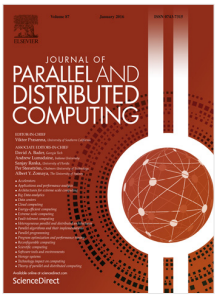
Accepted Manuscript



Prediction of human protein subcellular localization using deep learning

Leyi Wei, Yijie Ding, Ran Su, Jijun Tang, Quan Zou

PII: S0743-7315(17)30239-3
DOI: http://dx.doi.org/10.1016/j.jpdc.2017.08.009
Reference: YJPDC 3730

To appear in: *J. Parallel Distrib. Comput.*

Received date : 27 February 2017
Revised date : 12 June 2017
Accepted date : 18 August 2017

Please cite this article as: L. Wei, Y. Ding, R. Su, J. Tang, Q. Zou, Prediction of human protein subcellular localization using deep learning, *J. Parallel Distrib. Comput.* (2017), http://dx.doi.org/10.1016/j.jpdc.2017.08.009

Highlights

1. We propose a novel deep learning based predictor for human protein subcellular localization prediction.
2. graphical processing unites and CUDA software optimize the deep network architecture and efficiently train the deep networks.
3. The proposed deep learning network can automatically learn high-level and abstract feature representations of proteins by exploring non-linear relations among diverse subcellular locations.

Prediction of human protein subcellular localization using deep learning

Leyi Wei[1], Yijie Ding[1], Ran Su[2], Jijun Tang[1*], and Quan Zou[1*]

1. School of Computer Science and Technology, Tianjin University, Tianjin, China
2. School of Software, Tianjin University, Tianjin, China

*Corresponding author (Email address: zouquan@nclab.net)

Abstract

Protein subcellular localization (PSL), as one of the most critical characteristics of human cells, plays an important role for understanding specific functions and biological processes in cells. Accurate prediction of protein subcellular localization is a fundamental and challenging problem, for which machine learning algorithms have been widely used. Traditionally, the performance of PSL prediction highly depends on handcrafted feature descriptors to represent proteins. In recent years, deep learning has emerged as a hot research topic in the field of machine learning, achieving outstanding success in learning high-level latent features within data samples. In this paper, to accurately predict protein subcellular locations, we propose a deep learning based predictor called DeepPSL by using Stacked Auto-Encoder (SAE) networks. In this predictor, we automatically learn high-level and abstract feature representations of proteins by exploring non-linear relations among diverse subcellular locations, addressing the problem of the need of handcrafted feature representations. Experimental results evaluated with 3-fold cross validation show that the proposed DeepPSL outperforms traditional machine learning based methods. It is expected that DeepPSL, as the first predictor in the field of PSL prediction, has great potential to be a powerful computational method complementary to existing tools.

Keywords

Protein subcellular localization; Feature representation; Deep learning.

Introduction

Knowledge of the subcellular localization of proteins is critical for the understanding of their functions and biological processes in cells. Protein subcellular location information is of highly importance in various areas of research, such as drug design, therapeutic target discovery, and biological research, etc [1]. Accurate prediction of protein subcellular localization is the prequiste to help in-depth understanding and analysis of various protein functions.

As the wide applications of next sequencing techniques, protein sequences have accumulated rapidly during the last decades [2, 3]. Facing with such large-scale sequences, experimental determination of their protein subcellular locations is extremely inefficient and expensive in this post-genomic era. Therefore, effective and efficient computational methods are desired to assist biologists to address these experimental problems. During the past few years, many computational efforts have been done for predicting protein subcellular locations, thus generating a serious of computational methods. Most of high-performance computational methods use machine learning algorithms together with diverse feature representations to make predictions [4-8]. These machine learning based methods can be roughly divided into two classes: (1) sequence-based, and (2) annotation-based.

