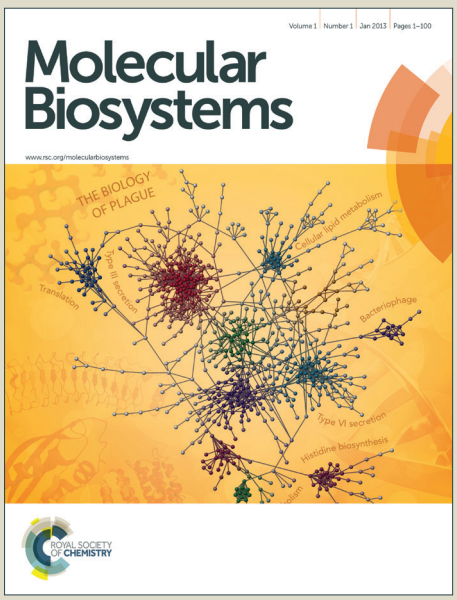
Molecular BioSystems

Accepted Manuscript



ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/molecularbiosystems

Cite this: DOI: 10.1039/c0xx00000x

www.rsc.org/xxxxxx

ARTICLE TYPE

lncRNA-MFDL: Identification of human long non-coding RNAs by fusing multiple features and using deep learning

Xiao-Nan Fan, Shao-Wu Zhang*

Key Laboratory of Information Fusion Technology of Ministry of Education, School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China

** To whom correspondence should be addressed. Tel: +86-29-88431308; Email: zhangsw@nwpu.edu.cn*

Abstract: Long noncoding RNAs (lncRNAs) are emerging as a novel class of noncoding RNAs and potent gene regulators, which play an important and varied role in cellular function. lncRNAs are closely related with the occurrence and development of some diseases. High-throughput RNA-sequencing techniques combined with de novo assembly have identified a large number of novel transcripts. Discovery of large and 'hidden' transcriptome urgently needs to develop the effective computational methods that can rapidly distinguish between coding and long noncoding RNAs. In this study, we developed a powerful predictor (named as lncRNA-MFDL) to identify lncRNAs by fusing multiple features of the open reading frame, k-mer, secondary structure and most-like coding domain sequence and using deep learning classification algorithm. Using the same human training dataset and 10 fold cross validation test, lncRNA-MFDL can achieve 97.1% prediction accuracy which is 5.7, 3.7, 3.4% higher than that of CPC, CNCI and lncRNA-FMFSVM predictors, respectively. Compared with CPC and CNCI predictors on other species (e.g., anole lizard, zebrafish, chicken, gorilla, macaque, mouse, lamprey, orangutan, xenopus and C. elegans) testing datasets, the new lncRNA-MFDL predictor is also much more effective and robust. These results show that lncRNA-MFDL is a powerful tool for identifying lncRNAs. The software package of lncRNA-MFDL can be freely available at http://compgenomics.utsa.edu/lncRNA_MDFL/ for academic users.

Introduction

A mass of evidence reveals that ~98% of the genome can be transcribed, of which only ~2% encodes protein genes [1, 2], and a majority of unexpected noncoding transcription has also been identified [3]. Therefore the vast majority of this unexpected transcription, sometimes referred to as "dark matter" [4, 5], has drawn a great deal of attention. In the mammalian noncoding transcriptome, long noncoding transcripts (>200nt) appear to comprise the largest portion, and show critical roles in diverse regulatory levels, such as transcriptional regulation and post-transcriptional regulation [6, 7].

