High Quality Protein Q8 Secondary Structure Prediction by Diverse Neural Network Architectures

Iddo Drori
Columbia University
idrori@cs.columbia.edu

Isht Dwivedi
Columbia University
isht.dwivedi@columbia.edu

Pranav Shrestha
Columbia University
ps2958@columbia.edu

Jeffrey Wan
Columbia University
jw3468@columbia.edu

Yueqi Wang
Columbia University
yw3169@columbia.edu

Yunchu He
Columbia University
yh3050@columbia.edu

Anthony Mazza
Columbia University
am4564@columbia.edu

Hugh Krogh-Freeman
Columbia University
hk2903@columbia.edu

Dimitri Leggas
Columbia University
ddl2133@columbia.edu

Kendal Sandridge
Columbia University
ks3311@columbia.edu

Linyong Nan
Columbia University
ln2401@columbia.edu

Kaveri Thakoor
Columbia University
kat2193@columbia.edu

Chinmay Joshi
Columbia University
caj2163@columbia.edu

Sonam Goenka
Columbia University
sg3625@columbia.edu

Chen Keasar
Ben Gurion University
keasar@cs.bgu.ac.il

Itsik Pe'er
Columbia University
itsik@cs.columbia.edu

Abstract

We tackle the problem of protein secondary structure prediction using a common task framework. This lead to the introduction of multiple ideas for neural architectures based on state of the art building blocks, used in this task for the first time. We take a principled machine learning approach, which provides genuine, unbiased performance measures, correcting longstanding errors in the application domain. We focus on the Q8 resolution of secondary structure, an active area for continuously improving methods. We use an ensemble of strong predictors to achieve accuracy of 70.5% (on the CB513 test set using the CB6133filtered training set). These results are statistically indistinguishable from those of the top existing predictors. In the spirit of reproducible research we make our data, models and code available [10][1], aiming to set a gold standard for purity of training and testing sets. Such good practices lower entry barriers to this domain and facilitate reproducible, extendable research.

[1] Codebase: https://github.com/idrori/cu-ssp

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

1 Introduction

Proteins are the major building blocks of life on earth, and the mediators of almost all chemical and biophysical events in living organisms. They are polymer chains of amino acid residues, whose sequences (aka primary structure) dictate stable spatial conformations, known as the native structures. These structures in turn enable the biological functions of proteins. The sequence space of proteins is vast, 20 possible residues per position, and evolution has been sampling it over billions of years. Thus, current proteins are highly diverse in sequences, structures and functions. Predicting the 3D structure of a protein (PSP) from its linear sequence of amino acid units is a fundamental problem in computational biology, which is open for 50 years already. Virtually all the diverse approaches to PSP use, as their stepping stone, a prediction of the protein's secondary structure, the focus of the current study.

Underneath the high diversity of protein structures, lies a relatively small set of recurrent patterns of torsion angles and hydrogen bonds that allow the protein to accommodate both local (*i.e.,* close chain positions) and non-local constraints. These patterns, which are known as secondary structure elements, imply the classification of the protein's residues to a relatively small number of structural classes known as the secondary structure. Since the mid-80s the dictionary of secondary structure patterns (DSSP) that suggested eight such classes has become the gold standard of the field [19]. Figure 1 show a protein residues as spheres colored by their Q3 and Q8 structures. As secondary structure elements are stabilized by both local and non-local interactions, the tendency of protein segments to adopt them is sequence dependent. Beta-strand, for example, is a common pattern that implies a stretch of residues of the "E" (extended) class. It is characterized by alternating hydrophobic (oil-like) and hydrophilic (water-loving) residues. Such correspondence between two alphabets calls for the development of prediction methods, and indeed as early as the mid 70s secondary structure prediction (SSP) has gained much interest and was tackled by a wide variety of statistical approaches [5, 12, 8, 15, 13, 20]. To ease the prediction challenge, these studies typically merged the eight DSSP classes to only three. In the early 90s Rost and Sander [26, 27], augmented protein sequences by profiles, derived from multiple sequence alignment of homologous proteins. They also introduced multi-tier neural networks and with these advances reached a landmark success of over 70% accuracy in the three-state prediction scheme (Q3), dramatically outperforming previous approaches. Their success paved the way to further studies that provided more elaborate implementations of these concepts [18, 37, 35, 24, 9, 23, 14, 32, 3], increasing the success rate of Q3 up to 84%. However, as performance approached the postulated theoretical limit (85%-88%) [36], interest in the problem declined and progress became negligible over almost a decade. Recently however, interest has rekindled, as scholars replaced the relatively modest goal of predicting three classes by the more ambitious prediction of eight classes (Q8) [34]. In the past five years there has been a steady and slow improvement in Q8 secondary structure prediction accuracy using deep neural networks [40, 28, 33, 21, 2, 17, 3, 11]. This work reports the integration of multiple ideas for improving Q8 secondary structure prediction using an ensemble of predictors to achieve state of the art accuracy on the CB513 [7] test set using a small training set of with less than 20% identity of sub-sequences.

