Prediction of Protein Backbone Torsion Angles Using Deep Residual Inception Neural Networks

Chao Fang [Student Member IEEE],

Department of Electrical Engineering and Computer Science, University of Missouri- Columbia, MO 65211

Yi Shang [Senior Member IEEE], and

Department of Electrical Engineering and Computer Science, University of Missouri- Columbia, MO 65211

Dong Xu [Member IEEE]

Department of Electrical Engineering and Computer Science and Bond Life Sciences Center, University of Missouri- Columbia, MO 65211

Abstract

Prediction of protein backbone torsion angles (Psi and Phi) can provide important information for protein structure prediction and sequence alignment. Existing methods for Psi-Phi angle prediction have significant room for improvement. In this paper, a new deep residual inception network architecture, called DeepRIN, is proposed for the prediction of Psi-Phi angles. The input to DeepRIN is a feature matrix representing a composition of physico-chemical properties of amino acids, a 20-dimensional position-specific substitution matrix (PSSM) generated by PSI-BLAST, a 30-dimensional hidden Markov Model sequence profile generated by HHBlits, and predicted eight-state secondary structure features. DeepRIN is designed based on inception networks and residual networks that have performed well on image classification and text recognition. The architecture of DeepRIN enables effective encoding of local and global interactions between amino acids in a protein sequence to achieve accurate prediction. Extensive experimental results show that DeepRIN outperformed the best existing tools significantly. Compared to the recently released state-of-the-art tool, SPIDER3, DeepRIN reduced the Psi angle prediction error by more than 5 degrees and the Phi angle prediction error by more than 2 degrees on average. The executable tool of DeepRIN is available for download at <http://dlsrv8.cs.missouri.edu/~cf797/MUFoldAngle/>.

Keywords

Protein structure prediction; protein backbone torsion angles; machine learning; deep neural networks; inception networks; residual networks

1. INTRODUCTION

Predicting a protein structure accurately from its amino acid sequence information alone has remained difficult for half a century [1, 2, 3]. One approach of helping predict a protein structure is to predict its backbone torsion angles (Psi and Phi). The backbone structure of a protein can be determined by its torsion angles (see Figure 1 for illustration of what Psi Phi angles are). Given an amino acid sequence, a number of machine-learning methods have been applied to predict the real-value torsion angles for each residue. The prediction accuracies have improved gradually over the years [4, 5, 6] and are approaching the ones estimated from NMR chemical shifts [6]. It has been demonstrated that the predicted torsion angles are useful to improve *ab initio* structure prediction [7, 8], sequence alignment [9], secondary structure prediction [10, 11], template-based tertiary structure prediction and fold recognition [12, 13, 14], protein structure classification and loop modeling [15,16], and protein conformation sampling [17].

Existing torsion angle prediction methods formulate the problem slightly differently. Some convert torsion angles to discrete classes [18, 19], while others treat them as continuous values [4, 5,11, 20]. Various machine-learning methods, especially deep neural networks in recent years, have been applied to this prediction problem. In [21], a deep recurrent restricted boltzman machine (DreRBM) was developed to achieve comparable results to SPIDER2 [22], a well-known software tool. SPIDER3 [23], a recently released tool that improved over SPIDER2, applied long-short term memory [24] bidirectional recurrent neural networks (LSTM-biRNN) [25] to achieve 5% to 10% reduction in the mean absolute error for Psi and Phi angle predictions, respectively, over SPI-DER2.

Traditional neural networks have been applied to torsion angle prediction for a long time. They typically used windows of neighboring amino acid residues to capture relationship between residues within a short range [5, 6, 26]. The ability of capturing the interactions between residues is limited by a small, fixed window size. Long-range, non-local residue relationships are not captured and used in these prediction methods. With the development of deep learning, deep neural networks are capable of capturing and using long-range information in prediction. For example, SPIDER3 used LSTM-biRNN to capture both short and long-range residue information effectively. In addition to deep recurrent networks, deep convolutional neural networks (CNN) has been applied in a large number of challenging machine learning tasks, such as image recognition [27, 28, 29], automatic machine translation [30], text generation [31], and achieved great success.

Many advanced deep-learning architectures, such as VGG [32], Inception [33], Xception [34], ResNet [35], DenseNet [36], have emerged in recent years with improved performance over previous methods in various applications. Inspired by these cutting-edge architectures, in this paper, a new deep residual inception network architecture, called DeepRIN, is proposed for the prediction of torsion angles. The input to DeepRIN is a carefully designed feature matrix representing the primary sequence of a protein, which consists of a rich set of information for individual amino acid and the context of the protein sequence. DeepRIN is designed based on the ideas of inception networks and residual networks, and predicts sine

and cosine values of Psi and Phi angles to remove periodicity, which can then be converted back to angle values. The contributions of this work include:

- (a) A new deep neural network architecture, DeepRIN, is proposed to effectively capture and process both local and global interactions between amino acids in a protein sequence.
- (b) Extensive experiments using popular benchmark datasets were conducted to demonstrate that DeepRIN outperformed the best existing tools significantly.
- (c) A software tool based on DeepRIN is made available to the research community for download.

The rest of the paper is organized as follows. In Section 2, the problem formation of protein backbone torsion angles is presented. In Section 3, DeepRIN is presented in detail. In Section 4, experimental results of DeepRIN and its performance comparisons with the existing methods are presented. Finally, in Section 5, we summarize the paper and discuss some future work.