With the development of high-throughput next-generation sequencing techniques, more and more novel transcripts are generated. It is highly desired to develop computational methods for efficiently and effectively identifying noncoding RNA, which leads to the development of theoretical and computational methods in recent few years. These approaches such as CONC (Coding Or Non-Coding) [8], CPC(Coding Potential Calculator) [9], PORTRAIT [10], PhyloCSF [11] and CPAT [12] typically identify noncoding genes that have short open reading frames (ORFs) and are less homologous with protein-coding genes [13]. However, they are not suitable for identifying long noncoding RNAs (lncRNAs), because lncRNAs may contain long putative ORFs or short protein-like sub-sequences [14, 15]. Recently, several approaches and tools [8, 16-18], were developed to identify lncRNAs. CNCI [8] extracted five features (i.e. the length and S-score of MLCDS, length-percentage, score-distance and codon-bias) by profiling adjoining nucleotide triples and used support vector machine (SVM) to distinguish protein-coding and long noncoding RNA sequences, but it did not consider the RNA structural information. Lv et al. [16] used the LASSO regularized logistic regression to select the chromatin and genomic features to identify lncRNAs over mouse brain development, however, relatively comprehensive chromatin data were only available for a handful of tissues/cells and species, and this method is not suitable for large-scale prediction of lncRNAs. iSeeRNA [17] used SVM model to identify the long intergenic noncoding RNAs (lincRNAs) by integrating multiple features (e.g. conservation, ORF, seven di- and tri-nucleotide sequence frequencies). Wang et al. [18] used GA-SVM algorithm to extract the optimized feature subset to identify the human lincRNAs. Although the existing lncRNAs predictors or methods can effectively identify lncRNAs, most of them used the support vector machine (SVM) to model the classifier based on the sequence and structural features of lncRNAs. SVM is a shallow learning model that consists of only one layer responsible for transforming the raw input features into

a problem-specific feature space, which has been shown effective in solving many simple or well-constrained problems, but its limited modelling and representational power can cause difficulties when dealing with more complicated real-world applications [19].

For further enhancing the prediction accuracy of lncRNAs, in this study, we developed a new powerful predictor (named as lncRNA-MFDL) by employing the deep learning classification algorithm and fusing multiple features of the open reading frame (ORF), k-mer, secondary structure and most-like coding domain. Deep Learning is a new area of machine learning research, which attempt to model high-level abstractions in data by using model architectures composed of multiple non-linear transformations[19-21]. Now, some deep learning architectures such as convolutional deep neural networks (CNNs), deep belief networks (DBNs), deep neural networks (DNNs) and deep stacking networks (DSNs) have been applied to computer vision, automatic speech recognition, natural language processing, and music/audio signal recognition, etc, where they have been shown to produce state-of-the-art results [19, 21-23]. In comparing with the existing tools, lncRNA-MFDL showed better performance on human and other ten species in 10-fold cross validation (10CV) and independent dataset tests.

Material and methods

Datasets

The lncRNAs and protein-coding genes (mRNAs) of human were downloaded from Gencode (v19) [24] and RefSeq database respectively. After removing the transcripts whose length is less than 200nt, we obtained 23,529 human lncRNAs and 35,607 mRNAs, from which the 10,000 lncRNAs and 10,000 mRNAs were selected to form the training dataset. The remaining lncRNAs and mRNAs were used to form the testing dataset. To validate the generalization performance of lncRNA-MFDL, we also created other species (e.g., anole lizard, zebrafish, chicken, gorilla, macaque, mouse, lamprey, orangutan, xenopus and C. elegans) testing datasets which were collected from Ensembl (v76). All training and testing datasets were summarized in Supplementary Table S1, and the accession numbers in these datasets were given in the Supplementary file A.

In general, establishing a highly useful biological molecular attribute predictor involves the following five steps [25, 26]: (1) constructing a valid benchmark dataset to train and test the predictor; (2) using effective mathematical expression to convert the nucleotide (or protein) alphabetic sequences into feature vectors that truly reflect their intrinsic correlation with the attribute to be predicted; (3) developing a powerful algorithm (or engine) to operate the prediction; (4) properly selecting the cross-validation tests to objectively evaluate the performance of the predictor; and (5) establishing a software tool. The lncRNA-MFDL predictor is divided into three stages: feature extraction, feature fusion and pattern classification. For feature extraction which is one of the most critical steps to build a classifier, the query nucleotide sequences are converted into a series of vectors with ORF (Open Reading Frame), *k-mer*, SS(secondary structure) and MLCDS(most-like coding domain sequence) descriptors. For feature fusion, the four kinds of features of ORF, k-mer, SS and MLCDS are integrated to represent the transcript sequence. For pattern classification, the vectors are classified by one deep learning architecture, deep stacking networks.