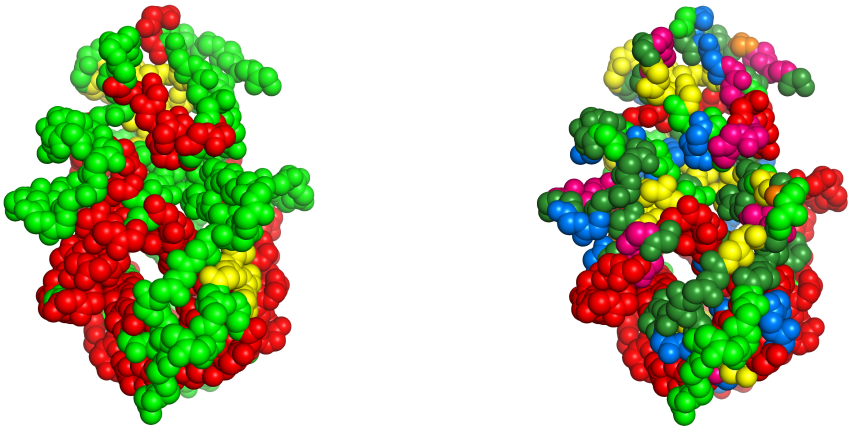

Figure 1: Q3 (left); Q8 (right) secondary structure spheres for protein 1AKD in CB513 dataset.

In 2014 Zhou and Troyanskaya published their GSN method [40] for SSP. To evaluate the performance of their method, they created a new benchmark termed CB6133 [39]. They used homologs-filtered subset of CB6133 to reach a then-record Q8 accuracy of 66.4% for CB513 [7] data-set. They also split CB6133 to training, validation and test sets and reported 72.1% Q8 accuracy. They made their benchmark publicly available in accessible numpy format, stirring a wave of studies and publications, including the current one. In comparison, we have reached state of the art accuracy for CB513 of 70.5% and best known published accuracy of 76.3% for CB6133. We report data issues with the CB6133 standard that we ran into. These were since quickly fixed by the responsive authors on 10.28.18. We report results of a common task challenge, predicting Q8 secondary structures using novel network architectures. We report evaluation of these models and their ensemble on the CB513 data-set, reaching accuracy equivalent to current top predictors. In the spirit of reproducible research we make our data, models, and code fully available.

2 Data and training

2.1 CB6133 dataset: correcting a long standing error

We began this work by using the CB6133 dataset [40] with the same train, validation, and test splits as used by other work for comparison [39]. We achieved the best known published performance on this dataset using the same published splits [39] as shown in Table 7. However, unfortunately, while using CB6133 dataset we couldn't but notice that it includes duplicate entries and the training, validation and test sets were not strictly disjoint. As a result of our finding the dataset splits were corrected by their creators and the valid splits re-published online by the 2014 authors on 10.28.18 [39]. Our contribution clears a long standing error in the field.

2.2 Training data used for testing CB513: setting up standards

We use the CB513 dataset [7] for testing which is valid, does not contain any duplicates, and is disjoint from the training set we use CB6133filtered (after removing duplicates). Recent work [38] performs a comprehensive performance comparison on this test set, however uses different training sets of different sizes as if they were the same, and therefore we do not report those results here. To standardize our comparison and minimize redundancy we considered the smaller training dataset, CB6133filtered, which multiple methods have in common. We achieve state of the art results as shown in Table 3.

