EasyChair Preprint

№ 8688

Geometric Algebra Models of Proteins for Three-Dimensional Structure Prediction

Alberto Pepe, Joan Lasenby and Pablo Chacon

August 17, 2022

Geometric Algebra Models of Proteins for Three-Dimensional Structure Prediction

No Author Given

No Institute Given

Abstract. A protein can be regarded as a chain of amino acids with unique folding in the three-dimensional (3D) space. Knowing the folding of a protein is highly desirable since the folding controls the protein properties. However, determining it experimentally is expensive and time consuming: estimating the 3D structure of a protein computationally - known as protein structure prediction (PSP) - can overcome these issues. In this paper, we explore the advantage of using Geometric Algebra (GA) to model proteins for PSP applications. In particular, we employ GA to define a metric of the orientation of the amino acids in the chain. We then encode this metric in matrix form and show how patterns in these images mirror folding patterns of proteins. Lastly, we prove that this metric is predictable through a standard deep learning (DL) architecture for the inference of pairwise amino acids distances. We demonstrate that GA is a powerful tool to obtain a compact representation of the protein geometry with potential to improve the prediction accuracy of standard PSP pipelines.

Keywords: Protein Structure Prediction · Deep Learning · Geometric Algebra.

1 Introduction

The 3D structure of a protein - known as tertiary structure - is the arrangement in space of its amino acid chain - the primary structure - and it determines the protein behaviour and cellular function. Determining the structure experimentally, however, is expensive and time consuming.

For this reason, there has been a great deal of recent interest in DL algorithms to predict the protein structure starting directly from the amino acid sequence [1, 2]. By cutting time and cost and achieving unprecedented accuracies, PSP has a huge potential impact on medicine and biotechnologies. The state of the art in PSP is represented by [3]: the AlphaFold2 pipeline can directly predict the 3D coordinates of heavy atoms and reach a median backbone accuracy of 0.96 Å on the CASP14 dataset [4]. The ensemble of its neural networks takes into account evolutionary, physical and geometrical constraints of the protein structures. From a geometrical point of view, proteins are represented as a residue gas: each amino acid - also called a residue - is associated with a rigid body (triangles) for the backbone and an angle for the sidechain. Similar processing

strategies are found in [5], where 1D, 2D and 3D data are combined in a pipeline of several neural networks producing mutual predictions.

Most PSP pipelines based on DL have contact and distance maps as their end goal, which are then used to predict the protein structure. However, in [6], it has been demonstrated that adding orientational information improves the accuracy of the structure prediction: adding angle maps (three in total, one for each dihedral angle associated with a residue) can improve the precision of the top L long-range contacts of up to 2.2% on the CASP13 dataset.

In this paper, we propose a single map based on a GA description of the protein geometry which is intuitive, compact, descriptive of the protein folding and easily predictable compared to standard angle maps, with the potential of simplifying both the protein modeling and the complex PSP pipelines.

The rest of the paper is structured as follows: in Section 2, the fundamentals of Conformal GA are introduced. In Section 3, the proposed protein model is presented and the GA cost and cost maps are introduced. In Section 4, the prediction algorithm and strategy are presented, while in Section 5 the prediction results are shown. Lastly, in Section 6, conclusions are drawn.

2 Conformal Geometric Algebra

Conformal Geometric Algebra (CGA) extends a GA $\mathcal{G}_{p,q,r}$ of dimension $n = p + q + r$ to $\mathcal{G}_{p+1,q+1,r}$ by introducing two basis vectors, e and $\bar{e}$, with $e^2 = +1$ and $\bar{e}^2 = -1$. Having introduced e and $\bar{e}$, we can compose the vectors

$$\begin{aligned} n_\infty &= e + \bar{e} \\ n_0 &= \frac{1}{2}(\bar{e} - e) \end{aligned} \tag{1}$$

which help define a mapping of the kind

$$x \in \mathcal{G}_{p,q,r} \longrightarrow F(x) \in \mathcal{G}_{p+1,q+1,r} \tag{2}$$

in which $F(x)$ is defined as

$$\begin{aligned} F(x) &= -\frac{1}{2}(x - e)n_\infty(x - e) \\ F(x) &= \frac{1}{2}(x^2 n_\infty + 2x - n_0) \end{aligned} \tag{3}$$

In the case in which we are dealing with a 3D space (i.e. $\mathcal{G}_{3,0,0}$), the equivalent CGA will be $\mathcal{G}_{4,1,0}$, i.e. a five-dimensional space. When working in CGA, point pairs, lines, planes, circles and spheres are all conveniently represented by blades in the 5D CGA. A summary is provided in Table 1.

