Real-value and confidence prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning

Yujuan Gao[1,2], Sheng Wang[2], Minghua Deng[1,3,4*], Jinbo Xu[2*]

December 19, 2017

[1] Center for Quantitative Biology, Peking University, Beijing, China
[2] Toyota Technological Institute at Chicago, Chicago, Illinois, United States of America
[3] School of Mathematical Sciences, Peking University, Beijing, China
[4] Center for Statistical Sciences, Peking University, Beijing, China

Abstract

Background. Protein dihedral angles provide a detailed description of protein local conformation. Predicted dihedral angles can be used to narrow down the conformational space of the whole polypeptide chain significantly, thus aiding protein tertiary structure prediction. However, direct angle prediction from sequence alone is challenging.

Method. In this study, we present a novel method to predict real-valued angles by combining clustering and deep learning. That is, we first generate certain clusters of angles (each assigned a label) and then apply a deep residual neural network to predict the label posterior probability. Finally, we output real-valued prediction by a mixture of the clusters with their predicted probabilities. At the same time, we also estimate the bound of the prediction errors at each residue from the predicted label probabilities.

Result. In this article, we present a novel method (named RaptorX-Angle) to predict real-valued angles by combining clustering and deep learning. Tested on a subset of PDB25 and the targets in the latest two Critical Assessment of protein Structure Prediction (CASP), our method outperforms the existing state-of-art method SPIDER2 in terms of Pearson Correlation Coefficient (PCC) and Mean Absolute Error (MAE). Our result also shows approximately linear relationship between the real prediction errors and our estimated bounds. That is, the real prediction error can be well approximated by our estimated bounds.

Conclusions. Our study provides an alternative and more accurate prediction of dihedral angles, which may facilitate protein structure prediction and functional study.

[*] Corresponding authors

1 Introduction

It has been shown that sequences contain rich information for protein tertiary structure prediction as well as functional study [1, 2]. But it is challenging to directly predict tertiary structure from primary sequence, so the hierarchical approach has been widely accepted as one of the most efficient methods. That means to transform the ultimate goal into several sub-problems, such as secondary structure prediction, solvent accessibility prediction, residue-residue contact prediction, etc. [3] reviewed the progress in the field of intermediate state or one-dimensional property prediction. It has been shown that predicted secondary structure is useful in the prediction of disordered and flexible regions, fold recognition and function prediction. However, secondary structure states are described as discrete classes and there is no clear boundary between coil and helical/strand states. It is a significant step towards establishing the structure and function of a protein to predict local conformation of the polypeptide chain. The local structural bias information restricts the possible conformations of a sequence segment and therefore narrows down the conformation space of the whole polypeptide chain significantly. Thus, prediction of dihedral angles is especially useful for protein tertiary structure prediction.

On the whole, dihedral angle prediction may benefit protein tertiary structure prediction in several aspects. Firstly, dihedral angle prediction may act as substitute or supplement for secondary structure prediction [4, 5, 6]. Secondly, It can be used in generation of sequence/structure alignment. For one thing, it can be directly applied to structure alignment methods based on dihedral angles [7, 8] and may aid refinement of target-template structure alignment. For another, considering predicted angles to refine multiple sequence alignment may narrow the gap between sequence and structure alignment, thus aiding de novo prediction of structural properties. In addition, dihedral angle prediction may also find applications in protein structure prediction that includes but not limits to fold recognition approaches [9, 10], fragment-free tertiary structure prediction [11], tertiary structure refinement and structure quality assessment [12] and functional study, such as ligand-binding site prediction [13].

There are mainly two kinds of problems in dihedral angle prediction: angle region prediction and real value prediction, which corresponds to two different representations of protein backbone local structural bias.

Initially, Ramachandran basin is an intuitive description of local structural bias [14]. A Ramachandran basin is a specific region of a Ramachandran plot and illustrates the preference of torsion angle values. Each angle pair can be assigned a basin label. With more basins, the assignment would be harder but the representation would be more accurate and vice versa. Colubri *et al.* tested the ability to recover the native structure from a given basin assignment for each residue to investigate the level of representation required to simulate folding and predict structure, resulting in five basins [15]. Gong *et al.* partitioned ϕ, ψ-space into a uniform grid of 36 squares, each $60^\circ \times 60^\circ$, thus resulting in 36 basins, and showed that they successfully reconstructed six proteins solely from their mesostate (basin label) sequences [16]. There are also some other methods to define basins and do angle region prediction with different definitions of basins [17, 18, 19, 20]. Although it is vital to determine the proper number of regions and clearly define the boundary, a universal algorithm to generate Ramachandran basins and assign basin labels remains to be developed. In our study, k-means clustering serves as the basin generator and label assigner.

While Ramachandran basin provides an overall description of conformation, it is a coarse-grained representation and lacks statistical explanations describing the torsion angle distributions of each basin. In consideration of the circular nature of angles, traditional parametric or non-parametric density estimation methods cannot work properly to approximate Ramachandran distributions. Fortunately, directional distributions such as von Mises distribution could solve the problem [21]. Bivariate von Mises distribution (mixtures) has been used to model protein dihedral angle distribution [22, 23], which removes arbitrariness in defining the boundary between discrete states. In this study, we assume angle pairs in each basin follow a bivariate von Mises distribution to derive the log-likelihood of each clustering.

Thanks to the rapid growth of Protein Data Bank and computational and algorithmic development in machine learning (especially deep learning), several supervised machine learning methods have been proposed to predict real values of dihedral angles. As ϕ values in α-helices and β-sheets are quite similar, ψ seems more informative. Wood *et al.* first developed a method DESTRUCT for prediction of real-valued dihedral angle ψ and used this information for prediction of the protein secondary structure with high accuracy [4]. Wu *et al.*proposed a composite machine-learning algorithm called ANGLOR to predict real-value protein backbone torsion angles from protein sequences [24]. The input features of ANGLOR include sequence profiles, predicted secondary structure and solvent accessibility. The mean absolute error (MAE) of the ϕ/ψ prediction was reported to be $28°/46°$. Later Song *et al.* developed TANGLE based on a two-level support vector regression approach using a variety of features derived from amino acid sequences, including the evolutionary profiles and natively disordered region as well as other global sequence features [25]. The MAE of the ϕ/ψ was $27.8°/44.6°$. Xue *et al.* established a neural network method called Real-SPINE, with sequence profiles generated from multiple sequence alignment and predicted secondary structures as inputs [26]. In 2015, they presented SPIDER2 [27] by improving SPIDER [28] through iterative learning, which used a deep artificial neural network (ANN) with three hidden layers of 150 nodes. They fed the predicted torsion angles of last layer as the input to the following generation and reported 19° and 30° for mean absolute errors of backbone ϕ and ψ angles, respectively. As it is impossible to introduce all methods here, interested readers can refer to excellent reviews [29, 30].

