Li et al. BMC Bioinformatics (2017) 18:443
DOI 10.1186/s12859-017-1842-2

BMC Bioinformatics

RESEARCH ARTICLE Open Access

Protein remote homology detection based on bidirectional long short-term memory

Shumin Li, Junjie Chen and Bin Liu*

Abstract

Background: Protein remote homology detection plays a vital role in studies of protein structures and functions. Almost all of the traditional machine leaning methods require fixed length features to represent the protein sequences. However, it is never an easy task to extract the discriminative features with limited knowledge of proteins. On the other hand, deep learning technique has demonstrated its advantage in automatically learning representations. It is worthwhile to explore the applications of deep learning techniques to the protein remote homology detection.

Results: In this study, we employ the Bidirectional Long Short-Term Memory (BLSTM) to learn effective features from pseudo proteins, also propose a predictor called **ProDec-BLSTM**: it includes input layer, bidirectional LSTM, time distributed dense layer and output layer. This neural network can automatically extract the discriminative features by using bidirectional LSTM and the time distributed dense layer.

Conclusion: Experimental results on a widely-used benchmark dataset show that **ProDec-BLSTM** outperforms other related methods in terms of both the mean ROC and mean ROC50 scores. This promising result shows that **ProDec-BLSTM** is a useful tool for protein remote homology detection. Furthermore, the hidden patterns learnt by **ProDec-BLSTM** can be interpreted and visualized, and therefore, additional useful information can be obtained.

Keywords: Protein sequence analysis, Protein remote homology detection, Neural network, Bidirectional Long Short-Term Memory

Background

Protein remote protein homology detection plays a vital role in the field of bioinformatics since remote homologous proteins share similar structures and functions, which is critical for the studies of protein 3D structure and function [1, 2]. Unfortunately, because of their low protein sequence similarities, the performance of predictors is still too low to be applied to real world applications [3]. During the past decades, some powerful computational methods have been proposed to deal with this problem. The earliest and most widely used methods are alignment-based approaches, including sequence alignment [4–8], profile alignment [9–14] and HMM alignment [15–17]. Later, discriminative methods have been proposed, which treat protein remote homology protein detection as a superfamily level classification task. These methods take the advantages of machine learning algorithms by using both positive and negative samples to train a classifier [18, 19]. A key of these methods is to find an effective representation of proteins. In this regard, several feature extraction methods have been proposed, for example, Top-n-gram extracted the evolutionary information from the profiles [20], Thomas Lingner proposed an approach to incorporate the distances between short oligomers [21], and some methods incorporated physicochemical properties of amino acids into the feature vector representation, such as SVM-RQA [22], SVM-PCD [23], SVM-PDT [24], disPseAAC [25]. Kernel tricks are also employed in discriminative methods, which are used to measure the similarity between protein pairs [26]. Several kernels have been proposed to calculate the similarity between protein samples, such as mismatch kernel [27], motif kernel [28], LA kernel [29], SW-PSSM [30], SVM-Pairwise [31], etc. For more information of these methods, please refer to a recent review paper [1].

* Correspondence: bliu@hit.edu.cn
School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, HIT Campus Shenzhen University Town, Xili, Shenzhen 518055, China

The aforementioned methods have obviously facilitated the development of this important field. However, further studies are still required. Almost all the machine learning methods require fixed length vectors as inputs. Nevertheless, the lengths of protein sequences vary significantly. During the vectorization process, the sequence-order information and the position dependency effects are lost, and this information is critical for protein sequence analysis and nucleic acid analysis [32–34]. Although some studies attempted to incorporate this information into the predictors [21, 24, 35, 36], it is never an easy task due to the limited knowledge of proteins.

Recently, deep learning techniques have demonstrated their ability for improving the discriminative power compared with other machine learning methods [37, 38], and have been widely applied to the field of bioinformatics [39], such as the estimation of protein model quality [40], protein structure prediction [41–43], protein disorder prediction [44], etc. Recurrent Neural Network (RNN) is one of the most successful deep learning techniques, which is designed to utilize sequential information of input data with cyclic connections among building blocks, such as Long Short-Term Memory (LSTM) [45, 46], and gated recurrent units (GRUs) [47]. LSTM can automatically detect the long-terms and short-terms dependency relationships in protein sequences, and decides how to process a current subsequence according to the information extracted from the prior subsequences [48]. LSTM has also been applied to protein remote homology detection to automatically to generate the representation of proteins [48]. Compared with other methods, it is able to identify effective patterns of protein sequences. Although this approach has achieved state-of-the-art performance, it has several shortcomings: 1) Hochreiter's neural network [48] only has two layers: LSTM and output layer. Its capacity is too limited to capture sequence-order effects, especially for the long proteins; 2) Features are generated only based on the last output of LSTM. However, as protein sequences contains hundreds of amino acids, it is hard to detect the dependency relationships of all the subsequences by only considering information contained in the last output of LSTM; 3) The last output generated from LSTM contains complex dependencies, which cannot be traced to any specific subsequence for further analysis.

