


Article

Protein Solvent-Accessibility Prediction by a Stacked Deep Bidirectional Recurrent Neural Network

Buzhong Zhang ^{1,2} , Linqing Li ¹ and Qiang Lü ^{1,*}

¹ School of Computer Science and Technology, Soochow University, Suzhou 215006, China; 20154027005@stu.suda.edu.cn (B.Z.); linqinglee@gmail.com (L.L.)

² School of Computer and Information, Anqing Normal University, Anqing 246011, China

* Correspondence: qiang@suda.edu.cn

Received: 21 April 2018; Accepted: 22 May 2018; Published: 25 May 2018



Abstract: Residue solvent accessibility is closely related to the spatial arrangement and packing of residues. Predicting the solvent accessibility of a protein is an important step to understand its structure and function. In this work, we present a deep learning method to predict residue solvent accessibility, which is based on a stacked deep bidirectional recurrent neural network applied to sequence profiles. To capture more long-range sequence information, a merging operator was proposed when bidirectional information from hidden nodes was merged for outputs. Three types of merging operators were used in our improved model, with a long short-term memory network performing as a hidden computing node. The trained database was constructed from 7361 proteins extracted from the PISCES server using a cut-off of 25% sequence identity. Sequence-derived features including position-specific scoring matrix, physical properties, physicochemical characteristics, conservation score and protein coding were used to represent a residue. Using this method, predictive values of continuous relative solvent-accessible area were obtained, and then, these values were transformed into binary states with predefined thresholds. Our experimental results showed that our deep learning method improved prediction quality relative to current methods, with mean absolute error and Pearson's correlation coefficient values of 8.8% and 74.8%, respectively, on the CB502 dataset and 8.2% and 78%, respectively, on the Manesh215 dataset.

Keywords: solvent-accessibility prediction; bidirectional recurrent network; sequence profile; merging operator

1. Introduction

Residue solvent accessibility (RSA) [1] in a protein is defined as the extent of accessible surface area of a given residue and is related to the residue spatial arrangement and packing. It reveals the folding state of proteins and has been considered as a significant quantitative measurement for three-dimensional structures of proteins [2]. Solvent accessibility is closely involved in structural domains' identification [3], fold recognition [4], binding region identification [5], protein-protein interactions [6] and protein-ligand interactions [7]. Therefore, predicting the RSA of a protein represents an important step in determining its structure and function. Traditionally, RSA prediction is performed in two forms: (1) as a binary or multi-class classification problem with varying thresholds (two-state (exposed or buried) [8] or three-state (exposed, intermediate, or buried)) [9]; and (2) based on the relative accessible solvent area (rASA) prediction [10]. For example, if the surface area of a residue exceeds a threshold of 25%, the residue is classified as exposed; however, there is no standard definition of the thresholds for solvent-accessible area. Generally, the later approach is preferred over the former since rASA provides more information compared to binary or multi-class classification. For instance, it provides numerical values, which is required to apply this characteristic in protein structure and

function prediction. In view of this, it is necessary to predict rASA. Recently, more and more studies have focused on rASA prediction.

Machine learning methods are extensively applied in RSA prediction and include those related to sequence alignment [11], neural networks [12–15], support vector machines [16,17], the use of multiple linear-regression models [18], Bayesian statistics [19], nearest-neighbor methods [20], energy optimization [21], gradient-boosted regression trees [22] and deep learning [23].

In 2003, Ahmad et al. [10] firstly detailed rASA prediction, and more attention has been paid to this research since that time. Ahmad et al. proposed a neural network method with only single sequence information as the input features, and the result of 18.8% mean absolute error (MAE) was achieved on the CB502 dataset. Wang et al. [18] applied the multiple linear regression method to predict rASA from the sequence information and position-specific scoring matrix (PSSM). This method achieved a result of 16.2% MAE on the CB502 dataset. Wang et al. [17] improved the result to 15.1% on the same dataset by accumulation cutoff set and support vector machine. Using a weighted sliding window scheme, Zhang et al. [24] obtained the result of 14% MAE on the CB502 dataset. Fan et al. [22] used gradient boosted regression trees to predict rASA and achieved a state-of-the-art performance, which is 9.4% MAE and 0.73 Pearson's correlation coefficient (PCC) on the CB502 dataset. Another benchmark dataset Manesh215 [25] is also widely used by researchers [10,13,14,22,24,26] to validate prediction methods. Table 1 summarizes the recent developments in predicting the values of rASA.

Table 1. The recent developments, in chronological order, for predicting the values of rASA.

Work	Algorithm	Description of Features	MAE (%)
Ahmad, 2003 [10]	Neural network	Amino acid composition	18.8
Wang, 2005 [18]	Multiple linear regression	PSSM	16.2
Garg, 2005 [13]	Neural network	PSSM, secondary structure	16.6
Nguyen, 2006 [27]	Two-stage SVR	PSSM	15.7
Wang, 2007 [17]	Support vector machine	PSSM	15.1
Dor, 2007 [14]	Neural network	PSSM, physical properties	14.3
Chang, 2008 [28]	Support vector regression	Enhances PSSM-based features	14.8
Faraggi, 2009 [15]	Neural networks	PSSM, physical properties, secondary structure	11.1
Meshkin, 2009 [29]	Pace regression	PSSM	13.4
Joo, 2012 [20]	k-nearest neighbor	PSSM	14.8
Kashefi, 2013 [30]	SVR and scatter search methods	PSSM, qualitative physicochemical features	12.31
Zhang, 2015 [24]	Weighted sliding window	PSSM, secondary structure, native disorder, physicochemical propensities, sequence-based features	14
Fan, 2016 [22]	Gradient boosted regression trees	PSSM, secondary structure, native disorder, conservation score, side-chain environment	9.4

The MAEs reported in this table were evaluated on a different dataset.

Position-specific scoring matrix has been widely used in proteomics and bioinformatics, such as protein structure prediction [31,32], backbone angles [23], protein subcellular localization prediction [33–35], membrane protein type prediction [36–38], protein subchloroplast localization [39], etc. Protein-sequence coding and PSSM have also been used for RSA prediction [12,16,20]. Recently, other sequence-based features, such as physical properties [40], conservation score [41] and side-chain environment [42], have been used [20–23]. Additionally, properties predicted by computational methods have also been used, including those associated with evaluating secondary structure and protein disorder [21,22,43]. For most of these, standard labels for RSA areas are calculated by DSSP software [44]. The rASA is obtained by normalizing the ASA value over the maximum value of exposed surface area obtained for either (1) each amino acid or for (2) an extended tripeptide conformation of Ala-X-Ala or Gly-X-Gly [10].

