

Enhancing protein backbone angle prediction by using simpler models of deep neural networks

Author

Mataeimoghadam, Fereshteh, Newton, MA Hakim, Dehzangi, Abdollah, Karim, Abdul, Jayaram, B, Ranganathan, Shoba, Sattar, Abdul

Published

2020

Journal Title

Scientific Reports

Version

Version of Record (VoR)

DOI

[10.1038/s41598-020-76317-6](https://doi.org/10.1038/s41598-020-76317-6)

Rights statement

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

Downloaded from

<http://hdl.handle.net/10072/402535>

Griffith Research Online

<https://research-repository.griffith.edu.au>



OPEN

Enhancing protein backbone angle prediction by using simpler models of deep neural networks

Fereshteh Mataeimoghadam^{1,7}✉, M. A. Hakim Newton^{1,2,7}✉, Abdollah Dehzangi^{3,4}, Abdul Karim¹, B. Jayaram⁵, Shoba Ranganathan⁶ & Abdul Sattar^{1,2}

Protein structure prediction is a grand challenge. Prediction of protein structures via the representations using backbone dihedral angles has recently achieved significant progress along with the on-going surge of deep neural network (DNN) research in general. However, we observe that in the protein backbone angle prediction research, there is an overall trend to employ more and more complex neural networks and then to throw more and more features to the neural networks. While more features might add more predictive power to the neural network, we argue that redundant features could rather clutter the scenario and more complex neural networks then just could counterbalance the noise. From artificial intelligence and machine learning perspectives, problem representations and solution approaches do mutually interact and thus affect performance. We also argue that comparatively simpler predictors can more easily be reconstructed than the more complex ones. With these arguments in mind, we present a deep learning method named Simpler Angle Predictor (SAP) to train simpler DNN models that enhance protein backbone angle prediction. We then empirically show that SAP can significantly outperform existing state-of-the-art methods on well-known benchmark datasets: for some types of angles, the differences are 6–8 in terms of mean absolute error (MAE). The SAP program along with its data is available from the website <https://gitlab.com/mahnewton/sap>.

Protein structure prediction (PSP) has been an unsolved problem for the last half century¹. Three dimensional structures of most proteins depend on their amino acid (AA) sequences. The PSP problem is to determine the three dimensional structures of given proteins just from their amino acid sequences. The difficulties come from the inevitability of searching an astronomically large conformation space and from the absence of a highly accurate energy function to evaluate potential protein conformations².

There are 20 types of amino acids. A protein might have any of the 20 types of amino acids any number of times in any order subject to stoichiometric constraints³. Each amino acid has three common atoms N , C_α and C among others. The C and N atoms of every two consecutive amino acids in a protein form a peptide bond and thus we obtain the backbone or main chain of the protein. As shown in Fig. 1, protein backbone structures can essentially be represented by dihedral angles ϕ , ψ , and ω , which are respectively defined by taking every four consecutive atoms from the sequence C_{i-1} , N_i , $C_{\alpha,i}$, C_i , N_{i+1} , $C_{\alpha,i}$. Typically ω is fixed at 180° for majority proteins⁴, and so only ϕ and ψ are to be determined. Besides being the parts of the main chain, each amino acid, starting from its C_α atom, has a side chain as well. The side chains have their own dihedral angles, but for this work we consider them to be out of scope. Once the backbone structures could be predicted with very high accuracy, side chain angles could be predicted or determined later. Besides ϕ , ψ , and ω angles, as shown in Fig. 1, θ and τ angles provide an alternative representation for protein backbone structures. While θ is a planar angle defined by three consecutive C_α atoms, τ is a dihedral angle defined by four consecutive C_α atoms. Such a representation is actually possible because of the nearly constant distance between consecutive C_α atoms. While ϕ and ψ are dihedral angles each involving four atoms from two consecutive residues, θ and τ involving three or four residues

¹School of Information and Communication Technology, Griffith University, Nathan, QLD, Australia. ²Institute of Integrated and Intelligent Systems, Griffith University, Nathan, QLD, Australia. ³Department of Computer Science, Rutgers University, Camden, NJ, USA. ⁴Center for Computational and Integrative Biology, Rutgers University, Camden, NJ, USA. ⁵Department of Chemistry and School of Biological Sciences, IIT Delhi, Delhi, India. ⁶Department of Chemistry and Biomolecular Sciences, Macquarie University, Macquarie Park, NSW, Australia. ⁷These authors contributed equally: Fereshteh Mataeimoghadam and M. A. Hakim Newton. ✉email: fereshteh.mataeimoghadam@griffithuni.edu.au; mahakim.newton@griffith.edu.au

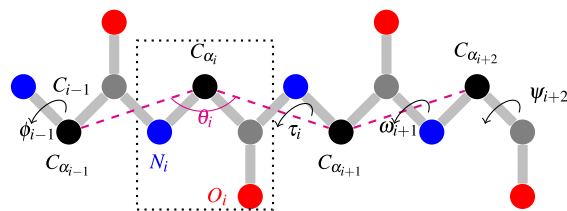


Figure 1. Backbone angles of a protein structure.

capture more local structures in a protein. In this work, we predict all the four types of backbone angles ϕ , ψ , θ and τ for each protein in a given protein using deep neural networks (DNN).

Prediction of protein backbone structures is very important since both template-based and template-free protein structure prediction rely strongly on that^{2,5}. From an abstraction based perspective, protein backbone structure prediction could be viewed as prediction of secondary structures (SSs). Protein secondary structure prediction has obtained significant success over the years through the use of various types of deep neural networks and their ensembles^{6–12} and ab initio methods¹³. For example, SSpro8¹⁴ achieves 79% accuracy on proteins with no homologs in the Protein Data Bank (PDB) and of 92% accuracy on proteins where homologs can be found in the PDB. However, this progress does not necessarily make backbone angle prediction trivial. With accurate SS predictions, one can obtain narrow ranges (about 20°) of ϕ and ψ angles, but only for helices and sheets. For coils, ϕ and ψ can take any value in $[-180, +180]$ and coils comprise about 40% residues in average proteins¹⁵. Moreover, errors in backbone angle prediction in one part of a protein has a cascaded effect in the construction of the entire protein structure. Overall, secondary structures, on one hand, are coarse-grained description of protein local structures in three (helices, sheets, and coils) or eight discrete states (including some variants of the three). On the other hand, secondary structures are somewhat arbitrarily defined with coils essentially having no well-defined structures. In contrast to secondary structures, backbone angles as being continuous variables can represent protein structures at greater accuracy levels. Moreover, predicted backbone dihedral angles, compared to the predicted secondary structures, have been found to be more useful in ab initio structure prediction or refinement by performing search^{16,17}. Protein backbone angle prediction has improved over the years. A number of methods have been developed to predict ϕ and ψ as both discrete^{18,19} and continuous^{9,20–27} labels.