Here, we are to propose a computational predictor for protein remote homology detection based on Bidirectional Long Short-Term Memory [45, 46, 49], called **ProDec-BLSTM**, to address the aforementioned disadvantages of the existing methods in this field. **ProDec-BLSTM** consisted of input layer, bidirectional LSTM layer, time distributed dense layer and output layer. With this neural network, both the long and short dependency information of pseudo proteins can be captured by tapping the information from every mediate hidden value of bidirectional LSTM. Experimental results on a widely used benchmark dataset and an updated independent dataset show that **ProDec-BLSTM** outperforms other existing methods. Furthermore, the patterns learnt by **ProDec-BLSTM** can be interpreted and visualized, providing additional information for further analysis.

Methods

SCOP benchmark dataset

A widely used benchmark dataset has been used to evaluate the performance of various methods [28], which was constructed based on the SCOP database [50] by Hochreiter [48]. This dataset can be accessed from http://www.bioinf.jku.at/software/LSTM_protein/.

The SCOP database [50] classifies the protein sequences into a hierarchy structure, whose levels from top to bottom are class, fold, superfamily, and family. 4019 proteins sequences are extracted from SCOP database, whose identities are lower than 95%, and they are divided into 102 families and 52 superfamilies. For each family, there are at least 10 positive samples. For the 102 families in the database, the training and testing datasets are defined as:

$$\begin{cases} \mathbb{S}_{\text{train}}(k) = \mathbb{S}^{+}_{\text{train}}(k) \cup \mathrm{E}^{+}_{\text{train}}(k) \cup \mathbb{S}^{-}_{\text{train}}(k) \\ \mathbb{S}_{\text{test}}(k) = \mathbb{S}^{+}_{\text{test}}(k) \cup \mathbb{S}^{-}_{\text{test}}(k) \end{cases} \quad (k = 1, 2, \ldots, 102) \tag{1}$$

where $\mathbb{S}^{+}_{\text{test}}(k)$ represents the k^{th} positive testing dataset with proteins in k^{th} family, and $\mathbb{S}^{+}_{\text{train}}(k)$ represents the k^{th} positive training dataset containing proteins in the same superfamily and not in the k^{th} family. $\mathrm{E}^{+}_{\text{train}}(k)$ denotes the extended positive training dataset for k^{th} training dataset. The added training samples are extracted from Uniref50 [51] by using PSI-BLAST [9] with default parameters except that the e-value was set as 10.0. For all of the superfamilies except which k^{th} family belongs to, select one family in each of the superfamilies respectively, to form the k^{th} negative testing dataset $\mathbb{S}^{-}_{\text{test}}(k)$ and the rest of proteins in these superfamilies are included in the negative training dataset $\mathbb{S}^{-}_{\text{train}}(k)$. The average number of samples of all the 102 training datasets is 9077.

Neural network architectures based on bidirectional LSTM

In this section, we will introduce the network architecture of **ProDec-BLSTM**, as shown in Fig. 1. This network has four layers: input layer, bidirectional LSTM layer, time distributed dense layer, and output layer. The input layer is designed to encode the pseudo protein by one-hot encoding [52].Bidirectional LSTM extracts the