Although these methods for RSA prediction have progressed, RSA prediction remains challenging. Improved performance in these areas will enable more precision in related protein studies.

Recently, deep learning methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains [45]. The Recurrent Neural Network (RNN), a type of deep neural network, processes an input sequence one element at a time, maintaining a hidden vector that implicitly contains the history information about the

past elements of the sequence. RNN is an extremely powerful sequence model for sequence modeling tasks [46], and many RNN methods have been applied to protein structure and function prediction [47–49]. We focused on a RNN model specific to protein sequences and applied a bidirectional recurrent neural network (BRNN) to predict RSA. To capture more local and long-range information, a merging operator was used to merge bidirectional information, given that this has rarely been addressed previously. Our deep learning method, a stacked deep bidirectional recurrent neural network (SDBRNN), is proposed, with three-layer bidirectional long-short memory (BLSTM) network with different merging operators used in each hidden layer. The public benchmark datasets CB502, Manesh215 and CASP10 were used for testing, with our experimental results showing that SDBRNN outperformed current state-of-the-art methods. Our method represents a more general approach and can be applied to a broad range of problems within and outside of computational biology. The rASA prediction tool of SDBRNN, training and testing datasets can be download from <http://210.45.175.81:8080/rsa/sdbrrn.html>.

2. Results

2.1. Measurement Evaluation

To evaluate model performance for RSA prediction, four widely-used measures were used: MAE, PCC, accuracy (ACC) and Matthews' correlation coefficient (MCC). The formulas are defined in Equations (1)–(4).

$$MAE = \frac{\sum_{i=1}^N |RSAr_i - RSAp_i|}{N} \quad (1)$$

$$PCC = \frac{\sum_{i=1}^N (RSAr_i - \overline{RSAr})(RSAp_i - \overline{RSAp})}{\sqrt{[\sum_{i=1}^N (RSAr_i - \overline{RSAr})^2][\sum_{i=1}^N (RSAp_i - \overline{RSAp})^2]}} \quad (2)$$

In Equations (1)–(2), $RSAr_i$ and $RSAp_i$ are the real and predicted rASA values of the i -th residue in the sequence, respectively, while \overline{RSAr} and \overline{RSAp} are the corresponding mean values. N is the length of the protein sequence to predict. MAE is used to evaluate the deviation between the predicted and real relative values of RSA, and PCC is employed to quantify the relationship between predicted and real values.

$$ACC = (TP + TN) / N \quad (3)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4)$$

MAE and PCC were commonly used to measure continuous values, whereas ACC and MCC were often used to measure binary classification. Prediction accuracy is the true predicted residues divided by the length of predicted sequence. The MCC is a correlation coefficient between the observed and predicted binary classifications. MCC is able to recover the drawback of accuracy regarding unbalance data. In Equations (3)–(4), TP , FP , TN and FN represent the number of true positives (exposed residues), true negatives (buried residues), false positives and false negatives, respectively.

2.2. Performance on Relative Solvent-Accessible Area Prediction

RSA predictors that use machine learning methods are converted into regressive problems. Results from the SDBRNN on the CB502 and Manesh215 datasets are shown in Table 2. Five predictors, including SARpred [13], SVR [13], Real-SPINE [15], NetSurfP [50] and PredRSA [22], the results of which are state-of-the-art, were used for comparison. Our method achieved an MAE of 8.8% and a PCC of 75% on CB502, an MAE of 8.2% and a PCC of 78% on Manesh215 respectively, both of which

are better than the state-of-the-art. Additionally, PCC values for the CB502 and Manesh215 datasets were 2% and 3% higher than that of PredRSA. To evaluate the generalization on individual sequences, the predicted MAEs of individual sequences from the Manesh215 dataset have been further analyzed in Figure 1. The x-axis represents the sequence length. The MAE values were mostly from 0.065–0.1. In general, prediction performances on long sequences are higher than those of short sequences.

Table 2. Performance comparison in predicting relative solvent-accessible areas (the best results are shown in bold).

Method	CB502		Manesh215	
	MAE (%)	PCC	MAE (%)	PCC
SARpred	17.4	0.6	16.6	0.61
SVR	14.8	0.68	14.2	0.69
Real-SPINE	14.5	0.68	13.8	0.7
NetSurfP	14.3	0.71	13.6	0.7
PredRSA	9.4	0.73	9.0	0.75
SDBRNN	8.8	0.75	8.2	0.78

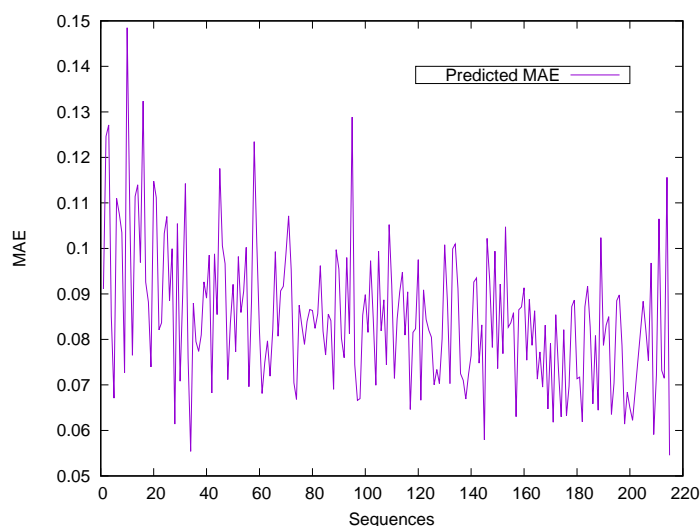


Figure 1. Predicted MAE based-on individual sequence from the Manesh215 dataset. The protein sequences are ordered by sequence length.

Two publicly-available datasets (CASP10 and CASP11) were also used to validate SDBRNN. CASP10 contains 90 proteins, and CASP11 contains 72 proteins, with the lengths of all sequences between 50 and 600 residues. The PCC value for the CASP10 dataset was 74.2%, which was 3.2% better than that for PredRSA, and that for CASP11 was 74.3%, which was slightly better than SPIDER2 [23].

