Bioinformatics, 31(21), 2015, 3492–3498
doi: 10.1093/bioinformatics/btv413
Advance Access Publication Date: 10 July 2015
Original Paper

OXFORD

Sequence analysis

Application of learning to rank to protein remote homology detection

Bin Liu[1,2,3,*], Junjie Chen[1] and Xiaolong Wang[1,2]

[1]School of Computer Science and Technology, [2]Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China and [3]Gordon Life Science Institute, Belmont, MA 02478, USA

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on June 4, 2015; revised on July 3, 2015; accepted on July 7, 2015

Abstract

Motivation: Protein remote homology detection is one of the fundamental problems in computational biology, aiming to find protein sequences in a database of known structures that are evolutionarily related to a given query protein. Some computational methods treat this problem as a ranking problem and achieve the state-of-the-art performance, such as PSI-BLAST, HHblits and ProtEmbed. This raises the possibility to combine these methods to improve the predictive performance. In this regard, we are to propose a new computational method called ProtDec-LTR for protein remote homology detection, which is able to combine various ranking methods in a supervised manner via using the Learning to Rank (LTR) algorithm derived from natural language processing.

Results: Experimental results on a widely used benchmark dataset showed that ProtDec-LTR can achieve an ROC1 score of 0.8442 and an ROC50 score of 0.9023 outperforming all the individual predictors and some state-of-the-art methods. These results indicate that it is correct to treat protein remote homology detection as a ranking problem, and predictive performance improvement can be achieved by combining different ranking approaches in a supervised manner via using LTR.

Availability and implementation: For users' convenience, the software tools of three basic ranking predictors and Learning to Rank algorithm were provided at http://bioinformatics.hitsz.edu.cn/ProtDec-LTR/home/

Contact: bliu@insun.hit.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Using sequence similarity between protein pairs to detect evolutionary relationships is one of the central tasks in bioinformatics, which can be applied to the protein 3D structure and function prediction (Bork and Koonin, 1998). Unfortunately, remote homology protein pairs have similar structures and functions, but they lack easily detectable sequence similarity, because the protein tertiary structure is more conserved than protein sequence. Therefore, it is often difficult to detect protein remote homology by computational approaches.

Some effective computational methods have been developed to address this challenging problem, which can be mainly divided into two groups, including discriminative methods and ranking methods. The first group discriminative methods treat protein remote homology detection as a classification problem using both the positive and negative samples to train the classification models, and then they are used to predict unseen samples. Among this kind of approaches, the methods based on Support Vector Machines (SVMs) achieve the state-of-the-art performance with appropriate kernel functions, which measure the similarity between any

pair of samples (Liu *et al.*, 2014a). These methods employ various features to represent the protein sequences, such as SVM-fisher (Leslie *et al.*, 2002), SVM-PDT (Liu *et al.*, 2012), SVM-pairwise (Muh *et al.*, 2009), SVM-LA (Saigo *et al.*, 2004) and some profile-based methods (Liu *et al.*, 2008, 2013, 2014c, 2015a).

In contract, the second group ranking methods treat protein remote homology detection as a ranking task or database searching task, where the query protein is searched against a protein database with known structures and functions, and a ranking list of the proteins in the database is returned according to their identified evolutionary relationships to the query protein. Early ranking methods were based on sequence alignment algorithms, such as Smith–Waterman algorithm (Smith and Waterman, 1981). Later, some more efficient algorithms were proposed so as to trade reduced accuracy for improved efficiency, such as Basic Local Alignment Search Tool (BLAST) (Altschul *et al.*, 1990) and FASTA (Pearson, 1991). Because the evolutionary relations of remote homology proteins cannot be easily detected by sequence similarity, these methods can only achieve limited performance. The predictive accuracy was significantly improved by using the profile-based alignment algorithms, e.g. PSI-BLAST method (Altschul *et al.*, 1997) iteratively builds a probabilistic profile of a query sequence and therefore a more sensitive sequence comparison score can be calculated. Several ranking methods employed the generative models iteratively trained by using positive samples of a protein family or superfamily and achieved better performance, e.g. HHblits (Remmert *et al.*, 2012) generates a profile hidden Markov model (profile hidden Markov model (HMM)) (Eddy, 1998; Karplus *et al.*, 1998) from the query sequence and iteratively searches through a large database. Recently, some ranking methods based on graph theory or semantic embedding techniques were proposed, e.g. motivated by the successful applications of Google's PageRank algorithm, an unsupervised algorithm called RankProp (Melvin *et al.*, 2009; Weston *et al.*, 2004) was proposed, which is a network-based inference method. Later, this method was further improved by a semi-supervised approach using labeled samples to learn a new network (Weston *et al.*, 2006). Based on the similarity between protein sequences and natural languages, the techniques derived from the field of natural language processing were applied to protein remote homology, e.g. ProtEmbed (Melvin *et al.*, 2011) converts a large-scale embedding of protein feature vectors into a low-dimensional 'semantic space'. Therefore, evolutionarily related proteins are embedded in close proximity in this 'semantic space'.