Although there has been tremendous development, their performance is still limited by their shallow architectures. Inspired by the excellent performance of convolution neural network in predicting secondary structure [31] and order/disorder regions [32] and also the success of residual framework to do contact prediction [33], we adopt the ultra deep residual framework of convolutional neural network to do k-means basin label probability prediction.

However, even though a protein backbone conformation can be highly accurately rebuilt from its respective native dihedral angles, accumulation of errors in predicted angles can lead to large deviation in three-dimensional structures, which prevents angle prediction from its direct use in building protein structures [27]. It is of great significance to produce the corresponding confidence scores for the real value predictions, i.e., we need to know the confidence level of the predictions. Otherwise the effect of predicted dihedral angles as restraints for three dimensional structure prediction would be limited [34]. Zhou *et al.* had developed SPIDER2 [27] to predict real-valued angles and then separately SPIDER2-Delta [35] to predict error of those predicted structural properties. Here we describe a simple hybrid technique to predict angles and confidence scores simultaneously.

Another problem that need to be considered is the periodicity of angles. For example, if an angle $\theta = 179°$ is predicted to be $-179°$, the error would be treated as 358° instead of 2°. There are some approaches proposed to reduce the impact of cyclic nature of angles. One was angle shifting to reduce confusion at 0° and 360° (or $-180°$ and 180°), e.g., shifting ψ by 100° and ϕ by $-10°$ [26] or adding 100° to the angles between $-100°$ and 180° and adding 460° to the angles between $-180°$ and $-100°$ [34]. But the improvement was limited and strongly depended on the angle range. For amino acids such as alanine that had minimal residues in the affected range, angle shifting made little difference [29]. A better choice was to take advantage of the inherent angle periodicity of trigonometric functions, that is, mapping the angles to their sine and cosine values [27], which has achieved best performance so far. Inspired by this, we deal with equivalent trigonometric representations of dihedral angle pairs, rather than real value angles.

Considering dihedral angles share similar patterns in alpha helix and beta strand, the acceptable (ϕ, ψ) patterns are limited. Moreover, it is much easier to do classification than regression. Also indebted to mixture models and Expectation-maximization algorithm, we develop a hybrid method of k-means clustering and deep learning to do angle prediction, combining advantages of discrete and continuous representation of dihedral angles. Specifically, we firstly generate a set of clusters of (ϕ, ψ) from training data, in which we could get the distribution of each cluster; then we use deep learning methods to predict discrete labels; lastly we predict real value angles by mixing empirical clusters with their predicted probabilities. We employ a residual framework of convolutional neuron network in RaptorX-Angle to predict the cluster label probabilities. We test our method on filtered PDB25 dataset as well as CASP (Critical Assessment of protein Structure Prediction) targets and compare with other three state-of-art methods. Tested on the subset of PDB25, our method gains about 0.5° and 1.4° for ϕ and ψ better MAE than SPIDER2, currently among the best backbone angle predictors. Our method also performs better than SPIDER2 on the CASP11 and CASP12 test targets. The advantage is even more obvious when looking into detailed secondary structural regions.

2 Methods

2.1 K-means clustering of angle vectors

Genearating k-means "centers" from angle vectors. For a dihedral angle pair (ϕ, ψ), we can equivalently denote it by an angle vector

$$\mathbf{v} = (\cos(\phi), \sin(\phi), \cos(\psi), \sin(\psi)).$$

Conversely, given the vector representation $\mathbf{v}$, we can easily derive the corresponding angles ϕ and ψ (Supplementary Material S1.1). We run k-means on angle vectors to cluster dihedral angle pairs in training set into $K = 10, 20, \ldots, 100$ clusters. Then we normalize the K centres $\{\mathbf{C}_k\}_{k=1}^{K}$ and get the final "centers" $\{\widetilde{\mathbf{C}}_k = (\widetilde{c}_{k0}, \widetilde{c}_{k1}, \widetilde{c}_{k2}, \widetilde{c}_{k3})\}_{k=1}^{K}$, so that each "centre" $\widetilde{\mathbf{C}}_k$ is a valid representation for some angle pair (Supplementary Material S1.2).

Predicting "true" labels from k-means. Given the K normalised vector "centres" $\{\widetilde{\mathbf{C}}_k\}_{k=1}^{K}$, we could assign the "true" label for each dihedral angle pair as the one whose corresponding normalised centre was closest to its respective vector representation. Then the "true" labels can be used as the training labels to build a deep learning model as a classifier to predict labels for testing data.

2.2 Deep learning model details

Deep Convolutional Neural Network (DCNN). DCNN consists of multiple convolutional blocks. A convolutional block is a neural network that implements a composite of linear convolution and nonlinear activation transformation. Convolution is used in place of general matrix multiplication, which can better capture local dependency. It has been widely accepted that protein torsion angles strongly depend on neighbour residues [36, 37, 38]. So DCNN is ideal to abstract angle information from sequence.

Residual Network (ResNet). DCNN can integrate features in hierarchical levels and some work has shown the significance of depth [39]. However, with the depth increasing, accuracy gets saturated and even degraded. That is because adding more layers may lead to higher training error as identity mapping is difficult to fit with a stack of nonlinear layers [40]. ResNet was proposed as a residual learning framework to ease the training of substantially deeper networks [41]. Figure 1 demonstrates the basic architecture of ResNet in RaptorX-Angle. Figure 1(A) is a residual block, which consists of 2 convolution layers and 2 activation layers, and the ResNet consists of stacked residual blocks (Figure 1B). The activation layer conducts a simple nonlinear transformation of its input depending on the activation function with no additional parameters. In this work, we used the ReLU activation function [42].