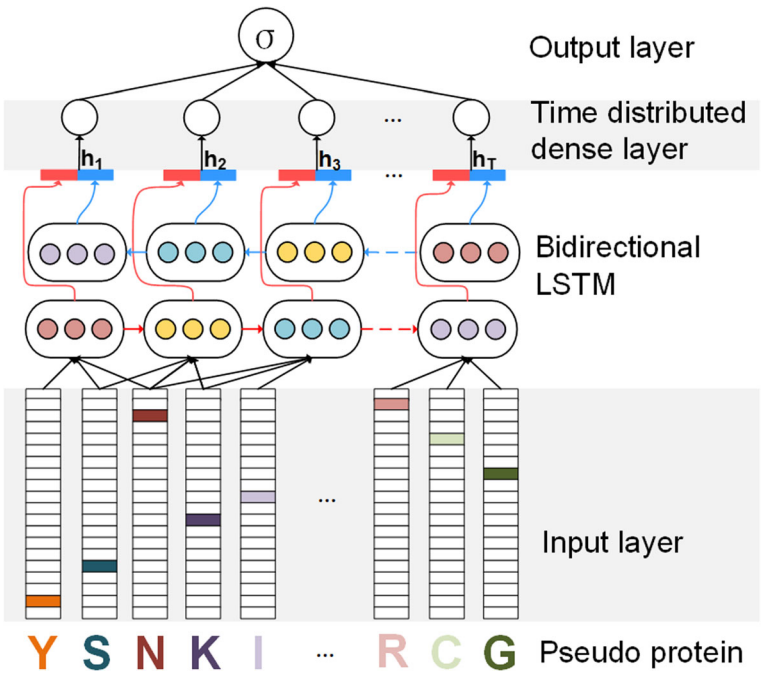

Fig. 1 The structure of ProDec-BLSTM. The input layer converts the pseudo proteins into feature vectors by one-hot encoding. Next, the subsequences within the sliding window are fed into the bidirectional LSTM layer for extracting the sequence patterns. Then, the time distributed dense layer weights the extracted patterns. Finally, the extracted feature vectors are fed into output layer for prediction

dependency relationships between subsequences. We take the advantages of every intermediate hidden value from bidirectional LSTM to better handling the long length of protein sequences. More comprehensive dependency information can be included into the hidden values by using bidirectional LSTM. Then, those intermediate hidden values are connected to the time distributed dense layer. Because memory cells in one block extract different levels of dependency information, the time distributed dense layer is designed to weight the dependency relationships extracted from different cells. The outputs of time distributed dense layer are concatenated into one feature vector and be fed into the output layer for prediction. Next, we will introduce the four layers in more details.

Input layer

The input layer transfers the protein sequence into a representing matrix, and fed it into the bidirectional LSTM layer.

Given a protein sequence **P**:

$$\mathbf{P} = \mathrm{R}_1, \mathrm{R}_2, ..., \mathrm{R}_l \tag{2}$$

where R_1 denotes the 1st residue, R_2 denotes the 2nd residue and so forth, l represents the length of **P**. Then the **P** is converted into pseudo protein **P′′** based on PSSM [26, 53] generated by PSI-BLAST with command line "-evalue 0.001 -num_iterations 3".

The input matrix at the t^{th} time step can be obtained by one-hot encoding of **P′′** [52], shown as:

$$\mathbf{M}_t = (\mathbf{v}_i, \mathbf{v}_{i+1}, ..., \mathbf{v}_{i+w-1}) \tag{3}$$

$$\mathbf{v}_i = (e_{i1}, e_{i2}, ..., e_{i20})^T, e_{ij} = \begin{cases} 1, \mathrm{R}_i = \mathrm{AA}_j \\ 0, \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{v}_i$ is the representing vector for R_i, w denotes the size of the sliding window, i represents the start position of the subsequence, AA_j denotes the j^{th} standard amino acid.

Bidirectional LSTM Layer

Bidirectional LSTM layer is the most important part in **ProDec-BLSTM**, aiming to extract the sequence patterns from pseudo proteins. The basic unit of LSTM is the memory cell. In this study, we adopted the memory cell described in [46], whose structure is shown in Fig. 2. The memory cell receives two input streams: the subsequence within the sliding window, and the output of LSTM from the last time step. Based on the two information streams, the three gates coordinate with each other to update and output the cell state. The input gate controls how much of new information can flow into the cell; The forget gate decides how much stored information in the cell will be kept. By coordination of input gate and forget gate, the cell state is updated. The output

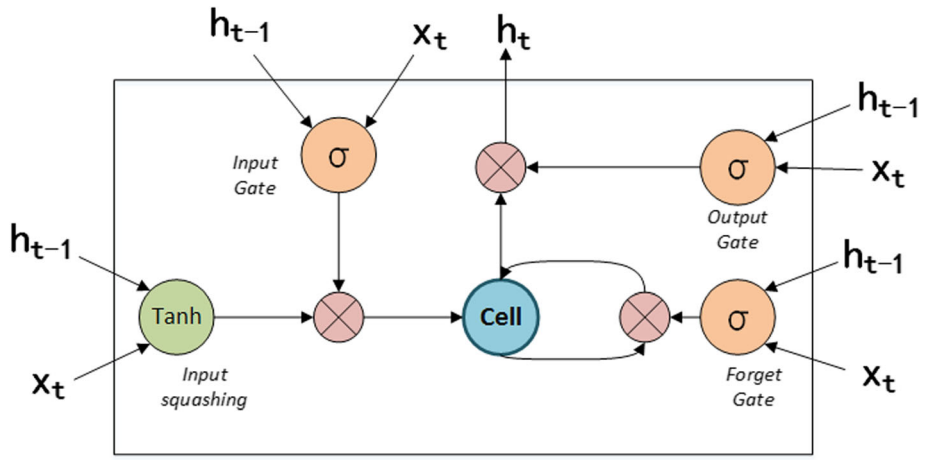

Fig. 2 The structure of LSTM memory cell. There are three gates, including input gate (marked as *i*), forget gate (marked as *f*), output gate (marked as *o*), to control the information stream flowing in and out the block. *σ* denotes the sigmoid function, which produces a value bounded by 0 and 1. The internal cell state is maintained and updated by the coordination of input gate and forget gate. The output gate controls outputting information stored in the cell. h is the output of the memory cell, x is representing matrix of the input subsequence and *t* mean the t^{th} time step