2.3 Features and output classes

Following Zhou and Troyanskaya [40] we use 46 features per residue *i.e., sequence positions* to classify each residue to one of nine classes. A subset of 22 features represent residue type by one-hot encoding. In addition to the standard 20 residue types: A, C, E, D, G, F, I, H, K, M, L, N, Q, P, S, R, T, W, V, and Y, we use X for non-standard residues (*e.g., selenium methionine*) and noSeq for padding. A second subset of 22 features represent residue's position in a position specific substitution matrix (PSSM aka profile) that was generated by PSI-BLAST [1]. Again the last two features represent non-standard residues and padding. Finally, two binary features indicate the first and last position of the sequence. All sequences are padded with one-hot encoding of noSeq to length 700. The output classes include the eight classes defined by DSSP [19]: L, B, E, G, I, H, S, and T.

3 Methods

The neural network architectures of our 6 models are diverse. This section provides a detailed description and an illustration of each architecture. The training time for each of the models is around one hour using an Nvidia 1080 GPU.

3.1 Bidirectional LSTMs with attention

Figure 2 shows the architecture for this model. An embedding of the bigram amino acid sequence input is concatenated with the profile features and passed to a bidirectional LSTM (with 75 units), followed by 4 unidirectional LSTMs (each with 150 units). The initial state of the latter LSTM is

initialized by the last hidden state of the former LSTM (concatenated in the case of biLSTM). For each possible pair of LSTMs, an attention mechanism [22] is applied using output of the latter LSTM as queries and output of the former LSTM as keys and values. This process generates 10 attention outputs, which are then added and passed to two fully-connected layers. This is the first time an attention mechanism [22, 31] is used for this problem achieving state of the art results without using convolutions.

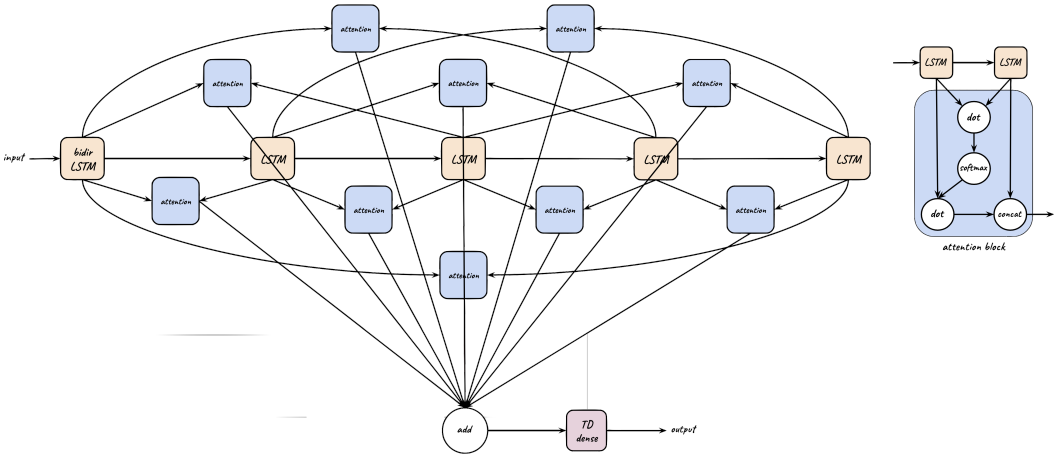

Figure 2: Bidirectional LSTMs with attention.

3.2 U-Net with convolution blocks

Figure 3 shows the architecture for this model. A fully convolutional model, using a one-dimensional U-Net [25] with dropout [29] and batch normalization [16]. The profile input matrix is concatenated with the output of the embedding layer and fed into the first layer of a 1D U-Net.