2 PROBLEM FORMULATION

For a given amino acid sequence of a protein, the goal is to predict the backbone torsion angles (Psi and Phi) of each residue. Psi is the $N(i)$, $Ca(i)$, $C(i)$, $N(i+1)$ torsion angle for residue i and Phi is the $C(i-1)$, $N(i)$, $Ca(i)$, $C(i)$ torsion angle for residue i (see Figure 1). The ranges of Psi-Phi angle are between -180 to 180 degrees. It is important to feed the deep neural networks with useful inputs in order to make accurate prediction. In the proposed method, the input is designed to include as much useful information that can be obtained or generated based on the amino acid sequence. Specifically, the input features consist of physicochemical properties of amino acids, PSI-BLAST profile, HHblits profile, and predicted secondary structures. Figure 2 shows an example of the physico-chemical feature vector. Each amino acid in the protein sequence is represented as a vector of 8 real numbers ranging from -1 to 1 . The vector consists of the seven physico-chemical properties as in [37] plus a number 0 or 1 representing the existence of an amino acid at this position as an input (called *NoSeq* label). The reason of adding the *NoSeq* label is because the proposed deep neural networks are designed to take a fixed size input, such as a sequence of length 700 residues in our experiment. Then, to run a protein sequence shorter than 700 through the network, the protein sequence will be padded at the end with 0 values and the *NoSeq* label is set to 1. For example, if the length of a protein is 500, then the first 500 rows are similar to the example in Figure 2 with *NoSeq* label set to 0 and the last 200 rows have 0 values in the first 7 columns and 1 in the last column. If the protein is longer than 700 residues, it can be split into multiple segments, each shorter than 700 residues. For the physico-chemical features, we applied the same seven phys_chemical terms from SPIDER3, such as molecular weight, hydrophobicity, etc. The data is available at the SPIDER3's website <http://sparks-lab.org/server/SPIDERS/>. The physico-chemical features describe hydrophobic, steric, and electric properties of amino acids and provide useful information for protein sequence analysis and prediction. Instead of representing the protein sequence as a one-hot sparse matrix, applying physico-chemical features represent the protein sequence as an informative dense matrix.

The second set of features comes from the protein profiles generated using PSI-BLAST [38]. In our experiments, the PSI-BLAST parameters were set to (*eval*: 0.001; *num_iterations*: 3; *db*: UniRef50) to generate PSSM, which was then transformed by the sigmoid function $S(x) = 1 / (1 + e^{-x})$ into the range (0, 1). Figure 3 shows an example of the PSI-BLAST profile. Each amino acid in the protein sequence is represented as a vector of 21 real values ranging from 0 to 1, representing the 20 amino acids PSSM values plus a *NoSeq* label in the last column. The features of the PSI-BLAST correspond to the 20 amino acids: “A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V”. The PSI-BLAST generates a PSSM or a position specific scoring matrix that represents evolutionary profiles derived from protein multiple sequence alignment.

The third set of features comes from the protein profiles generated using HHBlits [39]. In our experiments, the HHBlits software used database uniprot20_2013_03, which can be downloaded from http://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite_dbs/. Again, the profile values were transformed by the sigmoid function into the range (0, 1). Figure 4 shows an example of the HHBlits profile. Each amino acid in the protein sequence is represented as a vector of 31 real numbers, of which 30 from amino acids HMM profile values and 1 *NoSeq* label in the last column. The HHBlits, the names of the parameters are amino acids and some transition probabilities: “A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, M->M, M->I, M->D, I->M, I->I, D->M, D->D, Neff, Neff_I, and Neff_D”. HMM generates HMM profiles which is more sensitive than PSI-BLAST profile and provides useful evolutionary features for protein sequence.

The fourth set of features comes from the predicted eight-class secondary structures by DeepRIN itself. In our experiment, DeepRIN first takes the above-mentioned three sets of features as input to predict the eight-class secondary structure features. The prediction result is used as the fourth set of features for Psi-Phi angle prediction. The secondary structure prediction result is a vector of nine numbers, of which the first eight are the probabilities of the eight secondary structure classes and the last one is the *NoSeq* label.

The four sets of features are combined into one feature vector of length 66 for each amino acid as the input to our proposed deep neural networks. For the deep neural networks that take the fixed-size input, the input feature matrix would be of size 700 by 66.

The new deep neural network will not predict Psi-Phi angle values directly. Instead, the outputs for each amino acid position are $\sin(\phi)$, $\cos(\phi)$, $\sin(\varphi)$, and $\cos(\varphi)$, in order to remove the effect of angle's periodicity. The predicted angles can be calculated by using the *sine* and *cosine* predictions via equation $\alpha = \tan^{-1} [\sin \alpha / \cos \alpha]$.

3. NEW DEEP RESIDUAL INCEPTION NETWORKS (DEEPRIN) FOR TORSION ANGLE PREDICTION

In this section, a new deep inception residual network architecture is proposed for protein torsion angle prediction. The architecture consists of stringed inception modules and shortcut lines followed by some convolution and fully connected dense layers. Figure 5 shows the basic residual inception module of DeepRIN, which is designed based on InceptionNet [33] and ResNet [35]. The residual inception module consists of two stacked

inception modules to extract diverse features from the input. Convolutions with small windows are used to make the networks computationally efficient. The shortcut through Conv1 is designed to propagate feature maps into deeper layers of the network to address the vanishing-gradient problem. The DeepRIN block consists of inception modules, each of which is composed of stringed convolutional blocks. This design was intended to extract correlations among residues in a protein sequence effectively, together with small convolution units (Conv3). The two inception blocks use residual shortcut lines to propagate features into deep layers of the network.