Protein backbone angle prediction methods in recent years are mostly based on DNNs and their complex variants such as stacked sparse auto-encoder neural networks²³, long short-term memory (LSTM) bidirectional recurrent neural networks (BRNNs)^{6,25,27}, and Residual Networks (ResNets)²⁷, and their ensembles^{6,27} or layered iterations²⁴. In terms of input features, position specific scoring matrices (PSSM) produced by PSI-BLAST²⁸ have been used by most methods^{9,23–25,27}. Moreover, 7 physicochemical properties (7PCP) such as steric parameter (graph shape index), hydrophobicity, volume, polarisability, isoelectric point, helix probability, and sheet probability²⁹ have been used as well^{9,23–25,27}. Other input features that have been used include accessible surface area (ASA)²³, Hidden Markov Model (HMM) profiles^{9,27,30} produced by HHBlits³¹, contact maps²⁷, and PSP19⁶. In order to capture local structures around each given amino acids, sliding windows with various sizes have been used^{23–25}. Moreover, to capture the non-local or long-range interactions among amino acids in a protein, the entire protein sequence has been used as features^{9,24,26} or convolutional neural networks (CNNs)^{6,30} or LSTM-BRNNs^{25,27} have been used. In terms of datasets to be used to evaluate the prediction models, we refer to four datasets: PISCES³², SPOT-1D^{27,33}, PDB150³⁴ and CAMEO93³⁵. The first two datasets have respectively about 5.5K and 12.5K proteins with 1.2M and 2.7M residues. The last two datasets respectively have 150 and 93 proteins and have been used mainly for independent testing.

Given the literature explored above, we observe that in the protein backbone angle prediction research, there is an overall trend to employ more and more complex neural networks and then to throw more and more features to the neural networks. While more features might add more predictive power to the neural network, we argue that redundant features rather clutter the scenario and more complex neural networks then just counterbalance the noise. Similar results have been reported in other research areas. For example, in a Nature article in seismic aftershock prediction by deep learning methods³⁶, a simple two-parameter logistic regression (that is, one neuron) is shown to have obtained the same performance as that of the 13,451-parameter DNN. From artificial intelligence and machine learning perspectives, problem representations and solution approaches do mutually interact and thus affect performance. Nevertheless, we also argue that comparatively simpler predictors can more easily be reconstructed than the more complex ones. With these arguments in mind, we present a deep learning method named Simpler Angle Predictor (SAP) to train simpler DNN models that enhance protein backbone angle prediction. We then empirically show that SAP can significantly outperform the existing state-of-the-art methods SPOT-1D and OPUS-TASS⁶ on well-known benchmark datasets: for ψ and τ , the differences are 6–8 in terms of mean absolute error (MAE). With an ensemble of several types of DNNs using many input features, SPOT-1D and OPUS-TASS are very complex prediction methods compared to the SAP, which uses just a fully connected DNN and a few input features. The SAP program along with its data is available from the website <https://gitlab.com/mahnewton/sap>.

Methods

In this section, we describe the deep learning model proposed in this paper and the datasets used in this work.

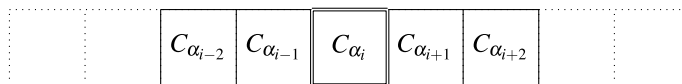


Figure 2. Sliding window of size 5: two residues on each side of a given residue.

Input features. As shown in Fig. 2, we use a sliding window of size W : up to $\frac{W}{2}$ amino acids at each side of a given amino acid. Depending on the window sizes, sliding windows can capture short or long range interactions between residues and secondary structures. Some backbone angle prediction methods that use recurrent neural networks (RNN) and CNNs take the whole protein sequences as input to capture interactions in the entire protein. However, with the absence of a firmly known energy function, it is not clear whether very long range interactions are really effective. So any choices regarding using sliding windows versus using entire proteins are to be made based on empirical evaluation. To make it clearer, in any distance-based energy components e.g. Lennard–Jones or charge-based potentials, the values are in effect zero after a certain distance. Moreover, if we look at the state-of-the-art backbone angle prediction method SPOT-1D, we see, besides using entire proteins, it still uses windowing to capture contact information. Our intent in this work is to explore simple models that can still achieve very good accuracy levels.

While window size effectively ensures context dependence of assumed local conformations, arguably there is not enough data in the training set, even in the protein data bank, to cover all possible combinations of amino acids (e.g. 20^5) with a given window size (e.g. 5). So the context has to be captured via a 3-state a 8-state model that can specify the average range of angle values for each amino acid in a given protein. The data deficiency for larger windows even further spoils the training. In this work, for each amino acid, we consider one of the 8 values G, H, I, T, S, E, B, and C to represent predicted 8-state SS and then encode that using an one-hot vector. The 8-state SS prediction is obtained by running SSpro8¹⁴ on each protein. The training set of SSpro8 comprises 5772 proteins that are released before August 20, 2013. SSpro8 uses sequence similarity and sequence-based structural similarity in SS prediction and achieves respectively 92% and 79% accuracy on proteins with and without homologs in the PDB. On one hand, we have already discussed that these highly accurate SS predictions do not necessarily solve the backbone angle prediction problem when high quality protein structures are to be constructed. On the other hand, we note that we have removed all SSpro8's training proteins from our training, validation, and tests sets and also use BLAST²⁸ for this purpose with e-value 0.01. In this aspect, our method differs from the state-of-the-art backbone angle predictor SPOT-1D, which uses homologous sequences to generate its HMM-based features.

For each amino acid, we consider 20 values obtained from the PSSM matrix generated by three iterations of PSI-BLAST²⁸ against the UniRef90 sequence database updated in April 2018. We also use 7PCP (seven physico-chemical properties) and ASA, and experiment with their various combinations. These features are very common in the literature.

In summary, we have $20 + 8 = 28$ PSSM and SS features plus various combinations of 7 or 1 feature values for 7PCP or ASA for each amino acid residue in each protein. This will be multiplied by the size of the sliding window used. We experiment with sliding windows of sizes 1, 5, 9, 13, 17, 21 as SPIDER²³ tried up to size 21.

Predicted outputs. We consider 4 outputs, one for each of ϕ , ψ , θ , and τ angles. Each ϕ and ψ can be associated with exactly one residue or C_{α} . A θ angle involving $C_{\alpha_{i-1}}$, C_{α_i} , $C_{\alpha_{i+1}}$ is associated with C_{α_i} . Similarly, a τ angle involving $C_{\alpha_{i-1}}$, C_{α_i} , $C_{\alpha_{i+1}}$, $C_{\alpha_{i+2}}$ is associated with C_{α_i} . In one set of experiments, we consider these angles directly, handling their periodicity (-180° to 180°) within the loss function of the DNN used. In another set of experiments, just like the state-of-the-art method SPOT-1D, we use both sine and cosine ratios for each of the 4 angles, and thus use 8 outputs. The trigonometric ratios handle the periodicity issue of the angles and the tangent values obtained from the sine and cosine values can give the predicted angle within -180° to 180° .

DNN architecture. Figure 3 shows the DNN architecture used in our method. The DNN in fact is a fully connected neural network (FCNN) with three hidden layers, each having 150 neurons. This architecture is similar to that used in SPIDER²³ and SPIDER2²⁴. SPIDER2, however, uses a series of 3 DNNs feeding a previous DNN's output as input to the next DNN. In our experiments, we have used only one DNN with three hidden layers, although we have trialed two and four hidden layers as well and showed the results later. The inputs and the outputs of the DNN are per amino acid basis. Depending on the size of the sliding window and the combinations of 7PCP and ASA, the input layer will have different numbers of inputs. The output layer has one output for each angle when we want to predict an angle directly. However, if we consider sine and cosine ratios of an angle and consequently later calculate the angle, then the output layer will have two outputs for each angle.

DNN implementation. The DNN has been implemented in Python language using Keras library and SGD optimiser with momentum 0.9. The learning rate starts from 0.01 and if the loss function does not improve in 3 iterations, then learning rate is reduced by a factor 0.5 until it reaches 10^{-15} . The activation function is linear in the output layer and sigmoid in the input and the hidden layers. The kernel initialiser is `glorot_uniform`. We run programs on NVIDIA Tesla V100-PCIE-32GB machines.