ORF Descriptor

Previous studies show that a lncRNA transcript is more likely to have low-quality ORF (e.g. either short ORF or small ORF proportion) than mRNAs [8]. Here, the txCdsPredict program from UCSC genome browser (https://genome.ucsc.edu/) was firstly employed to calculate the ORF for each transcript. Then we can use the following feature vector X_{ORF} to represent transcript sequences by serializing ORF length and ORF proportion.

$$X_{ORF} = [l, l/L] \quad (1)$$

where l is the ORF length, and L is the length of a transcript sequence.

k-mer Descriptor

Because lncRNAs exhibit poor protein-coding potential, the frequency of k neighboring (*k-mer*) bases may contain the statistical information for distinguishing between lncRNAs and mRNAs. Here, we used the following feature vectors X_{k-mer}^{ORF} and X_{k-mer}^{TS} to represent transcript sequences by serializing the *k-mer* (k=1, 2, 3) frequency in ORF and transcript sequence, respectively.

$$X_{k-mer}^{ORF} = [t_1, \ldots t_\alpha, \ldots, t_4, t_1', \ldots, t_\beta', \ldots, t_{16}', t_1'', \ldots, t_\gamma'', \ldots, t_{64}''] \quad (2)$$

$$X_{k-mer}^{TS} = [f_1, \ldots f_\alpha, \ldots, f_4, f_1', \ldots, f_\beta', \ldots, f_{16}', f_1'', \ldots, f_\gamma'', \ldots, f_{64}''] \quad (3)$$

where t_α ($\alpha = 1,2,\ldots,4$) is the frequency of a single base (A, C, G and T) in ORF; t_β' ($\beta = 1,2,\ldots,16$) is the frequency of two neighboring bases (e.g., AC, AG, AT, CG) in ORF; t_γ'' ($\gamma = 1,2,\ldots,64$) is the frequency of three neighboring bases (e.g., ACG, AGT, CGT) in ORF. f_α ($\alpha = 1,2,\ldots,4$) is the frequency of a single base (A, C, G and T) in transcript sequence; f_β' ($\beta = 1,2,\ldots,16$) is the frequency of two neighboring bases (e.g., AC, AG, AT, CG) in transcript sequence; f_γ'' ($\gamma = 1,2,\ldots,64$) is the frequency of three neighboring bases (e.g., ACG, AGT, CGT) in transcript sequence.

SS Descriptor

Secondary structures play a key role in the functions of lncRNAs and are more highly conserved than the primary sequences. In addition, the minimum free energy (MFE) is an index that evaluates the stability of the secondary structure of RNAs. Accordingly, we used a number of secondary structure-based features, in terms of the minimum free energy (MFE), paired bases and unpaired bases to form a feature vector X_{SS} to represent the transcript sequences. These feature values were given by the RNAfold program [27, 28].

$$X_{SS} = [v_{MFE}, v_{MFE}/L, n_p, n_{up}] \quad (4)$$

where, v_{MFE} is the minimum free energy; L is the length of a transcript sequence; n_p is the number of paired base; n_{up} is the number of unpaired base.

MLCDS Descriptor

Because the coding domain sequence (CDS) regions have been under a variety of competing selection pressures, especially the transcript optimization force that is associated with the juxtaposition of tRNAs but not required for non-coding regions [29], the features of the most-like CDS (MLCDS) were extracted to distinguish protein-coding sequences from the non-coding sequences [8]. The identification of the best MLCDS can be described as: 1) Firstly calculating the usage frequency of adjoining nucleotide triplets (ANT) with coding and non-coding transcript sequences, and using the log-ratio of the usage frequency of all kinds of ANT to constitute a 64*64 ANT score matrix; 2) Using a sliding window (e.g., 150nt) to scan each transcript sequence six times to generate six reading frames, meanwhile, calculating the sequence-score (S-score) of each window based on ANT score matrix; 3) Applying a dynamic programming of Maximum Interval Sum [30] to identify the MLCDS of each reading frame; 4) Defining one of the six candidate MLCDS regions with the maximum S-sore as the best MLCDS. Then, we used the following feature vector X_{MLCDS} to represent the transcript sequences by serializing the length and S-score of the best MLCDS, the length percentage and the score distance of MLCDS.