3 CGA in Protein Geometry

3.1 Cost Function

A protein can be simplified into a backbone chain and several side chains. The backbone is responsible for the 3D shape of the protein, and it is composed of

Table 1. Objects in CGA

Grade	Symbol	Object
1	A	point
2	A ∧ B	point pair
3	A ∧ B ∧ C	circle ($\mathcal{C}$)
3	A ∧ B ∧ n_∞	line (L)
4	A ∧ B ∧ C ∧ D	sphere (Σ)
4	A ∧ B ∧ C ∧ n_∞	plane (Π)

a series of carbon, nitrogen, and oxygen atoms. The α-carbons are the main feature of the backbone, to which the side chains that differentiate each amino acid are bonded. Each α-carbon is preceded by a nitrogen atom and followed by a carbon atom. Hence, to each amino acid i we can associate a triplet of atoms $\{N, C_\alpha, C\}_i$.

Each $\{N, C_\alpha, C\}$ triplet lies on a plane, constraining the protein folding (see Fig. 1). We can hence conveniently model a protein backbone in CGA so any three $\{N, C_\alpha, C\}$ atoms will lie on a plane (not too dissimilar to the residue gas of [3]): let A_i, B_i and C_i be the Euclidean coordinates expressed in Conformal space of the atoms $\{N, C_\alpha, C\}_i$, respectively. The plane associated with residue i can be expressed as the 4-blade:

$$\Pi_i = A_i \wedge B_i \wedge C_i \wedge n_\infty \tag{4}$$

Given two planes Π_i, Π_j corresponding to the amino acids i, j, we can compute the rotor that brings one to the other as described in [7]:

$$R_{ij} = \frac{1}{\sqrt{\langle K \rangle_0}}(1 - \Pi_i \Pi_j) \tag{5}$$

where $K = 2 - (\Pi_i \Pi_j + \Pi_j \Pi_i)$ and $\langle \cdot \rangle$ is the grade projector operator. We now use the cost function $C_\lambda(R)$ that measures how much the rotor R varies from the identity, as defined in [8]. $C_\lambda(R)$ is a weighted sum of a translational and a rotational term:

$$C_{\lambda_1 \lambda_2}(R) = \lambda_1 \langle R_\parallel \tilde{R}_\parallel \rangle_0 + \lambda_2 \langle (R_\perp - 1)(\tilde{R}_\perp - 1) \rangle_0 \tag{6}$$

in which the translational error is represented by $R_\parallel = R \cdot e$, and the rotational error by $\langle (R_\perp - 1)(\tilde{R}_\perp - 1) \rangle_0 = \langle (R - 1)(\tilde{R} - 1) \rangle_0$. As we are interested in an orientational feature, we will focus exclusively on the rotational part (case $\lambda_1 = 0, \lambda_2 = 1$).

3.2 Cost Maps

Inter-residue interactions are commonly represented as matrices - also called maps. A contact map $\mathbf{C}$ of a protein consisting of M residues, for example, is a binary $M \times M$ matrix of the type:

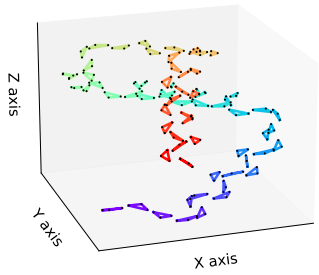

Fig. 1. First 70 $\{N, C_\alpha, C\}$ planar triplets of the haemoglobin backbone.

$$\mathbf{C}_{ij} = \begin{cases} 1 & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where d_{ij} is the distance between residues i, j expressed in Å measured as the Euclidean distance between the C_α coordinates of residues i and j. A cost map can be interpreted as: two residues are in contact if they are within a certain distance from each other. A more informative metric, usually real valued, is given by distance maps, which are similarly defined as:

$$\mathbf{D}_{ij} = d_{ij} \tag{8}$$

From either or both contact and distance maps it is possible to obtain accurate 3D shape estimation. However, when contact or distance maps are predicted and not exact, errors are introduced into the 3D reconstruction step. Having an additional map grasping the orientation between residues can help to further constrain the search space for the protein folding. We can hence employ our cost function to produce a cost map which contains orientational information as follows:

$$\mathbf{M}_{ij} = \begin{cases} \mathrm{C}_{\lambda_1 \lambda_2}(R_{ij}) & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