gate controls outputting the information stored in the cell, which is hidden value (denoted as $\mathbf{h}_t$ in Fig. 2).

The bidirectional LSTM is made up of two reversed unidirectional LSTM. To handle the long pseudo protein sequences, and better capture the dependency information of subsequences, we tap into all of the intermediate hidden values generated by bidirectional LSTM. The hidden values generated by the forward LSTM and backward LSTM for the same input subsequence are concatenated into a vector, which is shown in Eq. (5).

$$\mathbf{h}_t = \left(\mathbf{h}_t^{\mathrm{f}}, \mathbf{h}_t^{\mathrm{b}}\right) \tag{5}$$

where $\mathbf{h}$ is hidden value, f represents the forward LSTM, b represents the backward LSTM, t means the t^{th} time step.

In the bidirectional LSTM layer, the pseudo protein is processed N-terminus to C-terminus and C-terminus to N-terminus simultaneously. Therefore, $\mathbf{h}_t^{\mathrm{f}}$ contains dependencies between the target subsequence and its left neighbouring subsequence. $\mathbf{h}_t^{\mathrm{b}}$ contains dependencies between the target subsequence and its right neighbouring subsequence. These two dependency relationships are concatenated into one vector $\mathbf{h}_t$, which can be interpreted as the feature of the target subsequence. Therefore, more comprehensive dependencies can be included into the intermediate hidden values by using bidirectional LSTM.

Time distributed dense layer

Different memory cells in one block extracts different levels of dependency relationships. Thus, we add the time distributed dense layer after the bidirectional LSTM layer to give weights to the hidden values generated from different memory cells. The time distributed dense receives the hidden value generated from memory block, and outputs a single value for one subsequence. The outputs of time distributed dense layer at every position are then concatenated into one vector, which is fed into the output layer for prediction.

Output layer

The output layer is a fully connected network with one node and it performs the binary prediction based on the representing vectors generated by the time distributed dense layer. Therefore, for each protein, its probability of belonging to a specific superfamily is produced.

Implementation details

This network was implemented by using Keras 2.0.6 (https://github.com/fchollet/keras) with the backend of Theano (0.9.0) [54].

The size of the sliding window was set as 3, and the protein sequence length was fixed as 400. The bidirectional LSTM has 50 memory cells in one block. The time distributed fully dense layer was a fully connected layer with the one output node, using ReLu activation function [55]. All the initializations of weights and bias were set as the default in Keras. The model was optimized by the algorithm of RMSprop [56] with the loss function of binary crossentropy at learning rate 0.01. The batch size was 32. Dropout [57] was included in bidirectional LSTM layer and the proportion of disconnection was 0.2. Each model was optimized by training for 150 epochs.

Performance measure

In this study, ROC score and ROC50 score are used to evaluate the performance of various methods. Receiver

operating characteristics (ROC) curve is plotted by using the true positive rate as the x axis and the false positive rate as the y axis, which are calculated based on different classification threshold [58]. ROC score refers to the normalized area under ROC curve. ROC50 is the normalized area when the first 50 false positive samples occur. For a perfect classification, ROC score and ROC50 are equal to 1.