Another representative method (PSO-SVR [24]) was also compared. The maximum solvent-accessible area standard used in that study was according to Adhamd's work [10], which used extended tripeptides (Ala-X-Ala). We re-prepared data using the standard and re-trained model. Experiments using our method showed MAE and PCC values of 12.0%, and 0.765, respectively, on the Manesh215 dataset, which were better than 13.2% and 0.74 achieved by PSO-SVR. Additionally, on the CB502 dataset, our method showed MAE and PCC values of 13.3% and 0.739, which were better than 14% and 0.73 obtained by PSO-SVR.

2.3. Comparison of Different Classification Predictors

There are many methods proposed for predicting binary classification (exposed or buried) of residues. Thus, we have also examined the performance of our method in terms of two-state predictions. Similar to the two-layer predictor strategy [37,38], firstly the predicted rASA values were generated by the SDBRNN model. Then, the predicted relative values were transformed into binary states with predefined thresholds for comparison. For instance, the residue with a rASA value that is greater than or equal to the threshold, it can be considered as exposed, otherwise, it is considered as buried. The predictive ability between methods was compared against SARpred [13], PR [29], SVR [28], two-stage SVR [27], SS-SVM [30] and PredRSA using the Manesh215 dataset. Table 3 shows the performances of different methods. Results indicated that the SDBRNN model performed better at RSA prediction and generalization, except for the threshold of 50%.

Table 3. Binary classification prediction comparison between our method and other reported methods with different thresholds on the Manesh215 dataset.

Method	Accuracy for Two-State Prediction						
	5%	10%	20%	25%	30%	40%	50%
SARpred	74.9	77.2	77.7	-	77.8	78.1	80.5
PR	76.8	74.8	75.3	76.7	77.7	79.8	86.3
SVR	80.9	80.1	78.7	-	-	-	80.8
SS-SVM	79.2	78.2	77.6	77.6	77.5	79.7	86.5
Two-stage SVR	81.1	78.7	77.6	77.3	-	-	79.5
PredRSA	80	81.6	80.9	81.1	82.2	87.1	93.2
SDBRNN	83.5	82.4	82.3	82.6	83.5	87.4	93

Predicted accuracy on individual sequence from the Manesh215 dataset is analyzed in Figure 2 for evaluating the model generalization. The prediction accuracy of most proteins is between 75% and 90%. Only 14 sequences out of the 215 proteins have a prediction accuracy of less than 75%.

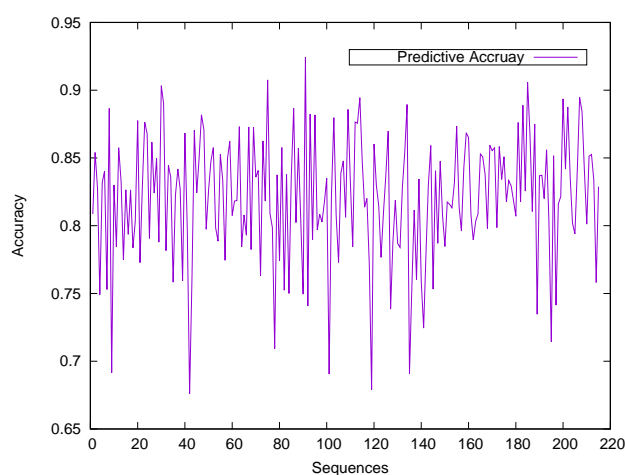
We also tested our model using three publicly-available datasets at different thresholds: CB502, Manesh215 and CASP10. ACC and MCC results are listed in Tables 4 and 5. To compare with PredRSA [22] in detail, its results are also listed in the tables. Our method showed mostly better accuracy on CB502 and Manesh215, except the threshold of 50%. At a threshold of 50%, the ACC associated with the PredRSA on the CB502 and Manesh215 datasets was 0.6% and 0.2% better than our method, respectively. The MCC results of our method are still better than PredRSA.

Table 4. Accuracy (%) performance comparison in binary classification prediction (the best results are shown in bold).

Threshold (%)	Manesh215		CB502		CASP10	
	PredRSA	SDBRNN	PredRSA	SDBRNN	PredRSA	SDBRNN
5	80.1	83.5	77.9	82.3	78.5	84
10	81.7	82.4	79	80.9	79.1	82.1
20	81	82.3	80.5	80.7	78.3	80.6
25	81.2	82.6	81	81.4	79.7	80.2
30	82.4	83.5	82.1	82.5	80.5	81.2
40	87.1	87.4	86.8	87	85	85.4
50	93.2	93	93	92.4	91.2	91.4

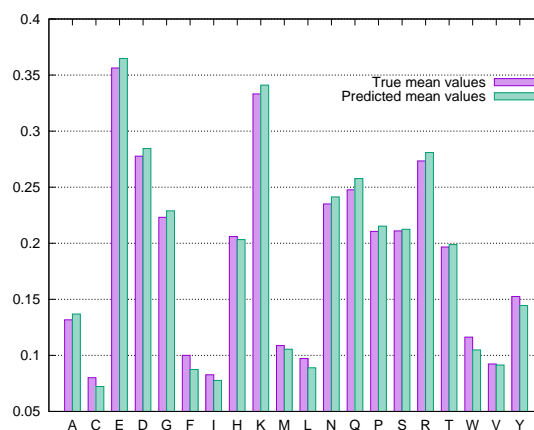
Table 5. Matthews' correlation coefficient performance comparison in binary classification prediction.

Threshold (%)	Manesh215		CB502		CASP10	
	PredRSA	SDBRNN	PredRSA	SDBRNN	PredRSA	SDBRNN
5	0.54	0.62	0.5	0.59	0.48	0.61
10	0.63	0.64	0.58	0.61	0.57	0.63
20	0.61	0.63	0.6	0.6	0.56	0.61
25	0.58	0.61	0.57	0.58	0.56	0.57
30	0.54	0.58	0.52	0.55	0.51	0.53
40	0.42	0.48	0.39	0.46	0.4	0.43
50	0.25	0.34	0.23	0.33	0.3	0.31

**Figure 2.** Predicted accuracy on individual sequences from the Manesh215 dataset. The rASA threshold is 25%. Protein sequences are ordered by sequence length.