Sequence-based methods use sequential information from primary sequences of proteins. For instance, Park et al. [9] trained a set of SVMs based on multiple sequence-based feature descriptors, such as amino acid composition, amino acid pair, and gapped amino acid pair composition, and proposed a voting scheme using the trained SVMs to predict protein subcellular locations. The upcoming problem is that features based on amino acid pair composition would lost sequence order effect. In order to address this problem, Chou et al. [10] proposed a modified feature encoding method, namely Pseudo Amino Acid

Composition (PseAAC), sufficiently taking the sequence order information for the remarkable improvement of the predictive performance. Most recently, Rahman et al. [11] proposed to fuse PseAAC, physiochemical property model (PPM), and amino acid index distribution (AAID) to improve the prediction accuracy. Moreover, other sequential information such as the sequence homology and sorting signals are often used to train machine learning models [12-15].

Annotation-based methods utilize information beyond protein sequences, such as GO (Gene Ontology) terms. The GO terms contain the description information of the cellular components, biological processes, and molecular functions of gene products, thus facilitating the accurate prediction of protein subcellular localization [16]. Shen *et al.* [17] proposed a predictor namely Hum-mPLoc 2.0 that encodes proteins into binary features to represent the GO (Gene Ontology) information. To further improve the representation quality of the GO information, Chou *et al.* [18] proposed a more effective way to use real values as the representations of proteins rather than binary values in Hum-mPLoc 2.0. More recently, to incorporate the advantages of sequence-based and annotation-based methods, other methods are developed in a hybrid way, which is based not only on the GO information but also other features including amino acid compositions (AAC) [19], pseudo amino acid compositions (PseAAC) [20-22], and semantic-similarity [23] etc.

In recent years, considerable progresses have been made for the accuracy improvements of machine learning based predictors. However, feature representation still needs handcrafted designation, which is the challenging research component, especially for those without prior field-related knowledge. In contrast to traditional machine learning approaches [24-32], deep learning is capable of automatically learning good feature representations from the input data with the multiple levels of hierarchical non-linear information processing. The representational capability of deep architectures has the great potential to

efficiently and effectively describe the highly non-linear and complex patterns. Classical deep learning architectures include deep belief networks (DBNs) [33], stacked auto-encoder (SAE) [34], and convolutional neural networks (CNNs) [35]. As compared to traditional machine learning architectures, deep learning has shown outstanding results in many applications, such as image classification, object recognition, and medical image analysis. In the last few years, deep learning has begun to be used in many fields of computational biology. For instance, Wen et al. [36] developed a deep-learning framework named DeepDTIs in prediction of drug−target interaction. DeepDTIs uses the DBN to build prediction models, and performs better than traditional machine learning algorithms. Likewise, Alipanahi et al. [37] used Conventional Neural Network (CNN) and proposed a deep-learning method called DeepBind for predicting sequence specificities of DNA- and RNA-binding proteins. DeepBind was reported to outperform state-of-the art methods. In addition, Jo et al. [38] constructed a deep-learning network model named DN-Fold, remarkably improving the performance of protein fold recognition. Zhang et al. [39] developed a deep learning based framework, named TIDE, for accurately predicting translation initiation sites (TISs) on a genome-wide scale data. It was found that TIDE is superior to existing prediction methods. In short, deep learning has exhibited extraordinary performance in computational biology. Accordingly, it is interesting to see whether it can succeed in the prediction of protein subcellular localization, which is still a major challenging task in computational biology.

Motivated by this, in this paper, we propose a new predictor using deep learning networks for predicting protein subcellular localization. To the best of our knowledge, this is the first-time use of deep learning architectures in the field of protein subcellular localization prediction. More specifically, for training, we initially use an unsupervised approach to automatically learn the high-level latent feature representations in the input data and initialize parameters, and then, use a supervised approach to optimize these parameters with the back propagation algorithm. Using the computational power of

graphical processing units (GPUs) and CUDA, we train the deep networks efficiently. After feature learning phase, we add an output layer at the top of deep learning networks. This layer includes a softmax regression classifier that trains the deep learning based features to predict protein subcellular localizations. The evaluation results show that our proposed prediction method achieves satisfactory performance in terms of overall accuracy in the prediction of protein subcellular localizations.