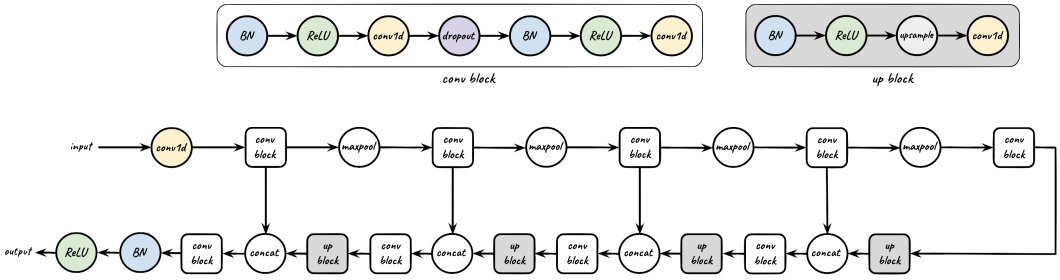

Figure 3: U-Net with convolution blocks.

3.3 Bidirectional GRU with convolutional blocks

Figure 4 shows the architecture for this model. The concatenation of one-hot encoded residue, residue embedding, and residue profiles are passed into multi-scale CNN layers with different kernel sizes (3, 5, 7) to obtain multiple local contextual feature maps [21]. This is followed by a series of cascading convolutional layers. A series of 3 concatenated 1D convolutions are applied. Each of the convolutions are followed by several layers [4]: time distributed ReLU activation, batch normalization and dropout layers (with probability 0.5). This passes through a single (256 unit) bidirectional (CuDNN) GRU [6] with a l_2 recurrent regularizer. The output is generated by two fully connected ReLU activated layers (of size 128 and 64) followed by a soft-max output layer.

3.4 Temporal convolutional network

Figure 5 shows the architecture for this model. Two embedding layers fed with bigrams of the original data are concatenated with profile features. One concatenated output is fed into a dense layer followed by dropout. Another concatenated output is fed into two bidirectional (CuDNN) GRUs. These two, separate layers (the dense and the 2nd bidirectional GRU) are concatenated. The

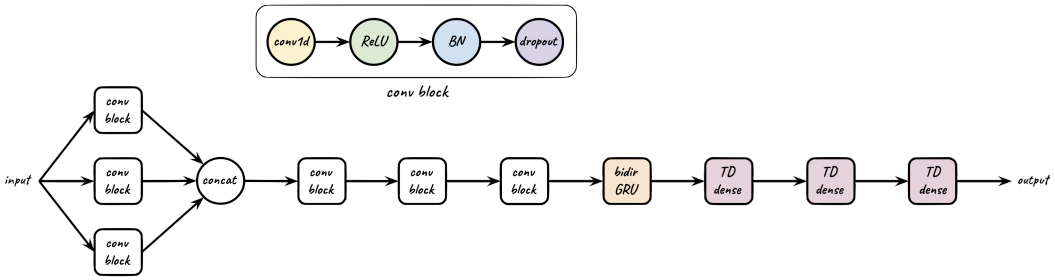

Figure 4: Bidirectional GRU with convolutional blocks.

concatenated output is fed into a dense layer, followed by dropout, a temporal convolutional network [30], and a time-distributed dense layer with softmax activation.

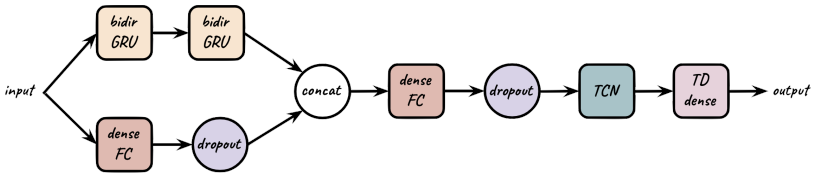

Figure 5: Temporal convolutional network (TCN).

3.5 Bidirectional GRU with 2D convolution

Figure 6 shows the architecture for this model. The model concatenates the following features as input: a linear combination of the onehot vectors of the preceding amino acids, a linear combination the onehot vectors of the following amino acids, the onehot vector corresponding to the current amino acid and the the profile features for the current amino acid. A fully-connected layer (128 units) removes sparsity from the features, and its outputs are fed into three convolutional layers (3, 7, 11) with 64 filters each. After batch normalization of the outputs, they are concatenated and passed through 3 stacked bidirectional GRUs (with 32 units each). The concatenation of the GRUs' outputs with the convolutional layers' outputs is passed through a two-layer fully connected network.