3.3 Examples

A comparison between contact map $\mathbf{C}$, distance map $\mathbf{D}$ and cost map $\mathbf{M}$ is given in Fig. 2 for an example protein

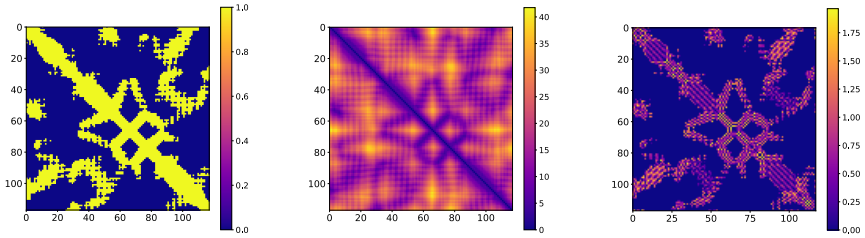

Fig. 2. From left to right: contact, distance and cost map for protein 2HC5A.

It is possible to establish a relation between patterns in cost maps and the protein secondary structure. By secondary structure we refer to the local folding of a segment of a protein, e.g. α-helices, β-sheets or turns). Secondary structure information is a common feature in PSP pipelines and one of the most important in predicting distance and contact maps, as shown in [5, 9].

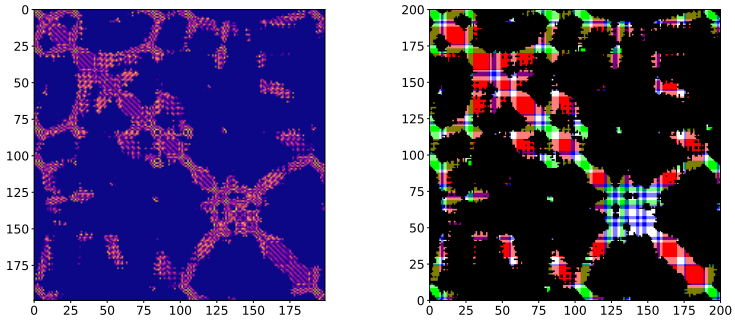

Fig. 3. Cost map (left) and secondary structure (right) for protein 4JZK. Visualizing the first 200 residues.

By assigning a colour to each secondary structure, it is possible visualize the secondary structure of each amino acid pair. We arbitrarily assigned red to α-helices, green to β-sheets, blue to turns and white to all the others. Any combination of these four colours gives the possible secondary structures of the pair, for a total of 10 different colour combinations. As shown in Fig. 3, it is possible to find a clear correspondence between secondary structures and patterns in the cost maps. To the best of our knowledge, this is the first example of an orientational map that also encodes the secondary structure of the protein.

4 Predicting Cost Maps

We verified the predictability of our cost maps by employing a deep residual network as presented in [9]. We will refer to both the network and the associated dataset as PDNET.

4.1 PDNET

PDNET is residual neural network composed of 128 blocks. Each residual block consists of a batch normalization layer, a ReLU activation function, a 2D convolutional layer with 3×3 kernel, a dropout layer with $\alpha = 0.3$, a ReLU activation function, and a 2D convolutional layer, for a total of ~ 9.5M tunable parameters.

PDNET was originally designed to predict either: (i) contact maps, (ii) binned distance maps or (iii) real-valued distance maps. We demonstrate that from the same features and with the same architecture originally presented in [11], cost maps can also be estimated. The task of distance map prediction is comparable to the problem of depth estimation: the three RGB channels of a colour image are replaced by tens of feature matrices derived from the amino acid sequence, and the depth map is replaced by the distance map.

Specifically, the total number of channels is $N = 57$, corresponding to 7 features: position specific scoring matrix (PSSM), secondary structure, entropy, FreeCon, CCMPred, surface area and potential energy. Of these CCMpred, FreeCon and potential energy are pairwise features, the rest are 1D features relative to a single amino acid. The 1D features are encoded twice as identical columns and rows for each amino acid in the sequence. The features are identical to those of the PDNET dataset of [9], which includes a more detailed description of their biochemical meaning. They are either derived from previous DL based prediction or multiple sequence alignment queries.