Methods and materials

Dataset preparation

Human protein sequences were collected from a well-known database – UniProtKB [40] (http://www.uniprot.org/help/uniprotkb). After eliminating repeat sequences, we yielded 11,689 protein sequences, covering approximately 200 subcellular locations. We ran statistics and sorted the number of proteins in each location, and then selected the top ten locations: cytoplasm, nucleus, cell membrane, membrane, secreted, cytoskeleton, cell projection, endoplasmic reticulum membrane, cell junction, and mitochondrion, respectively. To avoid the homolog bias, we employed the CD-HIT program [41] with a cutoff value of 0.7 to reduce the sequence similarity of the dataset. After this operation, any two of proteins in the dataset has $\leq$70% sequence similarity. At this end, 9,895 different samples are retained in the dataset.

In this study, we consider the prediction of protein subcellular localization as a single-label or multi-class classification problem. Considering the situation that a single protein may distribute in two or more subcellular locations simultaneously, thus it needs to introduce the definition of "locative protein". Given a same protein sequence that simultaneously exists in two different subcellular locations, it will be counted as 2 locative proteins; if simultaneously existing in three different locations, it will be counted as 3 locative proteins; and so forth. The number of locative proteins can be formulated as,

$$N_{loc} = \sum_{t=1}^{m} tN(t) \qquad (1)$$

where N_{loc} is the number of total locative proteins; m is the number of total subcellular locations; $N(t)$ is the number of proteins with t^{th} subcellular locations. Thus, of the 9,895 different proteins, 6,845 belongs to one location, 2,209 proteins to two locations, 583 to three locations, 158 to four locations, 48 to five locations, 11 to six locations, and 4 to seven locations. Hence, there are 13,978 (= 6,845 + 2,209 × 2 + 583 × 3 + 158 × 4 + 48 × 5 + 11 × 6 + 4 × 7) samples in this dataset, which is denoted as D_h. The distribution of protein subcellular locations is summarized in Table 1. For convenience of discussion, the dataset is denoted as D_h, and the 10 locations are labelled as 10 classes (see Table 1): Class1, Class2, Class3, Class4, Class5, Class6, Class7, Class8, Class9 and Class10, respectively.

Table 1. Protein subcellular location distribution of locative proteins in the D_h dataset.

Classes	Subcellular locations	Number of proteins
Class-1	Cytoplasm	3,374
Class-2	Nucleus	3,520
Class-3	Cell membrane	1,611
Class-4	Membrane	1,677
Class-5	Secreted	1,090
Class-6	Cytoskeleton	800
Class-7	Cell projection	521
Class-8	Endoplasmic reticulum membrane	550
Class-9	Cell junction	443
Class-10	Mitochondrion	392
	Total	**13,978**

Input feature descriptors

Given protein primary sequences, the first thing we should do is to extract features as the input of deep learning architecture. Here, we consider two well-known feature representation methods. The first one is based on physicochemical properties of proteins while the other is based on adaptive

skip dipeptide composition. Both of features have been proven to be effective in multiple Bioinformatics problems. Thus, they are considered in this study. The two feature types are briefly described as follows.

Features based on physicochemical properties

To capture the physicochemical information, we adopted a powerful feature descriptor that uses protein physicochemical properties to represent peptide sequences [4, 42]. This feature descriptor considers the following eight physicochemical properties: (1) normalized van der waals volume, (2) secondary structure, (3) solvent accessibility, (4) polarizability, (5) polarity, (6) hydrophobicity, (7) charge, and (8) surface tension. For each property, 20 standard amino acids {A,N,C,Q,G,H,I,L,M,F,P,S,T,W,Y,V,D,E,K,R} are divided into three groups, e.g., {ANCQGHILMFPSTWYV, DE, KR} for the charge property. To quantize the physicochemical information, the sequence **P** is encoded from the following three aspects: content, distribution and bivalent frequency for each physicochemical property. The details of encoding procedure can be referred to [4, 42]. To this end, the peptide sequence **P** is subsequently represented with a 188-dimension feature vector.