Benchmark datasets. We briefly describe the dataset used by SPOT-1D²⁷. This dataset has 12450 proteins that were culled from PISCES³² on Feb 2017 with the constraints of high resolution ($< 2.5\text{\AA}$), an R-free < 1 ,

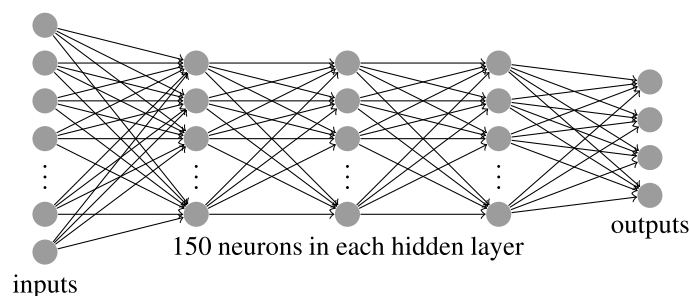


Figure 3. The fully connected deep neural network used in our method. It has three hidden layers, each having 150 neurons. The numbers of inputs and outputs could vary depending on the combinations of features used (e.g. PSSM plus SS and combinations of 7PCP and ASA) and the representation of the output angles (Direct Angles vs Trigonometric Ratios).

Datasets	Training	Validation	Testing	Total
Proteins	6721	667	1206	8594
Residues	1,670,605	165,530	282,461	2,118,596

Table 1. Numbers of proteins and residues in training, validation, and testing datasets. 27 testing proteins are in TEST2018 and 1179 are in TEST2016.

and a sequence identity cutoff of 25% according to BlastClust²⁸. Among those proteins, 1250 proteins deposited after June 2015 were separated into an independent test set, leaving 11200 proteins, which were then randomly divided into a training set (10200 proteins) and a validation set (1000 proteins). Then, some proteins were removed to obtain efficient calculation. This reduced the training, validation, and independent test sets to 10029, 983, and 1213 proteins, respectively. In the SPOT-1D dataset, another independent test set was obtained from the PDB. These proteins were released between January 01, 2018 and July 16, 2018 and solved with resolution $< 2.5\text{\AA}$ and R-free < 0.25 . In order to minimise evaluation bias associated with partially overlapping training data, proteins were removed for $> 25\%$ sequence identity to structures released prior to 2018. This dataset was also filtered to remove redundancy at a 25% sequence identity cutoff and another 13 proteins with length > 700 were removed, leaving 250 high-quality, non-redundant targets. For convenience, these two independent test sets were denoted as TEST2016 (1213 proteins) and TEST2018 (250 proteins) as they were deposited between June 2015 and Feb 2017 and between Jan 2018 and July 2018, respectively.

We use the same dataset used by SPOT-1D²⁷. However, we have performed additional filtering since it is not precisely clear to us how SPOT-1D handles the proteins that have mismatches in their amino acid sequences specified in various data source files (e.g. .t, .pssm, .dssp, and .fasta files). To be clearer, we have found that for some proteins, the amino acid sequence specified in one data source file has additional residues at the beginning or ending compared to that specified in another data source file. For such proteins, we have taken the part common in the amino acid sequences specified in various source files. However, when there is any mismatch at the middle of any two amino acid sequences specified in two different data source files for the same protein, we have removed the protein from the dataset. Also, we have removed proteins that have X in the secondary structure sequences in their corresponding DSSP files, although we do not use the secondary structure data from the DSSP files in our learning model. As mentioned before, apart from using subsets of features from SPOT-1D, we generate 8-state SS predictions using SSpro8¹⁴. The training set for SSpro8 comprised 5772 proteins released in the PDB before August 20, 2013. In order to avoid over-training with SSpro8 predictions as input of our method, we have removed 3259 proteins from SPOT-1D's proteins using Blast²⁸ against SSpro8's training set with e-value 0.01. We show in Table 1 the numbers of proteins and residues in training, validation, and testing datasets, after performing the abovementioned filtering. As we can see later in Table 5, the remaining dataset after performing the filtering does not degrade the performance of SPOT-1D.

While our main training and test proteins are from the SPOT1D dataset, for further independent testing, we use PDB150³⁴ and CAMEO93³⁵ datasets. The PDB150 dataset contains 150 proteins released between February 1, 2019 and May 15, 2019. For each protein, PSI-BLAST²⁸ was applied against the whole CullPDB³² dataset with e-value smaller than 0.005. The CAMEO93 dataset contains 93 proteins that are released between February 2020 and March 2020 and has been used by OPUS-TASS in its evaluation. For both datasets, we have applied 25% sequence similarity cutoff w.r.t. our and SSpro8's training and validation datasets and also have removed proteins having X in their fasta file. For proteins with discontinuity in their amino acid sequences, we have considered largest segment of each protein so that our sliding window method can still be applied. At the end, we have obtained 71 and 55 proteins from the PDB150 and CAMEO93 datasets and we use them for independent testing of our method and the state-of-the-art method OPUS-TASS and compare their performance.

Features		Encoding		ϕ MAE			ψ MAE			θ MAE			τ MAE		
ASA	7PCP	OR	NM	WS	Test	Valid	WS	Test	Valid	WS	Test	Valid	WS	Test	Valid
N	N	D	R	5	17.19	17.53	5	20.27	20.42	5	6.42	6.45	5	18.89	19.00
			Z	9	16.79	17.13	5	19.98	20.15	5	6.37	6.40	5	17.94	17.91
		R	R	17	37.39	37.75	17	15.90	16.10	17	31.30	31.37	17	25.36	25.59
			Z	17	34.76	35.14	17	15.13	15.30	9	30.54	30.69	9	23.70	23.87
Y	N	D	R	9	18.17	18.51	5	21.91	22.11	9	6.72	6.77	5	21.25	21.35
			Z	1	19.21	19.55	1	24.32	24.56	1	7.14	7.18	13	28.30	28.63
		R	R	9	36.03	36.54	5	15.80	16.00	5	31.14	31.18	9	24.89	25.06
			Z	9	33.47	33.78	9	14.82	14.94	9	30.24	30.33	9	22.65	22.81
N	Y	D	R	5	15.62	15.57	5	19.60	19.68	5	6.07	6.19	5	16.85	16.94
			Z	1	18.77	19.00	1	24.77	24.98	1	7.24	7.27	17	29.94	30.13
		R	R	13	34.20	34.56	13	14.94	15.03	13	30.36	30.45	13	23.15	23.30
			Z	13	33.11	33.39	13	14.23	14.55	5	30.22	30.30	9	22.77	22.92
Y	Y	D	R	9	18.79	18.13	9	22.00	22.15	5	7.21	7.27	9	21.00	21.09
			Z	1	21.55	21.55	9	28.55	28.85	9	8.13	8.19	1	36.64	37.01
		R	R	5	35.46	35.85	5	15.62	15.81	5	31.06	31.20	5	24.46	24.71
			Z	9	33.08	33.51	5	15.67	15.81	5	30.10	30.20	9	22.58	22.65

Table 2. Performance of SAP settings on 1206 testing proteins. In the table, column ASA denotes whether accessible surface area is used (Yes/No), column 7PCP denotes whether 7 physicochemical properties are used (Yes/No), column OR denotes output representation is in direct angles (D) or trigonometric ratios (R), column NM denotes normalisation method for input feature encoding is [0,1] range based (R) or Z-score based (Z), WS denotes the best size of the sliding window. Note that the emboldened cells denote the best performance for each combination of ASA and 7PCP while the boxed plus emboldened cells in each respective column denote the best performance among all SAP settings.