Figure 6 shows two DeepRIN networks with two residual inception modules followed by a convolutional layer and two dense layers. The first one predicts 8-class secondary structure and the second one predicts torsion angles. The input of the first network is a 700×58 feature matrix of the given protein sequence of length 700, whereas the input of the second network is a 700×66 feature matrix of the given protein sequence. Different numbers of residual inception modules were tried and compared in our experiments to find optimal values for our prediction tasks. The first part of the prediction system takes the protein physio-chemical property features, HHblits features and PSI-BLAST features, and concatenates them together. Then it goes through two residual inception modules to predict secondary structures (SS). The second part of the system takes the predicted SS along with the three input features from the first part and merges them together. Then it goes through two residual inception modules to predict the sine and cosine values of Psi and Phi angles. The deep neural networks were implemented, trained, and experimented using TensorFlow [40] and Keras [41].

Each convolution layer, such as ‘Conv (3)’ in Figure 5, consists of four operations in the following sequential order: 1) A one-dimensional convolution operation using the kernel size of three; 2) the batch normalization technique [42] for speeding up the training process and acting as a regularizer; 3) the activation operation, ReLU [43], for non-linear transformation; and 4) the dropout operation [44] to prevent the neural network from overfitting by randomly dropping connections during the deep network training process so that the network can avoid coadapting.

In our experiments, a large number of network parameters and training parameters were tried. For the experimental results reported in the next section, the dropout rate was set at 0.4. During training, the learning rate scheduler from Keras was used to control the learning rate. The initial learning rate was 0.0005, and after every 40 epochs, the learning rate dropped as shown by the following formulas: $learning\ rate = 0.001 * pow(0.8, floor_epoch + 1)/40$. An early stopping mechanism from Keras was used to stop training when the monitored validation performance (such as validation loss and/or validation accuracy) stopped improving. The “patience” (number of epochs with no improvement after which the training was stopped) was set between 5 and 8. The activation function used for the last layer is *tanh*, as the predicted *sine* and *cosine* value should be within the range $[-1, 1]$.

The new DeepRIN networks are different from existing deep neural networks. The networks in [23] consisted of stacked RNN and iterative training, while in DeepRIN, a stacked inception module and residual shortcuts were used to capture both local and global

interactions between residues efficiently. Also, DeepRIN is different from DReRBM and DRNN in [21] because they used a sliding window to create input to deep neural networks, which is unable to capture long-range interactions between amino acids. In DeepRIN, the input is the entire protein sequence.

At a convolutional layer, the previous layer's inputs are convolved with learnable kernels and go through the activation function to generate the output. The convolution operation can be expressed in the following formulas:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right)$$

where M_j represents the input maps of node j , b is the bias, k is the kernel, and x represents the feature maps, f is the activation function, and l is the current layer. The meaning of this formulas is that given an input from previous layer x_i^{l-1} the CNN kernel k convolves the input x_i^{l-1} , adds the bias b_j^l and then applies the activation function f to obtain the output, which is also the input for the next layer l of node j .

The mean squared error (MSE) loss function used during the training is defined as following:

$$Loss(w) = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2$$

where y^n is the true label, and $\hat{y}^n$ is the predicted label.

4. EXPERIMENTAL RESULTS

This section presents experimental results of the proposed deep neural networks on multiple benchmarks in comparison with the best existing methods. The following four publically available datasets were used in our experiments:

- 1) **CullPDB dataset.** CullPDB [45] was downloaded on 15 June 2017 with 30% sequence identity cutoff, resolution cutoff 3.0Å and R-factor cutoff 0.25. The raw Cull-PDB dataset contains 10,423 protein sequences. All sequences with length shorter than 50 or longer than 700 were filtered out, which left the remaining 9972 protein sequences. CD-HIT [46] was then used to remove similar sequences between CullPDB and CASP10, CASP11 and CASP12 datasets. After that, there were 9589 proteins left. Among them, 8 protein sequences were ignored because they are too short to generate PSI-BLAST profiles. For the final 9581 proteins, 9000 were randomly selected as the training set and the rest 581 as the validation set.
- 2) **CASP datasets.** CASP10, CASP11 and CASP12 datasets were downloaded from the official CASP website <http://predictioncenter.org>. The Critical Assessment of

protein Structure Prediction, or CASP, is a biannual worldwide experiment for protein structure prediction since 1994. The CASP datasets have been widely used in the bioinformatics community. To get the secondary structure labels of the proteins, the DSSP program [47] was used. Some of the PDB files (including T0675, T0677 and T0754) could not generate the DSSP results, and hence they were discarded. Some protein sequences (including T0709, T0711, T0816 and T0820) are too short and PSI-BLAST could not generate profiles so that they were also removed. Finally, 98 out of 103 in CASP10, 83 out of 85 in CASP11 and 40 out of 40 in CASP12 were used as our CASP dataset.

