nature portfolio

Peer Review File

Context-aware geometric deep learning for protein sequence design

REVIEWER COMMENTS

Reviewer #1 (Remarks to the Author):

Krapp et al. present a graph-based transformer method to recover a protein sequence given the atomic coordinates of a backbone. The methodology is akin to MPNN, with the added benefit that it has the capacity to consider ligands. The authors evaluated its accuracy in a number of tasks. They convincingly show the strength of including ligands in the graph, which improves functional sequence recovery, in particular in the vicinity of the ligand. The manuscript is well written, the work is well executed and presented. This method will certainly be useful for the community in design workflows, in particular where ligands are involved, as well as more generally in theoretical works aiming to map the structure-sequence space. I have only a few minor comments detailed below.

The training and test sets are processed to remove redundancy, e.g., at 30% sequence identity. How was overlap considered? For example, if a sequence is 40% identical and its overlap is only 50%. On a similar note, "no CATH similarity" means no single CATH domain is shared between both structures? (this could be clarified in the text).

Given that the methodology is similar to MPNN and ESM-if1, reflecting better on their similarity and differences in the recovered sequence spaces would be useful. For instance, a similarity could be computed between methods for each structure, and three distributions of these similarity values (i.e., MPNN-ESM-if1, MPNN-CARBONARA, ESM-if1-CARBONARA) could be shown.

The training was performed on biological assemblies from the PDB. Multiple assemblies can exist for a particular PDB code, and it is not clear how this was handled. Additionally, assemblies often contain artefactual interfaces that could be filtered out. I recommend retraining after such a filter is applied.

The architecture of the transformer could be illustrated in more detail in a supplementary figure.

The readme is cryptic. Some details regarding the main functions, their input-output and options would be useful for people to re-use the code. A gunzipped archive with the original structure files and original split used to train/test the network would also be important.

Reviewer #1 (Remarks on code availability):

I had a look at the repository but did not read, install, or run the code.

Reviewer #2 (Remarks to the Author):

"Context-aware geometric deep learning for protein sequence design" is a manuscript describing a novel deep-learning model (CARBonAra) for protein design that generalizes existing design methods by natively handling any type of molecular context within the design process. This is extremely important in protein design tasks, as information about non-protein molecules interacting with the candidate structure is typically available at design time. The authors show how this ability results in heavily increased median sequence recovery when provided with the correct molecular context. The authors validate the method in a real-case scenario by engineering a Beta-lactamase enzyme, retrieving four designs that, although weakly active against the substrate at 30C, display improved activity at 70C. This, combining with the finding that the four designs are part of a novel family of B lactamases enzymes, has very important implications for in-silico protein evolution analysis.

The manuscript is very well written and highly relevant, and results are presented with clarity and scientific rigor. Overall, I strongly recommend the manuscript for publication in Nature Communications.

I have some minor criticisms/observations that could hopefully help improving an already excellent work:

1) the authors perform the Beta lactamase design in a context-aware manner, where the context is provided with a nitrocefin molecule docked at the active site. I think the authors should specify A) why the docking is necessary here (I imagine that nitrocefin has never been crystallized neither in the reference pdb file nor in any of its homologs) B) how the docking was performed using Autodock Vina: how big was the search space? were the default settings used? how was the nitrocefin model generated? C) how was the analysed pose selected? In the methods there is some discussion about this selection, but it's not quantitative enough.

2) when discussing the context-aware performances of CARBonAra I would separate the glycans from the ligands. I believe glycoproteins could benefit even more than standard small molecules from the presence of the context, especially when the "context" (glycan here) itself is covalently linked to the protein.

3) The authors mention the GPU time needed to train CARBonAra. I do think that reporting (and maybe visualizing in a plot) the inference time would be very helpful to underline the competitiveness of the method. Maybe correlating it with the sequence length.

4) Alphafold2 is employed at various stages of the manuscript, from evaluating the foldability of the designed sequences to screening the obtained designs. It is usually used in single-sequence mode: why? In protein-design it is quite common to not input the multiple sequence alignment, but I think this should be explained.