Results

We compare various settings of SAP to find the best setting for each of the 4 types of angles to be predicted. This comparison helps us understand the impact of various features and encodings. Then, we compare the best settings with the current state-of-the-art predictors. Moreover, we show various other analyses of the results obtained for the best settings.

Calculating absolute errors. For each predicted angle P against the actual angle A , we calculate the difference $D = |P - A|$. Then, we take $AE = \min(D, 360 - D)$ as the absolute error (AE) for that predicted angle. This addresses the periodicity issue that each angle must be in the range -180° to 180° . When angles are predicted directly, we implement the AE calculation within the loss function for the training and validation, and also later for testing. When we use sine and cosine ratios, then we calculate AE only during testing. In all cases, the angles that are not defined for the amino acids at the beginning or ending of the proteins are ignored.

Determining best settings. We run 96 settings of SAP. All of these settings having 20 PSSM and 8 SS hot-vector features. The 96 settings are obtained by using or not using ASA, by using or not using 7PCP, by using range-based or Z-score based normalisation for input feature encoding, by using 6 window sizes (1, 5, 9, 13, 17, 21), and by using direct angles or trigonometric ratios to encode output angles. However, Table 2 presents performance of 16 settings only, selecting the best window size for each combination of the other parameters. From these results, it appears that window sizes 5 and 9 in most cases lead to better performances. Moreover, prediction of trigonometric ratios is better for ψ while prediction of direct angles is better for ϕ , θ , and τ . While not using ASA appears to be better than using, in contrast, using 7PCP appears to be better than not using.

Overall, the best SAP settings to predict the 4 types of angles are listed below. Henceforth, we use these angle specific settings in further analysis. It is worth noting here that in our observation, training a DNN simultaneously for several outputs is not much different from training the DNN separately for each output in terms of the accuracy level obtained for each output.

- ϕ : 7PCP, range-based normalisation, direct angle prediction, and window size 5
- ψ : 7PCP, z-score based normalisation, trigonometric ratio prediction, and window size 13
- θ : 7PCP, range-based normalisation, direct angle prediction, and window size 5
- τ : 7PCP, range-based normalisation, direct angle prediction, and window size 5

All results presented in Table 2 are for DNNs having 3 hidden layers. The choice of the number of layers was inspired by SPIDER²³. However, in Table 3, we show the performance of the best angle specific SAP settings when run with DNNs having 2 and 4 hidden layers. In most cases DNNs having 3 hidden layers show the best results (shown in bold in Table 3); where this is not the case, DNNs with three hidden layers are a close second (shown

Hidden Layer	ϕ MAE		ψ MAE		θ MAE		τ MAE	
	Test	Valid	Test	Valid	Test	Valid	Test	Valid
2	15.62	15.97	14.66	14.86	6.08	6.22	16.97	17.07
3	15.62	15.57	14.23	<i>14.55</i>	6.07	6.19	<i>16.85</i>	16.94
4	15.66	16.06	14.36	14.52	6.11	6.21	16.84	16.96

Table 3. Performance of the best angle specific SAP settings when the numbers of hidden layers in the DNNs are varied.

Dataset	Measure	ϕ	ψ	θ	τ
Validation	MAE	15.57	14.55	6.19	16.94
Testing	MAE	15.62	14.23	6.07	16.85
10-Fold	MAE	16.54	14.94	6.33	17.71
10-Fold	SDMAE	0.25	0.07	0.08	0.22

Table 4. Average performance of the best settings of SAP after 10-fold cross validation is performed.

in italics in Table 3), with the difference being < 0.05 . So for the rest of the paper, we have chosen DNNs with 3 hidden layers as the selected SAP settings.

Performing cross-validation. When we train a DNN, we specify the validation set. Consequently, the MAE values for the validation set as well as for the testing set for each SAP setting are shown in Table 2. In Table 4, we again show the MAE values but only for the best settings of SAP. However, to check the robustness of SAP, we perform 10-fold cross-validation, where the training and validation sets are merged. The merged proteins are then randomly divided into 10 folds. Then, 9 out of 10 folds are used in turn for training while the remaining one is used for testing. Table 4 shows the MAE value and the standard deviation of MAEs (SDMAE) for each type of angles to be predicted. As one can see, the small differences between MAE values and the small SDMAE values observed in the table shows the consistency and robustness of SAP.

Comparison with state-of-the-art predictors. We mainly compare the performance of SAP with that of SPIDER²⁴, SPOT-1D²⁷, and OPUS-TASS⁶ in Table 5. We have run these systems on the testing dataset that is used in this work and that is a subset of the SPOT-1D dataset because of more rigorous filtering. Moreover, we use 71 and 55 proteins from PDB150³⁴ and CAMEO93³⁵ datasets after performing filtering as mentioned before. However, we also compare SAP's performance with that of SPIDER2, SPOT-1D, and OPUS-TASS as they are reported in the respective publications. Below we briefly describe SPIDER2, SPOT-1D, and OPUS-TASS.

1. **SPIDER2** is similar to SAP in that both use similar FCNN and similar features. SPIDER2 uses three DNNs of its precursor SPIDER²³ in a series where the output of a previous DNN is fed as input to the next DNN in the series. Like SAP, SPIDER uses FCNN with 3 hidden layers each with 150 neurons. However, SPIDER uses stacked sparse auto-encoder for weight initialisation and 0-1 range normalisation for input values. SPIDER's input features are PSSM, 3-state predicted SS, ASA, and 7PCP and the outputs are represented by trigonometric ratios. The window size is 21 in SPIDER and 17 in SPIDER2. SPIDER and SPIDER2 use the PISCES³² dataset, which has 5840 proteins.
2. **SPOT-1D** is a recent protein backbone angle predictor. It uses an ensemble of 9 long short term memory (LSTM) bidirectional recurrent neural networks (BRNNs) and Residual Networks (ResNets). SPOT-1D's input features are PSSM, Hidden Markov Model (HMM), 7PCP, and contact maps. SPOT-1D obtains its predicted contact maps from SPOT-Contact³³. SPOT-1D then uses windowing of the predicted contact maps. Further, SPOT-1D generates HMM profiles that include information about homologous sequences. For this, SPOT-1D uses HHblits³¹ with the Uniprot sequence profile database from October 2017. SPOT-1D's inputs are mapped in the range of [0, 1] and the outputs are represented by trigonometric ratios. SPOT-1D's dataset is a superset of SAP's dataset.
3. **OPUS-TASS** is the current state-of-the-art protein backbone angle predictor and predicts ϕ and ψ only. Its architecture consists of CNN layers, LSTM layers, and Transformer³⁷ layers. It uses an input feature named PSP19³⁸, which classifies 20 residues into 19 rigid-body blocks depending on their local structures. It also introduces a new constrained/output feature named CSF3³⁹, which is a local backbone structure descriptor. Further, it uses a multi-task learning strategy⁴⁰ to maximise generalisation of the neural network and an ensemble of neural networks for further improvement.