- 3) **Recently released PDB proteins.** Because different tools used different training data, it is important to use PDB (Protein Data Bank) files that have not been seen by any previous predictors (including ours) to objectively compare their performance. For this purpose, recently published protein PDB files dated from 2017-7-1 to 2017-8-15 were downloaded. This set contains 614 proteins, each of which share less than 30% sequence identity. Then, 385 proteins with a length between 50 and 700 were kept. To perform a more objective test, each of the 385 proteins was searched against CullPDB using BLAST and was classified into two categories: 1) easy cases where e-value is less than or equal to 0.5; and 2) hard cases which have no hit or e-value higher than 0.5. After the classification, there are 270 easy cases and 115 hard cases.
- 4) **SPIDER3's training and test datasets.** These datasets are available from <http://sparks-lab.org/server/SPIDERS/>. The training data was used to train our model, and the test set performance was reported and compared with SPIDER3. In this experiment, only the sequences with length less than 700 residues were retained. In total 4532 out of 4590 training cases were used to train our DeepRIN model, and 1174 out of 1199 test cases were used to benchmark SPIDER3 and DeepRIN.

The hardware used to train the model is an Alienware Area-51 Desktop equipped with a NVidia Titan-X GPU (11GB graphic memory). The training time for the SS prediction model (Step 1 in Fig. 6) was around 2-2.5 hours per model. As a byproduct of the angle prediction model, the prediction accuracy of the Q8 eight-state SS was 0.7001(+/- 0.0044). The training time for the angle prediction model (Step 2 in Fig. 6) was around 2.6-3 hours per model. The time may vary when repeating our study because of the early stopping mechanism, size of training data, hardware, etc. The training speed can be much faster if using the recently released NVidia 1080-Ti graphic card.

In our experiment, the DeepRIN configuration shown in Figure 6 was used to generate the results reported in this paper. The DeepRIN networks consisted of two inception blocks and short-cut connected by residual network. We tried some different hyper-parameters, such as different numbers of inception blocks (ranging from 1 to 5), in our experiments. The configuration in Fig. 5 worked well in general and used in the final networks.

In most cases except the result in Table 4, the CullPDB dataset was used to train various deep neural networks, while the other datasets were only used for testing and performance comparison with different methods and with the results in the literature.

The Psi-Phi angle prediction was evaluated by the mean absolute error (MAE) between the truth angle (A_i^{true}) and the predicted angle (A_i^{pred}) for each residue i . To remove the periodicity in angles, the error in prediction of residue i is defined as the smaller value of d_i and $360d_i$ where $d_i = |A_i^{pred} - A_i^{true}|$. In addition, Pearson Correlation Coefficient was also used to evaluate the predicted Psi-Phi angles.

Table 1 compares the prediction results of DeepRIN and the best methods published in [21] using the same test dataset. In [21], Li *et al.* proposed DReRBM, which is a deep recurrent restricted Boltzmann machine [48] and DRNN, which is a deep recurrent neural network [49] to predict backbone torsion angle. SPIDER2 and SPIDER3 are two well-known tools representing the state-of-the-art. The testset contains 11 free modeling targets selected from CASP12 as described in [21]. The results show that DeepRIN outperformed all existing methods significantly, reducing the Psi MAE by over 2.6 degree over the second best, DReRBM, and the Phi MAE by nearly 1 degree over the second best, SPIDER3.

Table 2 compares the results of DeepRIN, SPIDER2 and SPIDER3 on three datasets, CASP10, CASP11 and CASP12. Again, DeepRIN outperformed SPIDER2 and SPIDER3 in all cases. The improvements are significant, reducing Psi angle prediction error by more than 5 degrees and Phi angle prediction error by more than 1.5 degrees. The improvement on the CASP12 dataset is slightly less than those on CASP10 and CASP11 datasets, which may be because targets in CASP12 are harder targets and PSI-BLAST or HHblits did not generate good profiles for them.

Figure 7 shows the average Psi-Phi angle prediction errors (MAEs) of DeepRIN for protein targets in the CASP12 dataset across eight types of secondary structures, which are 3_{10} -helix (G), α -helix (H), π -helix (I), β -strand (E), β -bridge (B), β -turn (T), bend (S) and loop or irregular (L). The result shows that for H (Helix), Psi-Phi angle prediction errors are the lowest, yet the errors for T (Turn), S (Bend) and C (Loop Region) are the highest. There is no “T” region in the targets.

Figure 8 shows the detailed Psi-Phi angle prediction results of two proteins in CASP12: T0860 and T0861. Predicted angle value, true angle value, and absolute error are plotted for each amino acid position, with its true secondary structure type labelled. T0860 is an example of hard cases with higher prediction errors, whereas T0861 is an example of accurate prediction. The H (helix) regions generally have lower prediction errors, whereas the C (coil) regions usually are harder to predict.

As shown in Figures 7 and 8, loop regions (T, S, C) generally have large prediction errors. The H (Helix) regions and strand (E) regions can be predicted accurately. Sometimes, Psi angles in beta sheets were predicted 180-degree reversely. For Phi angles, the predictions are relatively better in helix and beta sheet regions. T0861 was predicted quite well except for some turn and loop regions. An observation is that when the Psi-Phi angles were predicted

relatively well, the variance of the predicted Psi-Phi angles were relatively large, as it captured the diversity of different angles. When the Psi-Phi angles were not predicted accurately, the predicted angles had smaller variations.

Table 3 compares the results of DeepRIN, SPIDER2 and SPIDER3 on the recently released PDB dataset. On both the easy target subset and the hard target subset, DeepRIN outperformed both SPIDER2 and SPIDER3 significantly. These results provide an objective assessment because no predictors used these recently released PDB proteins in their training or development.