$$\mathrm{X}_{MLCDS} = [L_{BMLCDS}, S_{BMLCDS}, P_{MLCDS}, D_{MLCDS}] \tag{5}$$

where L_{BMLCDS}, S_{BMLCDS} are respectively the length and S-score of the best MLCDS; P_{MLCDS} and D_{MLCDS} respectively represent the length-percentage and the score distance of MLCDS, which are calculated by the following two formulas.

$$P_{MLCDS} = \frac{L_{BMLCDS}}{\sum_{i=1}^{n}(L_i)}, \quad i = 1, 2, \cdots, 6 \tag{6}$$

$$D_{MLCDS} = \frac{1}{5}\sum_{j=1}^{n}(S_{BMLCDS} - S_j), \qquad j = 1, 2, \cdots, 5 \tag{7}$$

where L_i represents the length of the *i-th* MLCDS; S_j represents the S-score of the *j-th* non-best MLCDS.

Feature Fusion

Feature fusion can derive the most discriminatory information from original multi-feature sets and eliminate the redundant information from the correlation between distinct feature sets, which benefits the final decision. Here, four kinds of feature set of ORF, k-mer, SS and MLCDS are concatenated into one set of feature vectors to represent the transcript sequences, which can be formulized as following:

$$X = [X_{ORF}, X_{k-mer}^{ORF}, X_{k-mer}^{TS}, X_{SS}, X_{MLCDS}] \tag{8}$$

Deep Learning Algorithm

Deep learning is a learning method with the deep architecture and the good learning ability, which can perform the intellectual learning like learning the features. The deep architecture refers to the multilayer network where each two adjacent layers are connected to each other in some way. Depending on the learning nature of layered module, the existing deep learning algorithms can be classified into generative (e.g. deep auto-encoder, deep Boltzmann machine and deep Belief networks), discriminate (e.g. convolutional neural networks and deep stacking networks) and hybrid architecture (e.g. deep neural networks) [31], which have been successfully applied to computer vision, speech recognition and signal processing [19, 21-23, 32]. The well-known DNNs require stochastic gradient descent which renders parallelization of network parameter learning virtually impossible. For overcoming the problem of parallelizing learning in DNN modules, the deep stacking networks (DSNs) were introduced [32, 33]. The basic DSN architecture consists of many stacking modules, each of which takes a simplified form of shallow multilayer perception using convex optimization for learning perceptron weights[34]. "Stacking" is accomplished by concatenating all previous modules' output predictions with the original input vector to form the new "input" vector in the new module [33]. The DSN weight parameters *W* (input weight matrices) and *U* (output network weight matrices) in each module are learned efficiently from the training data by using basic learning algorithm and fine tuning algorithm [32, 33].

Basic learning algorithm: Suppose $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N]$, $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_i, \cdots, \mathbf{t}_N]$ respectively denote the training vectors and target vectors, where N is the total number of training samples. The output of a DSN module is $\mathbf{y}_i = \mathbf{U}^T\mathbf{h}_i$, where $\mathbf{h}_i = \sigma(\mathbf{W}^T\mathbf{x}_i)$ is the hidden layer output, and $\sigma(\cdot)$ is the sigmoid function. The loss function of means square error is used to learn parameter $\mathbf{U}$ assuming $\mathbf{W}$ is given, that is, by minimizing the average of the total square error $E = \|\mathbf{Y} - \mathbf{T}\|^2 = Tr[(\mathbf{Y} - \mathbf{T})(\mathbf{Y} - \mathbf{T})^T]$ to learn parameter. If the lower layer weight matrix $\mathbf{W}$ is fixed, the hidden layer values $\mathbf{H}$ are also determined. Consequently, the upper-layer weight matrix $\mathbf{U}$ in each module can be determined by setting the gradient $\frac{\partial E}{\partial \mathbf{U}} = 2\mathbf{H}(\mathbf{U}^T\mathbf{H} - \mathbf{T})^T$ to zero, then leading to the closed-form solution