Since SPOT-1D and OPUS-TASS show their performance on two subsets namely TEST2016 and TEST2018 of the testing proteins, we also do the same although we show the accumulated results for all testing proteins. Note that both SPOT-1D's and OPUS-TASS's performances are not worse than their reported values as shown

Results below are as we run all of the systems on our datasets							
Dataset	Proteins	Residues	Method	ϕ MAE	ψ MAE	θ MAE	τ MAE
TEST2016	1179	278553	SPIDER2	18.80	29.93	8.08	31.72
			SPOT-1D	16.12	23.07	6.71	24.27
			OPUS-TASS	15.75	22.41	–	–
			SAP	15.63	14.25	6.07	16.87
TEST2018	27	3908	SPIDER2	17.17	27.69	7.22	28.86
			SPOT-1D	15.05	22.17	6.12	22.78
			OPUS-TASS	15.62	21.96	–	–
			SAP	14.58	13.22	5.60	15.22
Testing	1206	282461	SPIDER2	18.52	29.55	7.94	31.24
			SPOT-1D	15.94	22.92	6.61	24.02
			OPUS-TASS	15.74	22.41	–	–
			SAP	15.62	14.23	6.07	16.85
Results below are as they are reported in the respective publications							
Dataset	Proteins	Method	ϕ MAE	ψ MAE	θ MAE	τ MAE	
PISCES-test	1199	SPIDER2	19.7	30.3	8.2	32.6	
TEST2016	1213	SPOT-1D	16.27	23.26	6.89	25.38	
		OPUS-TASS	15.78	22.46	–	–	
TEST2018	250	SPOT-1D	16.89	24.87	6.91	25.94	
		OPUS-TASS	16.40	24.06	–	–	

Table 5. Performances of SPIDER2, SPOT-1D, SAP, and OPUS-TASS on our testing dataset and its subsets TEST2016 and TEST2018. The emboldened values are the winning numbers for the corresponding types of angles and datasets. OPUS-TASS does not predict θ and τ angles while the other three methods predict all four types of angles.

Results below are as we run all of the systems on our datasets							
Dataset	Proteins	Residues	Method	ϕ MAE	ψ MAE	θ MAE	τ MAE
PDB150	71	14964	SPIDER2	20.63	32.54	8.48	33.56
			SPOT-1D	18.37	25.48	7.74	26.54
			SAP	19.69	17.14	7.57	25.70
CAMEO93	55	13872	SPIDER2	20.35	31.80	8.34	33.83
			OPUS-TASS	16.76	24.04	–	–
			SAP	20.23	17.60	7.87	28.58
Results below are as they are reported in the respective publications							
Dataset	Proteins	Method	ϕ MAE	ψ MAE	θ MAE	τ MAE	
CAMEO	93	SPOT-1D	16.89	23.02			
		OPUS-TASS	16.56	22.56			

Table 6. Performances of SPIDER2, SPOT-1D, OPUS-TASS, and SAP on filtered PDB150 and CAMEO93 proteins. The emboldened values are the winning numbers for the corresponding types of angles and datasets. OPUS-TASS does not predict θ and τ angles while the other three methods predict all four types of angles.

in the bottom part of Table 5. Moreover, notice from the table that SAP significantly outperforms both SPOT-1D and OPUS-TASS in all cases. We have performed t-tests to compare the performances of SPOT-1D and OPUS-TASS with SAP and the p values are < 0.01 in all cases, indicating the differences are statistically significant. The differences are really huge for ψ and τ . These results demonstrate the effectiveness of SAP in enhancing protein backbone angle prediction accuracy.

To test the generality of performance of SAP over other datasets, we have run SAP on 71 proteins of PDB150 dataset and 55 proteins of CAMEO93 datasets. In Table 6, we also compare SAP's performance with SPOT-1D's performance on the PDB150 proteins and with OPUS-TASS's performance on the CAMEO93 proteins. Notice that SAP significantly outperforms SPOT-1D and OPUS-TASS in ψ , θ , and τ angles, but performs worse in ϕ prediction. We have performed t-tests to compare the performances of SPOT-1D and OPUS-TASS with SAP and the p values are < 0.01 in all cases, indicating the differences are statistically significant. Nevertheless, the margins in ψ and τ remain huge for SAP compared to the other methods.

Testing proteins		ϕ				ψ				θ			τ		
		SAP	OPUS-TASS	SPOT-1D	SPIDER2	SAP	OPUS-TASS	SPOT-1D	SPIDER2	SAP	SPOT-1D	SPIDER2	SAP	SPOT-1D	SPIDER2
Length	Count	MAE	Δ MAE	Δ MAE	Δ MAE	MAE	Δ MAE	Δ MAE	Δ MAE	MAE	Δ MAE	Δ MAE	MAE	Δ MAE	Δ MAE
001–100	210	14.43	+ 0.14	+ 0.34	+ 2.75	13.19	+ 7.84	+ 8.02	+ 13.53	5.63	+ 0.36	+ 1.61	15.19	+ 5.52	+ 11.55
101–200	381	15.33	+ 0.06	+ 0.35	+ 2.93	13.73	+ 7.94	+ 8.43	+ 14.77	6.09	+ 0.47	+ 1.82	16.54	+ 6.44	+ 13.42
201–300	264	15.21	+ 0.28	+ 0.56	+ 3.03	13.94	+ 8.01	+ 8.59	+ 14.99	5.96	+ 0.67	+ 1.93	16.25	+ 7.35	+ 14.46
301–400	180	15.73	– 0.26	+ 0.26	+ 3.36	14.29	+ 7.35	+ 8.30	+ 15.85	6.11	+ 0.57	+ 2.07	17.12	+ 7.08	+ 15.27
401–500	102	16.03	+ 0.37	+ 0.79	+ 3.44	14.70	+ 9.04	+ 9.45	+ 16.97	6.09	+ 0.81	+ 2.27	17.27	+ 8.33	+ 16.32
501–800	69	16.49	+ 0.28	+ 0.77	+ 3.25	15.22	+ 9.17	+ 10.21	+ 17.07	6.29	+ 0.86	+ 2.17	17.97	+ 9.15	+ 16.33
Overall	1206	15.62	+ 0.12	+ 0.41	+ 2.90	14.23	+ 8.18	+ 8.69	+ 15.32	6.07	+ 0.54	+ 1.87	16.85	+ 7.17	+ 14.39

Table 7. Performance of SAP, OPUS-TASS, SPOT-1D, and SPIDER2 when our testing proteins are grouped based on their lengths. In the table, Δ MAE of a system (e.g. OPUS-TASS, SPOT-1D or SPIDER2) is its MAE minus the MAE of SAP. As such, the greater the value of Δ MAE, the worse the performance of the system w.r.t. the performance of SAP. The horizontal lines in SAP columns split those columns such that the upper parts have MAE values less than the overall MAE values and the lower parts have MAE values greater (a slight exception for θ).

Comparison on protein length groups. In Table 7, we compare the performance of SAP, OPUS-TASS, SPOT-1D, and SPIDER2 when our testing proteins are grouped based on their lengths i.e. the number of amino acids each protein has. This is to observe how SAP's performance varies with the increase of the protein length. From the table, we see that for all four types of angles, SAP's prediction accuracy gradually decreases as the protein length increases. When protein lengths are 300 or below, the MAE values are less than the overall MAE values and for protein lengths above 300, the MAE values are greater than the overall MAE values. From the Δ MAE values (i.e. how far from SAP's MAE) of OPUS-TASS, SPOT-1D and SPIDER2, we see that with the increase of protein lengths, the performance difference increases; which essentially means compared to OPUS-TASS's or SPOT-1D's or SPIDER2's performance, SAP's performance rather gets better.