Reviewer #3 (Remarks on code availability):

Although I didn't review the whole code, I have some observations that I hope the authors will consider.

1) I have to commend the authors for making the conda environment setup very simple and straightforward.

2) I suggest that the authors add way more information to the README file

3) understanding how to run the program is not very intuitive, and I think that implementing one or more command line interfaces to launch the different design tasks on an input folder (or directly on a pdb file) would be very helpful. As an example, having something like "carbonara-design examples/2oob.pdb" doing the job of apply_model.ipynb

4) while running the model on a couple of dimers I encountered the following error "RuntimeError: Number of dimensions of repeat dims can not be smaller than number of dimensions of tensor". I don't know exactly what it means but it would be nice if the error message was somehow more informative. An example dimer I tried was pdb 8cyj, chains B and R.

Point-by-point response to reviewers' comments

We thank the reviewers for their constructive and overall positive feedback on our work. We reply point by point to their concerns here below and provide an emended version of our manuscript where additional figures are added and modified text is highlighted in red. To note, the user resources and support for our code have been heavily improved in order to respond to the reviewers' request and make our tool more accessible to the users. We think that this new version will allow more people to access our code using it in a more rational and efficient way.

Response to Reviewer #1

Krapp et al. present a graph-based transformer method to recover a protein sequence given the atomic coordinates of a backbone. The methodology is akin to MPNN, with the added benefit that it has the capacity to consider ligands. The authors evaluated its accuracy in a number of tasks. They convincingly show the strength of including ligands in the graph, which improves functional sequence recovery, in particular in the vicinity of the ligand. The manuscript is well written, the work is well executed and presented. This method will certainly be useful for the community in design workflows, in particular where ligands are involved, as well as more generally in theoretical works aiming to map the structure-sequence space. I have only a few minor comments detailed below.

We would like to thank the reviewer for the overall positive assessment of our work.

The training and test sets are processed to remove redundancy, e.g., at 30% sequence identity. How was overlap considered? For example, if a sequence is 40% identical and its overlap is only 50%.

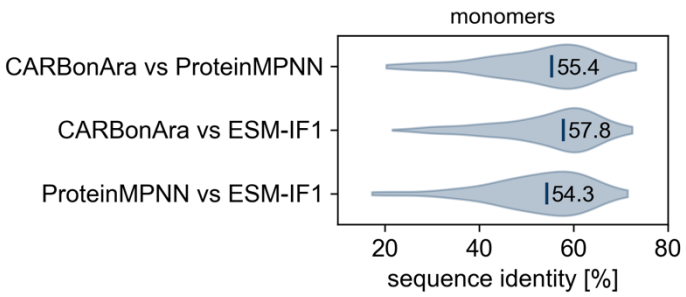
We calculate sequence identity based on the aligned portion of sequences. This implies that if two sequences exhibit 40% identity over a 50% overlap, the overlapping region still has a 40% identity. Consequently, these sequences would be regarded as sufficiently similar to be grouped together, even though much of their lengths might differ. This approach implies that we apply stricter criteria when segregating our training and testing datasets, ensuring that even partial similarities are accounted for to maintain the integrity of our data splits.

On a similar note, “no CATH similarity” means no single CATH domain is shared between both structures? (this could be clarified in the text).

We ensure that there are no shared CATH domains between the testing dataset and training dataset. We have better explained this point in the revised text.

Given that the methodology is similar to MPNN and ESM-if1, reflecting better on their similarity and differences in the recovered sequence spaces would be useful. For instance, a similarity could be computed between methods for each structure, and three distributions of these similarity values (i.e., MPNN-ESM-if1, MPNN-CARBONARA, ESM-if1-CARBONARA) could be shown.

As requested, we performed this comparative benchmark and found out that the sequences predicted by the three methods are as similar with each other than with the original scaffold sequence. The quantification is now shown in a new supplementary figure (Supplementary Figure 4) also attached here below.