Results and discussion

Comparison with various methods

We compared **ProDec-BLSTM** with various related methods, including GPkernel [28], GPextended [28], GPboost [28], SVM-Pairwise [31], Mismatch [27], eMOTIF [59], LA-kernel [29], PSI-BLAST [9] and LSTM [48]. The results are shown in Table 1, from which we can see that **ProDec-BLSTM** outperforms all of other methods. Both **ProDec-BLSTM** and LSTM [48] are based on deep learning techniques with smart representation of proteins, and all the other approaches are based on Support Vector Machines (SVMs). These results indicate that the LSTM method is a suitable approach for protein remote homology detection. As discussed above, the SVM-based methods rely on the quality of hand-made features and kernel tricks. However, due to the imited knowledges of proteins, their discriminative power is still low. In contrast, the deep learning algorithms, especially LSTM are able to automatically extract the features from proteins sequences, and capture the sequence-order effects. The t-test is employed to measure the differences between **ProDec-BLSTM** and LSTM [48]. The results show that **ProDec-BLSTM** significantly outperforms LSTM [48] in terms of ROC scores (P-value = 0.05) and ROC50 scores (P-value = 3.04e-09). There are four main reasons for **ProDec-BLSTM** outperforms LSTM: 1) **ProDec-BLSTM** taps into all of the intermediate hidden values generated by bidirectional LSTM to better handle the long proteins and pay attention to local as well as global dependencies; 2) **ProDec-BLSTM** used bidirectional LSTM layer which is able to include the dependency information from both N-terminal to C-terminal and from C-terminal to N-terminal into the intermediate hidden values; 3) the time distributed dense layer gives weights to different levels of dependency information to fuse information. 4) Evolutionary information extracted from PSSMs is incorporated into the predictor by using pseudo proteins.

Table 1 Mean ROC and ROC50 scores of various methods on the SCOP benchmark dataset (Eq. 1)

Methods	Mean ROC	Mean ROC50	classifier
GPkernel	0.902	0.591	SVM
GPextended	0.869	0.542	SVM
GPboost	0.797	0.375	SVM
SVM-Pairwise	0.849	0.555	SVM
Mismatch	0.878	0.543	SVM
eMOTIF	0.857	0.551	SVM
LA-kernel	0.919	0.686	SVM
PSI-BLAST	0.575	0.175	NA
LSTM	0.943	0.735	LSTM
ProDec-BLSTM	0.969	0.849	LSTM

Visualizations

The hidden patterns learnt by **ProDec-BLSTM** can be interpreted and visualized. We explore the reason why the proposed **ProDec-BLSTM** showed higher discriminative power based on the visualization of hidden patterns.

Given a pseudo protein $\mathbf{P}'$, it can be converted into a feature vector:

$$\mathbf{V} = [\alpha_1, \alpha_2, ..., \alpha_t] \tag{6}$$

where α_t indicates the output of time distributed dense layer at the t^{th} time step. The feature vector $\mathbf{V}$ is generated by concatenating all the outputs of time distributed dense layer and each value of $\mathbf{V}$ represents the fused dependency relationships of a subsequence. Thus, $\mathbf{V}$ contains global sequence characteristics.

Here, we demonstrate the testing set of the family *b.1.1.1* in SCOP benchmark dataset (**Eq. 1**), which has 538 positive samples and 543 negative samples, as an example: the representing vector of each sample are generated by the trained **ProDec-BLSTM** model, and then t-SNE [60] is employed to reduce the their dimensions into two in order to visualize their distributions (shown in Fig. 3). The ranges of x and y axis are both normalized. From Fig. 3, we can see that most of the positive and negative samples are clustered and clearly apart from each other, indicating that the feature vectors automatically generated by **ProDec-BLSTM** are effective for protein remote homology detection.

Independent test on SCOPe dataset

Moreover, as a demonstration, we also extend the comparison with other methods via an updated independent dataset set constructed based on SCOPe (latest version: 2.06) [61]. To avoid the homology bias, the CD-HIT [62] is used to remove those proteins from SCOPe that have more than 95% sequence identity to any protein in the SCOP benchmark dataset (**Eq. 1**). Finally, 4679 proteins in SCOPe are obtained using as the independent dataset (see Additional file 1). Trained with SCOP benchmark dataset, **ProDec-BLSTM** predictor is used to identify the proteins in the SCOPe independent dataset set. Four