When PDNET is employed to predict real valued distances, it employs the reciprocal logcosh as a loss function:

$$L_{\mathbf{D}}^{(i)} = \log\left(\cosh\left(\frac{K}{\mathbf{D}_P^{(i)} + \epsilon} - \frac{K}{\mathbf{D}_T^{(i)} + \epsilon}\right)\right) \tag{10}$$

where $\mathbf{D}_P^{(i)}$ is the predicted distance matrix, $\mathbf{D}_T^{(i)}$ the true distance matrix, ϵ a small positive number and K is a scalar set equal to 100. The inverse of the maps is taken in order to prioritize short-range interaction, for which higher accuracy is desirable, over long-range interaction, which is less relevant in terms of the overall 3D structure. The loss is evaluated pixel by pixel and summed over the total number of pixels.

4.2 Training Details

The GA-based cost maps are also real valued and bounded in the range $[0, 2]$, as we verified empirically by evaluating $C_{\lambda_1,\lambda_2}(R)$. However, since the cost does

not increase for residues further away as it is a purely orientational measure, we changed the loss to be:

$$L_{\mathbf{M}}(i) = \log\left(\cosh\left(\mathbf{M}_P^{(i)} - \mathbf{M}_T^{(i)}\right)\right) \tag{11}$$

where $\mathbf{M}_P^{(i)}, \mathbf{M}_T^{(i)}$ are the predicted and true cost maps for protein (i) in the training set, respectively.

For training the network, we kept the features unchanged from those of PDNET, namely a stack of images of the type $\{\mathbf{X}^{(i)}\}_{i=1}^{N}$, with $N = 57$ and $\mathbf{X}^{(i)} \in \mathbb{R}^{M\times M}$, in which M is the length of the protein sequence. The change comes in substituting the target $\mathbf{D}_T \in \mathbb{R}^{M\times M}$ - the true, real-valued distance maps, with $\mathbf{M}_T \in \mathbb{R}^{M\times M}$ - the true, real-valued cost maps, obtained from the protein coordinates in the protein database [10]. Again, the loss is evaluated per pixel.

The training set has been kept to 1000 proteins from the DEEPCOV dataset, and the testing set to 150 proteins from the PSICOV dataset, as in the original PDNET pipeline.

The code has been implemented using the Keras API of Tensorflow for the Machine Learning modules, the Clifford library for operations in Geometric Algebra and the PDB Module of the Biopython library for handling protein data. The code was written in the form of Jupyter Notebooks on Google Colaboratory and all the experiments have been run on an NVIDIA Tesla K80 GPU. All the scripts and data are available upon request to the authors.

We considered scenarios (see Fig. 4): (a) predicting cost maps with 57 feature channels (standard PDNET), (b) predicting cost maps with 57 feature channels + 1 (real) distance channel (ideal case, as distance maps would not be available), (c) predicting cost maps with 57 feature channels + 1 (predicted) distance channel also via PDNET (realistic case, as distance maps also need to be predicted in PSP).

5 Results

We evaluated two metrics, namely: (i) mean absolute error (MAE), as in common regression problems, and (ii) structural similarity index (SSIM) between $\mathbf{M}_P, \mathbf{M}_T$, since a low MAE does not necessarily mean that the patterns in the cost maps are captured successfully. The MAE is measured in Å, while the SSIM ranges between $[0, 1]$, with SSIM = 1 meaning fully similar matrices and SSIM = 0 fully dissimilar matrices. They are defined as follows:

$$MAE(\mathbf{M}_P, \mathbf{M}_T) : \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}|\mathbf{M}_{Pij} - \mathbf{M}_{Tij}| \tag{12}$$

$$SSIM(\mathbf{M}_P, \mathbf{M}_T) : \frac{(2\mu_{\mathbf{M}_P}\mu_{\mathbf{M}_T} + c_1)(2\sigma_{\mathbf{M}_P\mathbf{M}_T} + c_2)}{(\mu_{\mathbf{M}_P}^2 + \mu_{\mathbf{M}_T}^2 + c_1)(\sigma_{\mathbf{M}_P}^2 + \sigma_{\mathbf{M}_T}^2 + c_2)} \tag{13}$$