$$\mathbf{U} = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{T}^T \tag{9}$$

Note that the weight matrices $\mathbf{W}$ across all DNS modules need be set empirically. In general, there are two ways to set $\mathbf{W}$. First, using various distributions generate the random numbers to set $\mathbf{W}$. Second, applying contrastive divergence to separately train the restricted Boltzmann machines (RBM), and used the trained RBM weights to set $\mathbf{W}$. In this study, we used the trained RBM weights to set $\mathbf{W}$ for the bottom module.

Module-bound fine tuning: The weight matrices $\mathbf{W}$ of DSN in each module can be further learned using the batch-mode gradient descent. That is,

$$\frac{\partial E}{\partial \mathbf{W}} = 2\mathbf{X}[\mathbf{H}^T \circ (1 - \mathbf{H})^T \circ [\mathbf{H}^{\dagger}(\mathbf{H}\mathbf{T}^T)(\mathbf{T}\mathbf{H}^{\dagger}) - \mathbf{T}^T(\mathbf{T}\mathbf{H}^{\dagger})] \tag{10}$$

$$\mathbf{W}^{(j+1)} = \mathbf{W}^{(j)} + \eta \times \frac{\partial E^{(j)}}{\partial \mathbf{W}^{(j)}} \tag{11}$$

where $\mathbf{H}^{\dagger} = \mathbf{H}^{T}(\mathbf{H}\mathbf{H}^{T})^{-1}$, the symbol $\circ$ represents the element-wise matrix multiplication, and η is the learning rate of updating the weight matrices $\mathbf{W}$.

Then the batch-mode fine tuning algorithm updates $\mathbf{W}$ using Eq.10 and Eq.11, and $\mathbf{U}$ is subsequently updated using Eq.9 in a closed form with no iteration.

Assessment of Prediction System

The performance measures of overall accuracy (ACC), sensitivity (Sn), specificity (Sp) and the Matthew's correlation coefficient (MCC) were used to assess the prediction system, and they are defined as follows:

$$S_n = \frac{TP}{TP + FN} \tag{12}$$

$$S_p = \frac{TN}{TN + FP} \tag{13}$$

$$ACC = \frac{TP + TN}{TN + FP + TP + FN} \tag{14}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TN + FN)(TP + FN)(TN + FP)}} \tag{15}$$

where *TP* and *TN* are the number of correctly predicted lncRNAs and mRNAs, respectively, and *FP* and *FN* are the number of incorrectly predicted lncRNAs and mRNAs, respectively.

Results and Discussions

Performance of lncRNA-MFDL

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, *K*-fold (e.g. 5-fold, 10-fold) crossover or subsampling test, and jackknife test. However, of the three test methods, the jackknife test is deemed the least arbitrary that can always yield a unique result for a given benchmark dataset [25]. Accordingly, the jackknife test has been increasingly and widely used by investigators to examine the quality of various predictors [26, 35-38]. However, for large scale database, the jackknife test needs to spend lots of time to generate the prediction results. To reduce the computational time and evaluate the generalization performance of a predictor, we adopted the 10-fold cross-validation (10 CV) test and independent dataset test as done by many investigators.