The aforementioned ranking methods are based on different theories and achieve the state-of-the-art performance in the field of protein remote homology detection. Therefore, it is interesting to explore whether these approaches can be combined to further improve their performance. However, how to combine these ranking methods into one predictor is a challenging problem, because they are based on different techniques and their predictive results are often different. To address this problem, we employed the Learning to Rank (LTR) algorithm (Li, 2011) to combine different ranking methods in a supervised fashion. LTR is a supervised algorithm for training the model in a ranking task, which has been successfully applied to information retrieval, natural language processing, data mining, etc. (Li, 2011), e.g. LTR is severing as one of the key algorithms in many well-known searching engines, such as Bing (Liu *et al.*, 2007), Yahoo! (Figueroa and Neumann, 2013) and Google (Sculley, 2009).

In this study, we proposed a new computational method called ProtDec-LTR, which combines three state-of-the-art ranking methods by using LTR, including PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011). To our best knowledge, ProtDec-LTR is the first computational predictor that can combine various ranking methods via using a supervised framework.

2 Methods and algorithms

2.1 Method overview

Protein remote homology detection can be viewed as a ranking task or database searching problem. The aim is to search the query protein against a protein database and return the top ranked proteins in the database according to their evolutionary relationships to the query protein. This problem is particularly similar to the document retrieval task. In document retrieval, given a query, the system retrieves documents containing the query words from a collection of documents, ranks the documents and returns the top ranked documents. To apply information retrieval techniques to protein remote homology detection, the basic protein ranking algorithms should be identified. Here, PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011) were selected as the basic ranking algorithms, which are similar as the document retrieval engines in the field of information retrieve. Below is a brief description of the three methods.

PSI-BLAST (Altschul *et al.*, 1997) searches a query protein sequence against a database based on the profile generated from the multiple sequence alignment. In this study, PSI-BLAST version 2.2.29 (Altschul *et al.*, 1997) was employed to search the benchmark dataset with default parameters (*E* value and number of iterations were set as 0.001 and 3, respectively). HHblits (Remmert *et al.*, 2012) is another powerful computational method for protein remote homology detection, which generates a profile HMM from the query sequence and iteratively searches through a database of profile HMMs. HHblits version 2.0.16 (Remmert *et al.*, 2012) was used as the implementation of this method. All its parameters were set to default except that the number of iterations was set as 2. ProtEmbed (Melvin *et al.*, 2011) converts a query protein into a low dimensional 'semantic space' and measures the 'distance' between a query protein against all other proteins. We re-ran the ProtEmbed method (Melvin *et al.*, 2011) on the benchmark dataset with the optimized parameters (Melvin *et al.*, 2011).

The proposed ProtDec-LTR method is able to combine the aforementioned three methods in a supervised manner by using LTR. The flowchart of ProtDec-LTR is shown in Figure 1. Similar as the application of LTR in information retrieve, for protein remote homology detection, each protein sequence is treated as a 'document'. Three ranking lists are obtained by using the three aforementioned ranking methods, and then they are embedded as a feature matrix to train the LTR model. Finally, for an unseen query sample, its homologous proteins can be detected by the trained model of LTR. As a result, the three ranking predictors are combined in a supervised manner considering the advantages of all the three individual predictors for more accurate protein remote homology detection.

2.2 Feature representation

To apply LTR to protein remote homology detection, the protein sequences should be converted into feature matrices based on the three ranking methods, including PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011). Below is the description on how to generate the feature matrix. Given the benchmark dataset with n proteins,

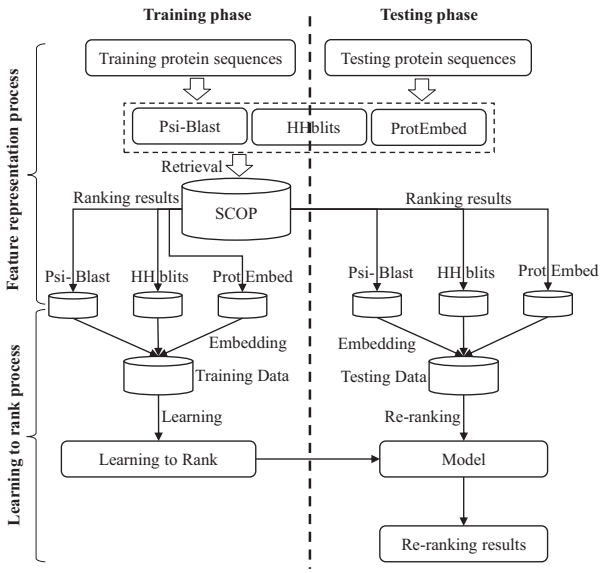

Fig. 1. The flowchart of ProtDec-LTR. There are three main phases in ProtDec-LTR, including feature representation, LTR training and testing phases. The proteins are first embedded into feature matrices constructed based on three basic ranking methods: PSI-BLAST, HHblits and ProtEmbed, and then they are fed into LTR for training the model. For an unseen protein, its homologous proteins can be detected by the trained model. Therefore, these three methods are combined in a supervised framework by using LTR