Features based on adaptive skip dipeptide composition

Dipeptide composition is the fraction of every two adjacent residues within a given peptide sequence. This is a measure of the correlation between two adjacent residues. However, the correlation information between the intervening two residues is lost. Here, we propose a modified dipeptide composition, called adaptive skip dipeptide composition, which is the fraction of every two residues with <=L intervening residues within a given peptide sequence, reflecting the correlation information of not only adjacent residues, but also different intervening residues. The adaptive skip dipeptide composition feature vector of a given peptide sequence is represented by,

$$FV = [fv_1, fv_2, \dots, fv_{400}]^T \qquad (2)$$

where

$$fv_i = \frac{O^i}{\sum_{k=1}^{L} n(k)} \qquad (3)$$

and where O^i represents the observed total number of *i*-th two residues with <=L intervening residues, and $n(k)$ represents the total number of all possible two residues with <=*k* intervening residues. If *k*=1, the feature vector is exactly the dinucleotide composition. In this study, the maximum value of *k* cannot succeed the minimum length of sequences in the dataset.

To this end, by fusing the above two feature types, we yielded a total of 588 features (=188+400) as the input of deep network.

Auto-Encoder (AE)

An auto-encoder is one type of artificial neural networks that include three layers: input layer, hidden layer, and output layer. It is a feedforward neural network that produces the output layer as close as possible to its input layer using a lower dimensional representation (hidden layer). The auto-encoder consists of an encoder and a decoder. The encoder is a nonlinear function (i.e., sigmoid function), applied to a mapping from the input layer to the hidden layer; while the decoder is also a nonlinear function that uses to map the feature representations from the hidden layer to the output layer.

Given an input feature vector $x \in R^{D_I}$, the encoder maps it to a compressed feature representation *y* through the following mapping formula:

$$y = f(Wx + b) \qquad (4)$$

where W is $m \times n$ weight matrix and b is bias vector with *m* dimension. The function *f* is a non-linear transformation on the linear mapping. Then, the decoder takes the hidden representation *y* from the input layer and decodes it as closely as possible to the original dimension. The decode transformation can be expressed as follows,

$$z = t(W'y + b') \quad (5)$$

The transformation is performed by a linear mapping followed by an arbitrary linear or non-linear function *t* that employs an $n \times m$ weight matrix W' and a bias vector of dimensionality *n*.

Stacked Auto-Encoder (SAE)

A stacked auto-encoder (SAE) is a hierarchical network that comprises of multiple auto-encoder layers. The SAE network uses the greedy layer-wise unsupervised learning algorithm for training. More specially, in the layer-wise training, the first auto-encoder layer (hidden layer) is trained on the original input data from the input layer. Then, the learned feature representations are fed to the second auto-encoder layer for training. This learning process is repeated for subsequent layers until the layer-wise pre-training is complete. This greedy layer-wise learning is so called 'pre-training'. It is worth noting that the pre-training process initializes the parameters of the deep network in an unsupervised manner. To improve the performance, it uses the back-propagation algorithm to further optimize the parameters generated by the pre-training process in a supervised manner. This supervised optimization step is referred to as 'fine-tuning'. The top output layer is used to represent the class label of an input data. In this layer, the number of units is set to 10, representing the 10 protein subcellular locations (classes). This layer can be also regarded as a classification layer, which the input data are classified into their corresponding classes (i.e., subcellular locations in this study). To avoid overfitting in the learning phase (both pre-training and fine-tuning) of the SAE, we used a dropout regularization factor, which is a method of randomly excluding fractions of hidden units in the training procedure by setting them to zero. This method prevents nodes from co-adapting too much and consequently avoids overfitting.

Implementation

The SAE algorithm of DeepPSL was implemented in MATLAB (2016b version), using the famous DeepLearningTutorials package. The algorithm is accelerated on the GPU (NVIDIA Tesla K80 using CUDA. The operate system is Windows 10 with 4.4 GHz Intel core i7 processor and 32G memory.

Measurements

We employed k-fold cross validation for performance evaluation of prediction models. In *k*-fold cross validation, the original dataset is randomly partitioned into *k* equal-size subsets. Of the *k* subsets, the $k-1$ subsets are used as raining data for model training, and the remaining one is retained as the validation dataset for testing the model. The cross-validation process is then repeated *k* times (the *folds*), with each of the *k* subsets used exactly once as the validation data. The *k* results from the folds can then be averaged to produce a single estimation. In this study, the value of *k* is set to 3.