To demonstrate the superiority of lncRNA-MFDL, we compared it with other three state-of-the-art predictors, CPC, CNCI and lncRNA-MFSVM. CPC used six biologically meaningful sequence features and SVM to discriminate coding from noncoding transcripts, which works well with known protein coding gene transcripts (PCTs) but may tend to classify novel PCTs into ncRNA if they have not recorded in the protein database used by CPC [9, 17]. CNCI used the profile information of adjoining nucleotide triples and SVM to effectively distinguish protein-coding and non-coding transcripts [8]. In order to show the outstanding performance of deep learning algorithm, we also design an lncRNA-MFSVM predictor by using SVM and the same features as lncRNA-MFDL predictor. The results of four predictors on the same human training dataset in 10CV test are shown on Table 1. It can be seen that the overall accuracy of lncRNA-MFDL is 97.1%, which is 5.7, 3.7 and 3.4 % higher than that of CPC, CNCI and lncRNA-MFSVM, respectively, and the deviation is also lower than CPC, CNCI and lncRNA-MFSVM; MCC of lncRNA-MFDL is 0.942, which is 0.109, 0.074 and 0.08 higher than that of CPC, CNCI and lncRNA-MFSVM, respectively; S_n of lncRNA-MFDL is 97.7%, which is 0.9, 6.5 and 4.4% higher than that of CPC, CNCI and lncRNA-MFSVM, respectively. These results show that the lncRNA-MFDL predictor has the powerful performance for distinguishing lncRNAs and mRNAs.

Comparing S_n, S_p of CPC, CNCI and lncRNA-MFDL, we found that S_n of CPC is 10% higher than its S_p, meaning that CPC tends to classify the new lncRNA into mRNA; S_n of CNCI is 4.3% lower than its S_p, meaning that CNCI tends to classify the new mRNA to lncRNA; while S_n and S_p of lncRNA-MFDL are approximately equal, meaning that the lncRNA-MFDL predictor is more robust than CPC and CNCI predictors.

The comparing results of lncRNA-MFSVM and lncRNA-MFDL indicate that the classifying performance of deep learning is superior to SVM. Comparing the results of CNCI and lncRNA-MFSVM which of them use the same SVM and different feature sets, we found that the distinguished power of our feature fusion method and CNCI feature extraction method is also same, but our feature fusion method can effectively predict lncRNAs.

Table 1. The performance of CPC, CNCI, lncRNA-MFSVM and lncRNA-MFDL on same human training dataset in 10CV test

	S_n (%)	S_p (%)	ACC (%)	MCC
CPC	96.8±1.7	86.0±1.7	91.4±0.9	0.833±0.018
CNCI	91.2±1.5	95.5±2.0	93.4±1.5	0.868±0.030
lncRNA-MFSVM	93.3±2.1	92.8±1.3	93.7±1.0	0.862±0.021
lncRNA-MFDL	97.7±1.3	96.5±1.0	97.1±0.8	0.942±0.016

In order to further evaluate the generalized performance of lncRNA-MFDL predictor, we also implemented lncRNA-MFDL predictor on human testing dataset and other species (e.g., anole lizard, zebrafish, chicken, gorilla, macaque, mouse, lamprey, orangutan, xenopus and C. elegans) testing datasets. The results of lncRNA-MFDL, CNCI and CPC on the 11 testing datasets are shown in Table 2. It can be seen that the overall accuracy of lncRNA-MFDL predictor is higher than that of CPC and CNCI predictors, suggesting that the lncRNA-MFDL predictor has better generalized performance.

Table 2. The overall accuracy (%) of CPC, CNCI and lncRNA-MFDL on 11 testing datasets.

	CPC	CNCI	LncRNA-MFDL
Human	92.6	95.8	96.0
Anole lizard	87.0	93.5	95.5
Zebrafish	85.8	90.3	90.5
Chicken	91.4	95.7	95.7
Gorilla	83.5	86.2	92.8
Macaque	89.0	94.1	96.2
Mouse	58.7	71.4	86.6
Lamprey	82.7	75.0	88.5
Orangutan	79.2	85.8	90.0
Xenopus	78.9	92.8	96.9
C. elegans	79.6	85.5	90.1