for a given protein q, its feature matrix can be represented as a matrix $\mathbf{\Pi}$:

$$\mathbf{\Pi} = \begin{bmatrix} s_1(q,p_1) & s_2(q,p_1) & \cdots & s_8(q,p_1) \\ s_1(q,p_2) & s_2(q,p_2) & \cdots & s_8(q,p_2) \\ \vdots & \vdots & \vdots & \vdots \\ s_1(q,p_n) & s_2(q,p_n) & \cdots & s_8(q,p_n) \end{bmatrix} \quad (1)$$

where $p_i (p_i \in S, 1 \leq i \leq n)$ represents the ith protein in the ranking list, which is a potential homologous protein related with q retrieved by three basic ranking methods; $s_1(q,p_i)$ and $s_2(q,p_i)$ represent the bitscore and E value calculated by PSI-BLAST, respectively; $s_3(q,p_i)$ and $s_4(q,p_i)$ represent the probability and E value calculated by HHblits, respectively; $s_5(q,p_i)$ is the distance between two proteins in the 'semantic space' generated by ProtEmbed and the last three elements $s_6(q,p_i)$, $s_7(q,p_i)$ and $s_8(q,p_i)$ represent the reciprocal of ranking position in three ranking results.

Each element $s_j(q,p_i)$ in feature matrix $\mathbf{\Pi}$ [Equation (1)] should be normalized by using the following equation:

$$s'_j(q,p_i) = \frac{s_j(q,p_i)}{\max\limits_{i=1}^{n}\{s_j(q,p_i)\}} \quad (2)$$

The label list $\mathbf{Y}$ of $\mathbf{\Pi}$ can be represented as:

$$\mathbf{Y} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{bmatrix} \quad (3)$$

where l_i ($l_i \in \mathbb{N}$) represents the grade of relevance of a protein pair (q,p_i) labeled based on the Structure Classification of Proteins (SCOP) (Koehl and M, 2000). If the protein pair q and p_i are in the same superfamily, l_i is set as 1, otherwise, it is 0.

2.3 Learning to rank

As an efficient supervised ranking algorithm, LTR algorithm has been successfully applied to document retrieval, question answering, computational advertising (Li, 2011), etc. Three different versions of LTR have been proposed, including pointwise approach, pairwise approach and listwise approach (Liu, 2009). They are different in input representations and loss functions. Among the three methods, the listwise approach outperforms the other two methods because it can take ranking lists as input samples in both learning and prediction steps. Therefore, the group structure of ranking can be maintained and ranking evaluation measures can be more directly incorporated into the loss functions.

In this study, the listwise approach of LTR algorithm was used to construct the ProtDec-LTR. LTR is a supervised learning algorithm and thus has training and testing phases. Given m training samples, each sample should be first converted into feature matrix $\mathbf{\Pi}$ based on Equations (1) and (2). The goal of LTR is to automatically learn a ranking function $F(\cdot)$ from feature matrix $\mathbf{\Pi}$ to a list of labels $\mathbf{Y}$ of training samples [Equation (3)]. A loss function $L(\cdot,\cdot)$ is utilized to evaluate the predictive results of $F(\cdot)$. Each row x in feature matrix $\mathbf{\Pi}$ is ranked according to $F(x)$, then the top k remote homologous proteins in the ranking list are evaluated by using the grades y in label list $\mathbf{Y}$. The loss function is represented as $L(F(x), y)$. The risk function $R(\cdot)$ can be defined as the expected loss function:

$$R(F) = \frac{1}{m}\sum_{i=1}^{m} L\Big(F(x_i), y_i\Big) \quad (4)$$

where x_i represents the ith row in $\mathbf{\Pi}$ [Equation (1)], y_i is the grade of protein pair in label list $\mathbf{Y}$ [Equation (3)]. The learning task then becomes the minimization of the empirical risk function that can be solved by using Gradient Descent (Bottou, 2010; Burges *et al.*, 2005).