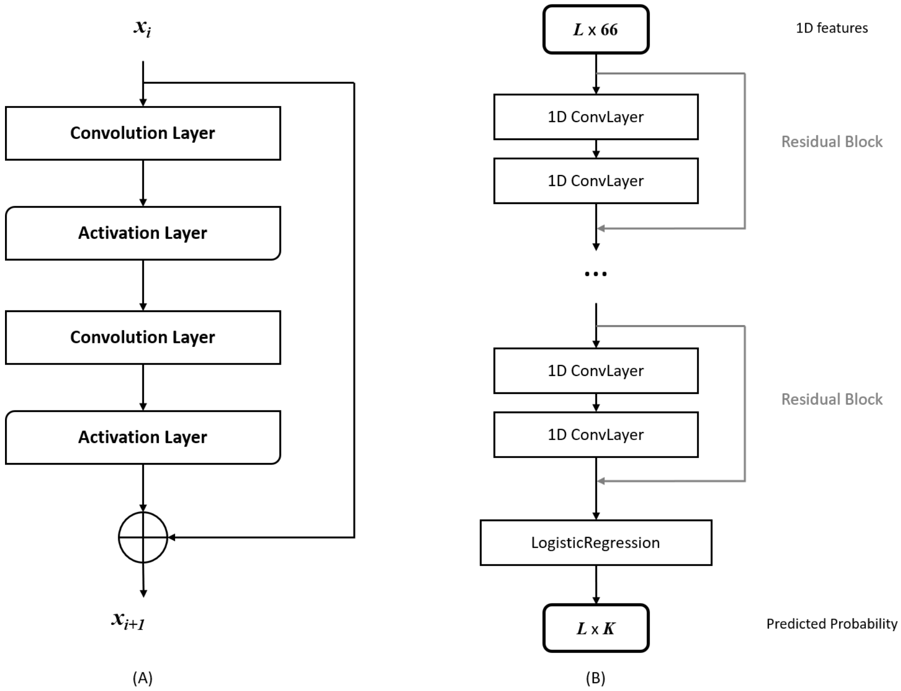

Figure 1: Illustration of the ResNet model in RaptorX-Angle. (A) A building block of ResNet with x_i and x_{i+1} being input and output, respectively. (B) The ResNet model architecture as a classifier with stacked residual blocks and a logistic regression layer. Here L is the sequence length of the protein or total number of residues under prediction and K is the number of clusters.

Logistic Regression layer. DCNN and ResNet can capture information from data and output abstract features. To do classification for residues, a logistic regression layer is added as the final layer in RaptorX-Angle, which could output the marginal probability of K labels (Figure 1(B)).

Loss function. We train model parameters through maximizing the probability of angle pairs belong to the "true" labels. Naturally, the loss function is defined as the negative log-likelihood averaged over all residues of the training proteins.

Regularization and optimization. As is widely used in machine learning, the log-likelihood objective function is penalized with a L_2-norm of the model parameters to prevent overfitting. Thus, the final objective function has two items: loss function and regularization item, with a regularization factor λ to balance the two items. That is, the final objective function is:

$$\max_{\theta} \quad \log P_\theta(Y|X) - \lambda\|\theta\|^2$$

where X is the input features, Y is the output labels, θ is the model parameters and λ is the regularization factor used to balance the log likelihood and regularization. We use Adam [43] to minimize the objective function, which usually can converge within 20 epochs. The whole algorithm has been implemented by Theano [44] and mainly run on a GPU card.

Input features. For each residue in each protein sequence, we generate a total of 66 input features, of which 20 from position specific scoring matrix(PSSM) of PSI-BLAST [45], 20 from position-specific frequency matrix (PSFM) of HHpred [46, 47], 20 from primary sequence, 3 from predicted solvent accessibility (ACC) and 3 from predicted secondary structure(SS) probabilities (Supplementary Material S1.3).

2.3 Predicting real-value angles from predicted marginal probability

From the last logistic regression layer of the deep learning model, we could predict the marginal probability $\mathbf{P} = (p_1, p_2, \ldots, p_K)$ of an angle pair for each label. We use the marginal probability rather than the single predicted label to reduce bias. Concretely, we calculate the weighted mean by:

$$\widehat{\mathbf{v}} = (v_0, v_1, v_2, v_3) = \sum_{k=1}^{K} p_k \widetilde{\mathbf{C_k}},$$

Finally, we normalise $\widehat{\mathbf{v}}$ to get

$$\widehat{\cos(\phi)} = \frac{v_0}{\sqrt{{v_0}^2 + {v_1}^2}}, \widehat{\sin(\phi)} = \frac{v_1}{\sqrt{{v_0}^2 + {v_1}^2}},$$

$$\widehat{\cos(\psi)} = \frac{v_2}{\sqrt{{v_2}^2 + {v_3}^2}}, \widehat{\sin(\psi)} = \frac{v_3}{\sqrt{{v_2}^2 + {v_3}^2}}.$$

and we could derive the predicted real values $\widehat{\phi}, \widehat{\psi}$ from this angle vector (Supplementary Material S1.1). We also tried to predict real-value angles from labels with top $R (R < K)$ probabilities when K is well chosen (Supplementary Material S2.3).

2.4 Programs to compare and evaluation metrics

We compare our method with three available standalone softwares SPIDER2 [27], SPINE X [11], and ANGLOR [24]. All the programs are run with parameters suggested in their respective papers.

We evaluate the performance by Pearson Correlation Coefficient (PCC) and Mean Absolute Error (MAE) as described by [48], for assessing the prediction of ϕ/ψ angles. Considering the periodicity of angles, PCC is calculated between the cosine (sine) values of predicted and experimentally determined angles. MAE is the average absolute difference between predicted and experimentally determined angles. The periodicity of an angle has been taken care of by utilizing the smaller value of the absolute difference $d(= |\theta_{pred} - \theta_{exp}|)$ and $360 - d$ for average, where θ_{pred} is the predicted angle and θ_{exp} is the true angle value.

3 Results

3.1 Datasets

We use the targets from PDB25 updated in February, 2016. The set consists of 10820 non-redundant protein chains, in which any two chains share no more than 25% sequence identity. To remove impact of disordered regions, we filter out proteins with internal disordered regions by DSSP [49]. Finally we get 7604 proteins. We then randomly select 5070 proteins as the candidate training set, 1267 as validation set (VL1267, see VL1267_list.txt) and the remaining 1267 as test set (TS1267, see TS1267_.txt). We also test on 85 CASP11 targets (see casp11_list.txt) and the latest 40 CASP12 targets (see casp12_list.txt) with publicly released native structures. To remove redundancy between training proteins and CASP targets, we run MMseqs2 [50], which is similar but more sensitive and faster than BLAST (PSI-BLAST) for protein sequence homology search, with *seqID* cutoff 0.25 and also E-value cutoff 0.001 to filter 5070 the candidate training proteins, resulting in 5046 training proteins (TR5046, see TR5046_list.txt).

3.2 Choosing a proper number of clusters

A vital problem is how to select the number of clusters, which can be reduced to defining measures for clustering evaluation. Here we adopt two measures: (i) entropy loss based on discrete distribution; (ii) loglikelihood based on continuous distribution to evaluate 10 different clusterings ($K = 10, 20, \ldots, 100$).

Firstly, we do k-means clustering on TR5046 and get K empirical clusters. Secondly, we train the deep learning models and do classification on VL1267, then we can obtain the predicted marginal probability of the K clusters $\mathbf{P_i} = (p_{i1}, p_{i2}, \ldots, p_{iK}), i = 1, 2, \ldots, N$, where i is the index of residue and N is the total number of residues in VL1267.