2.4. Residue-Specific Variation in Predictive Error

To evaluate the predictive performance for various types of residues, we calculated the average rASA values using the Manesh215 and CB502 datasets for all 20 types of amino acids. SDBRNN accurately predicted RSA using the CB502 dataset with an MAE <1%, except that phenylalanine (F) and tryptophan (W) were predicted with an MAE <1.3% (Figure 3). Most amino acids in the Manesh215 dataset were predicted with an MAE <1%, with tyrosine, histidine, tryptophan and phenylalanine predicted with <2% MAE.

**Figure 3.** Comparison of true mean values and predicted mean values for 20 types of amino acids using the CB502 dataset.

3. Discussion

To improve the performance on different combinations of sequence-derived features, we validated five types of input variables using the TR7000 dataset for training and an independent test set (TS261). The performance of the method on different feature combinations was compared, with the results (Table 6) indicating that all five features contributed to RSA prediction.

Table 6. Different combinations of sequence-derived features for SDBRNN predictors on an independent test set (TS261).

Feature	MAE (%)	PCC
PSSM	9.33	0.732
PSSM + SC	9.03	0.749
PSSM + SC + CS	9.00	0.750
PSSM + SC + CS + PP	8.95	0.750
PSSM + SC + CS + PP + PC	8.86	0.753

PSSM: position specific scoring matrix, 20 dimensions; SC: protein sequence coding, 22 dimensions; CS: residue conservation score, 1 dimension; PP: physical properties, 7 dimensions; PC: physicochemical characteristics, 3 dimensions.

When bidirectional information from a hidden node is merged in the BRNN, concatenation (concat) [51] and sum [47] are commonly used. In the SDBRNN, three merging operators (“concat”, “sum” and “weighting sum”) were used in different hidden layers. To assess the generalization of this BRNN model using hybrid merging operators, we compared the SDBRNN with a BLSTM using the “concat” operator (BLSTM_C), with an BLSTM using the “sum” operator (BLSTM_S) and with a unidirectional LSTM with 800 hidden nodes. Hyperparameters for the BLSTM_C and BLSTM_S were the same as those used for the SDBRNN, and rASA prediction was used to compare the different models. The results listed in Table 7 show that hybrid merging operators effectively promoted better predictive performance.

Table 7. Comparison of different LSTM models on relative RSA area prediction. MAE is the value percentage (%).

Method	CB502		Manesh215		CASP10		TS261	
	MAE (%)	PCC	MAE (%)	PCC	MAE (%)	PCC	MAE (%)	PCC
LSTM	9.8	0.694	9.4	0.722	10.0	0.698	10.0	0.695
BLSTM_C	9.0	0.74	8.44	0.772	9.33	0.734	9.0	0.748
BLSTM_S	8.93	0.744	8.33	0.775	9.26	0.739	8.96	0.747
SDBRNN	8.84	0.748	8.24	0.777	9.19	0.742	8.86	0.753

4. Materials and Methods

4.1. Datasets and Input Features

A large, non-homologous sequence dataset, produced using the PISCES server [52], was obtained. Structures exhibited <25% similarity and showed a resolution >3.0 Å, with R factors of 1.0. Three-dimensional structure files were downloaded from the RCSB Protein Data Bank. After removing redundancy in the test datasets using cd-hit [53], 7361 proteins (CullPDB7361) comprising 1,596,728 residues in proteins with sequence lengths between 50 and 600 residues were retained. Among these, 7000 proteins (TR7000) were used for training, and 261 proteins (TS261) and 100 proteins (VD100) were randomly selected for testing and validating the model. In addition to CullPDB7361, three public testing datasets, CB502, Manesh215 and CASP10, were used to evaluate the performance of our model. CB502 (82,420 residues) was selected from CB513 [54] and ordered by sequence length

from long to short. The Manesh215 dataset [25] contains 47,243 residues. CASP10 from the Protein Sequence Prediction Center (<http://predictioncenter.org/>) has 123 domain fragments and extracts from 103 chains. The CASP10 dataset was selected according to protein identity and contained 20,778 residues from 90 proteins. The sequence lengths of the proteins in test datasets were all ≤ 600 .

Five types of features, PSSM, protein sequence coding, conservation score, physical properties and physicochemical characteristics, were used as input features. PSSM-derived features have been widely used to perform protein-related predictions. PSSMs provide the effective frequency of occurrence of all 20 amino acid residues at each position of the sequence. PSSMs can be obtained by performing multiple sequence alignments on a large protein database (NCBI NR database) using PSI-BLAST [55]. PSI-BLAST involves an iterative search process in searching the profile of a query protein. The expectation value (E-value) and the number of iterations for PSI-BLAST are set to 0.001 and three, respectively. The PSSM profile is in the form of a $20 \times L$ matrix where L is the length and each amino acid in the sequence is described by 20 features.

Physical properties [40] include: a steric parameter (graph-shape index), polarizability, volume (normalized van der Waals volume), hydrophobicity, isoelectric point, helix probability and sheet probability. Specific values were derived from the study [40]. Protein physicochemical characteristics [56] include the number of atoms, electrostatic charges and potential hydrogen bonds.

To ensure smooth transitions in changes to the network gradient, the above features were normalized using a logistic regression function $y = 1/(1 + e^{-x})$.

Residue conservation was derived from the amino acid frequency distribution in the corresponding column of a multiple-sequence alignment of homologous proteins. A one-dimensional conservation score was computed according to Quan's [41] previously described Equation (5):

$$R = \log 20 + \sum_{i=1}^{20} Q_i \log Q_i \quad (5)$$

Commonly-used protein coding involves an orthogonal code. In addition to the 20 known residues, "X" represents unknown residues, and "NoSeq" represents non-protein sequences in our coding scheme. A $22 \times L$ matrix represents a sequence, where L is the sequence length and a 22-dimensional (dim) vector represents a residue in the sequence. However, the 22-dimensional coding vector represents a parsed, one-hot vector, where only one of 22 elements is a non-zero value. For instance, for the residue "A", the coding is: "1 0". We adopted an embedding operation from natural-language processing to transform sparse sequence features into denser representations. This embedding operation was implemented as a simple auto-encoder (a feed-forward neural-network layer) along with an embedding matrix mapping a sparse 22-dimensional vector into a denser 22-dimensional vector. This transformation was just converted into a one-hot vector, which coded one residue into a dense vector.

In our scheme, an input residue was represented by 53-dimensional features: 20-dim PSSM, 7-dimensional physical properties, 3-dimensional physicochemical characteristics, 1-dimensional conservation score and 22-dimensional protein codings. These features are all derived from the protein sequence.