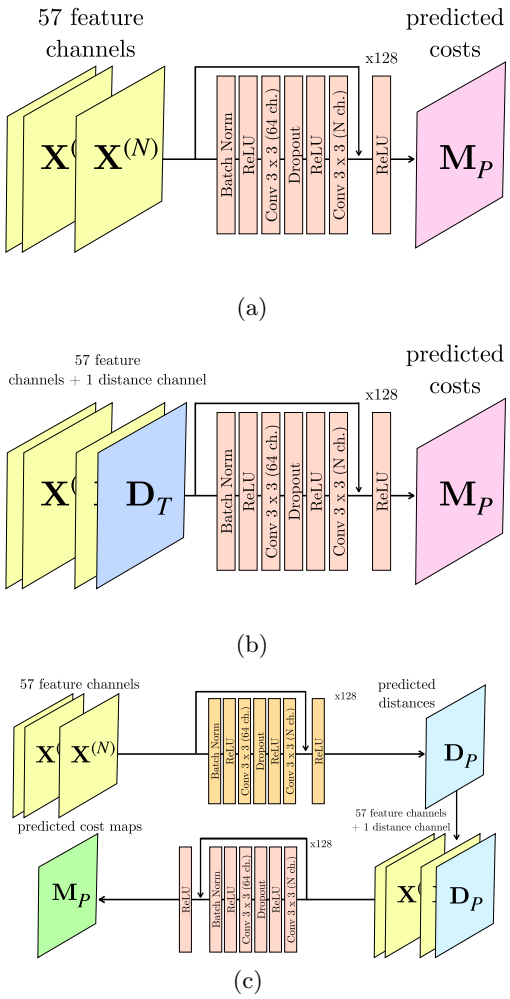

Fig. 4. The three processing schemes: (a) predicting costs from PDNET; (b) predicting costs from PDNET + true distance maps; (c) predicting costs from PDNET + predicted distances, themselves predicted from PDNET.

with $\mu_{\mathbf{M}_T}$ being the mean of $\mathbf{M}_T$, $\mu_{\mathbf{M}_P}$ the mean of $\mathbf{M}_P$, $\sigma_{\mathbf{M}_P\mathbf{M}_T}$ the covariance of $\mathbf{M}_P$ and $\mathbf{M}_T$, $\sigma^2_{\mathbf{M}_P}$ the variance of $\mathbf{M}_P$, $\sigma^2_{\mathbf{M}_T}$ the variance of $\mathbf{M}_T$, $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ with $k_1 = 0.01, k_2 = 0.03$ and L being the dynamic range, set to $L = 255$.
Results are summarized in Tables 2-3.

Table 2. MAE between original and predicted cost maps (Å)

	no distance			with distance			with pred. distance		
	Max	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**
DEEPCOV (val)	0.1080	0.0218	0.0009	0.0418	0.0108	0.0005	0.0607	0.01825	0.0005
PSICOV (test)	0.0342	0.0158	0.0029	0.0275	0.0125	0.0028	0.0327	0.01490	0.0029

Table 3. SSIM between original and predicted cost maps.

	no distance			with distance			with pred. distance		
	Max	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**
DEEPCOV (val)	0.9946	0.9041	0.4990	0.9986	0.9652	0.8387	0.9991	0.9360	0.7442
PSICOV (test)	0.9937	0.9431	0.8592	0.9941	0.9632	0.9130	0.9936	0.9519	0.8851

It can be noticed that cost maps are indeed predictable based on features commonly used to predict distances. However, when predicting cost maps without distance information, only close range contacts (i.e. the pixels close to the diagonal) are predicted accurately. Adding predicted distance information, on the other hand, allows us to significantly improve the prediction of the patterns in cost maps, with a mean MAE decrease by 16.3% for the training set and by 5.7% for the testing set. The average SSIM increased by 3.5% and by 1% for the training and testing sets, respectively. The better the prediction of the distance information (i.e., the closer the predicted distance maps are to the original ones), the higher the improvement on cost prediction.
Examples of the predicted cost maps in comparison with the original cost maps over the testing set are given in Fig. 5.