Entropy loss. Entropy $H(\cdot)$ is always used to measure the information of a distribution. From k-means clustering on TR5046, the background distribution among clusters $\mathbf{P_0} = (p_{01}, p_{02}, \ldots, p_{0K})$ could be derived. Then the entropy loss of this clustering on VL1267 can be calculated as the mean difference between entropy of background distribution and predicted marginal distribution:

$$\begin{aligned} EL &= \frac{1}{N}\sum_{i=1}^{N}(H(\mathbf{P_0}) - H(\mathbf{P_i})) \\ &= \frac{1}{N}\sum_{i=1}^{N}(\sum_{k=1}^{K} p_{0k}\log(p_{0k}) - \sum_{k=1}^{K} p_{ik}\log(p_{ik})) \end{aligned}$$

which can roughly evaluate the information gain from the clustering. Here N is the number of residues in VL1267.

Loglikelihood. To demonstrate the detailed information of each cluster, we need a continuous angular(circular) distribution defined on the torus. Mixture bivariate von Mises distributions are successfully used to describe the local bias of torsion angle pair (ϕ, ψ) [21, 22, 23], we assume that angle pairs belong to the same cluster k obey a common bivariate von Mises distribution f_k with parameters $\Theta_k = (\kappa_1^k, \kappa_2^k, \kappa_3^k, \mu^k, \nu^k)$. Here,

$$\begin{aligned} f_k(\phi, \psi) &= c(\kappa_1^k, \kappa_2^k, \kappa_3^k)\exp\{\kappa_1^k\cos(\phi - \mu^k) \\ &\quad +\kappa_2^k\cos(\psi - \nu^k) + \kappa_3^k\cos(\phi - \mu^k - \psi + \nu^k)\} \end{aligned}$$

where μ^k and ν^k are the mean value of ϕ and ψ, respectively; κ_1^k, κ_2^k are the concentrations, κ_3^k allows for the dependency between the two angles and $c(\kappa_1^k, \kappa_2^k, \kappa_3^k)$ is a normalization constant:

$$c(\kappa_1^k, \kappa_2^k, \kappa_3^k) = (2\pi)^2\{I_0(\kappa_1^k)I_0(\kappa_2^k)I_0(\kappa_3^k) + 2\sum_{p=1}^{\infty} I_p(\kappa_1^k)I_p(\kappa_2^k)I_p(\kappa_3^k)\}$$

in which $I_p(\kappa)$ is the modified Bessel function of the first kind and order p. Parameters $\{\Theta_k = (\kappa_1^k, \kappa_2^k, \kappa_3^k, \mu^k, \nu^k)\}_{k=1}^K$ can be intuitively estimated from the empirical clusters $\{(\phi, \psi)_k\}_{k=1}^K$ [51]. Then the density function for the torsion angle pair (ϕ, ψ) can be approximately described as:

$$f(\phi, \psi) = \sum_{k=1}^{K} p_k f_k(\phi, \psi)$$

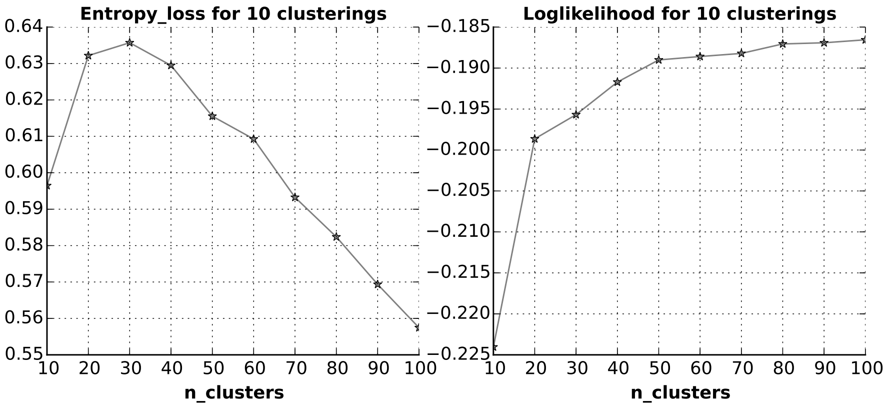

Figure 2: Selecting proper number of clusters. Left: entropy loss of discrete label probabilities; Right: loglikelihood of mixture bivariate von Mises distribution.

where p_k is the predicted marginal probability of (ϕ, ψ) belongs to cluster k. Then the log-likelihhod for the VL1267 can be calculated as:

$$LL = \frac{1}{N}\sum_{i=1}^{N} \log f(\phi_i, \psi_i) = \frac{1}{N}\sum_{i=1}^{N} \log \sum_{k=1}^{K} p_{ik} f_k(\phi_i, \psi_i)$$

Selecting proper K. Figure 2 shows the result of entropy loss and loglikelihood with respect to the number of clusters. As expected, the loglikelihood increases along with K, which means it can better describe the data with more clusters. But when K goes larger than 30, there is an obvious decrease in entropy loss. Maybe that is because the more clusters are used, the more challenging it would be to do angle prediction. As there is a soaring information gain when K goes from 10 to 20 and little difference when K increases from 20 to 30, we test every single clustering between 20 and 30 and there is no significant benefit with more clusters. So we just choose $K = 20$ to do following studies.

3.3 Feature contribution study

The features can be divided into three categories: sequence information including amino acid (aa) and profile, predicted secondary structure (SS) and solvent accessibility (ACC). Sequence profile information are generated from PSI-BLAST (PSSM) and HHpred (PSFM) (See Supplementary Material S1.3 for more details). To test the impact of different feature combinations, we design six experiments: (1) basic1 = 20 PSSM + 20 aa; (2) basic2 = 20 PSFM + 20 aa; (3) basic = 20 PSSM + 20 PSFM + 20 aa; (4) basic + 3 ACC; (5) basic + 3 SS; (6) basic + 3 ACC + 3 SS. The network architecture is fixed as $N_{layers} = 5, N_{nodes} = 100, halfWinSize = 3$ (ResNet 5-100-3), and the regularization factor is fixed to be 0.0001.

Table 1 shows the MAE performance of different feature combinations on TS1267. From the first three experiments with only sequence information involved, there is little performance difference between PSSM and PSFM, and the combination of PSSM and PSFM gains the best accuracy. So PSSM and PSFM are complementary and both unignorable. ACC and SS both contribute significantly and also the combination gain the best accuracy. Finally we use the whole set of features.

Table 1: The Mean Absolute Error of different feature combinations with ResNet 5-100-3 on TS1267.