The rASA of a residue in a protein chain is calculated by dividing the accessible surface area derived from DSSP [44] by the maximum solvent accessibility. However, there is no standard for maximum solvent accessibility. According to previous results [22,25], Gly-X-Gly extended tripeptides were used.

4.2. BRNN and Merging Operator

For sequence data $X = (x_1, x_2, x_3 \dots x_{t-1}, x_t, x_{t+1} \dots x_n)$, where x_i is context dependent and strongly reliant on forward and backward information. The label vector $Y = (y_1, y_2, y_3 \dots y_{t-1}, y_t, y_{t+1} \dots y_n)$ is the target output space. Compared to a forward neural network, the current output from the recurrent

neural network will be reverted backward for subsequent time-specific inputs. The recurrent neural network structure can be described as Equation (6):

$$\begin{aligned} h_t &= f(W_x x + W_h h_{t-1} + b_n) \\ y_t &= \sigma(W_y h_t + b_y) \end{aligned} \tag{6}$$

At the time $T = t$, the recurrent network can remember the information from previous $x_1, x_2, x_3 \dots x_{t-1}$ and the present input x_t . However, in many applications, the output y_t might be dependent on the entire input sequence, as in protein sequences and handwriting-recognition examples. BRNN [57] combines an RNN that moves forward through time beginning from the start of the sequence along with another RNN that moves backward through time beginning from the end of the sequence. In this BRNN, time-increasing input is represented by $\vec{f}(x_1, x_2, x_3, \dots, x_t)$, and time-decreasing input is represented by $\overleftarrow{f}(x_t, x_{t+1}, \dots, x_n)$. Bidirectional information is merged as the current node's output. Therefore, BRNN is more suitable for context-related applications, and it is capable of outperforming unidirectional recurrent neural network. The BRNN can be described as Equation (7):

$$\begin{aligned} O_t &= \eta(F_t, B_t, x) \\ \text{s.t. } F_t &= (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_t), \\ B_t &= (\overleftarrow{h}_t, \overleftarrow{h}_{t+1}, \dots, \overleftarrow{h}_n) \end{aligned} \tag{7}$$

The BRNN structure is capable of efficiently learning sequence information [45]. Previous studies focused primarily on the BRNN methodology with little research on incorporating bidirectional-information merging. In a conventional neural network, pooling operations are computationally important, with the pooling layer responsible for accumulating convolutional results. The merging operator in a BRNN acts similarly to pooling operations. At time $T = t$, input forwarded to the current node is represented by $\vec{f}(x_1, x_2, x_3, \dots, x_t)$, and the backward input is represented by $\overleftarrow{f}(x_t, x_{t+1}, \dots, x_n)$. Therefore, a common formula for the output of the current node is presented in Figure 4 and as follows:

$$\begin{aligned} h_t &= W_f F_t \Theta W_b B_t + \gamma b_t \\ W_f, W_b &\in [0, 1], \gamma \in \{0, 1\} \end{aligned} \tag{8}$$

where Θ is the merging operator, $\Theta \in R, R = \{\otimes, +, \oplus, \infty, \max, \min, \text{avg}, \dots\}$. \otimes represents element-wise multiplication; $+$ is the sum; \oplus is the element-wise weighting sum; ∞ is concatenation; and \odot is the reshape operator. The merging operator can perform more computations than necessary when the information is merged via the aggregation operation.

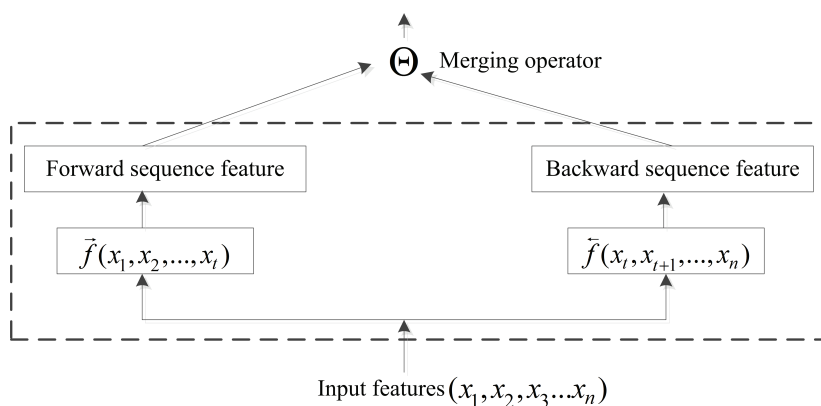


Figure 4. In order to remember more long-range information in the sequence study, when the past computing information and the future computing information are merged, the merging operator is proposed to execute the merging operation.

4.3. Model

4.3.1. SDBRNN

Protein primary structure represents a strongly time-ordered conformation, suggesting that it contains adequate information for the protein to fold into its native conformation. To capture more features in the primary structure, we use three types of computing operators in the merging operation. During the initial period, the computing node in the BRNN needs to remember backward, forward and current input information; therefore, the merging operator “concat” is used in the first layer. In the second layer, the computing node no longer needs to remember previous and future information; however, the “sum” operator reinforces bidirectional input aggregation, with bidirectional information again aggregated in the third layer. Therefore, a “weighting sum” operator is used for filtering unnecessary information and extracting key features. The equations associated with these operations are as follows (9)–(11):

$$h_t^1 = F_t^1 \circ B_t^1 \quad (9)$$

$$h_t^2 = F_t^2 + B_t^2 \quad (10)$$

$$\begin{aligned} h_t^3 &= W_f F_t^3 + W_b B_t^3 + b_t \\ O(h_t) &= \tanh(h_t^3) \end{aligned} \quad (11)$$

The recurrent neural network accepts a time-stepping sequence that makes the network extremely deep, with the depth making the network difficult to train because of the exploding or vanishing gradient [46]. Long short-term memory (LSTM) [58], which consists of a variety of gate structures (a forget gate, an input gate and an output gate) and a memory cell are used to address the vanishing gradient problem. In the SDBRNN architecture, the unidirectional computing node is performed by LSTM. The widely-accepted LSTM definition was provided by Graves [59], and the details of LSTM are described in Equation (12), which involves no peep-hole connections, mainly to improve computing performance.

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ c_t &= f_t c_{t-1} + i_t \tanh(W_c x_t + U_c h_{t-1} + b_c) \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (12)$$

where σ is the sigmoid function; i , f , o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the hidden vector h .