In this study, we employed a listwise algorithm of LTR, LambdaMART (Burges, 2010), to learn a ranking model. Given a set of training samples, LambdaMART trained a boosted tree model (a linear combination of a set of regression trees) with Normalized Discounted Cumulative Gain (NDCG) loss function (Burges, 2010; Donmez *et al.*, 2009). Because the size of the ranking list was very large in the current study, only the top 50 detected remote homologous proteins were considered by the LTR, so as to reduce the computational cost. The loss function can be represented as:

$$L(F(x), y) = \exp(-\mathrm{NDCG}(50)) \quad (5)$$

$$\mathrm{NDCG}(50) = G_{\max}^{-1}(50) \sum_{i=1}^{\mathbf{\Pi}(\mathrm{i}) \leq 50} \frac{2^{y_i} - 1}{\log_2\Big(1 + \mathbf{\Pi}(\mathrm{i})\Big)} \quad (6)$$

where $\mathbf{\Pi}(i)$ is the ranking position of protein in predictive results and $G_{\max}(50)$ is the normalizing factor. For a perfect ranking, the proteins with higher grades are always ranked higher. The corresponding empirical risk function of LambdaMART can be written as followings:

$$R(F) = \frac{1}{m}\sum_{i=1}^{m} \exp\left(-G_{\max,i}^{-1}(50) \sum_{j=1}^{\mathbf{\Pi}_i(j) \leq 50} \frac{2^{y_{i,j}} - 1}{\log_2(1 + \mathbf{\Pi}_i(j))}\right) \quad (7)$$

The pseudo codes of the training phase of LTR algorithm for protein remote homology detection are shown in Algorithm 1.

Algorithm 1. The training phase of LTR for protein remote homology detection

Parameters: number of basic methods m, number of training samples n
Input: training protein sequences
Output: the ranking model $F(x)$
Initialize the basic methods and LambdaMART model
For i=0 to n **do**
  For $j = 0$ to m **do**
    Retrieve the benchmark database by using the basic methods
  End for
    Embed the feature vectors by using Equations (1) and (2)
    Label the feature vectors based on Equation (3)
End for
For $k = 1$ to n **do**
  Train LambdaMART ranking model $F(x)$ by using Equations (5) and (7)
End for

For an unseen query protein, its three ranking lists generated by PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011) were converted into the feature matrix **Π** [Equation (1)], then its homologous proteins can be identified based on learnt ranking function $F(\cdot)$.

To help the readers to understand its processes, the training and testing phases of LTR are illustrated in Figure 2.

2.4 Dataset

An updated benchmark dataset (Melvin *et al.*, 2011) was employed to evaluate the performance of different ranking methods, because it can provide good comparability with previous methods. This benchmark dataset was constructed based on SCOP database (Koehl and M, 2000) containing 7329 proteins from 1070 different superfamilies. These proteins were extracted from the Astral database (Brenner *et al.*, 2000), and the identity of any two sequences was lower than 95%. For readers' convenience, the codes of the 7329 proteins and their sequences as well as the attributes of their families and superfamilies are given in Supplementary Material S1.

2.5 Evaluation methodology

How to evaluate the prediction quality is a key for developing a new predictor and estimating its potential application value. The 5-fold cross-validation has been used to evaluate the performance of each method. The benchmark dataset was randomly divided into five subgroups with approximately equal number of proteins. Each method was trained and tested five times with five different training and test sets. For each time, four subsets were used as training data and the remaining one was used as test data.

Two performance measures were employed to evaluate the performance of each method, including ROC1 score and ROC50 score (Gribskov and Robinson, 1996). ROC1 and ROC50 scores represent the area under ROC curve up to the first false positive and the 50th false positives, respectively. Both ROC1 and ROC50 scores were normalized. A socre of 1 means perfect perdiction, whereas a score of 0 means that none of the protens is correctly identified.

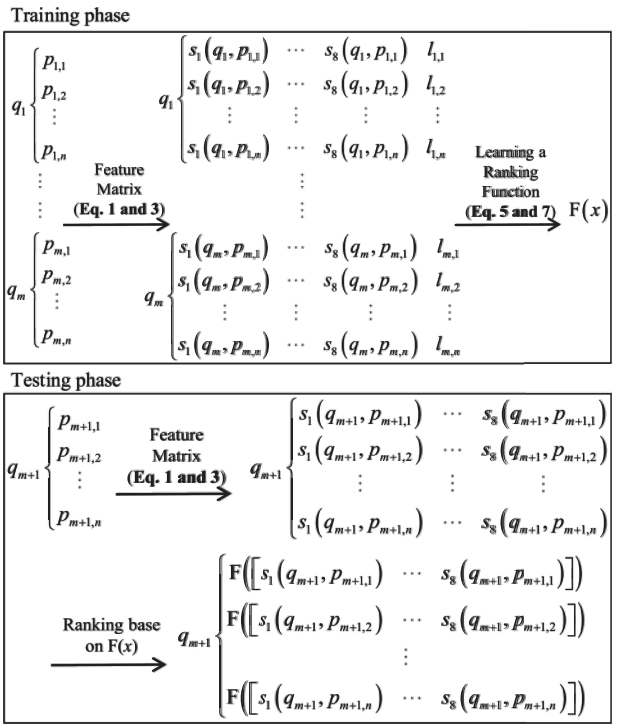

Fig. 2. The training and testing phases of LTR. In training phases, the training samples are represented as feature matrices based on Equations (1) and (3), and then they are used to train a ranking function $F(\cdot)$ based on Equations (5) and (7). In the testing phases, the testing samples are re-ranked by using the learnt ranking function $F(\cdot)$ to detect their homologous proteins