Comparison on secondary structure groups. Table 8 (Left) shows the residue distribution over the testing proteins when the residues are grouped on their SS types. Types C, E, H, S and T are the most frequent groups. Figure 4 (Top Four) shows the MAE values of SAP, OPUS-TASS, SPOT-1D, and SPIDER2 when the residues are grouped on their SS types. From the charts, frequent SS type H appears to have the best MAE values while other frequent SS types C, E, and S have significantly worse MAE values than the overall MAE values.

Comparison on amino acid groups. Table 8 (Right) shows the residue distribution over the testing proteins when the residues are grouped on their AA types. Types A, D, E, G, I, K, L, P, R, S, T, and V are the most frequent groups having at least 4.5% residues. Figure 4 (Bottom Four) shows the MAE values of SAP, OPUS-TASS, SPOT-1D, and SPIDER2 when the residues are grouped on their AA types. From the charts, frequent AA types A, E, I, L appear to have the best MAE values in all 4 types of angles. Among other frequent AA types C, D, G have worse MAE values than the overall MAE values in some types of angles.

Using angle ranges from predicted secondary structures. Given the SS predictions and their suggested ranges of ϕ and ψ values as shown in Table 8 left, particularly for helices (G, H, I) and sheets (B, E), one might just use the mid values of the respective ranges as the predicted values and expect an MAE of about 10 for the respective SS type. When we do that for the residues that belong to SS types G, H and I, we get MAE values respectively 16.91, 8.78, and 24.02 for ϕ and 27.71, 9.12, 22.04 for ψ . In contrast the MAE values for SAP predictions are respectively 12.39, 5.43, 11.34 for ϕ for SS types G, H, and I, and 12.41, 5.73, 12.06 for ψ . The situations worsens for sheets such as SS types B and E. These results clearly show that just achieving higher accuracy in SS prediction would not be sufficient for backbone angle prediction.

Comparison of angle distributions. Figure 5 shows the distributions of the actual angles and predicted values obtained from SAP, OPUS-TASS, SPOT-1D, and SPIDER2. As we can see from the charts, the distribution of values predicted by SAP aligns very well with the distribution of the actual values. The peaks and troughs of the distributions align quite well, even multiple peaks and troughs are captured well. While the peaks of the predicted distributions are larger and narrower than those of the actual distributions, the troughs of the predicted distributions are rather smaller and wider than those of the actual distributions. When SAP's curves are compared with OPUS-TASS's, SPOT-1D's and SPIDER2's, we see SAP's curves are occasionally closer to the curves for the actual values. We also see that the distributions of ϕ and ψ angles for are OPUS-TASS and SPOT-1D are almost similar. Notice that the largest peaks of the predicted values are higher than the largest peaks of the actual values except in predicted ψ distributions of OPUS-TASS, SPOT-1D and SPIDER2. One noticeable fact in the ψ chart is OPUS-TASS, SPOT-1D and SPIDER2 predicted values are outside of $[-90, 90]$ but actual values are roughly within the range. Another noticeable fact is in the θ chart: there are actual values between 0 and 90

SS	Residues	Percentage	ϕ Range	ψ Range
B	2955	1.05	[− 130, − 110]	[110, 130]
C	56,250	19.91	[− 180, + 180]	[− 180, + 180]
E	61,041	21.61	[− 130, − 110]	[110, 130]
G	10,581	3.75	[− 59, − 39]	[− 36, − 16]
H	96,993	34.34	[− 67, − 47]	[− 57, − 37]
I	47	0.02	[− 67, − 47]	[− 80, − 60]
S	22,984	8.14	[− 180, + 180]	[− 180, + 180]
T	31,610	11.19	[− 180, + 180]	[− 180, + 180]
Total	282,461	100.00		
AA	Residues	Percentages		
A	22,406	7.93		
C	3874	1.37		
D	16,697	5.91		
E	18,752	6.64		
F	12,022	4.26		
G	19,593	6.94		
H	6904	2.44		
I	16,152	5.72		
K	16,024	5.67		
L	26,909	9.53		
M	5963	2.11		
N	12,161	4.31		
P	12,752	4.51		
Q	10,567	3.74		
R	14,619	5.18		
S	17,387	6.16		
T	15,492	5.48		
V	19,622	6.95		
W	4170	1.48		
Y	10,395	3.68		
Total	282,461	100.00		

Table 8. Residue distribution over the testing proteins when residues are grouped on their (Left) SS and (Right) AA types. Also, on the left, typical ranges suggested for the torsion angles ϕ and ψ for various secondary structures⁴¹.

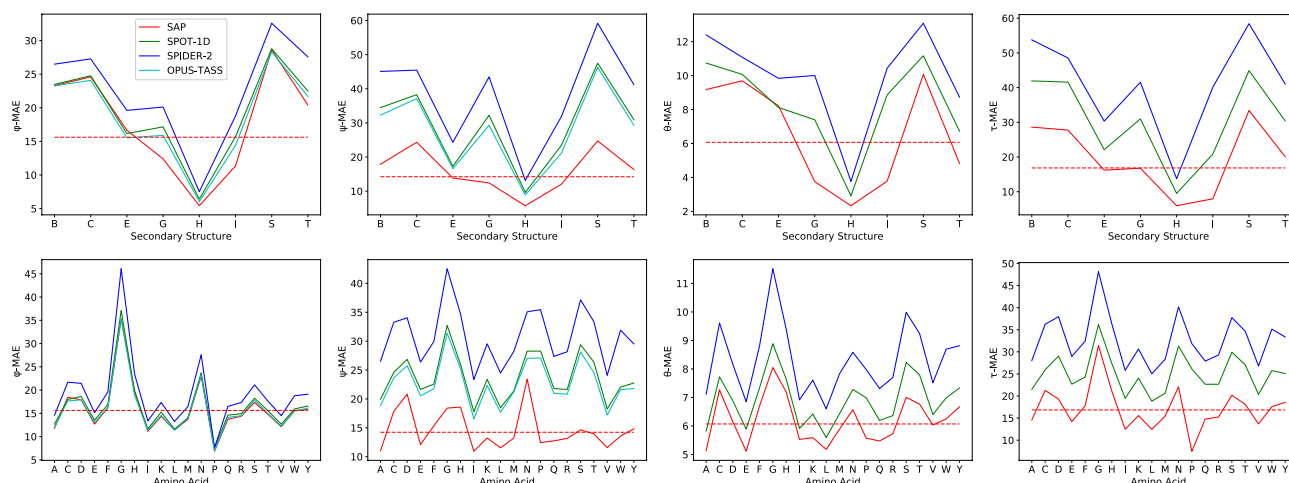


Figure 4. Performance of SAP, OPUS-TASS, SPOT-1D, SPIDER2 on the testing proteins when residues are grouped based (Top Four) on their SS types and (Bottom Four) on their AA types. In the charts, y-axis shows MAE values and x-axis shows SS or AA types. The dashed horizontal line in each chart shows the overall MAE value for SAP.