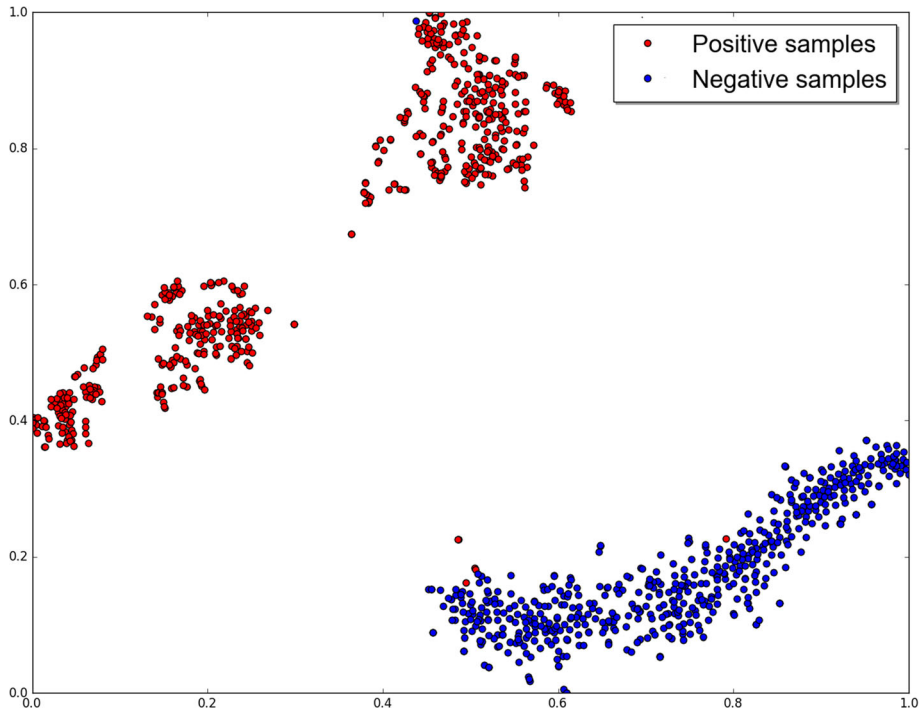

Fig. 3 Feature visualization of ProDec-BLSTM for the protein family *b.1.1.1*. The positive samples and negative samples are shown in red color and blue color, respectively

related methods are compared with **ProDec-BLSTM**, including HHblits [16], Hmmer [15], PSI-BLAST [9] and ProDec-LTR [3, 63]. HHblits and PSI-BLAST are employed in the top-performing methods in CASP [64] and ProDec-LTR [3] is a recent method that combines different alignment-based methods. The results thus obtained are given in Table 2**,** and their implementations are listed below. It can be clearly seen from there that the new predictor outperforms all the existing approaches for protein remote homology detection.

Conclusion

In this study, we propose a predictor **ProDec-BLSTM** based on bidirectional LSTM for protein remote homology detection, which can automatically extract the discriminative features and capture sequence-order effects. Experimental results showed that **ProDec-BLSTM** achieved the top performance comparing with other existing methods on an SCOP benchmark dataset and a SCOPe independent dataset. Comparing with handmade protein features used by traditional machine learning methods, the features learnt by **ProDec-BLSTM** have more discriminative power.

Such high performance of **ProDec-BLSTM** benefits from bidirectional LSTM, and time distributed dense layer, by which it is able to extract the global and local sequence order effects. Every intermediate hidden values of bidirectional LSTM are also incorporated into the proposed predictor so as to capture context dependency information of subsequences. The time distributed dense layer gives weights to different level of dependency relationships, and fuses the dependency information.

In the future, we will focus on exploring new features to further improve the performance of **ProDec-BLSTM**, such as directly learning from PSSM [65].

Table 2 Mean ROC and ROC50 scores of related methods on the SCOPe independent dataset

Method	Mean ROC	Mean ROC50
HHblits[a]	0.725	0.443
Hmmer[b]	0.556	0.145
PSI-BLAST[c]	0.668	0.096
ProtDec-LTR[d]	0.742	0.445
ProDec-BLSTM	0.970	0.714

[a]the command line of HHblits is '-e 1 -p 0 -E inf -Z 10000 -B 10000 -b 10000'
[b]The parameters of Hmmer are set as default
[c]The paramters of PSI-BLAST are set as default
[d]The above three alignment-based methods are combined by ProDec-LTR. The model is trained with SCOP benchmark dataset (Eq. 1)

Additional files

Additional file 1: The SCOP ID of the independent SCOPe testing dataset. (PDF 7601 kb)

Additional file 2: The source code and its document of ProDec-BLSTM. (ZIP 316 kb)

Abbreviations
GRU: Recurrent gated unit; HMM: Hidden Markov model; LSTM: Long-Short Term Memory; ReLu: Rectified Linear Units; RMSProp: Root Mean Square Propagation; RNN: Recurrent neural network; ROC: Receiving operating characteristics; SVM: Support vector machine

Acknowledgements
Not applicable.

Funding
This work was supported by the National Natural Science Foundation of China (No. 61672184), the Natural Science Foundation of Guangdong Province (2014A030313695), Guangdong Natural Science Funds for Distinguished Young Scholars (2016A030306008), Scientific Research Foundation in Shenzhen (Grant No. JCYJ20150626110425228, JCYJ20170307152201596), and Guangdong Special Support Program of Technology Young talents (2016TQ03X618). The funding bodies do not play any role in the design or conclusion of the study.

Availability of data and materials
The SCOP benchmark dataset used in this study was published in [48], which is available on http://www.bioinf.jku.at/software/LSTM_protein/. The SCOPe independent dataset is listed in Additional file 1. The source code of **ProDec-BLSTM** and its document are in Additional file 2.