ID	Phi	Psi	Phi_H	Psi_H	Phi_E	Psi_E	Phi_C	Psi_C
(1)	19.97	31.97	9.82	17.57	20.70	26.97	29.84	49.66
(2)	20.02	31.78	9.86	17.68	20.46	26.38	30.10	49.39
(3)	19.27	30.04	9.11	15.58	19.70	24.64	29.35	48.02
(4)	19.08	29.30	9.07	15.44	19.36	23.18	29.11	47.10
(5)	19.19	28.73	8.56	13.76	19.29	22.43	31.00	47.95
(6)	**18.58**	**27.98**	**8.45**	**13.37**	**19.03**	**22.14**	**28.61**	**46.21**

Phi and Psi denote MAE for all residues;
Phi_H and Psi_H denote MAE for residues in helix region;
Phi_E and Psi_E denote MAE for residues in beta strand region;
Phi_C and Psi_C denote MAE for residues in coil region.
$(\cdot)$ indicates the id of the feature contribution experiments.

3.4 Overall PCC performance of cosine values compared with other methods

To tune proper regularization factor and also network architectures, we perform 5-fold cross validation on TR5046 (Supplementary Material S2.1, S2.2). Finally, we choose an ensemble of 6 networks (Supplementary Material S2.2). We test our method on TS1267 and also the popular CASP targets, including 85 CASP11 targets and 40 CASP12 targets. Table 2 shows the PCC performance of cosine values on the three benchmarks. RaptorX-Angle has gained the highest PCC on all datasets. We also evaluate PCC performance of sine values (See Supplementary Material S2.4) and get similar results.

Table 2: Pearson correlation coefficient of cosine values between predicted and true angles.

	TS1267	**CASP11**	**CASP12**
	$\cos(\phi)/\cos(\psi)$	$\cos(\phi)/\cos(\psi)$	$\cos(\phi)/\cos(\psi)$
RaptorX-Angle	0.7111/0.7576	0.6585/0.7103	0.6539/0.6979
SPIDER2	0.6893/0.7427	0.6485/0.7095	0.6299/0.6761
SPINE X	0.6410/0.6543	0.5015/0.4891	0.4990/0.5039
ANGLOR	0.4775/0.6226	0.4437/0.5868	0.4431/0.5772

3.5 Overall MAE performance compared with other methods

Table 3 shows the MAE performance on the three benchmarks in different secondary structural regions of our RaptorX-Angle comparing with other three methods. All methods have larger MAE on CASP targets than on TS1267. It is reasonable since CASP targets are usually hard to predict. It can be seen that RaptorX-Angle performs the best on all benchmarks, with about 0.5° and 1.4° for ϕ and ψ better MAE on both TS1267 and CASP12 and slightly better performance on CASP11 than the second best method SPIDER2. We perform Student's t test of absolute errors between RaptorX-Angle and SPIDER2. As a result, the p-values for ϕ/ψ are $8.65e-12/2.79e-33$, $5.13e-2/8.36e-2$ and $1.28e-5/2.59e-8$ on TS1267, CASP11 and CASP12, respectively. That is , the advantage of RaptorX-Angle over SPIDER2 on TS1267 and CASP12 is statistically more significant than on CASP11. These results demonstrate the rationality of representing the Ramachandran plot with a limited number of clusters, say 20 clusters, and also reflect the power of deep learning methods.

Table 3: Mean absolute error of four methods for different secondary structural regions on three benchmarks: TS1267, 85 CASP11 targets and 40 CASP12 targets.

(°)	Phi	Psi	Phi_H	Psi_H	Phi_E	Psi_E	Phi_C	Psi_C
TS1267								
RaptorX-Angle	$\underline{18.08}$	$\underline{26.68}$	$\underline{8.35}$	$\underline{12.98}$	$\underline{18.24}$	$\underline{20.94}$	$\underline{27.88}$	$\underline{44.11}$
SPIDER2	18.57	28.02	8.59	14.52	19.28	23.09	28.28	44.73
SPINE X	20.31	34.05	9.32	16.69	22.23	31.23	30.32	53.42
ANGLOR	24.01	43.59	9.29	26.41	27.47	40.88	36.89	62.72
CASP11								
RaptorX-Angle	$\underline{20.00}$	$\underline{30.14}$	$\underline{9.49}$	$\underline{15.65}$	$\underline{18.82}$	$\underline{23.58}$	$\underline{29.87}$	46.89
SPIDER2	20.18	30.32	9.53	16.05	19.77	24.50	29.88	$\underline{46.84}$
SPINE X	24.85	46.58	13.57	29.65	26.25	43.65	33.88	63.49
ANGLOR	25.69	46.17	9.99	27.72	28.08	43.85	37.96	64.03
CASP12								
RaptorX-Angle	$\underline{20.69}$	$\underline{32.73}$	9.28	$\underline{16.73}$	$\underline{19.94}$	$\underline{26.06}$	$\underline{31.22}$	$\underline{51.02}$
SPIDER2	21.13	34.17	$\underline{9.13}$	17.19	21.35	28.56	31.95	52.76
SPINE X	24.85	46.57	11.52	26.34	26.98	46.04	35.85	65.33
ANGLOR	25.79	47.37	9.69	28.81	29.11	44.79	38.65	65.74

Same notations with Table 1

3.6 Mean absolute error performance study in VL1267

In methodology, the conversion from angle pair to trigonometric vector is nonlinear, the prediction error may depend on angles. And in biology, prediction

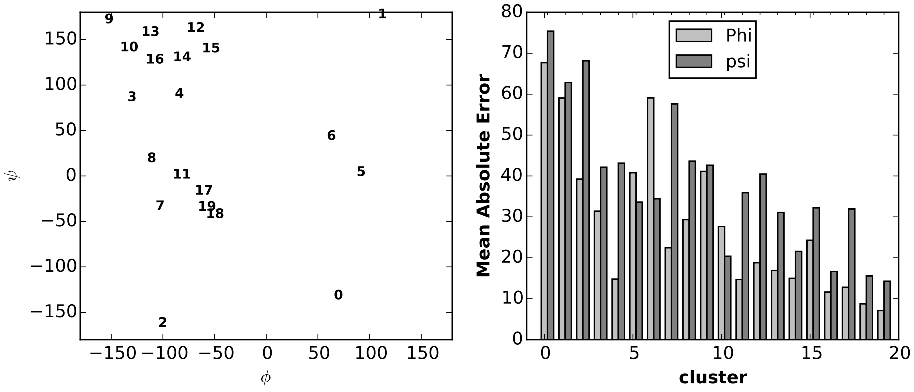

Figure 3: Mean absolute error performance for different clusters in VL1267. Left: visualization of 20 cluster centers on the Ramachandran plot with smaller number indicating smaller size. Right: mean absolute error for different clusters.

error may differ for different amino acids with different microscopic biochemical properties, and also for different protein classes with different macroscopic structures. So we perform detailed studies on prediction error in VL1267.