To quantify the predictive performance of a predictor, two metrics are used, which are accuracy (ACC) and overall accuracy (OA). The two metrics are calculated as follows:

$$\begin{cases} ACC(i) = \frac{C(i)}{N(i)} \\ OA = \frac{\sum_{i=1}^{10} C(i)}{\sum_{i=1}^{10} N(i)} \end{cases} \qquad (6)$$

where $C(i)$ represents the number of sequences correctly predicted in the *i*-th subcellular locations, and $N(i)$ represents the number of sequences in the *i*-th subcellular locations. The OA is the average ratio of correctly predicted subcellular locations over the total locations including the predicted and the real ones.

Results and Discussion

The aim of this study is to apply and estimate the accuracy of deep learning-based method to predict protein subcellular localizations.

To evaluate the prediction quality of a predictor, we performed the proposed DeepPSL on the dataset we mentioned in Section "dataset preparation". This dataset includes 10 different subcellular locations (classes), denoted from class-1 to class-10. The 3-fold cross validation test was used for the performance valuation of a predictor. The performance of the proposed DeepPSL is listed in Table 2. As seen from Table 2, the DeepPSL achieved satisfactory overall performance, obtaining 37.4% in terms of overall accuracy (OA) for the 10-class subcellular localization prediction. To be specific, the DeepPSL had relatively high performance for the "Cytoplasm", "Nucleus", and "Secreted" subcellular locations with 45.9%, 53.6%, and 47.8% in terms of accuracy (ACC), respectively. For the "Cytoskeleton", "Cell projection", and "Cell junction" subcellular locations, the proposed predictor had extremely poor performance with the ACCs in the range of 0.38%-4.9%. This is probably because we trained the deep learning based predictor on the imbalance dataset, i.e., the ratio of the major against few class is roughly 9:1. Thus, it easily results in that the proteins in the few class are incorrectly predicted to the major class. More specially, in the "Cell projection" location, only 2 protein samples are correctly predicted among the 521 protein samples. The imbalance problem will be addressed in our future work.

In the last layer of our proposed deep learning network, the softmax regression classifier is used for prediction by default. To investigate the effect of classification algorithms, we chose two typical and high-efficiency classification algorithms, SVM and RF, to compare with the default softmax regression classifier. We used the feature representations generated from the layer before the last layer to train SVM and RF to make predictions, respectively. The comparison results of the underlying softmax regression classifier and the two classifiers (SVM and RF) are illustrated in Figure 1. As shown in Figure 1, we can see that the softmax classifier achieved 37.4% in terms of overall accuracy, significantly outperforming the other two classifiers (34.5% for SVM and 35.1% for RF) by 2.9% and 2.3%. This indicates that the

softmax classifier have stronger classification ability than the other two classifiers for prediction of protein subcellular localizations. Thus, there is no need to substitute the underlying softmax classifiers with other classifiers in the deep networks. It is worth noting that all the classifiers are performed under their default parameter settings.

Table 2. The performance of the proposed DeepPSL on the D_h dataset.

Subcellular locations	Number of proteins	Number of correctly predicted proteins	ACC (%)
Cytoplasm	3,374	1,548	45.9
Nucleus	3,520	1,887	53.6
Cell membrane	1,611	502	31.2
Membrane	1,677	514	30.6
Secreted	1,090	521	47.8
Cytoskeleton	800	39	4.9
Cell projection	521	2	0.38
Endoplasmic reticulum membrane	550	117	21.3
Cell junction	443	6	1.35
Mitochondrion	392	90	23.0
Total	**13,978**	**5,226**	**37.4**