Authors' contributions
SML carried out remote homology detection studies, participated in coding and drafting the manuscript. BL conceived of this study, and participated in writing this manuscript. BL and JJC and participated in the design of the study and performed statistical analysis. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 19 July 2017 Accepted: 21 September 2017
Published online: 10 October 2017

References

1. Chen J, Guo M, Wang X, Liu B. A comprehensive review and comparison of different computational methods for protein remote homology detection. Brief Bioinform. 2016. https://doi.org/10.1093/bib/bbw108.
2. Wei L, Zou Q. Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition. Int J Mol Sci. 2016;17(12):2118.
3. Liu B, Chen J, Wang X. Application of learning to rank to protein remote homology detection. Bioinformatics. 2015;31(21):3492–8.
4. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970; 48(3):443–53.
5. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981;147(1):195–7.
6. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
7. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci. 1988;85(8):2444–8.
8. Zou Q, Hu Q, Guo M, Wang G. HAlign: Fast Multiple Similar DNA/RNA Sequence Alignment Based on the Centre Star Strategy. Bioinformatics. 2015;31(15):2475–81.
9. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402.
10. Margelevicius M, Laganeckas M, Venclovas C. COMA server for protein distant homology search. Bioinformatics. 2010;26(15):1905–6.
11. Jaroszewski L, Li Z, Cai XH, Weber C, Godzik A. FFAS server: novel features and applications. Nucleic Acids Res. 2011;39(Web Server issue):W38–44.
12. Sadreyev R, Grishin N. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. J Mol Biol. 2003; 326(1):317–36.
13. Yang Y, Faraggi E, Zhao H, 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. 2011;27(15): 2076–82.
14. Yan K, Xu Y, Fang X, Zheng C, Liu B. Protein fold recognition based on sparse representation based classification. Artif Intell Med. 2017;79:1–8.
15. Finn RD, Clements J, Eddy SR. HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. 2011;39(Web Server issue):W29–37.
16. Remmert M, Biegert A, Hauser A, Soding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods. 2011; 9(2):173–5.
17. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote protein homologies. Bioinformatics. 1998;14(10):846–56.
18. Wei L, Liao M, Gao X, Zou Q. Enhanced Protein Fold Prediction Method through a Novel Feature Extraction Technique. IEEE Trans Nanobiosci. 2015; 14(6):649–59.
19. Zhao X, Zou Q, Liu B, Liu X. Exploratory predicting protein folding model with random forest and hybrid features. Curr Proteomics. 2014;11(4):289–99.
20. Liu B, Wang X, Lin L, Dong Q, Wang X. A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. BMC Bioinform. 2008;9(1):510.
21. Lingner T, Meinicke P. Remote homology detection based on oligomer distances. Bioinformatics. 2006;22(18):2224–31.
22. Yang Y, Tantoso E, Li K-B. Remote protein homology detection using recurrence quantification analysis and amino acid physicochemical properties. J Theor Biol. 2008;252(1):145–54.
23. Webb-Robertson B-JM, Ratuiste KG, Oehmen CS. Physicochemical property distributions for accurate and rapid pairwise protein homology detection. BMC Bioinform. 2010;11(1):145.
24. Liu B, Wang X, Chen Q, Dong Q, Lan X. Using amino acid physicochemical distance transformation for fast protein remote homology detection. PLoS One. 2012;7(9):e46633.
25. Liu B, Chen J, Wang X. Protein remote homology detection by combining Chou's distance-pair pseudo amino acid composition and principal component analysis. Mol Gen Genomics. 2015;290(5):1919–31.
26. Liu B, Zhang D, Xu R, Xu J, Wang X, Chen Q, Dong Q, Chou KC. Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. Bioinformatics. 2014; 30(4):472–9.
27. Leslie CS, Eskin E, Cohen A, Weston J, Noble WS. Mismatch string kernels for discriminative protein classification. Bioinformatics. 2004;20(4):467–76.
28. Håndstad T, Hestnes AJ, Sætrom P. Motif kernel generated by genetic programming improves remote homology and fold detection. BMC Bioinform. 2007;8(1):23.
29. Saigo H, Vert JP, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics. 2004;20(11):1682–9.
30. Rangwala H, Karypis G. Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics. 2005;21(23):4239–47.
31. Liao L, Noble WS. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. J Comput Biol. 2003;10(6):857–68.
32. Liu B, Liu F, Wang X, Chen J, Fang L, Chou KC. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res. 2015;43(W1):W65–71.
33. Liu B, Fang L, Liu F, Wang X, Chou KC. iMiRNA-PseDPC: microRNA precursor identification with a pseudo distance-pair composition approach. J Biomol Struct Dyn. 2016;34(1):220–32.
34. Liu B, Liu F, Fang L, Wang X, Chou KC. repRNA: a web server for generating various feature vectors of RNA sequences. Mol Gen Genom. 2016;291(1): 473–81.
35. Liu B, Chen J, Wang X. Protein remote homology detection by combining Chou's distance-pair pseudo amino acid composition and principal component analysis. Mol Gen Genom. 2015;290(5):1919–31.