Study Mean absolute error performance for different clusters. As each cluster corresponds to a certain angle region, we calculate the MAE for each cluster in VL1267. We observe that the 20 clusters are well consistent with Ramachandran plot and also the two peaks for ϕ and ψ [11] (Figure 3 Left). And the prediction errors differ a lot between clusters. It turns out that clusters with more residues in coil region tend to result in larger prediction errors. Moreover, prediction error for ϕ is smaller than for ψ. But there are three uncommon clusters with larger MAE for ϕ, i.e., 5, 6 and 10 (Figure 3 Right). Clusters 5 and 6 are totally in one of the peak areas in Ramachandran plot, which may indicate some interesting biological discoveries.

Mean absolute error performance for each amino acid type. As different amino acids have different stereochemical and physiochemical properties, they are anticipated to have different degrees of difficulty for the torsion angle prediction. In Table 4, we examine the MAE performance for each of 20 amino acid types. Glycine, with no side-chain atom except for a proton, has least steric restriction to backbone dihedral angle motions. As a result, it has the largest prediction error ($43.32°/39.59°$ for ϕ/ψ). In contrast, Proline has the least MAE ($8.84°$) for ϕ but has an unusually large MAE ($33.00°$) for ψ prediction due to its special side-chain structure, which is consistent with [24]. In addition, three of the amino acids (Ile, Leu and Val) with the smallest MAE are all hydrophobic.

Table 4: Mean absolute error performance for each amino acid type in VL1267.

Amino acids	Abundance	Frequency(%)	$\phi(^\circ)$	$\psi(^\circ)$
A (Ala)	22527	8.46	13.87	22.92
C (Cys)	3151	1.18	20.50	28.66
D (Asp)	15946	5.99	20.71	30.80
E (Glu)	18326	6.89	14.75	23.97
F (Phe)	10812	4.06	18.13	26.10
G (Gly)	19133	7.19	**43.32**	**39.59**
H (His)	5989	2.25	22.04	31.12
I (Ile)	15302	5.75	12.79	20.12
K (Lys)	15299	5.75	16.71	25.83
L (Leu)	24731	9.29	12.49	21.37
M (Met)	5833	2.19	16.71	24.86
N (Asn)	11383	4.28	27.38	32.04
P (Pro)	11977	4.50	**8.84**	**33.00**
Q (Gln)	10163	3.82	15.96	24.72
R (Arg)	13529	5.08	16.81	25.45
S (Ser)	15991	6.01	20.83	33.92
T (Thr)	14309	5.38	17.12	30.92
V (Val)	18612	6.99	13.70	20.94
W (Trp)	3854	1.45	18.05	27.61
Y (Tyr)	9287	3.49	18.83	27.02
Total	266154	100	18.32	27.15

Mean absolute error performance for different protein classes. After studying on MAE performance in microcosmic view, we intend to study the performance for different macroscopical structures. We abstract 99, 117, 171, 117 proteins from VL1267 (resulted in 17696, 24874, 47304 and 19645 residues) in all α, all β, α/β and $\alpha+\beta$ classes, respectively. We calculate the absolute error for every residue in each class. Figure 4 shows the violin plot of prediction error for ϕ (Left) and ψ (Right). A violin plot is similar to box plot except that it also shows the probability density of the data. We can see although the MAE for ϕ are smaller for all protein classes, prediction errors belong to each protein class have their own distribution pattern and the pattern is similar between ϕ and ψ. Overall, prediction errors are smallest in all α proteins and largest in all β for both ϕ and ψ predictions.

3.7 Estimating confidence score of predicted angles

Generally, variance σ^2 includes variance within cluster σ_w^2 and variance between cluster σ_b^2. To produce the confidence score of our predicted angles, we calculate the standard deviation from variances within a cluster. Specifically, for each cluster k, we can get the in-cluster variance $\sigma_k^2(\theta)$ from training data, where

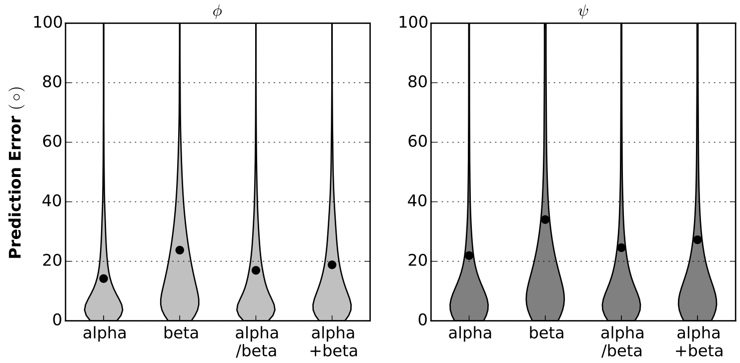

Figure 4: Mean absolute error performance for different protein classes in VL1267. Left: for ϕ prediction. Right: for ψ prediction.

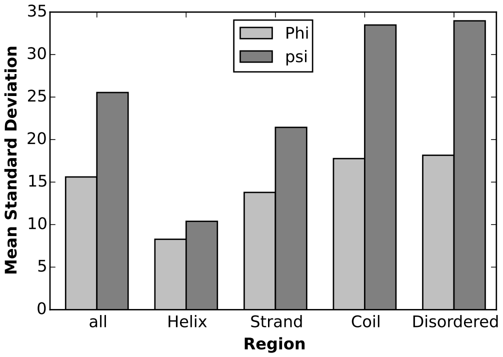

Figure 5: Mean standard deviation for different secondary structural regions in TS1267.

$\theta = \phi$ or ψ. Then we derive the variance of prediction by:

$$var(\theta) = \sigma^2(\theta) = \sum_{k=1}^{K} p_k \sigma_k^2(\theta)$$

Figure 5 shows the mean standard deviation for ϕ and ψ in different regions. As expected, the smallest variance appears in helix region, and then strand and lastly coil region. The standard deviation in disordered regions are rather large and quite similar to coil regions, which is consistent with our prior knowledge that disordered region resembles loop region and is rather flexible.

Figure 6 demonstrates the relationship between MAE and mean standard deviation for ϕ and ψ in different regions on VL1267. Roughly, the relationship is linear ($R^2 = 0.8911$). So the MAE can be bounded well by the standard

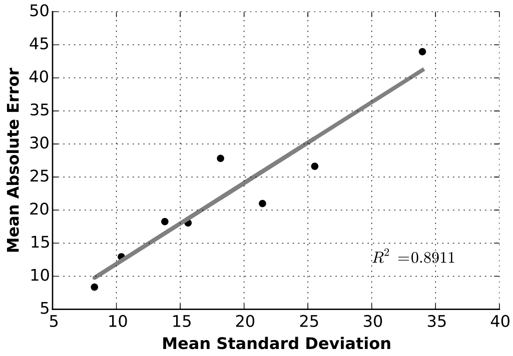

Figure 6: Relationship between prediction error and standard deviation. Eight points are for two kinds of angles in four secondary structural regions (total, helix, strand, coil).

deviation. We predict the error for each residue in each target from TS1267 and calculate corresponding Pearson and Spearman correlation coefficients (PCC and SCC) between prediction errors and true errors, and also the mean absolute error for prediction errors (MAEPE). Finally, we obtain $PCC = 0.3109, SSC = 0.5427, MAEPE = 13.94$ for ϕ and $PCC = 0.2597, SCC = 0.4751, MAEPE = 26.21$ for ψ. We also try to fit two linear models for ϕ and ψ separately on the all data points in VL1267 and get similar testing results. This indicates that the mean for different secondary structural regions almost contains enough information about the relationship between the estimated standard deviation and prediction error (Supplementary Material S2.6).