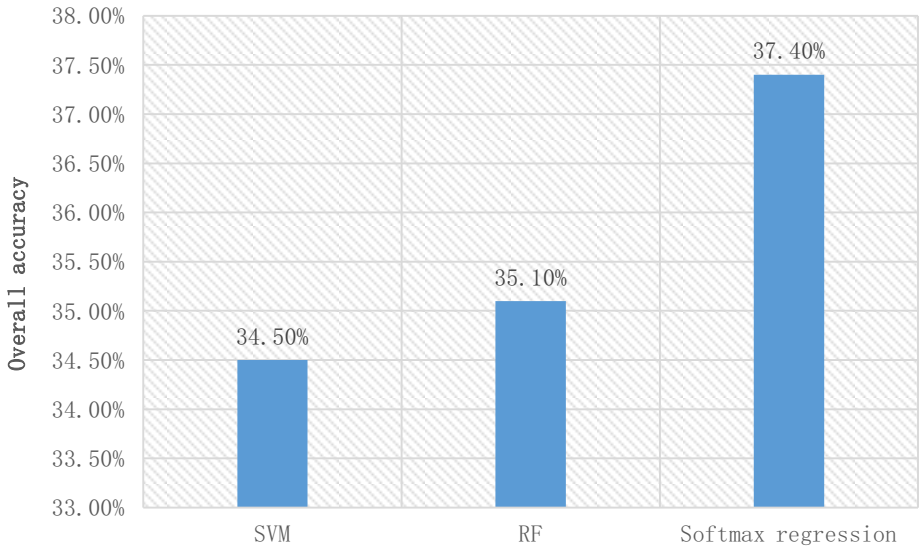


Figure 1. Performance of the underlying softmax regression classifier and other classifiers with the learning feature representations on the D_h dataset

Conclusion

In this paper, we have proposed a deep architecture, namely DeepPSL, for the classification of protein subcellular localizations. Unlike existing machine learning based methods that consider only handcrafted features extracted directly from protein primary sequences, the proposed predictor can automatically learn and extract meaningful feature representations such as non-linear correlations among features that enable to improve the prediction accuracy. We evaluated and compared the performance of the proposed method with traditional machine learning algorithms. The experimental results show that the proposed DeepPSL has better prediction performance, indicating that deep learning has great potential for the performance improvement in Bioinformatics. In future work, we expect that as more protein subcellular location data becomes available, this model will further improve in performance and reveal more useful and meaningful feature patterns. Deep learning based methods require large-scale data sets.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61370010).

Competing interests

The authors declare that they have no competing interests.

Author contributions

LYW analyzed the data and wrote the manuscript, YJD and PWX implemented the algorithms and carried out the experiments. RS, JJT, and QZ edited the manuscript and approved it.

References

1. Eisenhaber F, Bork P: **Wanted: subcellular localization of proteins based on sequence**. *Trends in cell biology* 1998, **8**(4):169-170.
2. Ansorge WJ: **Next-generation DNA sequencing techniques**. *New biotechnology* 2009, **25**(4):195-203.
3. Morozova O, Marra MA: **Applications of next-generation sequencing technologies in functional genomics**. *Genomics* 2008, **92**(5):255-264.