The following BRNN layer represents a multi-layer perceptron (MLP) network that reduces the network scale. The MLP layer with logistic activation ultimately outputs the prediction, with logistic regression used for fitting the relative surface-area value. The SDBRNN architecture is shown in Figure 5. The cross-entropy loss function is used for model training:

$$L(x_i) = \frac{1}{N} \sum_{i=1}^N y_i \times \log(\hat{y}_i) \quad (13)$$

where \hat{y}_i are the predicted probabilities of secondary structure labels, y_i represent ground-truth labels of the secondary structure and N is the number of input residues. \hat{y}_i is a vector of output activations prior to normalization with the logistic function. These derivatives are then fed back through the network using back propagation through time to determine the weight gradient.

In our model, rASA prediction is simulated as a regressive problem. With the predicted rASA values, the classifications are carried out in two states (buried and exposed). We have tried seven thresholds of 5, 10, 20, 25, 30, 40 and 50% in the two-state classification.

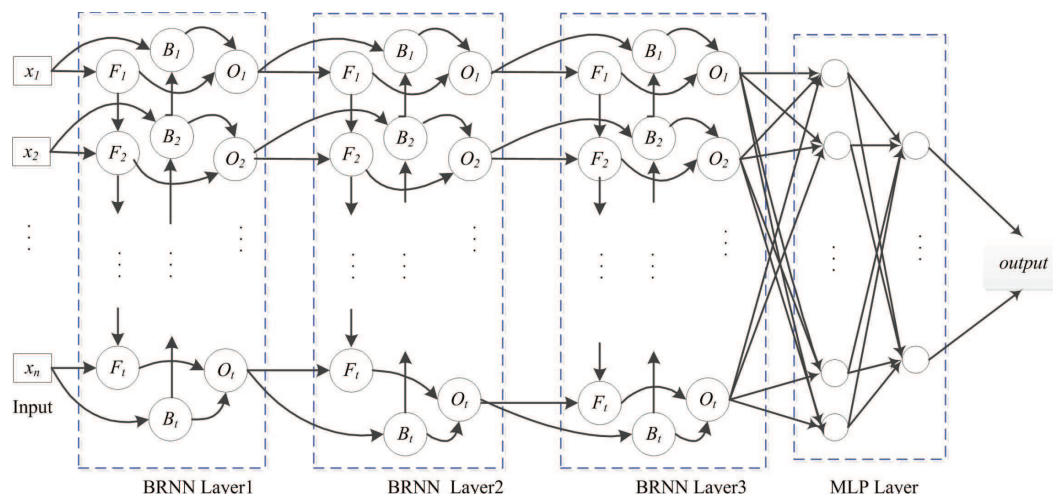


Figure 5. SDBRNN architecture. The merging operator “concat” is used in the first BRNN layer. The “sum” and “weighting sum” operator are used in the second and third layer. Two multi-perception networks are connected to BRNN.

4.3.2. Model Hyperparameters

SDBRNNs have three BRNN layers and two MLP layers. The first BRNN layer has 300 nodes, with 500 nodes in the other two layers. The first MLP layer has 128 nodes, and the second has 60 nodes. The Adam optimizing function was used for training the network using default settings, with the default learning rate set to 0.0008. This was reduced by 50%, whereas the validation accuracy decreased by more than 10-times. The threshold of the learning rate was set to 0.0001.

A natural stopping policy was accepted while the validation stopped decreasing. To balance model performance between fitting and generalization, a regularization or penalty term was attached to the loss function (e.g., $L1$ or $L2$). Unlike other models, regularization was not attached, and our experiment displayed dropout efficiency [60]. Except for the classification layer, each layer was regularized according to a dropout ($p = 0.5$) to avoid overfitting. Our model was implemented in Keras, which is a public deep learning software based on Theano [61]. Weights in the SDBRNN were initialized with default values in Keras. The network was trained using a single NVIDIA Tesla K40 GPU with 12 GB memory. Training the model requires about 16.5 min per epoch, and it takes about 0.04 seconds to test one sequence.

Proteins shorter than 600 AA were padded with all-zero features. The outputs corresponding to padded inputs are labeled as 0.5 according to the logistic function. The advantage of padding proteins is that it enables training the model on a GPU in batches.

5. Conclusions

Accessible RSA is often used as an important measure in proteomics study for describing protein properties. Compared with traditional machine learning techniques, deep learning exhibits wider generalization and is applied in our study to predict RSA area. Deep neural networks are stacked by various complicated neural networks, and more generalized representation capacity can be obtained. However, the deep neural networks own enormous parameters and need more samples and computing resources to train models. Use of the merging operator was proposed for merging computations in the BRNN hidden layer. We redesigned BLSTM merging using three types of merging operators (“concat”, “sum”, and “weighting sum”) in SDBRNN. Compared with the commonly-used BLSTM method using a single merging operator, as well as other predictors, SDBRNN captured more protein features and was more generalizable. Our results on test datasets verified this.

Relative solvent-accessible areas are greatly affected by maximum solvent-accessible areas, as more maximum solvent-accessible areas will lead to lower relative solvent-accessible areas and lower MAEs.

However, there are no standard maximum solvent accessible areas. In this work, a novel deep learning method (SDBRNN) was presented for the prediction of RSA area, as well as for binary state classification. Our methods can be applied to a broad range of problems within and outside of computational biology.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2218-273X/8/2/33/s1>, File f1: PDB-ID list of CullPDB7361. Table t1: Predicted MAE and accuracy at the threshold of 25% of individual sequences on the Manesh215 dataset.

Author Contributions: Q.L. conceived of the study. B.Z. performed the experiments, analyzed the data and initially drafted the manuscript. L.L. collected the features. All authors contributed to the revision and approved the final manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number No. 61170125, the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) and the Natural Science Research Project of Anhui Provincial Department of Education grant number No. KJ2018A0383.