Lastly, we evaluated the permutation feature importance (PFI) to rank the most relevant features in the prediction of cost maps. We did so by training the network by permuting one feature at a time and then taking the ratio of our metric with and without permutation of that feature. By permutation we refer to the shuffling of a single feature across the training set, meaning that when we evaluate the PFI for feature n, each protein will have associated an erroneous feature n belonging to a different protein during training, while leaving the

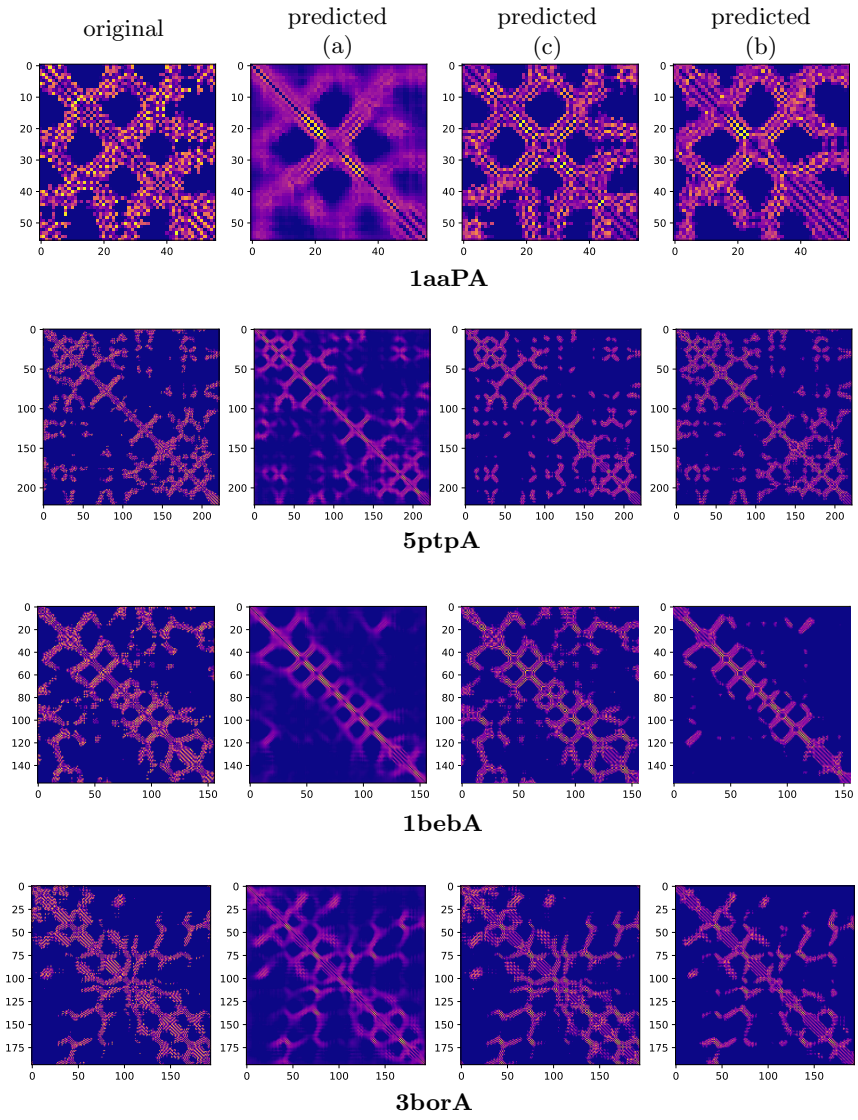

Fig. 5. Examples of the GA cost map for four protein chains predicted with the three approaches (a,b,c) of Fig. 4. Note how adding distances significantly improves the quality of the prediction.

testing set unchanged. We then measured the PFI of feature n as:

$$PFI^{(n)}_{MAE} = \frac{MAE(\mathbf{M}_P, \mathbf{M}_T)}{MAE^{(n)}(\mathbf{M}_P, \mathbf{M}_T)} \tag{14}$$

$$PFI^{(n)}_{SSIM} = \frac{SSIM^{(n)}(\mathbf{M}_P, \mathbf{M}_T)}{SSIM(\mathbf{M}_P, \mathbf{M}_T)} \tag{15}$$

In which $f(\mathbf{M}_P, \mathbf{M}_T)$ is the metric f measured with standard training procedure, and $f(\mathbf{M}_P, \mathbf{M}_T)^{(n)}$ is the metric f measured when permuting feature n during training.

Results for the validation set (DEEPCOV) and for two testing set (PSICOV and CAMEO HARD) are shown in Figure 5. The PSSM and secondary structures appear to be the two most relevant features, a result which mirrors that found for distance maps in [9]. This is in agreement with the findings of Section 3, where we saw the close relationship between cost patterns and secondary structures.