3 Results and discussion

3.1 The ranking methods are complementary

In the field of protein remote homology detection, several ranking methods have been proposed. Among these approaches, PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011) showed the state-of-the-art performance. Therefore, it is particularly interesting to explore if these three predictors are complementary or not. In this regard, pairwise comparisons among the three basic methods in the superfamily-level and sequence-level were conducted, respectively, and the results are shown in Figure 3. Identical results of the two methods will fall on the diagonal line, from which we can see that the three methods are complementary because only a few points fall on the diagonal line. These results are not surprising, because they are based on different techniques and theories, e.g. PSI-BLAST is based on sequence-sequence alignment, while HHblits is based on profile–profile alignment. ProtEmbed employs the semantic embedding, a technique from natural language processing, to construct a low-dimension 'semantic space'.

3.2 ProtDec-LTR can improve the performance by combining three ranking methods in a supervised manner

The predictive results of the three basic ranking methods on the benchmark dataset are listed in Table 1, from which we can see that ProtEmbed achieved the best performance and HHblits outperformed PSI-BLAST. These results are fully consistent with previous studies (Melvin *et al.*, 2011).

As mentioned above, these three ranking methods are complementary. Therefore, performance improvement would be achieved

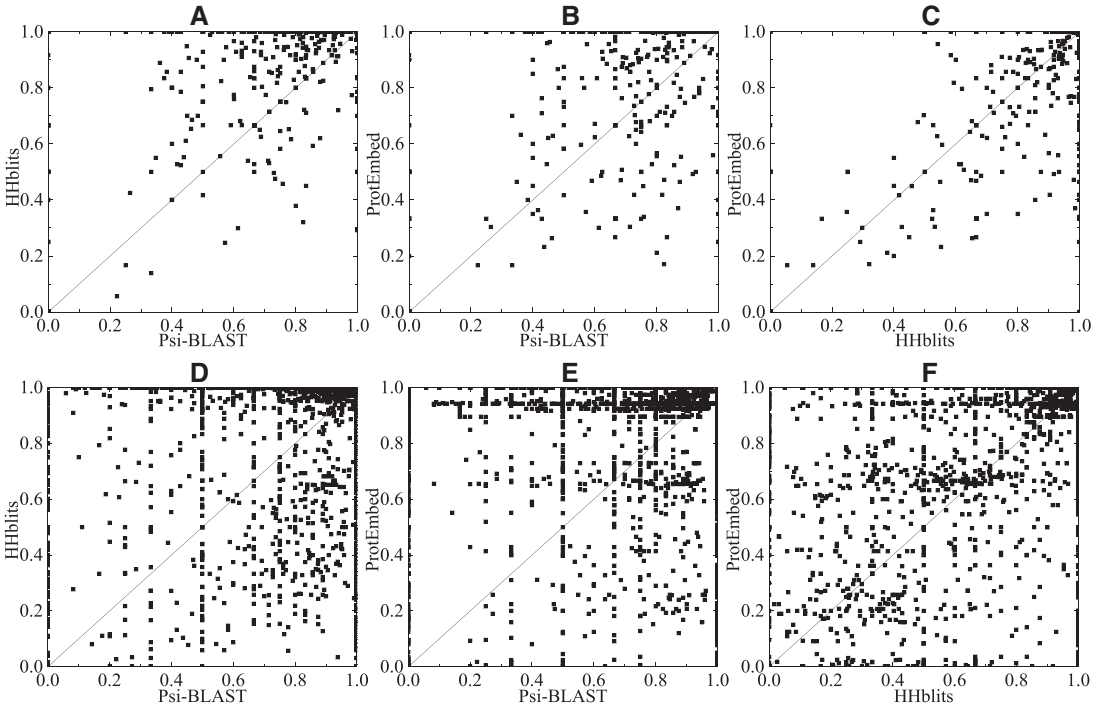

Fig. 3. Comparisons of three basic ranking methods. (**A–C**) The superfamily-level comparisons between two methods labeled near the axis, and the sequence-level comparison results are shown in the (**D**–F). The coordinates of the points in the plot represent the ROC1 scores obtained by the two methods labeled near the axis. Identical results will fall into the diagonal line

Table 1. A comparison of the 5-fold cross-validation results by computational predictors for protein remote homology detection on the benchmark dataset