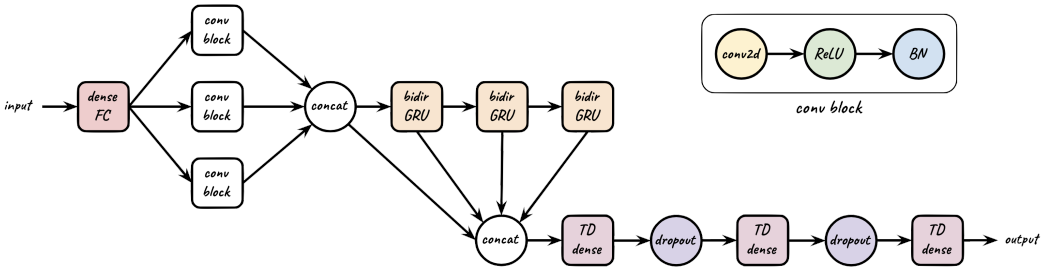

Figure 6: Bidirectional GRUs.

3.6 Convolutions and bidirectional LSTM

Figure 7 shows the architecture for this model. The model uses skip connections, feeding the encoded input, to two independent convolution layers of 64 channels each (with 11 and 7 kernel sizes respectively). Further, we concatenate both with the input. Now, we again use two independent convolution layers each of 64 channels (with 5 and 3 kernel size respectively). Again, we concatenate the input from the previous concatenation and the output of the two convolution layers. Next, this concatenation is fed to a bidirectional LSTM layer that produces a 128 unit output which is finally used to generate the output using a TD dense layer.

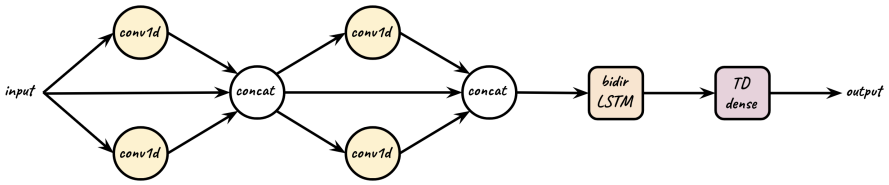

Figure 7: Convolutions and bidirectional LSTM.

Table 1: Hyperparameters of each of our models.

Model	Optimizer	Learning Rate	Decay	Epochs	Batch
U-net with convolution blocks	RMSprop	0.002	0.5	80	128
Bidirectional GRU with conv. blocks	Nadam	0.002	0.004	75	128
Temporal convolutional network	Adam	0.001	0.0001	5	16
Bidirectional GRUs	Nadam	0.002	0.004	10	64
Bidirectional LSTMs with attention	RMSprop	0.003	0.5	20	64
Convolution and bidirectional LSTM	RMSprop	0.001	0	30	128

3.7 Model hyperparameters

Table 1 summarizes the hyperparameters used for each of our models.

4 Results

4.1 Unbiased accuracy evaluation using CB513

Table 2 compares mean accuracy between each of our 6 models and their ensemble model. The ensemble is computed by taking the argmax over the average of probabilities over all the models for each Q8 structure class, $y = \arg\max_j \frac{1}{m}(\sum_{i=1}^{m} p_i^{(j)})$, for $m = 6$ models and $j = 1, \ldots, 8$ classes. Table 3 compares accuracy with other work on the best single model and ensemble for the CB513 dataset. Table 4 shows the confusion matrix for each of the Q8 structures for the CB513 dataset. Table 5 shows the precision, recall, and f-score for each of the Q8 structures for the CB513 dataset.

4.2 Accuracy evaluation on CB6133

For completeness of comparison we provide results on the CB6133 dataset. Table 6 compares mean accuracy between each of our 6 models and their ensemble model. Table 7 compares accuracy with other work on the best single model and ensemble. Table 8 shows the confusion matrix for each of the Q8 structures for the CB6133 dataset. Table 9 shows the precision, recall, and f-score for each of the Q8 structures for the CB6133 dataset.

5 Conclusions and future work

We present new diverse architectures for protein structure prediction, some of which have not been used in the field before, and perform with state of the art accuracy. In future work, these architectures

Table 2: Q8 mean accuracy of our models and their ensemble on the CB513 dataset.