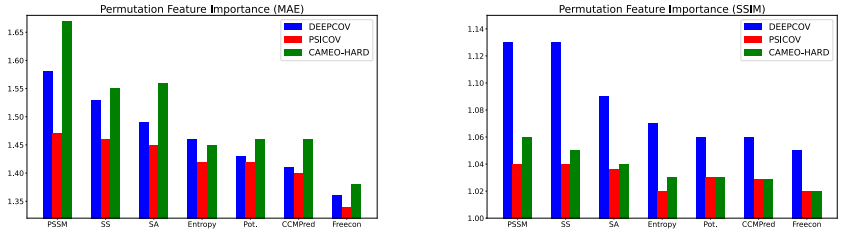

Fig. 6. Permutation Feature Importance over MAE and SSIM for each of the 7 features across validation and test sets.

6 Conclusions and Future Work

In this paper, we have introduced a new feature based on GA describing the relative amino acid orientation for PSP applications. We firstly presented the criterion behind the modeling of a protein backbone as a collection of planes. We then evaluated the rotor between each pair of planes and associated a cost to it. The pairwise costs were then arranged in matrix form to produce cost maps. We proceeded to show how patterns in cost maps can be directly associated to the protein secondary structure and verified how standard features and algorithms employed in PSP to predict distance maps can also be used to predict our proposed GA cost maps. Adding distance information - even if only predicted - can further improve the predicted cost maps in terms of MAE and SSIM.

Our cost maps therefore constitute a useful tool for protein modelling and may provide new orientation-based features that could improve the the final

3D structure prediction.We believe that GA could hence constitute a successful tool to model proteins and provide new orientational features that can improve the precision of the 3D structure prediction and reduce the number of required features.

Future work might include employing predicted costs, along with feature and distance maps, to predict the 3D coordinates of C_α atoms in the protein backbone on the basis of [5, 12] and verify whether the cost maps can further constrain the search space and improve the accuracy of the 3D coordinates, or employing different GA modeling choices and hence new costs, and verify whether they can capture different protein features compared to the cost proposed in this paper.

References

1. AlQuraishi M. Machine learning in protein structure prediction. Current opinion in chemical biology. 2021 Dec 1;65:1-8.
2. Pearce R, Zhang Y. Deep learning techniques have significantly impacted protein structure prediction and protein design. Current opinion in structural biology. 2021 Jun 1;68:194-207.
3. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A. Highly accurate protein structure prediction with AlphaFold. Nature. 2021 Aug;596(7873):583-9.
4. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., Moult, J. (2021). Critical assessment of methods of protein structure prediction (CASP)—Round XIV. Proteins: Structure, Function, and Bioinformatics, 89(12), 1607-1617.
5. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, Wang J, Cong Q, Kinch LN, Schaeffer RD, Millán C. Accurate prediction of protein structures and interactions using a three-track neural network. Science. 2021 Aug 20;373(6557):871-6.
6. Yang J, Anishchenko I, Park H, Peng Z, Ovchinnikov S, Baker D. Improved protein structure prediction using predicted interresidue orientations. Proceedings of the National Academy of Sciences. 2020 Jan 21;117(3):1496-503.
7. Lasenby J, Hadfield H, Lasenby A. Calculating the rotor between conformal objects. Advances in Applied Clifford Algebras. 2019 Nov;29(5):1-9.
8. Eide, E.R., Master's Degree Thesis, University of Cambridge, Camera Calibration using Conformal Geometric Algebra, 2018
9. Adhikari B. A fully open-source framework for deep learning protein real-valued distances. Scientific reports. 2020 Aug 7;10(1):1-0.
10. Burley, S. K., Berman, H. M., Kleywegt, G. J., Markley, J. L., Nakamura, H., Velankar, S. (2017). Protein Data Bank (PDB): the single global macromolecular structure archive. Protein Crystallography, 627-641.
11. Adhikari B. DEEPCON: protein contact prediction using dilated convolutional neural networks with dropout. Bioinformatics, 2020, 36.2: 470-477.
12. Costa A, Ponnapati M, Jacobson JM, Chatterjee P. Distillation of MSA Embeddings to Folded Protein Structures with Graph Transformers. bioRxiv. 2021 Jan 1.