Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

A deep learning framework with hybrid encoding for protein coding regions prediction in biological sequences

Chao Wei, Junying Zhang*, Xiguo Yuan, Zongzhen He and Guojun Liu

School of Computer Science and Technology, Xidian University, Xi'an, 710071, PR China

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Protein coding regions prediction is a very important but overlooked subtask for tasks such as prediction of complete gene structure, coding/noncoding RNA. Many machine learning methods have been proposed for this problem, they first encode a biological sequence into numerical values and then feed them into a classifier for final prediction. However, encoding schemes directly influence the classifier's capability to capture coding features and how to choose a proper encoding scheme remains uncertain. Recently, we proposed a protein coding region prediction method in transcript sequences based on a bidirectional recurrent neural network with non-overlapping kmer, and achieved considerable improvement over existing methods, but there is still much room to improve the performance. In fact, kmer features that count the occurrence frequency of trinucleotides only reflect the local sequence order information between the most contiguous nucleotides, which loses almost all the global sequence order information.

Results: We here present a deep learning framework with hybrid encoding for protein coding regions prediction in biological sequences, which effectively exploit global sequence order information, non-overlapping kmer features and statistical dependencies among coding labels. Evaluated on genomic and transcript sequences, our proposed method significantly outperforms existing state-of-the-art methods.

Availability: The source code and the dataset used in the paper are publicly available at: https://github.com/xdcwei/DeepCoding/.

Contact: jyzhang@mail.xidian.edu.cn

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

1 Introduction

Genome annotation helps in understanding complicated biological mechanisms underlying gene regulation and remains a challenging problem in biology. The development of next-generation sequencing (NGS) technologies give rise to an exponential increase of sequence data. Many efforts have been dedicated to the identification of genomic mutations by using NGS datasets Yuan *et al.* (2017); Tuo *et al.* (2020) in the past few years, it is urgent to find effective genome annotation techniques for predicting genes Catherine *et al.* (2002).

The prediction of protein coding regions in genomic or transcript sequences is a very important but overlooked subtask for genome annotation. Many well-known gene prediction tools (e.g., GenScan Burge and Karlin (1997), Augustus Stanke *et al.* (2004)) are integrated models, in which the task of identifying gene structure is first divided into subtasks such as the prediction of functional sites and coding regions, and then these subtasks are integrated into a structured learning framework for the prediction of gene structure Al-Turaiki *et al.* (2011). Moreover, coding features is also very important for computational methods to discriminate mRNAs from long non-coding RNAs Li *et al.* (2014); Tong and Liu (2019). However, prediction of protein coding regions from uncharacterized biological sequences (e.g., genomic or transcript sequences) is a very challenging task. This is because (1) genomic sequences contain introns that disrupt the coding structure Catherine *et al.* (2002). (2) there exists a considerable number of short exons bordered by large introns, which is easily missed by computational methods Catherine *et al.* (2002). (3) unlike consensus motifs, coding features often exhibit higher-order distant interactions among nucleotides and more difficult to capture Rajapakse and Ho (2005).

Many existing computational methods Hatzigeorgiou *et al.* (1996); Guigó (1997); Zhang *et al.* (1998); Hatzigeorgiou (2002); Shuo and Yisheng (2009); Tzanis *et al.* (2012); Wei *et al.* (2020) have been proposed for protein coding regions prediction in genomic or transcript sequence during the past decades. They first encode a biological sequence into numerical values and then feed them into a classifier for final prediction. There are mainly two types according to the encoding scheme they use: sequential model and discrete model. The sequential model converts each nucleotide of a biological sequence into a numerical value one by one, which preserves the original order of the bases that appears in the biological sequence. A widely used encoding scheme is one-hot representation (also called C4 encoding) Voss (1992) that encodes four nucleotides with a binary vector of four bits (A-[1,0,0,0], C-[0,1,0,0], etc.). The binary numbers for each nucleotide are orthonormal to each other and have identical Hamming distance. The one-hot encoding scheme is not only applied to protein coding regions prediction, but also a large number of applications Alipanahi *et al.* (2015); Min *et al.* (2017); Du *et al.* (2018); Zuallaert *et al.* (2018). In contrast, the discrete model exerts efforts on engineering a set of features based on prior knowledge from a biological sequence. Some widely used biological features include the codon usage Staden and McLachian (1982), codon prototype Shepherd and J. (1981), hexmer usage Claverie *et al.* (1990), and Z curves of biological sequence Chun-Ting and Ren (1991), which have been comprehensively reviewed by Fickett and Tung (1992).