Ensemble	70.5
U-Net with convolution blocks	68.9
Bidirectional GRU with convolution blocks	68.9
Temporal convolutional network	68.7
Bidirectional LSTMs with attention	68.4
Convolutions and bidirectional LSTM	67.8
Bidirectional GRUs	67.5

Table 3: Q8 mean accuracy using the best single model and ensemble for different methods on CB513 dataset. One apparently relevant study, CRRNN [38], which also reports results on CB513, is excluded from the table as its training set is twice as large as the one used by the other methods.

Model	Best Single	Ensemble
MUFOLD-SS [11]	70.5	70.6
NCCNN [3]	70.3	71.4
biRNN-CRF [17]	69.4	70.9
DeepMSCNN [2]	70.0	70.6
DCRNN [21]	69.4	69.7
BLSTM [28]	67.4	N/A
GSN [40]	66.4	N/A
DeepCNF [33]	N/A	68.3
Ours	68.9	70.5

Table 4: Confusion matrix for each of the Q8 structures for the CB513 dataset. The rows represent the predicted output and the columns represent the ground truth labels.

	L	B	E	G	I	H	S	T
L	11,828	618	1,880	629	4	738	3,192	1,619
B	7	31	6	0	0	3	4	0
E	3,167	316	15,419	234	2	334	997	565
G	134	8	24	851	0	233	109	328
I	0	0	0	0	0	0	0	0
H	762	77	216	777	22	24,126	554	1,585
S	871	49	201	78	0	77	2,039	502
T	1,151	82	270	563	2	646	1,421	5,414

Table 5: Precision, recall, and f-scores for each of the Q8 structures for the CB513 dataset.

	Precision	Recall	F-score
L	0.58	0.66	0.62
B	0.61	0.03	0.05
E	0.73	0.86	0.79
G	0.50	0.27	0.35
I	0.0	0.0	0.0
H	0.86	0.92	0.89
S	0.53	0.25	0.34
T	0.57	0.54	0.55

Table 6: Q8 mean accuracy of our models and their ensemble on the CB6133 dataset.

Ensemble	76.3
U-Net with convolution blocks	75.4
Temporal convolutional network	75.4
Bidirectional GRU with convolution blocks	74.8
Bidirectional GRUs	72.9
Convolutions and bidirectional LSTM	71.6
Bidirectional LSTMs with attention	68.3

Table 7: Q8 mean accuracy using best single model and ensemble for different methods on the CB6133 dataset. Both our best single model 75.4% and ensemble 76.3% perform best compared with previously known published methods.

	Best single	Ensemble
GSN [40]	72.1	N/A
DCRNN [21]	N/A	73.2
biRNN-CRF [17]	73.4	74.8
CRRNN [38]	N/A	74
Ours	75.4	76.3

Table 8: Confusion matrix for each of the Q8 structures for the CB6133 dataset. The rows represent the predicted output and the columns represent the ground truth labels.

	L	B	E	G	I	H	S	T
L	7,218	322	1,220	373	0	389	1,855	894
B	3	46	17	1	0	1	1	0
E	1,445	142	10,344	106	0	152	395	233
G	146	5	28	754	0	164	77	209
I	0	0	0	0	0	0	0	0
H	591	34	251	661	0	19,085	337	1,062
S	406	22	104	37	0	36	1,010	165
T	719	55	255	370	0	394	815	3,737

Table 9: Precision, recall and f-scores for each of the Q8 structures for the CB6133 dataset.

	Precision	Recall	F-score
L	0.58	0.68	0.63
B	0.66	0.07	0.13
E	0.80	0.85	0.83
G	0.54	0.33	0.41
I	0.0	0.0	0.0
H	0.87	0.94	0.90
S	0.59	0.23	0.32
T	0.58	0.59	0.59

may be used as a starting points for meta learning improved architectures, in a neural architecture search. Finally, in the spirit of reproducible research we make our data, models, and code publicly available [10].

Acknowledgments

We would like to thank the 100 CS/DSI/Stats graduate students at Columbia University of the Fall 2018 Deep Learning course for their participation in an in class protein secondary structure prediction competition. The models which achieved top performance in the competition were invited to participate in this follow-up work, which lead to the discovery of new architectures with state of the art performance. We would like to thank Tomer Sidi of BGU for thorough examination of the correct measures used for performance comparison. We would like to thank Jian Zhou and Olga Troyanskaya of Princeton for making their CB6133 dataset available and for updating their CB6133 dataset splits following our work. Chen Keasar is partially supported by grants 1122/14 from the Israel Science Foundation (ISF).