3.8 Computational cost analysis

All mentioned methods could do angle prediction target by target, so the computational cost is bounded by the longest protein (i.e., protein with the largest number of residues). To generate angle predictions for 1xdoA, the largest protein in TS1267 with 685 residues, it takes 726s, 123s, 370s and 524s for ANGLOR, spineX, SPIDER2 and RaptorX-Angle, respectively.

As far as we see, the computational cost is mainly determined by method outline, network complexity, feature engineering and technical resources. ANGLOR is a composite method and the technology was not so developed at that time, it needs the most time. While spineX just adopted a simple network, SPIDER2 used more features iteratively in a more complex network and it takes longer than spineX.

Compared with the second best SPIDER2, RaptorX-Angle used much deeper networks and also adopted profile information from hhblits (PSFM), besides PSSM from PSI-BLAST harnessed by spineX and SPIDER2. As a result, it takes SPIDER2 360s to generate features with 4 CPUs and 20s to predict angles using a CPU, while it takes RaptorX-Angle 385s to generate features with 4 CPUs, and 200s to predict angles from the features using a GPU card.

However, we can integrate the features of a total batch of proteins and run them all at once. Actually, it just takes 750s to do angle prediction for all proteins in TS1267, while other methods needs many CPUs in parallel. Overall, our method is faster for prediction of many proteins and has gained better prediction accuracy.

4 Discussion

We have transformed the hard real-valued prediction problem into a discrete label assignment problem, which has simplified the problem and also gained better results. Overall, our RaptorX-Angle gains the best PCC in terms of cosine and sine of angles on all datasets. It has about 0.5° and 1.4° for ϕ and ψ better MAE than the second best method SPIDER2 on a subset of PDB25. We have also calculated the two-state accuracy to see how much improvement there would be in large angle errors. RaptorX-Angle performs the best and has about 0.15 and 1 percent improvement over SPIDER2 for ϕ and ψ on TS1267(See Supplementary Material S2.5). Our method also works very well on the CASP targets. Moreover, we have estimated the prediction errors at each residue by a mixture of the clusters with their predicted probabilities. It has been shown that there is approximately linear relationship between the real prediction error and in-cluster standard deviation. That is a unique feature of our method. In addition, we check the prediction for disordered regions. As there is no angle information, we just analyze the standard deviation and get quite large values and similar patterns to coil region. It is consistent with our prior knowledge that disordered region is rather flexible and resembles loop region. We also do comprehensive studies on prediction performance in VL1267, both in microscopic and macroscopic view.

This simple technique has gained better performance than other state-of-art methods. It demonstrates that for protein structures, the 20 clusters contain enough information for (ϕ, ψ), which is an efficient compression of information. The idea that to predict dihedral angles from clustering has turned out to be successful due to three aspects. The first is the continuous growth of the solved structures [52], so we have enough training data. The second is the novel idea to predict real-value angles by mixing a set of clusters with their respective predicted probabilities. Conversely, such good performance demonstrated that the distribution of protein backbone dihedral angles can be described through a set of clusters. Last but not the least, the everlasting development of deep learning models and optimization methods proves to be a powerful tool to promote new ideas and exploit new methods.

But there is still room for improvement. RaptorX-Angle just used one-dimensional features and adopted 1D CNN. It cannot extract information of long range interaction. Heffernan *et al.* has developed more accurate SPIDER3 employing Long Short-Term Memory (LSTM) Bidirectional Recurrent Neural

Networks (BRNNs), which are capable of capturing long range interactions [53]. That is, considering pairwise interaction can further increase prediction accuracy. We will include two-dimensional features and exploit 2D CNN to see how much improvement could be achieved.

Moreover, as mentioned before, accumulation of prediction errors has buried the usefulness of torsion angles to construct 3D models. There is a great demand to develop a proper technique to deal with the errors. A general pipeline to add angle restraints and confidence to improve protein tertiary structure prediction need to be developed.

In conclusion, this study has made a more reliable prediction of dihedral angles and may facilitate protein structure prediction and functional study. In the near future, we can use the angle restraints to do tertiary structure prediction, which should be considered carefully to deal with errors and flexibility. We can also adopt the angle prediction to aid structure alignment and fold recognition.

Acknowledgements

This work has been partly supported by the National Key Research and Development Program of China(No.2016YFA0502303), the National Key Basic Research Project of China (No. 2015CB910303), the National Natural Science Foundation of China (No.31471246) and China Scholarship Council. It is also supported by National Institutes of Health grant R01GM089753 to JX and National Science Foundation grant DBI-1564955 to JX. The authors are also grateful to the support of Nvidia Inc.

References

[1] Debora S Marks, Thomas A Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature biotechnology*, 30(11):1072–1080, 2012.

[2] David De Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013.

[3] Lukasz Kurgan and Fatemeh Miri Disfani. Structural protein descriptors in 1-dimension and their sequence-based predictions. *Current Protein and Peptide Science*, 12(6):470–489, 2011.

[4] Matthew J Wood and Jonathan D Hirst. Protein secondary structure prediction with dihedral angles. *PROTEINS: Structure, Function, and Bioinformatics*, 59(3):476–481, 2005.

[5] Petros Kountouris and Jonathan D Hirst. Predicting β-turns and their types using predicted backbone dihedral angles and secondary structures. *BMC bioinformatics*, 11(1):407, 2010.

[6] Eshel Faraggi, Tuo Zhang, Yuedong Yang, Lukasz Kurgan, and Yaoqi Zhou. Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, 33(3):259–267, 2012.

[7] Xijiang Miao, Peter J Waddell, and Homayoun Valafar. Tali: local alignment of protein structures using backbone torsion angles. *Journal of bioinformatics and computational biology*, 6(01):163–181, 2008.

[8] S Jung, SE Bae, and HS Son. Validity of protein structure alignment method based on backbone torsion angles. *J Proteomics Bioinform*, 4:218–226, 2011.