The abovementioned two models have both merits and demerits. In fact, data representation in genome analysis plays an important role in computational methods to learn relevant biological features Yu *et al.* (2018); Kalkatawi *et al.* (2019), however, effectively encoding biological sequences and building computational methods for features learning remains uncertain Yu *et al.* (2018). The sequential model preserves the global sequences order information Chen *et al.* (2014) but computational methods could not fully capture biological features by this model. As mentioned by Rajapakse and Ho (2005); Li *et al.* (2004), it is not easy for neural networks to learn high-order correlations from extremely low-level inputs, e.g., a string of nucleotides. The work Fu *et al.* (2019) also claim that sequential model like one-hot encoding may contain limited useful information compared to other objects like images or sounds that is more suitable for deep learning. Recently, Choong and Lee (2017) claims that one-hot encoding is unable to capture the frequency domain of features like kmer. On the contrary, a discrete model like kmer representation of biological sequences is a feature that has proved to be a successful means to discriminate between coding and non-coding regions for the fact that the distribution over the 64 different codons is significantly different in coding regions compared to noncoding regions Axelson-Fisk (2010). Despite the effectiveness of kmer, it can only incorporate local sequence order information between the most contiguous nucleotides and none of the global sequence order information can be reflected Chen *et al.* (2014).

Based on the aforementioned analysis, we explore how to enhance the prediction of protein coding regions in genomic and transcript sequences by integrating sequential with the discrete model. We propose a novel method for protein coding regions prediction by using a hybrid convolutional neural network Lecun *et al.* (2015) and bidirectional recurrent neural network Schuster and Paliwal (1997) framework (CNN-BRNN), which effectively exploits global sequence order information, non-overlapping kmer features Hatzigeorgiou (2002), and statistical dependencies among coding labels. Evaluated on genomic and transcript sequences, our method gives an excellent prediction performance, which significantly outperforms existing state-of-the-art method. There are three contributions which may explain the excellent performance of our proposed framework:

- We present a CNN-BRNN framework for protein coding regions prediction both in genomic and transcript sequences, which significantly outperforms existing state-of-the-art methods.
- We exploit a hybrid encoding (e.g., C2 Arniker *et al.* (2011) and kmer) for protein coding regions prediction for the first time, it fuses global sequence order information and kmer features simultaneously, which demonstrate improved prediction performance over using each single encoding.
- Inspired by our previous work Wei *et al.* (2020), we extend label dependencies to genomic sequences and significantly improve the prediction performance on genomic sequences over existing methods.

2 Related Work

We here review the most relevant works to us. A few previous works demonstrate that combining sequence information with biological features can bring considerable performance improvement in specific applications. It is firstly introduced for promoter prediction in Xie *et al.* (2006), who proposes a method called PromoterExplorer which integrates various biological features (e.g., local distribution of pentamers, positional CpG island features) with digitized DNA sequence in a cascade AdaBoost-based classifier and achieves promising performance. The work Chen *et al.* (2014) presents a sequence-based predictor, called iTIS-PseTNC, for identifying translation initiation site in human genes and claims that using kmer representation of DNA sequences only reflect local sequence order information but lose all global sequence order information. They Chen *et al.* (2014) remedy this by using a collaborative representation called pseudo trinucleotide composition which incorporates the physicochemical properties into DNA sequence and combines with kmer features. Recently, in the work of Fu *et al.* (2019), they propose a hybrid sequence-based deep learning model called MHCpG, which integrates MeDIP-seq data with Histone information to predict DNA methylated CpG states, it exceeds the other approaches and gained more satisfactory promoter prediction performance owing to fusing multiple biological relevant features and sequence information. Moreover, the work Pan and Shen (2017) proposes a deep learning-based framework (iDeep) by using a novel hybrid convolutional neural network and deep belief network to predict the RNA-binding proteins (RBP) interaction sites and motifs on RNAs. This new protocol is featured by transforming the original observed data into a high-level abstraction feature space using multiple layers of learning blocks, where the shared representations across different domains are integrated. All the above works give us a strong intuition that there exists a complementary relationship between sequence information and biological features like kmer feature, and we could improve the prediction performance of kmer based protein coding regions prediction methods by incorporating global sequence order information.

3 Method

In this section, definitions of problems, data representation of biological sequences, the CNN-BRNN framework for protein coding regions prediction are introduced. The graphical illustration of the proposed method is shown in Figure 1.