Table 4 compares the performance between DeepRIN and SPIDER3 using SPIDER3's training data. Only in this experiment, the DeepRIN was trained using SPIDER3's training data. Due to our network architecture, we only used the protein samples with sequence length less than 700 residues, which accounts for 4532 out of 4590 proteins in total, i.e., DeepRIN actually used less training data than SPIDER3. After training, both methods were run on six different test datasets, including SPIDER3's test sets (with sequence length less than 700, which accounts for 1174 out of 1199 test proteins in total). The results are shown in Table 4. Across all test datasets, DeepRIN outperformed SPIDER3.

5. SUMMARY

In this work, a new deep neural network architecture DeepRIN was proposed for protein Psi-Phi angle prediction. A software tool has been developed based on DeepRIN and is made available to the research community for download at <http://dlsrv8.cs.missouri.edu/~cf797/MUFoldAngle/>. Extensive experimental results show that DeepRIN consistently generated more accurate predictions than the best state-of-the-art methods. The experiments were designed carefully and the datasets used in training, e.g., the CullPDB dataset, were processed to remove any significantly similar sequences with the test sets using CD-HIT to avoid any bias. Compared to previous deep-learning methods for Psi-Phi angle prediction, this work developed a more sophisticated, yet efficient, deep-learning architecture. DeepRIN utilized hierarchical deep Inception blocks to effectively process local and non-local interactions between residues and the residual shortcuts help propagate high-level features into deep networks.

For this particular application, the improvement over other networks is substantial. A major reason is probably that compared to recurrent networks, such as recurrent RBM or LSTM-biRNN, stacked hierarchical layers of inception networks have a stronger capability to extract correlated features among long-range residues in a protein sequence. We have seen this property in our previously work of protein secondary structure prediction using deep inception inside inception [50] and neighbor-residual network for protein secondary structure prediction [51].

An interest phenomenon is that comparing between Table 2 and Table 4 shows that using a larger dataset (Cull-PDB instead of SPIDER3) does not significantly improve the performance of DeepRIN. This is also observed when we compare between Table 3 and Table 4. In fact, for PDB hard cases, DeepRIN's performance is better when trained using

SPIDER3 database than CullPDB. This may be because SPIDER3's training dataset contains some similar sequence to the hard cases that we labeled. The measurement of "hard cases" in our training dataset is against CullPDB, but some of these sequences may not be really "hard cases" for SPIDER3 as they may be similar to proteins in the SPIDER3's training dataset. We applied CD-Hit between SPIDER3's training dataset (total 4590 proteins) and CASPs 10, 11, 12 with 30% sequence similarity cutoff and we found there were 298, 298, and 231 proteins filtered out from 4590, respectively. Some proteins in SPIDER3's TR4590 training set had significant sequence similarities with proteins in CASPs 10, 11, 12. Another point is that DeepRIN converged well to achieve good performance using a sufficient dataset so that a larger dataset may not necessarily further improve the model. Hence, DeepRIN is a good architecture which can train effectively using a relatively small dataset.

In the future, DeepRIN and its variants will be applied to predict other protein structure related properties, such as solvent accessibility, contact number, and protein order / disorder region. These predicted features are also useful for protein tertiary structure prediction and protein model quality assessment.

Acknowledgment

This work was partially supported by the National Institutes of Health grant R01-GM100701. The high-performance computing infrastructure was partially supported by the National Science Foundation under grant number CNS-1429294. We like to take this opportunity to thank the anonymous reviewers for their constructive suggestions.

Biographies

Chao Fang received the B.S. and M.S. degrees in computer science from the University of Missouri - Columbia, in 2012, 2014, respectively. He is currently pursuing the PhD degree in computer science from the University of Missouri - Columbia. His research interest is deep learning and bioinformatics.

Yi Shang is a Professor and an Associate Chair and Director of Graduate Studies in the Department of Computer Science, University of Missouri, Columbia, Missouri. He received his Ph.D. degree in computer science from University of Illinois at Urbana-Champaign in

1997, M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 1991, and B.S. degree from the University of Science and Technology of China, Hefei, in 1988. He joined the Computer Science Department at MU in 1997 as an assistant professor. From 2001 to 2003, he took a leave from MU and worked at the renowned Xerox Palo Alto Research Center (Xerox PARC), the birth place of personal computer and Ethernet among many other revolutionary technological innovations, on embedded systems control and optimization and wireless sensor networks. Yi Shang has extensive research experience on nonlinear and constrained optimization, intelligent and distributed computing, mobile computing, and bioinformatics. He has published over 150 refereed journal and conference papers, and received 6 US patents. Many of his works are well received. For example, he pioneered the multidimensional scaling based approach for ad hoc network localization and one of his papers has been cited over 1100 times. He has served on program and/or organizing committees of over 40 international conferences. His research projects have been funded by federal and industrial grants and contracts from NSF, NIH, DARPA, Microsoft, Raytheon, and University of Missouri Research Board.

Dong Xu is a James C. Dowell Professor in the Electrical Engineering and Computer Science Department, Director of Information Technology Program, with appointments in the Christopher S. Bond Life Sciences Center and the Informatics Institute at the University of Missouri. After earning his doctorate in 1995, Xu completed two years of postdoctoral work at the U.S. National Cancer Institute and then was a staff scientist at Oak Ridge National Laboratory until joining the university in 2003. His research includes protein structure prediction, high-throughput biological data analyses and in silico studies of plants, microbes and cancers. He has published more than 300 papers and is a recipient of the 2001 R&D 100 Award, 2003 Federal Laboratory Consortium's Award of Excellence in Technology Transfer and 2010 Outstanding Achievement Award from International Society of Intelligent Biological Medicine. Xu is Associate Editor in Chief of IEEE/ACM Transactions on Computational Biology and Bioinformatics. He is a Fellow of American Association for the Advancement of Science (AAAS).