[9] Chao Zhang, Jingtong Hou, and Sung-Hou Kim. Fold prediction of helical proteins using torsion angle dynamics and predicted restraints. *Proceedings of the National Academy of Sciences*, 99(6):3581–3585, 2002.

[10] Wei Zhang, Song Liu, and Yaoqi Zhou. Sp 5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model. *PloS one*, 3(6):e2325, 2008.

[11] Eshel Faraggi, Yuedong Yang, Shesheng Zhang, and Yaoqi Zhou. Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure*, 17(11):1515–1527, 2009.

[12] Gregory E Sims and Sung-Hou Kim. A method for evaluating the structural quality of protein models by using higher-order φ–ψ pairs scoring. *Proceedings of the National Academy of Sciences of the United States of America*, 103(12):4428–4432, 2006.

[13] Chen Cao and Shutan Xu. Improving the performance of the plb index for ligand-binding site prediction using dihedral angles and the solvent-accessible surface area. *Scientific Reports*, 6, 2016.

[14] GN t Ramachandran and V Sasisekharan. Conformation of polypeptides and proteins. *Advances in protein chemistry*, 23:283–437, 1968.

[15] Andrés Colubri, Abhishek K Jha, Min-yi Shen, Andrej Sali, R Stephen Berry, Tobin R Sosnick, and Karl F Freed. Minimalist representations and the importance of nearest neighbor effects in protein folding simulations. *Journal of molecular biology*, 363(4):835–857, 2006.

[16] Haipeng Gong, Patrick J Fleming, and George D Rose. Building native protein conformation from highly approximate backbone torsion angles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(45):16227–16232, 2005.

[17] David L Dowe, Lloyd Allison, Trevor I Dix, Lawrence Hunter, Chris S Wallace, and Timothy Edgoose. Circular clustering by minimum message length of protein dihedral angles. 1995.

[18] Rui Kuang, Christina S Leslie, and An-Suei Yang. Protein backbone angle prediction with machine learning approaches. *Bioinformatics*, 20(10):1612–1621, 2004.

[19] Olav Zimmermann and Ulrich HE Hansmann. Support vector machines for prediction of dihedral angle regions. *Bioinformatics*, 22(24):3009–3015, 2006.

[20] Shesheng Zhang, Shengping Jin, and Bin Xue. Accurate prediction of protein dihedral angles through conditional random field. *Frontiers in biology*, 8(3):353–361, 2013.

[21] Harshinder Singh, Vladimir Hnizdo, and Eugene Demchuk. Probabilistic model for two dependent circular variables. *Biometrika*, pages 719–723, 2002.

[22] Kanti V Mardia, Charles C Taylor, and Ganesh K Subramaniam. Protein bioinformatics and mixtures of bivariate von mises distributions for angular data. *Biometrics*, 63(2):505–512, 2007.

[23] Shuai Cheng Li, Dongbo Bu, Jinbo Xu, and Ming Li. Fragment-hmm: A new approach to protein structure prediction. *Protein Science*, 17(11):1925–1934, 2008.

[24] Sitao Wu and Yang Zhang. Anglor: a composite machine-learning algorithm for protein backbone torsion angle prediction. *PLoS One*, 3(10):e3400, 2008.

[25] Jiangning Song, Hao Tan, Mingjun Wang, Geoffrey I Webb, and Tatsuya Akutsu. Tangle: two-level support vector regression approach for protein backbone torsion angle prediction from primary sequences. *PloS one*, 7(2):e30361, 2012.

[26] Bin Xue, Ofer Dor, Eshel Faraggi, and Yaoqi Zhou. Real-value prediction of backbone torsion angles. *Proteins: Structure, Function, and Bioinformatics*, 72(1):427–433, 2008.

[27] Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, Yuedong Yang, and Yaoqi Zhou. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, 5, 2015.

[28] James Lyons, Abdollah Dehzangi, Rhys Heffernan, Alok Sharma, Kuldip Paliwal, Abdul Sattar, Yaoqi Zhou, and Yuedong Yang. Predicting backbone cα angles and dihedrals from protein sequences by stacked sparse

auto-encoder deep neural network. *Journal of computational chemistry*, 35(28):2040–2046, 2014.

[29] Harinder Singh, Sandeep Singh, and Gajendra PS Raghava. Evaluation of protein dihedral angle prediction methods. *PloS one*, 9(8):e105667, 2014.

[30] Olav Zimmermann. Backbone dihedral angle prediction. *Prediction of Protein Secondary Structure*, pages 65–82, 2017.

[31] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6, 2016.

[32] Sheng Wang, Shunyan Weng, Jianzhu Ma, and Qingming Tang. Deepcnf-d: predicting protein order/disorder regions by weighted deep convolutional neural fields. *International journal of molecular sciences*, 16(8):17315–17330, 2015.

[33] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):e1005324, 2017.

[34] Eshel Faraggi, Bin Xue, and Yaoqi Zhou. Improving the prediction accuracy of residue solvent accessibility and real-value backbone torsion angles of proteins by guided-learning through a two-layer neural network. *Proteins: Structure, Function, and Bioinformatics*, 74(4):847–856, 2009.

[35] Jianzhao Gao, Yuedong Yang, and Yaoqi Zhou. Predicting the errors of predicted local backbone angles and non-local solvent-accessibilities of proteins by deep neural networks. *Bioinformatics*, 32(24):3768–3773, 2016.

[36] Marcos R Betancourt and Jeffrey Skolnick. Local propensities and statistical potentials of backbone dihedral angles in proteins. *Journal of molecular biology*, 342(2):635–649, 2004.

[37] O Keskin, D Yuret, A Gursoy, M Turkay, and B Erman. Relationships between amino acid sequence and backbone torsion angle preferences. *Proteins: Structure, Function, and Bioinformatics*, 55(4):992–998, 2004.

[38] Abhishek K Jha, Andres Colubri, Muhammad H Zaman, Shohei Koide, Tobin R Sosnick, and Karl F Freed. Helix, sheet, and polyproline ii frequencies and strong nearest neighbor effects in a restricted coil library. *Biochemistry*, 44(28):9691–9702, 2005.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[40] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[42] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[43] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[45] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

[46] Johannes Söding. Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7):951–960, 2004.

[47] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

[48] Petros Kountouris and Jonathan D Hirst. Prediction of backbone dihedral angles and protein secondary structure using support vector machines. *BMC bioinformatics*, 10(1):437, 2009.

[49] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[50] Martin Steinegger and Johannes Söding. Sensitive protein sequence searching for the analysis of massive data sets.

[51] Thomas Hamelryck, Kanti Mardia, and Jesper Ferkinghoff-Borg. *Bayesian methods in structural bioinformatics*. Springer, Cambridge, 2012.

[52] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[53] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. 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*, page btx218, 2017.