36. Deng S-P, Huang DS. SFAPS: An R package for structure/function analysis of protein sequences based on informational spectrum method. Methods. 2014;69(3):207–12.
37. Huang DS. A constructive approach for finding arbitrary roots of polynomials by neural networks. IEEE Trans Neural Netw. 2004;15(2):477–91.
38. Zhao ZQ, Huang DS, Sun BY. Human face recognition based on multi-features using neural networks committee. Pattern Recogn Lett. 2004;25(12):1351–8.
39. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
40. Cao R, Bhattacharya D, Hou J, Cheng J. DeepQA: improving the estimation of single protein model quality with deep belief networks. BMC Bioinform. 2016;17(1):495.
41. Yang Y, Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Zhou Y. SPIDER2: A Package to Predict Secondary Structure, Accessible Surface Area, and Main-Chain Torsional Angles by Deep Neural Networks. Methods Mol. 2017;1484:55–63.
42. Heffernan R, Yang Y, Paliwal K, Zhou Y. Capturing Non-Local Interactions by Long Short Term Memory Bidirectional Recurrent Neural Networks for Improving Prediction of Protein Secondary Structure, Backbone Angles, Contact Numbers, and Solvent Accessibility. Bioinformatics. 2017. doi: 10.1093/bioinformatics/btx218.
43. Wang S, Peng J, Ma J, Xu J. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. Sci Rep. 2016;6:18962.
44. Hanson J, Yang Y, Paliwal K, Zhou Y. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. Bioinformatics. 2017;33(5):685–92.
45. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9(8):1735–80.
46. Gers FA, Schmidhuber J, Cummins F. Learning to Forget: Continual Prediction with LSTM. Neural Comput. 2000;12(10):2451–71.
47. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:14123555 2014.
48. Hochreiter S, Heusel M, Obermayer K. Fast model-based protein homology detection without alignment. Bioinformatics. 2007;23(14):1728–36.
49. Graves A, Fernández S, Schmidhuber J. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In: Duch W, Kacprzyk J, Oja E, Zadrożny S, editors. Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005: 15th International Conference, Warsaw, Poland, September 11–15, 2005 Proceedings, Part II. Berlin: Springer Berlin Heidelberg; 2005. p. 799–804.
50. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol. 1995;247(4):536–40.
51. Suzek BE, Huang H, McGarvey P, Mazumder R, Wu CH. UniRef: comprehensive and non-redundant UniProt reference clusters. Bioinformatics. 2007;23(10):1282–8.
52. Quang D, Xie X. Dan Q. a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic Acids Res. 2016;44(11):e107.
53. Chen J, Long R, Wang X, Liu B, Chou K-C. dRHP-PseRA: detecting remote homology proteins using profile based pseudo protein sequence and rank aggregation. Sci Rep. 2016;6:32333.
54. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y. Theano: A CPU and GPU Math Compiler in Python. Proceedings of the 9th Python in Science Conference. Austin, Texas. 2010:3–10.
55. Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks. In: 14th International Conference on Artificial Intelligence and Statistics: 2011; 2011. p. 315–23.
56. Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning. 2012;4(2):26-31.
57. 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(6):1929–58.
58. Gribskov M, Robinson NL. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. Comput Chem. 1996;20(1):25–33.
59. Ben-Hur A, Brutlag D. Remote homology detection: a motif based approach. Bioinformatics. 2003;19(suppl 1):i26–33.
60. van der Maaten L, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res. 2008;9(11):2579–605.
61. Chandonia JM, Fox NK, Brenner SE. SCOPe: Manual Curation and Artifact Removal in the Structural Classification of Proteins – extended Database. J Mol Biol. 2017;429(3):348–55.
62. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006;22(13):1658–9.
63. Chen J, Guo M, Li S, Liu B. ProtDec-LTR2.0: An improved method for protein remote homology detection by combining pseudo protein and supervised Learning to Rank. Bioinformaitcs. https://doi.org/10.1093/bioinformatics/btx429.
64. Moult J. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol. 2005;15(3):285–9.
65. Wang B, Chen P, Huang DS, Li JJ, Lok TM, Lyu MR. Predicting protein interaction sites from residue spatial sequence profile and evolution rate. FEBS Lett. 2006;580(2):380–4.