Methods	ROC1	ROC50
PSI-BLAST	0.7504	0.8007
HHblits	0.8399	0.8820
ProtEmbed	0.8136	0.8897
ProtDec-LTR (PSI-BLAST+HHblits)	0.6631	0.7910
ProtDec-LTR (PSI-BLAST+ProtEmbed)	0.8133	0.8907
ProtDec-LTR (HHblits+ProtEmbed)	0.8442	0.9023
ProtDec-LTR (PSI-BLAST+Hhblits+ProtEmbed)	0.8437	0.9021

by combining them in a supervised fashion by using LTR algorithm. We applied LTR to combine the three ranking methods, and the results of different combinations are listed in Table 1, from which we can see that the best performance of ProtDec-LTR was obtained when combining HHblits and ProtEmbed. When using PSI-BLAST as a basic predictor for ProtDec-LTR, the predictive performance decreased. It is not surprising because the performance of PSI-BLAST is much lower than that of other two approaches as listed in Table 1. Much noise can be introduced when incorporating it into the framework of ProtDec-LTR.

The performance of various methods is plotted in Figure 4. In each graph, a higher curve corresponds to better performance. As shown in this figure, ProtDec-LTR (HHblits + ProtEmbed) achieved the best performance, outperforming other three ProtDec-LTR approaches and it also outperformed all the three basic ranking methods, including PSI-BLAST (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) and ProtEmbed (Melvin *et al.*, 2011), indicating that it is correct to combine different ranking methods in a supervised manner by using the LTR algorithm and performance improvement can be achieved by using this approach. This conclusion is further supported by Figure 5, which compares the performance of ProtDec-LTR against the three individual ranking methods. For most query proteins and superfamilies, ProtDec-LTR outperformed the basic methods, especially, ProtDec-LTR can obviously improve the predictive performance for some superfamilies, e.g. PSI-BLAST, HHblits and ProtEmbed achieved ROC50 scores of 0.5997, 0.8613 and 0.8474 on superfamily a.74.1, respectively, while ProtDec-LTR achieved an ROC50 score of 0.9276, which significantly outperformed all the three basic methods.

4 Conclusion

Based on the similarities between protein sequences and natural languages, some studies have applied the theories and techniques from natural language processing to the field of protein remote homology detection, e.g. Latent Semantic Analysis was used to extract the contextual usage of the word-document matrix based on several building blocks of protein sequences and improved performance for protein remote homology detection has been acquired in comparison with basic formalisms (Dong *et al.*, 2005; Liu *et al.*, 2008). Recently, the semantic embedding method was used to convert a large-scale embedding of protein feature vectors into a low-dimensional 'semantic space' and the homologous proteins can be inferred based on this 'semantic space' (Melvin *et al.*, 2011).

Motivated by the successful applications of these natural language processing techniques, in this study, we applied the LTR algorithm, a technique derived from natural language processing, to detect remote homologous proteins and a new computational predictor called ProtDec-LTR was proposed, which combines three state-of-the-art ranking methods (PSI-BLAST, HHblits and ProtEmbed) in a supervised manner. In this approach, protein remote homology detection is treated as a document retrieval task, where the protein sequences are viewed as documents. Experimental

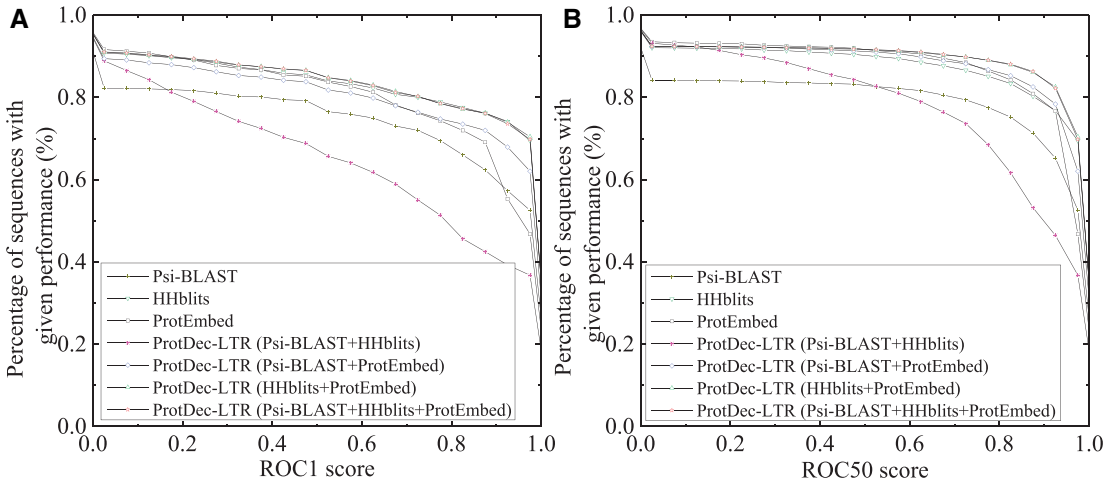

Fig. 4. Relative performance of various computational predictors. The graph plots the percentage of sequences for which the method exceeds a given performance. The higher curve means the method performs better. ROC1 and ROC50 are used as the performance measures for (**A**) and (**B**), respectively