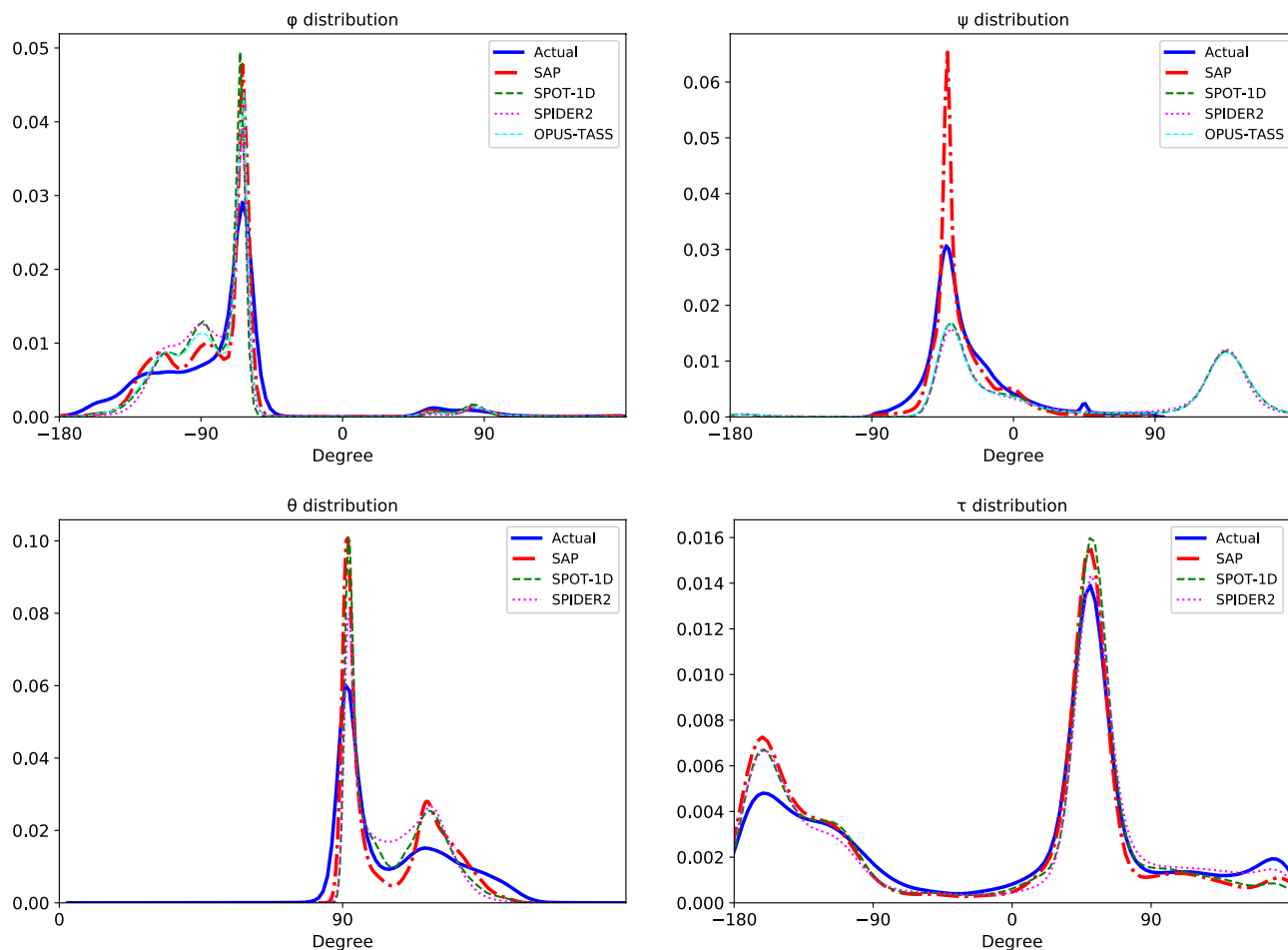


Figure 5. Distributions of actual angles of testing proteins and predictions of SAP, OPUS-TASS, SPOT-1D, and SPIDER2.

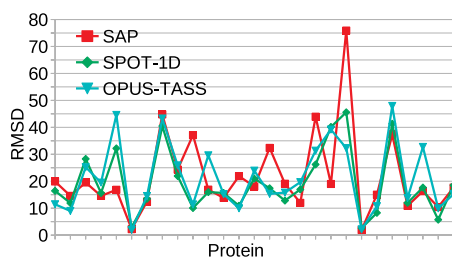


Figure 6. RMSD values for SAP, SPOT-1D, and OPUS-TASS on TEST2018 proteins.

although with almost zero probability, and these values are not much captured by the predictors. Overall, there is a tendency to predict the peak values with probabilities larger than that of the actual values.

Protein structure generation and refinement. Given the improvement in angle prediction accuracy, an interesting question is as follows: “Can predicted angles be directly employed in building accurate protein structures?” The direct answer to this question is yes if we reach to a very high accuracy level. This is actually the aim of this study to enhance the performance gradually to the level that would predict protein structures with very high accuracy; which is very challenging. Given the 27 proteins in our TEST2018 set, we have tried to generate entire protein structures from the predicted values obtained from SAP, OPUS-TASS, and SPOT-1D, and assuming $\omega = 180^\circ$ and standard bond distances. From Fig. 6, we can see very high root mean square distance (RMSD) for more proteins and only for 2–3 proteins, RMSD values are less than 6Å , a distance considered to be practically meaningful. Although this is the case with protein structure generation, for structure refinement via ab initio structure sampling and evaluation by using perturbation techniques would obtain significant help. This is because given a prediction ρ and estimated error ϵ , with some level of certainty, one can focus searching within the region $[\rho - \epsilon, \rho + \epsilon]$. These soft constraints can thus reduce the search space significantly. With more

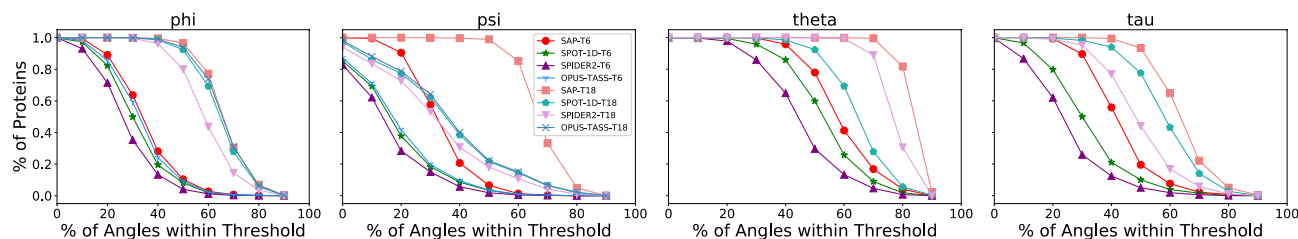


Figure 7. Percentages of proteins (y-axis) that have a given percentage of residues (x-axis) with AE at most a given threshold T where T is 6 and 18 and are denoted by T6 and T18. The lower the threshold, the better the prediction quality.

proteins having more dihedral angles predicted with less absolute errors, *ab initio* or refinement search for protein structures would be benefited more from SAP's prediction than OPUS-TASS's or SPOT-1D's.

Comparison on correct prediction per protein. Having the discussion regarding structure generation and refinement, we compare SAP, OPUS-TASS, and SPOT-1D on what portions of the angles of the proteins are predicted within certain error levels. Figure 7 shows the percentages of proteins that have a given percentage of particular angles with absolute errors at most a given threshold. We choose the threshold values to be 6 and 18 in the charts because SAP's minimum and maximum MAE values are close to 6 and 18 respectively, for example for θ and τ . Notice that SPOT-1D's and OPUS-TASS's performances are very close in the charts for ϕ and ψ . Moreover, SAP outperforms the other three methods in all angles in all threshold levels.

