" or "Cagiada, et al., 2023 [75]" to indicate the method proposed in their article.