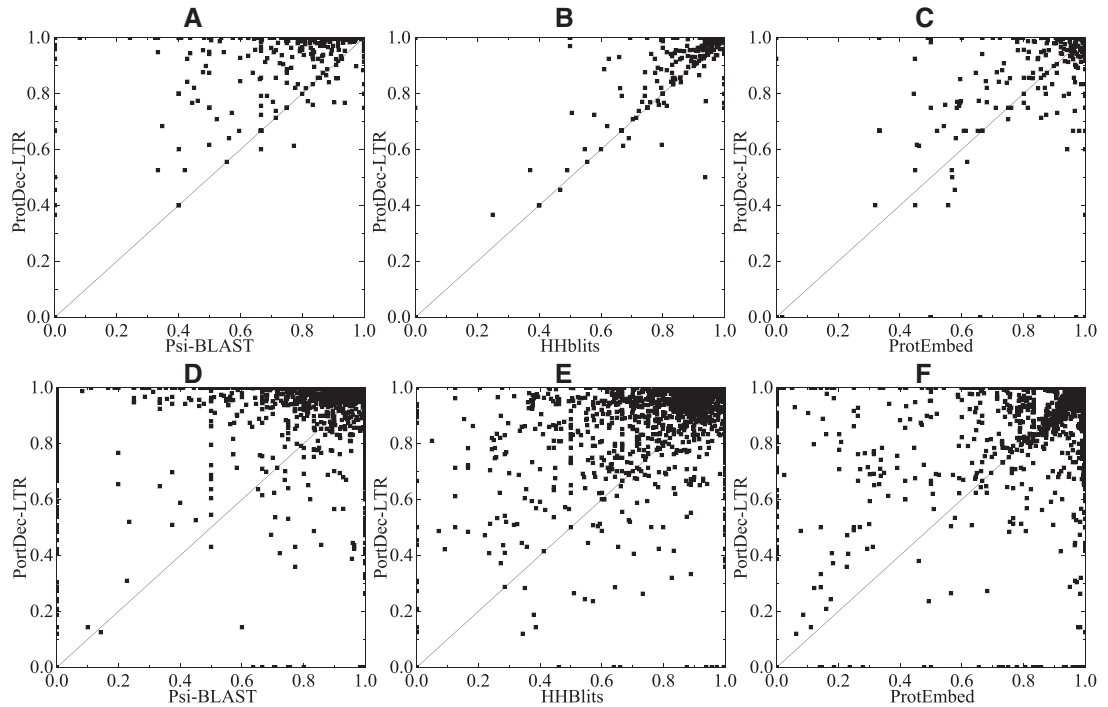

Fig. 5. Performance comparison between ProtDec-LTR against three predictors. (**A–C**) The superfamily-level comparisons between two methods labeled near the axis, and the sequence-level comparison results are shown in the (**D–F**). The coordinates of the points in the plot represent the ROC50 scores obtained by the two methods labeled near the axis

results on a widely used benchmark dataset showed that ProtDec-LTR outperformed other competing methods, especially for some protein superfamilies, performance improvement was obvious. ProtDec-LTR employs a supervised approach to train a model based on the labeled data and the three complementary ranking methods, which considers the advantages of all these individual predictors. It is the main reason for the better performance of ProtDec-LTR. These results further confirm that application of techniques from natural language processing is an efficient way for protein remote homology detection.

It has not escaped our notice that the current approach can be easily applied to other tasks in bioinformatics, because many problems in this field can be formulated as ranking tasks, such as fold recognition (Dong *et al.*, 2009; Lin *et al.*, 2013), etc. Our future studies will focus on exploring new features or 'grammar rules' of protein sequences (Liu *et al.*, 2014b, 2015b) and applying other language processing techniques to protein remote homology detection, such as deep learning (Bengio, 2009; Hinton *et al.*, 2006), etc.

Acknowledgements

The authors would like to thank Hang Li and Jun Xu for their helpful discussions. The authors also wish to thank the three anonymous reviewers for their constructive comments, which are very helpful in strengthening the presentation of this study. This work was supported by the National Natural Science Foundation of China [61300112 and 61272383], the Scientific Research

Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the Natural Science Foundation of Guangdong Province [2014A030313695], Strategic Emerging Industry Development Special Funds of Shenzhen [JCYJ20140508161040764], Shenzhen Municipal Science and Technology Innovation Council [CXZZ20140904154910774] and National High Technology Research and Development Program of China (863 Program) [2015AA015405].

Conflict of Interest: none declared.