REFERENCES

- [1]. Gibson KD and Scheraga HA, "Minimization of polypeptide energy. I. Preliminary structures of bovine pancreatic ribonuclease S-peptide.," *Proceedings of the National Academy of Sciences*, vol. 58, no. 2, pp. 420–427, 1 1967.
- [2]. Dill KA and MacCallum JL, "The Protein-Folding Problem, 50 Years On," *Science*, vol. 338, no. 6110, pp. 1042–1046, 2012. [PubMed: 23180855]
- [3]. Zhou Y, Duan Y, Yang Y, Faraggi E, and Lei H, "Trends in template/fragment-free protein structure prediction," *Theoretical Chemistry Accounts*, vol. 128, no. 1, pp. 3–16, 1 2010. [PubMed: 21423322]

- [4]. Dor O and Zhou Y, "Real-SPINE: An integrated system of neural networks for real-value prediction of protein structural properties," *Proteins: Structure, Function, and Bioinformatics*, vol. 68, no. 1, pp. 76–81, 2007.
- [5]. Xue B, Dor O, Faraggi E, and Zhou Y, "Real-value prediction of backbone torsion angles," *Proteins: Structure, Function, and Bioinformatics*, vol. 72, no. 1, pp. 427–433, 2008.
- [6]. Faraggi E, Xue B, and Zhou Y, "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*, vol. 74, no. 4, pp. 847–856, 2009.
- [7]. Faraggi E, Yang Y, Zhang S, and Zhou Y, "Predicting Continuous Local Structure and the Effect of Its Substitution for Secondary Structure in Fragment-Free Protein Structure Prediction," *Structure*, vol. 17, no. 11, pp. 1515–1527, 2009. [PubMed: 19913486]
- [8]. Simons KT et al., "Improved recognition of native- like protein structures using a combination of sequence- dependent and sequence- independent features of proteins," *Proteins: Structure, Function, and Genetics*, vol. 34, no. 1, pp. 82–95, 1 1999.
- [9]. Huang Y-M and Bystroff C, "Improved pairwise alignment of proteins in the Twilight Zone using local structure predictions," 2005 IEEE Computational Systems Bioinformatics Conference - Workshops (CSBW05).
- [10]. Mooney C, Vullo A, and Pollastri G, "Protein Structural Motif Prediction in Multidimensional- ψ Space Leads to Improved Secondary Structure Prediction," *Journal of Computational Biology*, vol. 13, no. 8, pp. 1489–1502, 2006. [PubMed: 17061924]
- [11]. Wood MJ and Hirst JD, "Protein secondary structure prediction with dihedral angles," *Proteins: Structure, Function, and Bioinformatics*, vol. 59, no. 3, pp. 476–481, 2005.
- [12]. Yang Y, Faraggi E, Zhao H, and Zhou Y, "Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates," *Bioinformatics*, vol. 27, no. 15, pp. 2076–2082, 11 2011. [PubMed: 21666270]
- [13]. Karchin R, Cline M, Mandel-Gutfreund Y, and Karplus K, "Hidden Markov models that use predicted local structure for fold recognition: Alphabets of backbone geometry," *Proteins: Structure, Function, and Bioinformatics*, vol. 51, no. 4, pp. 504–514, 2003.
- [14]. Zhang W, Liu S, and Zhou Y, "SP5: Improving Protein Fold Recognition by Using Torsion Angle Profiles and Profile-Based Gap Penalty Model," *PLoS ONE*, vol. 3, no. 6, 4 2008.
- [15]. Najibi S, Maadooliat M, Zhou L, Huang J and, Gao X, "Protein structure classification and loop modeling using multiple Ramachandran distributions." *Computational and structural biotechnology journal* vol. 15: pp. 243–254, 2017. [PubMed: 28280526]
- [16]. Maadooliat M, Zhou L, Najibi S, Gao X, and Huang J, "Collective estimation of multiple bivariate density functions with application to angular-sampling-based protein loop modeling." *Journal of the American Statistical Association*, vol. 111, no. 513, pp. 43–56, 2016.
- [17]. Maadooliat M, Gao X, and Huang J, "Assessing protein conformational sampling methods based on bivariate lag-distributions of backbone angles." *Briefings in bioinformatics*, no. 14, no. 6, pp. 724–736, 2012. [PubMed: 22926831]
- [18]. Kuang R, Leslie C, and Yang A, "Protein backbone angle prediction with machine learning approaches," *Bioinformatics*, vol. 20, no. 10, pp. 1612–1621, 2004. [PubMed: 14988121]
- [19]. Kang HS, Kurochkina NA, and Lee B, "Estimation and Use of Protein Backbone Angle Probabilities," *Journal of Molecular Biology*, vol. 229, no. 2, pp. 448–460, 1993. [PubMed: 8429556]
- [20]. Lyons J et al., "Predicting backbone $C\alpha$ angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network," *Journal of Computational Chemistry*, vol. 35, no. 28, pp. 2040–2046, 12 2014. [PubMed: 25212657]
- [21]. Li H, Hou J, Adhikari B, Lyu Q, and Cheng J, "Deep learning methods for protein torsion angle prediction," *BMC Bioinformatics*, vol. 18, no. 1, 2017.
- [22]. Yang Y et al., "SPIDER2: A Package to Predict Secondary Structure, Accessible Surface Area, and Main-Chain Torsional Angles by Deep Neural Networks," *Methods in Molecular Biology Prediction of Protein Secondary Structure*, pp. 55–63, 2016.