Acknowledgments: Our thanks are given to the reviewers for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RSA	Residue Solvent Accessibility
rASA	relative accessible solvent area
BLSTM	bidirectional long-short memory
SDBRNN	stacked deep bidirectional recurrent neural network
RNN	recurrent neural network
BRNN	bidirectional recurrent neural network
LSTM	long short-term memory
MLP	multi-layer perceptron
MAE	mean absolute error
PCC	Pearson's correlation coefficient
MCC	Matthews' correlation coefficient
ACC	Accuracy

References

1. Lee, B.; Richards, F.M. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.* **1971**, *55*, 379–400. [[CrossRef](#)]
2. Rost, B.; Sander, C. Conservation and prediction of solvent accessibility in protein families. *Proteins Struct. Funct. Bioinform.* **1994**, *20*, 216–226. [[CrossRef](#)] [[PubMed](#)]
3. Wodak, S.J.; Janin, J. Location of structural domains in protein. *Biochemistry* **1981**, *20*, 6544–6552. [[CrossRef](#)] [[PubMed](#)]
4. Liu, S.; Zhang, C.; Liang, S.; Zhou, Y. Fold recognition by concurrent use of solvent accessibility and residue depth. *Proteins Struct. Funct. Bioinform.* **2007**, *68*, 636–645. [[CrossRef](#)] [[PubMed](#)]
5. Mooney, C.; Pollastri, G.; Shields, D.C.; Haslam, N.J. Prediction of short linear protein binding regions. *J. Mol. Biol.* **2012**, *415*, 193–204. [[CrossRef](#)] [[PubMed](#)]
6. Connolly, M.L. Solvent-accessible surfaces of proteins and nucleic acids. *Science* **1983**, *221*, 709–713. [[CrossRef](#)] [[PubMed](#)]
7. Huang, B.; Schroeder, M. LIGSITE csc: Predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.* **2006**, *6*, 19. [[CrossRef](#)] [[PubMed](#)]
8. Janin, J. Surface and inside volumes in globular proteins. *Nature* **1979**, *277*, 491. [[CrossRef](#)] [[PubMed](#)]
9. Rose, G.; Geselowitz, A.; Lesser, G.; Lee, R.; Zehfus, M. Hydrophobicity of amino acid residues in globular proteins. *Science* **1985**, *229*, 834–838. [[CrossRef](#)] [[PubMed](#)]

10. Ahmad, S.; Gromiha, M.M.; Sarai, A. Real value prediction of solvent accessibility from amino acid sequence. *Proteins Struct. Funct. Bioinform.* **2003**, *50*, 629–635. [[CrossRef](#)] [[PubMed](#)]
11. Holbrook, S.R.; Muskal, S.M.; Kim, S.H. Predicting surface exposure of amino acids from protein sequence. *Protein Eng.* **1990**, *3*, 659–665. [[CrossRef](#)] [[PubMed](#)]
12. Ahmad, S.; Gromiha, M.M. NETASA: Neural network based prediction of solvent accessibility. *Bioinformatics* **2002**, *18*, 819–824. [[CrossRef](#)] [[PubMed](#)]
13. Garg, A.; Kaur, H.; Raghava, G.P. Real value prediction of solvent accessibility in proteins using multiple sequence alignment and secondary structure. *Proteins Struct. Funct. Bioinform.* **2005**, *61*, 318–324. [[CrossRef](#)] [[PubMed](#)]
14. Dor, O.; Zhou, Y. Real-SPINE: An integrated system of neural networks for real-value prediction of protein structural properties. *Proteins Struct. Funct. Bioinform.* **2007**, *68*, 76–81. [[CrossRef](#)] [[PubMed](#)]
15. Faraggi, E.; Xue, B.; 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 Struct. Funct. Bioinform.* **2009**, *74*, 847–856. [[CrossRef](#)] [[PubMed](#)]
16. Kim, H.; Park, H. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor. *Proteins Struct. Funct. Bioinform.* **2004**, *54*, 557–562. [[CrossRef](#)] [[PubMed](#)]
17. Wang, J.Y.; Lee, H.M.; Ahmad, S. SVM-Cabins: Prediction of solvent accessibility using accumulation cutoff set and support vector machine. *Proteins Struct. Funct. Bioinform.* **2007**, *68*, 82–91. [[CrossRef](#)] [[PubMed](#)]
18. Wang, J.Y.; Lee, H.M.; Ahmad, S. Prediction and evolutionary information analysis of protein solvent accessibility using multiple linear regression. *Proteins Struct. Funct. Bioinform.* **2005**, *61*, 481–491. [[CrossRef](#)] [[PubMed](#)]
19. Thompson, M.J.; Goldstein, R.A. Predicting solvent accessibility: Higher accuracy using Bayesian statistics and optimized residue substitution classes. *Proteins Struct. Funct. Bioinform.* **1996**, *25*, 38. [[CrossRef](#)]
20. Joo, K.; Lee, S.J.; Lee, J. Sann: Solvent accessibility prediction of proteins by nearest neighbor method. *Proteins Struct. Funct. Bioinform.* **2012**, *80*, 1791. [[CrossRef](#)] [[PubMed](#)]
21. Iqbal, S.; Mishra, A.; Hoque, M.T. Improved prediction of accessible surface area results in efficient energy function application. *J. Theor. Biol.* **2015**, *380*, 380–391. [[CrossRef](#)] [[PubMed](#)]
22. Fan, C.; Liu, D.; Huang, R.; Chen, Z.; Deng, L. PredRSA: A gradient boosted regression trees approach for predicting protein solvent accessibility. *BMC Bioinform.* **2016**, *17*, S8. [[CrossRef](#)] [[PubMed](#)]
23. Heffernan, R.; Paliwal, K.; Lyons, J.; Dehzangi, A.; Sharma, A.; Wang, J.; Sattar, A.; Yang, Y.; Zhou, Y. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* **2015**, *5*, 11476. [[CrossRef](#)] [[PubMed](#)]
24. Zhang, J.; Chen, W.; Sun, P.; Zhao, X.; Ma, Z. Prediction of protein solvent accessibility using PSO-SVR with multiple sequence-derived features and weighted sliding window scheme. *BioData Min.* **2015**, *8*, 3. [[CrossRef](#)] [[PubMed](#)]
25. Naderi-Manesh, H.; Sadeghi, M.; Arab, S.; Moosavi Movahedi, A.A. Prediction of protein surface accessibility with information theory. *Proteins* **2001**, *42*, 452–459. [[CrossRef](#)]
26. Nepal, R.; Spencer, J.; Bhogal, G.; Nedunuri, A.; Poelman, T.; Kamath, T.; Chung, E.; Kantardjiev, K.; Gottlieb, A.; Lustig, B. Logistic regression models to predict solvent accessible residues using sequence- and homology-based qualitative and quantitative descriptors applied to a domain-complete X-ray structure learning set. *J. Appl. Crystallogr.* **2015**, *48*, 1976–1984. [[CrossRef](#)] [[PubMed](#)]
27. Nguyen, M.N.; Rajapakse, J.C. Prediction of protein relative solvent accessibility with a two-stage SVM approach. *Proteins Struct. Funct. Bioinform.* **2005**, *59*, 30–37. [[CrossRef](#)] [[PubMed](#)]
28. Chang, D.T.; Huang, H.Y.; Syu, Y.T.; Wu, C.P. Real value prediction of protein solvent accessibility using enhanced PSSM features. *BMC Bioinform.* **2008**, *9*, 1–12. [[CrossRef](#)] [[PubMed](#)]
29. Meshkin, A.; Sadeghi, M.; Ghasem-Aghaee, N. Prediction of relative solvent accessibility using pace regression. *EXCLI J.* **2009**, *8*, 211–217.
30. Kashefi, A.H.; Meshkin, A.; Zargoosh, M.; Zahiri, J.; Taheri, M.; Ashtiani, S. Scatter-search with support vector machine for prediction of relative solvent accessibility. *Excli J.* **2013**, *12*, 52–63. [[PubMed](#)]
31. Qian, N.; Sejnowski, T.J. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.* **1988**, *202*, 865–884. [[CrossRef](#)]