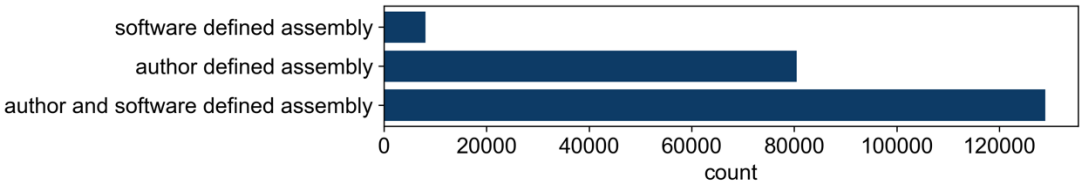


Supplementary Figure 4 | Comparison of predicted sequences between methods. Sequence identity between sequences predicted by two different methods for 142 monomers. The sequences predicted by the three methods are as similar with each other than with the original scaffold sequence.

The training was performed on biological assemblies from the PDB. Multiple assemblies can exist for a particular PDB code, and it is not clear how this was handled. Additionally, assemblies often contain artefactual interfaces that could be filtered out. I recommend retraining after such a filter is applied.

We only took the first biological assembly as provided by RCSB PDB. The first biological assembly is in many cases provided by the authors or both by the software and confirmed by the authors. Therefore, this would suggest that most of the oligomers have been annotated with the correct biological interface. In the cases where the authors didn’t know the oligomerization state of the protein, the assignment is only done automatically by software like PISA.

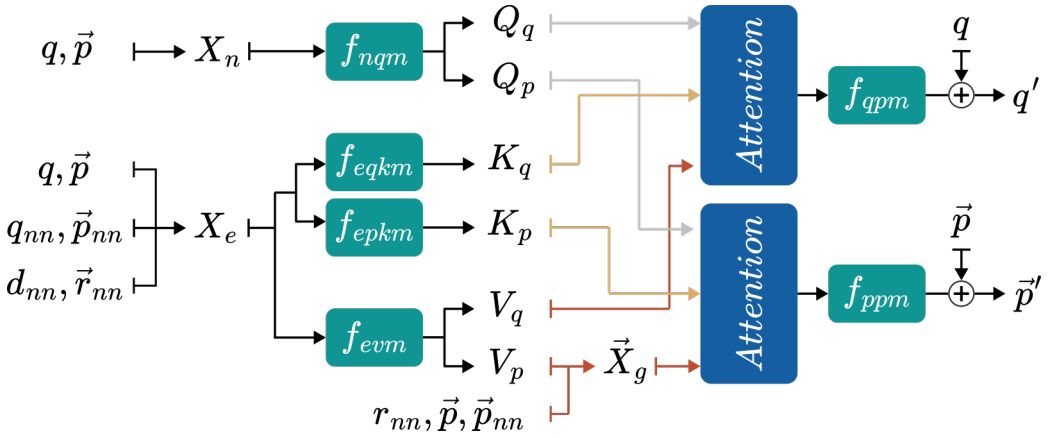
However, our analysis reported in the figure here below indicates that most (95%) of the interfaces present in the first biological assembly are properly annotated and can be considered trustworthy. The ones relying only on automatic assignments of oligomerization states are a minority (5%) that should not bias the model significantly.



RCSB PDB assembly assessment. We downloaded the information provided for all structures on RCSB PDB and extracted the assignment method of biological assembly 1. We show here the number of biological assemblies assigned automatically (software), manually (author) or hybrid (author and software).

The architecture of the transformer could be illustrated in more detail in a supplementary figure.

We prepared an additional supplementary figure to better illustrate the architecture of our model.



Supplementary Figure 1 | CARBonAra geometric transformer. The inputs of the geometric transformer are scalar state (q) and vector state (**p**) for the central atom and the neighbors scalar states (q_{nn}), vector states ($\mathbf{p}_{nn}$), distances (d_{nn}) and relative displacement vectors ($\mathbf{r}_{nn}$). First, we extract the scalar information of the central node features (X_n) and edges features (X_e) from the inputs. The central node features produce the queries (Q_q, Q_p) through an MLP (f_{nqm}). The edge node features produce the keys (K_q, K_p) and values (V_q, V_p) through multiple MLP (f_{eqkm}, f_{epkm}, f_{evm}). We project the vector track values (V_p) on relative displacement vectors (r_{nn}) and concatenate the vector states to create the geometric features ($\mathbf{X}_g$). We compute the multi-heads key, query and value attention for the

scalar and vector track. We reduce the outputs of the attention operation with an MLP for the scalar quantities (f_{qpm}) and a weighted sum (f_{ppm}) for the vector track to preserve the rotation equivariance of the operation. Lastly, we add the input states as residual connections.