- [23]. Heffernan R, Yang Y, Paliwal K, and 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*, vol. 33, no. 18, pp. 2842–2849, 2017. [PubMed: 28430949]
- [24]. Hochreiter S and Schmidhuber J, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [PubMed: 9377276]
- [25]. Mikolov T and Zweig G, “Context dependent recurrent neural network language model,” 2012 IEEE Spoken Language Technology Workshop (SLT), 2012.
- [26]. Wu S and Zhang Y, “ANGLOR: A Composite Machine-Learning Algorithm for Protein Backbone Torsion Angle Prediction,” *PLoS ONE*, vol. 3, no. 10, 2008.
- [27]. Krizhevsky A, Sutskever I, and Hinton G, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [28]. Pfister T, Simonyan K, Charles J, and Zisserman A, “Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos,” *Computer Vision -- ACCV 2014 Lecture Notes in Computer Science*, pp. 538–552, 2015.
- [29]. Lawrence S, Giles CL, Tsoi AC and Back AD, “Face recognition: a convolutional neural-network approach,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997. [PubMed: 18255614]
- [30]. Collobert R and Weston J, “A unified architecture for natural language processing,” *Proceedings of the 25th international conference on Machine learning - ICML 08*, 2008.
- [31]. Vinyals O, Toshev A, Bengio S and Erhan D, “Show and tell: A neural image caption generator,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
- [32]. Pfister T, Simonyan K, Charles J, and Zisserman A, “Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos,” *Computer Vision -- ACCV 2014 Lecture Notes in Computer Science*, pp. 538–552, 2015.
- [33]. Szegedy C, Ioffe S, Vanhoucke V, and Alemi AA, “Inceptionv4, Inception-ResNet and the Impact of Residual Connections on Learning,” *AAAI* 2017.
- [34]. Chollet F “Xception: Deep Learning with Depthwise Separable Convolutions.” *arXiv preprint arXiv:1610.02357* 2016.
- [35]. He K, Zhang X, Ren S, and Sun J, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [36]. Huang G, Liu Z, Weinberger KQ, and van der Maaten L, “Densely connected convolutional networks.” *arXiv preprint arXiv:1608.06993*. 2016.
- [37]. Fauchère J-L, Charton M, Kier LB, Verloop A, and Pliska V, “Amino acid side chain parameters for correlation studies in biology and pharmacology,” *International Journal of Peptide and Protein Research*, vol. 32, no. 4, pp. 269–278, 12 2009.
- [38]. Altschul S et al., “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1 1997. [PubMed: 9254694]
- [39]. Remmert M, Biegert A, Hauser A, and Soding J, “HHblits: lightning-fast iterative protein sequence searching by HMMHMM alignment,” *Nature Methods*, vol. 9, no. 2, pp. 173–175, 2011. [PubMed: 22198341]
- [40]. Abadi M et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems.” *arXiv preprint arXiv:160304467*, 2016.
- [41]. Chollet F, Keras. A, 2015.
- [42]. Ioffe S and Szegedy C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” *International Conference on Machine Learning*, 2015.
- [43]. Radford A, Metz L, and Chintala S, “Unsupervised representation learning with deep convolutional generative adversarial networks.” *arXiv preprint arXiv:151106434*, 2015.
- [44]. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, and Salakhutdinov R, “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of machine learning research*, vol 15, no.1, pp. 1929–1958. 2014.
- [45]. Wang G and Dunbrack RL Jr. “PISCES: a protein sequence culling server.” *Bioinformatics Vol* 19, no. 12, pp. 1589–1591. 2003. [PubMed: 12912846]

- [46]. Li W and Godzik A. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences." *Bioinformatics* vol 22, no.13 pp. 1658–1659. 2006. [PubMed: 16731699]
- [47]. Kabsch W and Sander C. "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features." *Biopolymers* Vol 22 No. 12, pp 2577–2637, 1983. [PubMed: 6667333]
- [48]. Nair V and Hinton GR, "Rectified linear units improve restricted boltzmann machines". In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
- [49]. Mikolov T, Karafiat M, Burget L, Cernocky J, and Khudanpur S, September. Recurrent neural network based language model. *Interspeech* Vol. 2, no. 3, 2010.
- [50]. Fang C, Shang Y, and Xu D, "MUFold-SS: Protein Secondary Structure Prediction Using Deep Inception-Inside-Inception Networks." *arXiv preprint arXiv:170906165*, 2017.
- [51]. Fang C, Shang Y, and Xu D, "A New Deep Neighbor-Residual Neural Network for Protein Secondary Structure Prediction." *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on. IEEE*, 2017.

Fig. 1.
An illustration of what Psi Phi angles are.

S	-0.337	-0.637	-0.544	-0.364	-0.265	-0.466	-0.212	0
R	0.105	0.373	0.466	-0.9	0.9	0.528	-0.371	0
M	0.11	0.066	0.087	0.337	-0.262	0.652	-0.001	0
P	0.247	-0.9	-0.294	0.055	-0.01	-0.9	0.106	0
S	-0.337	-0.637	-0.544	-0.364	-0.265	-0.466	-0.212	0
			...					
			...					