32. Rost, B.; Sander, C. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. USA* **1993**, *90*, 7558–7562. [[CrossRef](#)] [[PubMed](#)]
33. Wan, S.; Mak, M.W.; Kung, S.Y. Transductive Learning for Multi-Label Protein Subchloroplast Localization Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2016**, *14*, 212–224. [[CrossRef](#)] [[PubMed](#)]
34. Chou, K.C.; Shen, H.B. Cell-PLoc: A package of Web servers for predicting subcellular localization of proteins in various organisms. *Nat. Protoc.* **2008**, *3*, 153–162. [[CrossRef](#)] [[PubMed](#)]
35. Wan, S.; Mak, M.W.; Kung, S.Y. FUEL-mLoc: Feature-unified prediction and explanation of multi-localization of cellular proteins in multiple organisms. *Bioinformatics* **2017**, *33*, 749–750. [[CrossRef](#)] [[PubMed](#)]
36. Hayat, M.; Khan, A. MemHyb: Predicting membrane protein types by hybridizing SAAC and PSSM. *J. Theor. Biol.* **2012**, *292*, 93. [[CrossRef](#)] [[PubMed](#)]
37. Chou, K.C.; Shen, H.B. MemType-2L: A web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem. Biophys. Res. Commun.* **2007**, *360*, 339–345. [[CrossRef](#)] [[PubMed](#)]
38. Wan, S.; Mak, M.W.; Kung, S.Y. Mem-ADSVM: A two-layer multi-label predictor for identifying multi-functional types of membrane proteins. *J. Theor. Biol.* **2016**, *398*, 32–42. [[CrossRef](#)] [[PubMed](#)]
39. Wan, S.; Mak, M.W.; Kung, S.Y. Ensemble linear neighborhood propagation for predicting subchloroplast localization of multi-location proteins. *J. Proteome Res.* **2016**, *15*, 4755–4762. [[CrossRef](#)] [[PubMed](#)]
40. Meiler, J.; Müllerl, M.; Zeidler, A.; Schmäschke, F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *J. Mol. Model.* **2001**, *7*, 360–369. [[CrossRef](#)]
41. Quan, L.; Lv, Q.; Zhang, Y. STRUM: Structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics* **2016**, *32*, 2936. [[CrossRef](#)] [[PubMed](#)]
42. Bowie, J.U.; Luthy, R.; Eisenberg, D. A Method to Identify Protein Sequences that Fold into a Known Three-Dimensional Structure. *Science* **1991**, *253*, 164. [[CrossRef](#)] [[PubMed](#)]
43. Wu, W.; Wang, Z.; Cong, P.; Li, T. Accurate prediction of protein relative solvent accessibility using a balanced model. *Biodata Min.* **2017**, *10*, 1. [[CrossRef](#)] [[PubMed](#)]
44. Kabsch, W.; Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **1983**, *22*, 2577–2637. [[CrossRef](#)] [[PubMed](#)]
45. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
46. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An Empirical Exploration of Recurrent Network Architectures. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 171–180.
47. Li, Z.; Yu, Y. Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks. In Proceedings of the 25th International Joint Conference on Artificial Intelligence(IJCAI), New York, NY, USA, 9–15 July 2016; pp. 2560–2567.
48. Wan, F.; Zeng, J. Deep learning with feature embedding for compound-protein interaction prediction. *bioRxiv* **2016**. [[CrossRef](#)]
49. Zhou, J.; Troyanskaya, O.G. Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction. In Proceedings of the 31st International Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014; pp. 745–753.
50. Petersen, B.; Petersen, T.N.; Andersen, P.; Nielsen, M.; Lundegaard, C. A generic method for assignment of reliability scores applied to solvent accessibility predictions. *BMC Struct. Biol.* **2009**, *9*, 51. [[CrossRef](#)] [[PubMed](#)]
51. Chen, J.; Chaudhari, N.S. Cascaded Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2007**, *4*, 572–582. [[CrossRef](#)] [[PubMed](#)]
52. Wang, G.; Dunbrack, R.L., Jr. PISCES: A protein sequence culling server. *Bioinformatics* **2003**, *19*, 1589–1591. [[CrossRef](#)] [[PubMed](#)]
53. Li, W.; Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **2006**, *22*, 1658. [[CrossRef](#)] [[PubMed](#)]
54. Cuff, J.; Barton, G. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* **2000**, *40*, 502–511. [[CrossRef](#)]
55. Altschul, S.F.; Gertz, E.M.; Agarwala, R.; Schaäffer, A.A.; Yu, Y.K. PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic Acids Res.* **2009**, *37*, 815–824. [[CrossRef](#)] [[PubMed](#)]

56. Nan, L.; Zhonghua, S.; Fan, J. Prediction of protein-protein binding site by using core interface residue and support vector machine. *BMC Bioinform.* **2009**, *9*, 553.
57. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
58. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735. [[CrossRef](#)] [[PubMed](#)]
59. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2014; pp. 273–278.
60. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
61. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv* **2016**, arXiv:1605.02688.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).