References

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bengio,Y. (2009) Learning deep architectures for AI. *Foundations Trends Machine Learn.*, **2**, 1–127.

Bork,P. and Koonin,E.V. (1998) Predicting functions from protein sequences—where are the bottlenecks? *Nat. Genet.*, **18**, 313–318.

Bottou,L. (2010) Large-scale machine learning with stochastic gradient descent. In: Lechevallier,Y. and Saporta,Gi. (eds) *Proceedings of COMPSTAT'2010*, Springer-Verlag Berlin Heidelberg, pp. 177–186.

Brenner,S.E. *et al*. (2000) The ASTRAL compendium for sequence and structure analysis. *Nucleic Acids Res.*, **28**, 254–256.

Burges,C. *et al*. (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning. ACM, New York, NY, USA, pp. 89–96.

Burges,C.J. (2010) From ranknet to lambdarank to lambdamart: an overview. *Learning*, **11**, 23–581.

Dong,Q. *et al*. (2005) Application of latent semantic analysis to protein remote homology detection. *Bioinformatics*, **22**, 285–290.

Dong,Q. *et al*. (2009) A new taxonomy-based protein fold recognition approach based on auto-cross covariance transformation. *Bioinformatics*, **25**, 2655–2662.

Donmez,P. *et al*. (2009) On the local optimality of LambdaRank. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 460–467.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Figueroa,A. and Neumann,G. (2013) Learning to rank effective paraphrases from query logs for community question answering. In: *AAAI Press, Palo Alto, California*. Citeseer.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (Roc) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Hinton,G.E. *et al*. (2006) A fast learning algorithm for deep belief nets. *Neural Comput.*, **18**, 1527–1554.

Karplus,K. *et al*. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Koehl,P. and M, M.L. (2000) The ASTRAL compendium for sequence and structure analysis. *Nucleic Acids Res.*, **28**, 254–256.

Leslie,C.S. *et al*. (2002) The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput.*, **7**, 566–575.

Li,H. (2011) A short introduction to learning to rank. *IEICE Trans. Inf. Syst.*, **94**, 1854–1862.

Lin,C. *et al*. (2013) Hierarchical classification of protein folds using a novel ensemble classifier. *PLoS One*, **8**, e56499.

Liu,B. *et al*. (2008) A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. *BMC Bioinformatics*, **9**, 510.

Liu,B. *et al*. (2012) Using amino acid physicochemical distance transformation for fast protein remote homology detection. *PLoS One*, **7**, e46633.

Liu,B. *et al*. (2013) Protein remote homology detection by combining Chou's pseudo amino acid composition and profile—based protein representation. *Mol. Inform.*, **32**, 775–782.

Liu,B. *et al*. (2014a) Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. *Bioinformatics*, **30**, 472–479.

Liu,B. *et al*. (2014b) iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. *PLoS One*, **9**, e106691.

Liu,B. *et al*. (2014c) Using distances between top-n-gram and residue pairs for protein remote homology detection. *BMC Bioinformatics*, **15**, S3.

Liu,B. *et al*. (2015a) Protein remote homology detection by combining Chou's distance-pair pseudo amino acid composition and principal component analysis. *Mol. Genet. Genomics*, doi: 10.1007/s00438-00015-01044-00434.

Liu,B. *et al*. (2015b) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res.*, **W1**, W65–W71.

Liu,T. (2009) Learning to rank for information retrieval. *Foundations Trends Inf. Retrieval*, **3**, 225–331.

Liu,T. *et al*. (2007) Letor: benchmark dataset for research on learning to rank for information retrieval. In: *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*. ACM, New York, NY, USA, pp. 3–10.

Melvin,I. *et al*. (2009) RANKPROP: a web server for protein remote homology detection. *Bioinformatics*, **25**, 121–122.

Melvin,I. *et al*. (2011) Detecting remote evolutionary relationships among proteins by large-scale semantic embedding. *PLoS Comput. Biol.*, **7**, e1001047.

Muh,H.C. *et al*. (2009) AllerHunter: a SVM-pairwise system for assessment of allergenicity and allergic cross-reactivity in proteins. *PLoS One*, **4**, e5861.

Pearson,W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, **11**, 635–650.

Remmert,M. *et al*. (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, **9**, 173–175.

Saigo,H. *et al*. (2004) Protein homology detection using string alignment kernels. *Bioinformatics*, **20**, 1682–1689.

Sculley,D. (2009) Large scale learning to rank. In: *NIPS Workshop on Advances in Ranking*. pp. 1–6.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Weston,J. *et al*. (2004) Protein ranking: from local to global structure in the protein similarity network. *Proc. Natl. Acad. Sci. USA*, **101**, 6559–6563.

Weston,J. *et al*. (2006) Protein ranking by semi-supervised network propagation. *BMC Bioinformatics*, **7**, S10.