Conclusions

Input features and neural network architectures interact with each other when employed in prediction systems. Consequently, inclusions of just more features might cause cluttering and the complex networks might then be needed to counterbalance. In the protein backbone angle prediction research, the existing state-of-the-art prediction method uses ensembles of several types of deep neural networks and a number of features. In this paper, we present simpler deep neural network models for protein backbone angle prediction. Our models use fewer features and simpler neural networks but on a standard benchmark dataset obtain significantly better mean absolute errors than the state-of-the-art predictor. Our program named Simpler Angle Predictor (SAP) along with its data is available from the website <https://gitlab.com/mahnewton/sap>.

Received: 6 May 2020; Accepted: 23 October 2020

Published online: 10 November 2020

References

- Gibson, K. D. & Scheraga, H. A. Minimization of polypeptide energy. I. Preliminary structures of bovine pancreatic ribonuclease S-peptide. *Proc. Natl. Acad. Sci. U.S.A.* **58**, 420 (1967).
- Zhou, Y., Duan, Y., Yang, Y., Faraggi, E. & Lei, H. Trends in template/fragment-free protein structure prediction. *Theor. Chem. Acc.* **128**, 3–16 (2011).
- Mittal, A., Jayaram, B., Shenoy, S. & Bawa, T. S. A stoichiometry driven universal spatial organization of backbones of folded proteins: are there Chargaff's rules for protein folding? *J. Biomol. Struct. Dyn.* **28**, 133–142 (2010).
- Cutello, V., Narzisi, G. & Nicosia, G. A multi-objective evolutionary approach to the protein structure prediction problem. *J. R. Soc. Interface* **3**, 139–151 (2005).
- Guo, J.-T., Ellrott, K. & Xu, Y. A historical perspective of template-based protein structure prediction. In *Protein Structure Prediction*, 3–42 (Springer, 2008).
- Xu, G., Wang, Q. & Ma, J. OPUS-TASS: a protein backbone torsion angles and secondary structure predictor based on ensemble neural networks. *Bioinformatics (Oxford, England)* (2020).
- Hu, H., Li, Z., Elofsson, A. & Xie, S. A Bi-LSTM based ensemble algorithm for prediction of protein secondary structure. *Appl. Sci.* **9**, 3538 (2019).
- Torrisi, M., Kaleel, M. & Pollastri, G. Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. *Sci. Rep.* **9**, 1–12 (2019).
- Fang, C. *Applications of deep neural networks to protein structure prediction*. Ph.D. thesis, University of Missouri-Columbia (2018).
- Torrisi, M., Kaleel, M. & Pollastri, G. Porter 5: fast, state-of-the-art *ab initio* prediction of protein secondary structure in 3 and 8 classes. *bioRxiv* 289033 (2018).
- Faraggi, E., Zhang, T., Yang, Y., Kurgan, L. & Zhou, Y. SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J. Comput. Chem.* **33**, 259–267 (2012).
- Kihara, D. The effect of long-range interactions on the secondary structure formation of proteins. *Protein Sci.* **14**, 1955–1963 (2005).
- Rost, B. Protein secondary structure prediction continues to rise. *J. Struct. Biol.* **134**, 204–218 (2001).
- Magnan, C. N. & Baldi, P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics* **30**, 2592–2597 (2014).
- Lodish, H. *et al.* Hierarchical structure of proteins. In *Molecular Cell Biology*, 4th edn (WH Freeman, 2000).
- Faraggi, E., Yang, Y., Zhang, S. & Zhou, Y. Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure* **17**, 1515–1527 (2009).
- Simons, K. T. *et al.* Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins: Struct. Funct. Bioinform.* **34**, 82–95 (1999).
- Kuang, R., Leslie, C. S. & Yang, A.-S. Protein backbone angle prediction with machine learning approaches. *Bioinformatics* **20**, 1612–1621 (2004).

19. Kang, H. S., Kurochkina, N. A. & Lee, B. Estimation and use of protein backbone angle probabilities. *J. Mol. Biol.* **229**, 448–460 (1993).
20. Wood, M. J. & Hirst, J. D. Protein secondary structure prediction with dihedral angles. *Proteins: Struct. Funct. Bioinform.* **59**, 476–481 (2005).
21. Dor, O. & Zhou, Y. Real-SPINE: an integrated system of neural networks for real-value prediction of protein structural properties. *Proteins: Struct. Funct. Bioinform.* **68**, 76–81 (2007).
22. Xue, B., Dor, O., Faraggi, E. & Zhou, Y. Real-value prediction of backbone torsion angles. *Proteins: Struct. Funct. Bioinform.* **72**, 427–433 (2008).
23. Lyons, J. *et al.* Predicting backbone α angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *J. Comput. Chem.* **35**, 2040–2046 (2014).
24. Heffernan, R. *et al.* Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* **5**, 11476 (2015).
25. Heffernan, R., Yang, Y., Paliwal, K. & Zhou, Y. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics* **33**, 2842–2849 (2017).
26. Heffernan, R. *et al.* Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning. *J. Comput. Chem.* **39**, 2210–2216 (2018).
27. Hanson, J., Paliwal, K., Litfin, T., Yang, Y. & Zhou, Y. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics* **35**, 2403–2410 (2018).
28. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
29. Meiler, J., Müller, M., Zeidler, A. & Schmäschke, F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Mol. Model. Annu.* **7**, 360–369 (2001).
30. Klausen, M. S. *et al.* NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins: Struct. Funct. Bioinform.* **87**, 520–527 (2019).
31. Remmert, M., Biegert, A., Hauser, A. & Söding, J. HHblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nat. Methods* **9**, 173 (2012).
32. Wang, G. & Dunbrack, R. L. PISCES: recent improvements to a pdb sequence culling server. *Nucleic Acids Res.* **33**, W94–W98 (2005).
33. Hanson, J., Paliwal, K., Litfin, T., Yang, Y. & Zhou, Y. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics* **34**, 4039–4045 (2018).
34. Fang, C., Shang, Y. & Xu, D. Prediction of protein backbone torsion angles using deep residual inception neural networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **16**, 1020–1028 (2018).
35. Haas, J. *et al.* The protein model portal—a comprehensive resource for protein structure and model information. *Database.* **2013** (2013).
36. Mignan, A. & Broccardo, M. One neuron versus deep learning in aftershock prediction. *Nature* **574**, E1–E3 (2019).
37. Vaswani, A. *et al.* Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008 (2017).
38. Xu, G. *et al.* OPUS-DOSP: a distance-and orientation-dependent all-atom potential derived from side-chain packing. *J. Mol. Biol.* **429**, 3113–3120 (2017).
39. Xu, G., Ma, T., Zang, T., Wang, Q. & Ma, J. OPUS-CSF: ac-atom-based scoring function for ranking protein structural models. *Protein Sci.* **27**, 286–292 (2018).
40. Lounici, K., Pontil, M., Tsybakov, A. B. & Van De Geer, S. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468* (2009).
41. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolym. Orig. Res. Biomol.* **22**, 2577–2637 (1983).

Acknowledgements

This research is partially supported by Australian Research Council Discovery Grant DP180102727.

Author contributions

F.M. and M.A.H.N. contributed equally and in all parts of the work. A.D. helped prepare data, took part in discussions, analysed results, and reviewed the manuscript. A.K. helped implement the program in Python. B.J. and S.R. reviewed the manuscript. A.S. took part in discussions and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to F.M. or M.A.H.N.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020