References

[1] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[2] Akosua Busia, Jasmine Collins, and Navdeep Jaitly. Protein secondary structure prediction using deep multi-scale convolutional neural networks and next-step conditioning. *arXiv preprint arXiv:1611.01503*, 2016.

[3] Akosua Busia and Navdeep Jaitly. Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. *arXiv preprint arXiv:1702.03865*, 2017.

[4] Francois Chollet. Deep learning with python. *Manning Publications*, 2017.

[5] Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS*, 2014.

[7] James A. Cuff and Geoffrey J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519.

[8] G Deleage and B Roux. An algorithm for protein secondary structure prediction based on class prediction. *Protein Engineering, Design and Selection*, 1(4):289–294, 1987.

[9] Ofer Dor and Yaoqi Zhou. Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins: Structure, Function, and Bioinformatics*, 66(4):838–845, 2007.

[10] Iddo Drori, Isht Dwivedi, Pranav Shrestha, Jeffrey Wan, Yueqi Wang, Yunchu He, Anthony Mazza, Hugh Krogh-Freeman, Dimitri Leggas, Kendal Sandridge, Chinmay Joshi, Sonam Goenka, Linyong Nan, Kaveri Thakoor, Itsik Pe'er, and Chen Keasar. Github repository for high quality protein Q8 secondary structure prediction by diverse neural network architectures. `https://github.com/idrori/cu-ssp`, 2018.

[11] Chao Fang, Yi Shang, and Dong Xu. Mufold-SS: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598, 2018.

[12] Jean Garnier, David J Osguthorpe, and Barry Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, 120(1):97–120, 1978.

[13] J-F Gibrat, J Garnier, and B Robson. Further developments of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *Journal of molecular biology*, 198(3):425–443, 1987.

[14] 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*, 5:11476, 2015.

[15] L Howard Holley and Martin Karplus. Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences*, 86(1):152–156, 1989.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] Alexander Rosenberg Johansen, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Deep recurrent conditional random field network for protein secondary prediction. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 73–78. ACM, 2017.

[18] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

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

[20] DG Kneller, FE Cohen, and R Langridge. Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of molecular biology*, 214(1):171–182, 1990.

[21] Zhen Li and Yizhou Yu. Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *International Joint Conference on Artificial Intelligence*, 2016.

[22] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *Emperical Methods in Natural Language Processing*, 2015.

[23] Christophe N Magnan and Pierre Baldi. Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 2014.

[24] Uros Midic, A Keith Dunker, and Zoran Obradovic. Improving protein secondary-structure prediction by predicting ends of secondary-structure segments. In *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on*, pages 1–8. IEEE, 2005.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993.

[27] Burkhard Rost and Chris Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics*, 19(1):55–72, 1994.

[28] Søren Kaae Sønderby and Ole Winther. Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828*, 2014.

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[30] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[32] Sheng Wang, Wei Li, Shiwang Liu, and Jinbo Xu. Raptorx-property: a web server for protein structure property prediction. *Nucleic acids research*, 44(W1):W430–W435, 2016.

[33] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6(1):18962, 2016.

[34] Z. Wang, F. Zhao, J. Peng, and J. Xu. Protein 8-class secondary structure prediction using conditional neural fields. In *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 109–114, Dec 2010.

[35] Claire L Wilson, Paul E Boardman, Andrew J Doig, and Simon J Hubbard. Improved prediction for n-termini of α-helices using empirical information. *Proteins: Structure, Function, and Bioinformatics*, 57(2):322–330, 2004.

[36] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, 19(3):482–494, 2018.

[37] Adam Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[38] Buzhong Zhang, Jinyan Li, and Qiang Lü. Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC bioinformatics*, 19(1):293, 2018.

[39] Jian Zhou and Olga G Troyanskaya. CB6133 dataset, 2014.

[40] Jian Zhou and Olga G Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. *International Conference on Machine Learning*, pages 745–753, 2014.