4. Lin C, Zou Y, Qin J, Liu X, Jiang Y, Ke C, Zou Q: **Hierarchical Classification of Protein Folds Using a Novel Ensemble Classifier**. *PLoS One* 2013, **8**(2):e56499.
5. Liu X, Li Z, Liu J, Liu L, Zeng X: **Implementation of Arithmetic Operations with Time-free Spiking Neural P Systems**. 2015.
6. Wu Y, Krishnan S: **Combining least-squares support vector machines for classification of biomedical signals: a case study with knee-joint vibroarthrographic signals**. *Journal of Experimental & Theoretical Artificial Intelligence* 2011, **23**(1):63-77.
7. Song L, Li D, Zeng X, Wu Y, Guo L, Zou Q: **nDNA-prot: Identification of DNA-binding Proteins Based on Unbalanced Classification**. *BMC Bioinformatics* 2014, **15**:298.
8. Wei L, Liao M, Gao Y, Ji R, He Z, Zou Q: **Improved and Promising Identification of Human MicroRNAs by Incorporating a High-quality Negative Set**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2014, **11**(1):192-201
9. Park K-J, Kanehisa M: **Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs**. *Bioinformatics* 2003, **19**(13):1656-1663.
10. Chou KC: **Prediction of protein cellular attributes using pseudo-amino acid composition**. *Proteins: Structure, Function, and Bioinformatics* 2001, **43**(3):246-255.
11. Rahman J, Mondal MNI, Islam KB, Hasan AM: **Feature Fusion Based SVM Classifier for Protein Subcellular Localization Prediction**. *Journal of Integrative Bioinformatics* 2016, **13**(1):288.
12. Emanuelsson O, Nielsen H, Brunak S, Von Heijne G: **Predicting subcellular localization of proteins based on their N-terminal amino acid sequence**. *Journal of molecular biology* 2000, **300**(4):1005-1016.
13. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, Poulin B, Anvik J, Macdonell C, Eisner R: **Predicting subcellular localization of proteins using machine-learned classifiers**. *Bioinformatics* 2004, **20**(4):547-556.
14. Mak M-W, Guo J, Kung S-Y: **PairProSVM: protein subcellular localization based on local pairwise profile alignment and SVM**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 2008, **5**(3):416-422.
15. Nielsen H, Engelbrecht J, Brunak S, Heijne GV: **A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites**. *International journal of neural systems* 1997, **8**(05n06):581-599.
16. Consortium GO: **The Gene Ontology (GO) database and informatics resource**. *Nucleic Acids Research* 2004, **32**(Database issue):D258-261.
17. Shen H-B, Chou K-C: **A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0**. *Analytical biochemistry* 2009, **394**(2):269-274.
18. Chou K-C, Wu Z-C, Xiao X: **iLoc-Hum: using the accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites**. *Molecular Biosystems* 2012, **8**(2):629-641.

19. Cedano J, Aloy P, Perez-Pons JA, Querol E: **Relation between amino acid composition and cellular location of proteins**. *Journal of molecular biology* 1997, **266**(3):594-600.
20. Chen J, Tang Y, Chen C, Bin F, Lin Y, Shang Z: **Multi-Label Learning With Fuzzy Hypergraph Regularizition for Protein Subcellular Location Prediction**. 2014.
21. Liu B, Liu F, Wang X, Chen J, Fang L, Chou K-C: **Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences**. *Nucleic Acids Research* 2015, **W1**:W65-W71.
22. Liu B, Xu J, Fan S, Xu R, Zhou J, Wang X: **PseDNA-Pro: DNA-Binding Protein Identification by Combining Chou's PseAAC and Physicochemical Distance Transformation**. *Molecular Informatics* 2015, **34**(1):8-17.
23. Wan S, Mak M-W, Kung S-Y: **HybridGO-Loc: mining hybrid features on gene ontology for predicting subcellular localization of multi-location proteins**. *PloS one* 2014, **9**(3):e89545.
24. Wei L, Tang J, Zou Q: **Local-DPP: An Improved DNA-binding Protein Prediction Method by Exploring Local Evolutionary Information**. *Information Sciences* 2017, **384**:135-144.
25. Wei L, Wan S, Guo J, Wong KK: **A novel hierarchical selective ensemble classifier with bioinformatics application**. *Artificial Intelligence in Medicine* 2017.
26. Wei L, Xing P, Shi G, Ji Z, Zou Q: **Fast prediction of methylation sites using sequence-based feature selection technique**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2017.
27. Wei L, Xing P, Su R, Shi G, Ma Z, Zou Q: **CPPred-RF: a sequence-based predictor for identifying cell-penetrating peptides and their uptake efficiency**. *Journal of Proteome Research* 2017.
28. Wei L, Xing P, Tang J, Zou Q: **PhosPred-RF: a novel sequence-based predictor for phosphorylation sites using sequential information only**. *IEEE Transactions on NanoBioscience* 2017.
29. Wei L, Xing P, Zeng J, Chen J, Su R, Guo F: **Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier**. *Artificial Intelligence in Medicine* 2017.
30. Wei L, Zhang B, Chen Z, Xing G, Liao M: **Exploring Local Discriminative Information from Evolutionary Profiles for Cytokine-Receptor Interaction Prediction**. *Neurocomputing* 2016, **217**:37-45.
31. Wei L, Zou Q: **Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition**. *International Journal of Molecular Sciences* 2016, **17**(12):2118.
32. Xing P, Su R, Guo F, Wei L: **Identifying N6-methyladenosine sites using multi-interval nucleotide pair position specificity and support vector machine**. *Scientific Reports* 2017, **7**.
33. Rahhal MMA, Bazi Y, Alhichri H, Alajlan N, Melgani F, Yager RR: **Deep learning approach for active classification of electrocardiogram signals**. *IEEE Transactions on Industrial Electronics* 2016, **345**(1):340-354.