Comparison with individual feature classifier

To further verify the effectiveness of LncRNA-MFDL predictor, we compared it with five other individual feature deep learning classifiers based on the ORF, k-mer^{ORF}, k-mer^{TS}, SS and MLCDS feature descriptors, respectively. The results on human training dataset in 10CV test are shown in Table 3, from which we can see that the overall accuracy of LncRNA-MFDL is 3, 10.6, 18.2, 13.4, 5.1% higher than that of ORF-DL, k-mer^{ORF}-DL, k-mer^{TS}-DL, SS-DL and MLCDS-DL classifiers, respectively, suggesting our feature fusion method is effective for distinguishing the lncRNAs and mRNAs. The ORF and MLCDS feature descriptors are more powerful than the other three feature descriptors, meaning that ORF and MLCDS features contribute the most to the overall performance of LncRNA-MFDL predictor. Comparing with the individual feature DL classifier, these results show that LncRNA-MFDL predictors are effective and robust for predicting lncRNAs.

Table 3. Results of LncRNA-MFDL and five other individual feature deep learning classifiers on human training dataset in 10CV test

	Sn (%)	Sp (%)	ACC (%)	MCC
ORF-DL	95.1±1.2	93.1±1.6	94.1±0.8	0.883±0.016
k-mer^{ORF}-DL	87.6±4.6	85.5±5.7	86.5±4.2	0.732±0.084
k-mer^{TS}-DL	76.1±3.7	81.8±10.8	78.9±5.5	0.583±0.109
SS-DL	79.3±3.0	88.1±9.0	83.7±5.0	0.679±0.105
MLCDS-DL	95.7±2.2	88.3±1.2	92.0±1.3	0.842±0.026
LncRNA-MFDL	97.7±1.3	96.5±1.0	97.1±0.8	0.942±0.016

Conclusions

Large-scale of transcriptome sequencing technology have identified a great amount of transcripts, that attract the attention on the study of lncRNAs. However, for most species, it remains a challenge to identify lncRNAs from protein coding genes, because of the lack of necessary information such as whole-genome sequence, known protein regions and comprehensive chromatin data. Therefore, it is important to develop a method to distinguish lncRNAs and protein-coding genes based on the RNA sequences. In this study, based on the RNA sequences, we introduced five kinds of feature descriptors (e.g. ORF, *k-mer*^{ORF}, *k-merTS*, SS and MLCDS) and fused them forming a vector to represent the RNA sequence. Instead of the shallow learning models (e.g. SVM, HMM, CRF), we used DSN deep learning architecture model to design the lncRNA-MFDL predictor for discriminating lncRNAs and mRNAs. Comparing with the existing CPC, CNCI predictors of lncRNA on the human training dataset and other 10 species testing datasets, lncRNA-MFDL predictor show strong robust and powerful ability for distinguishing lncRNAs and mRNAs, and it represents an intriguing and promising avenue for predicting lncRNAs. lncRNA-MFDL software package is available at http://compgenomics.utsa.edu/lncRNA_MDFL/.

Funding

This paper was supported by the National Natural Science Foundation of China (61170134, 61473232, 91430111) and the Graduate Starting Seed Fund of Northwestern Polytechnical University (Z2014145, Z2014152)