The readme is cryptic. Some details regarding the main functions, their input-output and options would be useful for people to re-use the code. A gunzipped archive with the original structure files and original split used to train/test the network would also be important.

We spent most of the time on this revision to improve the README file with more details on how to install and use the software. We also included the original split used to train, test and validate the method in the GitHub repository. The input structures can be easily downloaded from the FTP server of RCSB PDB. We think that this improved version will be instrumental in allowing a more user-friendly experience of our code.

Response to Reviewer #2

"Context-aware geometric deep learning for protein sequence design" is a manuscript describing a novel deep-learning model (CARBonAra) for protein design that generalizes existing design methods by natively handling any type of molecular context within the design process. This is extremely important in protein design tasks, as information about non-protein molecules interacting with the candidate structure is typically available at design time. The authors show how this ability results in heavily increased median sequence recovery when provided with the correct molecular context. The authors validate the method in a real-case scenario by engineering a Beta-lactamase enzyme, retrieving four designs that, although weakly active against the substrate at 30C, display improved activity at 70C. This, combining with the finding that the four designs are part of a novel family of B lactamases enzymes, has very important implications for in-silico protein evolution analysis.

The manuscript is very well written and highly relevant, and results are presented with clarity and scientific rigor. Overall, I strongly recommend the manuscript for publication in Nature Communications.

We would like to thank this reviewer for the overall positive assessment of our work.

I have some minor criticisms/observations that could hopefully help improving an already excellent work:

1) the authors perform the Beta lactamase design in a context-aware manner, where the context is provided with a nitrocefin molecule docked at the active site. I think the authors should specify A) why the docking is necessary here (I imagine that nitrocefin has never been crystallized neither in the reference pdb file nor in any of its homologs)

Docking was indeed necessary because we do not have any TEM-1 structure in complex with any substrate. This information was probably too hidden in the methods section. We have better explained the protocol in the revision. The docking was guided also by the several TEM structures complexed by inhibitors as well as biomolecular models of the enzyme in complex with its substrates.

B) how the docking was performed using Autodock Vina: how big was the search space? were the default settings used? how was the nitrocefin model generated?

We obtained the 3D coordinates of nitrocefin from the PubChem database (PubChem CID: 6436140) and used a search space of size 40x40x40 $Å^3$ centered on the enzyme's active site (determined by visual inspection). The exhaustiveness parameter was set to

200, and 30 models were generated. All these additional details are now reported in the revised methods.

C) how was the analysed pose selected? In the methods there is some discussion about this selection, but it's not quantitative enough.

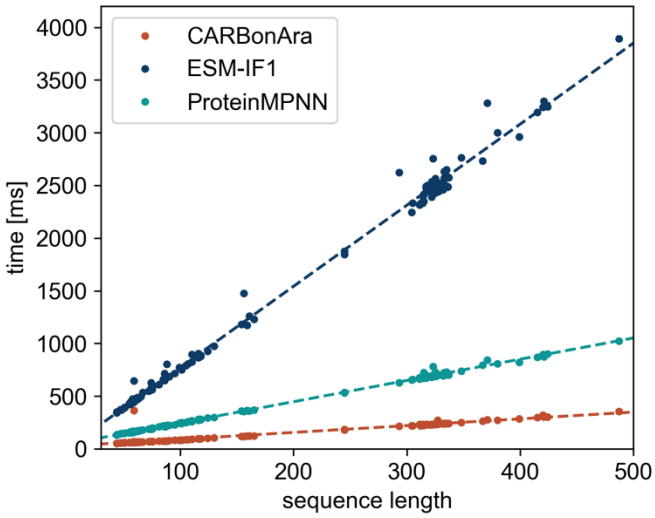
The analyzed pose was selected based on both the DDG score and the proximity of the carbonyl group of the β-lactam ring to the catalytic residue S70. We also looked for interactions between nitrocefin and residues R244 and N132, known to stabilize cephalosporin binding in TEM-1.