Fig. 2.
An example of the physico-chemical feature vector of a protein sequence. Each amino acid in the protein sequence is represented as a vector of 8 real numbers ranging from -1 to 1 . The feature vectors of the first 5 amino acids are shown.

S	0.731	0.268	...	0.119	0
R	0.268	0.993	...	0.119	0
M	0.268	0.119	...	0.731	0
P	0.268	0.119	...	0.047	0
S	-0.731	-0.268	...	-0.119	0
			...		
			...		

Fig. 3.
An example of the PSI-BLAST profile of a protein sequence. Each amino acid in the protein sequence is represented as a vector of 21 real numbers ranging from 0 to 1. The feature vectors of the first 5 amino acids are shown. The third through 19th columns are omitted.

S	0.016	0.018	...	1	0
R	0.004	0.011	...	1	0
M	0	0.006	...	0.5	0
P	0.130	0.003	...	0.483	0
S	0.081	0.022	...	1	0
			...		
			...		

Fig. 4.
An example of the HHBlits profile of a protein sequence. Each amino acid in the protein sequence is represented by a vector of 31 real values ranging from 0 to 1. The feature vectors of the first five amino acids are shown.

Fig. 5.

A residual-inception module of DeepRIN. The green rhombus “Conv(1)” denotes a convolution operation using window size 1 and the number of filters is 100. The orange square “Conv(3)” denotes a convolution operation using window size 3 and the number of filters is 100. The blue oval “Concatenate” denotes feature concatenation.

Step 1: Eight-state secondary structure prediction

Step 2: Psi-Phi angle prediction

Fig. 6.

Architecture of the proposed deep residual inception network architecture (DeepRIN).

CASP 12 Average Psi MAE

CASP12 Average Phi MAE

Fig. 7.

Average Psi-Phi angles prediction errors (MAEs) for each of the eight types of secondary structure on the CASP12 dataset by DeepRIN.

Fig. 8.

Psi-Phi angle prediction results of two proteins in CASP12, T0860 (top) and T0861 (bottom). Predicted angle value, true angle value, and absolute error are plotted for each amino acid position, which is labelled by its true secondary structure type. The low blue bars, which denote the error between predicted angles and truth angles.

Table 1.

Comparison of mean absolute errors (MAE) of Psi-Phi angle prediction between DeepRIN and best existing predictors using the test sets from [21], (DeepRIN trained using CullPDB)

	Psi MAE	Phi MAE
DReRBM [18] *	35.99	22.47
DRNN [18] *	37.54	22.38
SPIDER2 *	37.67	22.61
SPIDER3	39.06	21.24
DeepRIN	33.34	20.30

* Published results in [21]

Table 2.

Comparison of mean absolute errors (MAE) and Pearson Correlation Coefficient (PCC) of Psi-Phi angle prediction between DeepRIN, SPIDER2 and SPIDER3 using the CASP10, CASP11, and CASP12 datasets. The PCC is written in parentheses. Bold-font numbers in each column represent the best result on a particular dataset. (DeepRIN trained using CullPDB)

	Angle	CASP10	CASP11	CASP12
SPIDER2	Psi	49.18(0.693)	48.13 (0.699)	50.59 (0.688)
	Phi	25.83 (0.766)	26.06 (0.756)	26.85 (0.758)
SPIDER3	Psi	31.17(0.824)	33.05 (0.805)	35.67 (0.783)
	Phi	20.41 (0.831)	21.38(0.815)	21.12(0.809)
DeepRIN	Psi	25.79 (0.861)	27.96 (0.841)	31.39 (0.813)
	Phi	17.39 (0.874)	18.97 (0.853)	20.21 (0.838)

Table 3.

Comparison of mean absolute errors (MAE) and Pearson Correlation Coefficient (PCC) of Psi-Phi angle prediction between DeepRIN and SPIDER2 and SPIDER3 using the recently released PDB dataset The PCC is shown in parentheses. (DeepRIN trained using CullPDB)

	Angle	Recent PDB easy cases (226)	Recent PDB hard cases (94)
SPIDER2	Psi	28.13 (0.849)	33.42 (0.808)
	Phi	18.46 (0.850)	20.24 (0.830)
SPIDER3	Psi	24.93 (0.867)	29.53 (0.827)
	Phi	17.49 (0.861)	19.35 (0.838)
DeepRIN	Psi	22.67 (0.882)	28.94 (0.834)

Table 4.

Comparison of mean absolute errors (MAE) of Psi-Phi angle prediction between DeepRIN and SPIDER3 using the same training dataset (SPIDER's training set) and the same test dataset. (DeepRIN trained using SPIDER3's training data)

	Angle	CASP10 (98 out of 103)	CASP11 (83 out of 85)	CASP12 (40 out of 40)	SPIDER3's test set (1174 out of 1199)	Recent PDB easy cases (226)	Recent PDB hard cases (94)
SPIDER3	Psi	31.17	33.05	35.67	27.33	24.93	29.53
	Phi	20.41	21.38	21.12	18.73	17.49	19.35
DeepRIN	Psi	25.64	28.27	30.93	22.74	23.25	28.78
	Phi	17.22	18.99	19.72	16.40	16.04	18.01

* T0675, T0677 and T0754 could not generate DSSP results. T0709, T0711, T0816 and T0820 are too short and PSI-BLAST did not have a hit. The SPIDER3's training set (sequences with length less than 700) 4532 out of 4590 proteins in total were used to train our model and the SPIDER3's test set (sequences with length less than 700) 1174 out of 1199 proteins in total were used for testing on both tools.