References

1. M. C. Frith, M. Pheasant and J. S. Mattick, *European Journal of Human Genetics*, 2005, **13**, 894-897.
2. I. H. G. S. Consortium, *Nature*, 2004, **431**, 931-945.
3. T. IUM, 2012.
4. G. Riddihough, *Science*, 2005, **309**, 1507-1507.
5. J. M. Johnson, S. Edwards, D. Shoemaker and E. E. Schadt, *Trends in Genetics*, 2005, **21**, 93-102.
6. F. Aguilo, M.-M. Zhou and M. J. Walsh, *Cancer research*, 2011, **71**, 5365-5369.
7. M. Cesana, D. Cacchiarelli, I. Legnini, T. Santini, O. Sthandier, M. Chinappi, A. Tramontano and I. Bozzoni, *Cell*, 2011, **147**, 358-369.
8. L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen and Y. Zhao, *Nucleic acids research*, 2013, **41**, e166-e166.
9. L. Kong, Y. Zhang, Z.-Q. Ye, X.-Q. Liu, S.-Q. Zhao, L. Wei and G. Gao, *Nucleic acids research*, 2007, **35**, W345-W349.
10. R. T. Arrial, R. C. Togawa and M. M. Brigido, *BMC bioinformatics*, 2009, **10**, 239.
11. M. F. Lin, I. Jungreis and M. Kellis, *Bioinformatics*, 2011, **27**, i275-i282.
12. L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher and W. Li, *Nucleic acids research*, 2013, **41**, e74-e74.
13. M. Clamp, B. Fry, M. Kamal, X. Xie, J. Cuff, M. F. Lin, M. Kellis, K. Lindblad-Toh and E. S. Lander, *Proceedings of the National Academy of Sciences*, 2007, **104**, 19428-19433.
14. M. Guttman and J. L. Rinn, *Nature*, 2012, **482**, 339-346.
15. M. E. Dinger, K. C. Pang, T. R. Mercer and J. S. Mattick, *PLoS computational biology*, 2008, **4**, e1000176.
16. J. Lv, H. Liu, Z. Huang, J. Su, H. He, Y. Xiu, Y. Zhang and Q. Wu, *Nucleic acids research*, 2013, **41**, 10044-10061.
17. K. Sun, X. Chen, P. Jiang, X. Song, H. Wang and H. Sun, *BMC genomics*, 2013, **14**, S7.
18. Y. Wang, Y. Li, Q. Wang, Y. Lv, S. Wang, X. Chen, X. Yu, W. Jiang and X. Li, *Gene*, 2014, **533**, 94-99.
19. D. Yu and L. Deng, *Signal Processing Magazine, IEEE*, 2011, **28**, 145-154.
20. G. E. Hinton, S. Osindero and Y.-W. Teh, *Neural computation*, 2006, **18**, 1527-1554.
21. Y. Bengio, A. Courville and P. Vincent, 2013.
22. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, *arXiv preprint arXiv:1207.0580*, 2012.
23. L. Deng, Proc. Asian-Pacific Signal and Information Processing–Annual Summit and Conference (APSIPA-ASC), 2011.
24. J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa and S. Searle, *Genome research*, 2012, **22**, 1760-1774.
25. K.-C. Chou, *Journal of theoretical biology*, 2011, **273**, 236-247.
26. S.-W. Zhang, Y.-F. Liu, Y. Yu, T.-H. Zhang and X.-N. Fan, *Analytical biochemistry*, 2014, **449**, 164-171.
27. I. L. Hofacker, *Nucleic acids research*, 2003, **31**, 3429-3431.
28. I. L. Hofacker, *Current Protocols in Bioinformatics*, 2004, 12.12. 11-12.12. 12.
29. J. R. Buchan, L. S. Aucott and I. Stansfield, *Nucleic acids research*, 2006, **34**, 1015-1027.
30. S. Mukherjee and Y. Zhang, *Nucleic acids research*, 2009, **37**, e83-e83.
31. L. Deng and D. Yu.
32. L. Deng, D. Yu and J. Platt, Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, 2012.
33. L. Deng, X. He and J. Gao, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013.
34. L. Deng and D. Yu, Proceedings of the Interspeech, 2011.
35. S.-W. Zhang, Y.-L. Zhang, H.-F. Yang, C.-H. Zhao and Q. Pan, *Amino Acids*, 2008, **34**, 565-572.
36. S.-W. Zhang, W. Chen, F. Yang and Q. Pan, *Amino Acids*, 2008, **35**, 591-598.
37. W. Chen, S.-W. Zhang, Y.-M. Cheng and Q. Pan, *Proteome science*, 2011, **9**, S16.
38. W. Chen, P.-M. Feng, H. Lin and K.-C. Chou, *Nucleic acids research*, 2013, gks1450.