2) when discussing the context-aware performances of CARBonAra I would separate the glycans from the ligands. I believe glycoproteins could benefit even more than standard small molecules from the presence of the context, especially when the "context" (glycan here) itself is covalently linked to the protein.

We followed this reasonable suggestion of the reviewer, as we have also recently addressed the possibility to treat carbohydrate-protein interfaces specifically (see PeSTo-Carbs, Bibekar et al. JCTC 2024). We updated Figure 3, panel a and b where ligands and glycans are now separated, showing indeed significant differences.

3) The authors mention the GPU time needed to train CARBonAra. I do think that reporting (and maybe visualizing in a plot) the inference time would be very helpful to underline the competitiveness of the method. Maybe correlating it with the sequence length.

We quantitatively assessed in the revision the performance of our method with respect to ESM-IF1 and ProteinMPNN. As it is clear from the new supplementary figure, our method has a small advantage in terms of computational performance over the other two methods.



Supplementary Figure 3 | Run time analysis on GPU. Model run time as a function of the sequence length tested on a Nvidia RTX 2080 Ti and Intel i9-9900K. ESM-IF1 runs out of memory on the GPU with larger system so we compared the three methods on 142 structures with sequence length under 500 amino acids.

4) Alphafold2 is employed at various stages of the manuscript, from evaluating the foldability of the designed sequences to screening the obtained designs. It is usually used in single-sequence mode: why? In protein-design it is quite common to not input the multiple sequence alignment, but I think this should be explained.

In our study, we employed AlphaFold2 in single-sequence mode to evaluate and validate the structural integrity of the designed protein sequences. Primarily, the use of single-sequence mode allows for streamlined computational efficiency and simplification of the process by avoiding the computation of multiple sequence alignments (MSAs). Additionally, single-sequence predictions are shown to correlate well with experimental success, offering a convenient metric for assessing the foldability and functional potential of new designs [Bennett, N.R., Coventry, B., Goreshnik, I. *et al.* Improving de novo protein binder design with deep learning. *Nat Commun* **14**, 2625 (2023)]. We better explained this point in the revised text.

Response to Reviewer #3

Although I didn't review the whole code, I have some observations that I hope the authors will consider.

We would like to thank the reviewer for the critical assessment of our work.

1) I have to commend the authors for making the conda environment setup very simple and straightforward.

2) I suggest that the authors add way more information to the README file

We spent significant time in the effort to improve the README file with more details on how to install and use the software. We think that this new version will allow the users more convenient and easier use of our method.

3) understanding how to run the program is not very intuitive, and I think that implementing one or more command line interfaces to launch the different design tasks on an input folder (or directly on a pdb file) would be very helpful. As an example, having something like "carbonara-design examples/2oob.pdb" doing the job of apply_model.ipynb

We implemented and added a command line tool to use the software more easily. Here is in an example of command line to generate 100 sequences using a PDB file as input scaffold:

carbonara --num_sequences 100 --imprint_ratio 0.5 examples/pdbs/2oob.pdb outputs

We provide more examples and details on the GitHub repository page to produce sequences with different fixed parts, using different sampling methods and interpreting the output.

4) while running the model on a couple of dimers I encountered the following error "RuntimeError: Number of dimensions of repeat dims can not be smaller than number of dimensions of tensor". I don't know exactly what it means but it would be nice if the error message was somehow more informative. An example dimer I tried was pdb 8cyj, chains B and R.

To reply to this point, we added several examples and a notebook as a quick start guide with examples of usage with files and expected outputs.

REVIEWERS' COMMENTS

Reviewer #1 (Remarks to the Author):

The authors have thoroughly addressed my comments. This is a great piece of work.

Reviewer #3 (Remarks to the Author):

I thank the authors for answering all my questions and I fully recommend the manuscript for publication.