34. Suk HI, Shen D: **Deep learning-based feature representation for AD/MCI classification**. In: *Medical Image Computing & Computer-assisted Intervention: Miccai International Conference on Medical Image Computing & Computer-assisted Intervention: 2013*; 2013: 583.
35. Swietojanski P, Ghoshal A, Renals S: **Convolutional Neural Networks for Distant Speech Recognition**. *IEEE Signal Processing Letters* 2014, **21**(9):1120-1124.
36. Wen M, Zhang Z, Niu S, Sha H, Yang R, Yun Y, Lu H: **Deep-Learning-Based Drug-Target Interaction Prediction**. *Journal of Proteome Research* 2017, **16**(4):1401.
37. Alipanahi B, Delong A, Weirauch MT, Frey BJ: **Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning**. *Nature Biotechnology* 2015, **33**(8):831.
38. Jo T, Hou J, Eickholt J, Cheng J: **Improving Protein Fold Recognition by Deep Learning Networks**. *Scientific Reports* 2015, **5**:17573.
39. Zhang S, Hu H, Jiang T, Zhang L, Zeng J: **TITER: predicting translation initiation sites by deep learning**. *bioRxiv* 2017:103374.
40. Boutet E, Lieberherr D, Tognolli M, Schneider M, Bairoch A: **Uniprotkb/swiss-prot**. *Plant Bioinformatics: Methods and Protocols* 2007:89-112.
41. Huang Y, Niu B, Gao Y, Fu L, Li W: **CD-HIT Suite: a web server for clustering and comparing biological sequences**. *Bioinformatics* 2010, **26**(5):680-682.
42. Zou Q, Wang Z, Guan X, Liu B, Wu Y, Lin Z: **An approach for identifying cytokines based on a novel ensemble classifier**. *BioMed research international* 2013, **2013**.

***Author Biography & Photograph**

Leyi Wei is an Assistant Professor of School of Computer Science and Technology at Tianjin University, China. He received the BSc degree in Computing Mathematics in 2010, the MSc degree in Computer Science in 2013, and the Ph.D. degree in Computer Science from Xiamen University, China, respectively. His research interests include bioinformatics, machine learning, and parallel computing.

Yijie Ding received his B.S. degree in Marine Remote Sensing and Information Processing from Tianjin University of Science and Technology. Currently, he is a Ph.D. candidate student of the School of Computer Science and Technology at Tianjin University, China. His research interests include bioinformatics and machine learning.

Ran Su is an Associate Professor of School of Computer Software at Tianjin University, China. She received the Ph.D. degree in Computer Science from The University of New South Wales, Australia in 2013. Her research interests include pattern recognition, machine learning and bioinformatics.

Jijun Tang is a Professor of Computer Science at Tianjin University, China. He received his Ph.D. degree from Computer Science, University of New Mexico, in 2004. His research is in the areas of high-performance computing, engineering simulation, and bioinformatics.

Quan Zou is a Professor of Computer Science at Tianjin University, China. He received his Ph.D. in Computer Science from Harbin Institute of Technology, P.R.China in 2009. His research is in the areas of bioinformatics, machine learning and parallel computing. Now he is putting the focus on genome assembly, annotation and functional analysis from the next generation sequencing data with parallel computing methods. Several related works have been published by Briefings in Bioinformatics, Bioinformatics, PLOS ONE and IEEE/ACM Transactions on Computational Biology and Bioinformatics. He serves on many impacted journals and NSFC (National Natural Science Foundation of China)