3.1 Preliminaries

In what follows, $\boldsymbol{s} = s_1 s_2 ... s_n$ is a biological sequence (e.g., DNA or mRNA), where $s_i \in \{A, C, T, G\}$, and $\boldsymbol{y} = y_1 y_2 ... y_n$ is the label sequence of $\boldsymbol{s}$, where $y_i \in \{1, 0\}$ denote the position i in $\boldsymbol{s}$ is coding ($y_i = 1$) or not ($y_i = 0$). Then the protein coding regions prediction is equivalent to solve the following *maximum a posteriori* (MAP) estimation

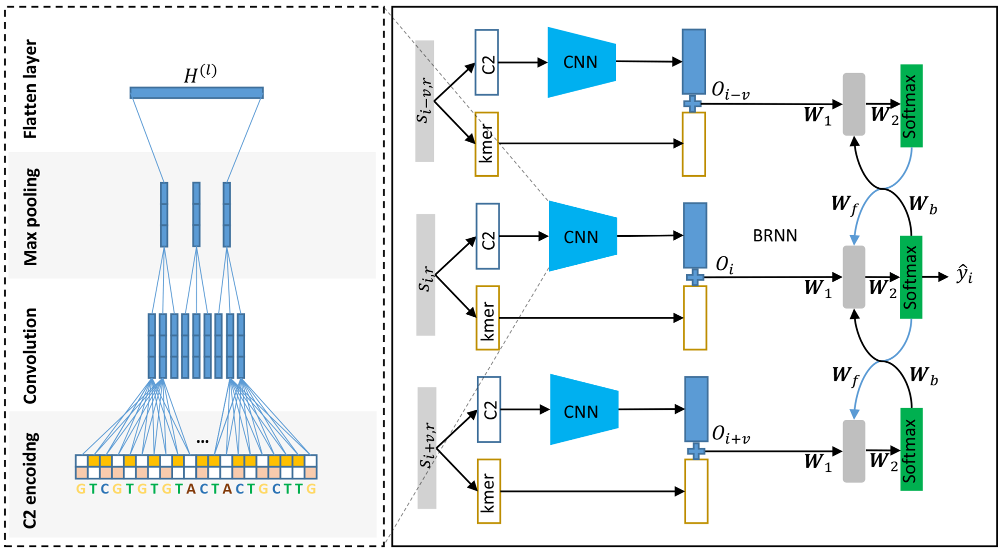

Fig. 1. A graphical illustration of the proposed CNN-BRNN architecture for protein coding regions prediction. For each position in a biological sequence, the current subsequence and its neighboring subsequences are firstly encoded into C2 and kmer encoding, then C2 encoding into a CNN and merges with kmer, which are finally fed into a BRNN for protein coding regions prediction.

problem

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{s}) \tag{1}$$

Almost all previous machine learning methods regard protein coding regions prediction as an independent binary classification problem and adopt a sliding window strategy to discriminate coding or non-coding, then the Eq. 1 can be formulated as:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} \prod_{i=1}^{n} p(y_i|\boldsymbol{s}_{i,r}) \tag{2}$$

and the predictions can be made separately, in the form

$$y_i^* = \arg\max_{y_i} p(y_i|\boldsymbol{s}_{i,r}) \tag{3}$$

where $\boldsymbol{s}_{p,r}$ indicate a subsequence of $\boldsymbol{s}$ centered at position p with a fixed length window $2 \times r + 1$. In our previous work, we demonstrate the significance of exploiting label dependencies among coding labels and improve the prediction performance in transcripts. In this work, we also extend the label dependencies to genomic sequences. Hence, for a position i in a genomic or transcript sequence, we here consider the following MAP problem:

$$y_i^* = \arg\max_{y_i} p(y_i|\boldsymbol{s}_{i,r}, \boldsymbol{y}_{\neg i,u}) \tag{4}$$

where $\neg$ denote not operator and $\boldsymbol{y}_{\neg i,u} = y_{i-u}...y_{i-1}y_{i+1}...y_{i+u}$. It can be observed from Eq. 4 that whether the position i in $\boldsymbol{s}$ is coding or not depends on not only its local region $\boldsymbol{s}_{i,r}$, but also its neighboring coding labels $\boldsymbol{y}_{\neg i,u}$. This characteristic resembles the linear-chain conditional random fields (CRF) Lafferty *et al.* (2001) that encodes state features and transition features. State features encode the content properties of the current position, while transition features focus on state transition information (e.g., in our work, coding to coding or coding to non-coding). It is worth emphasizing that the main difference of our model from CRF is that, we here consider long-range dependencies between labels while CRF consider the most neighboring two labels (e.g., y_i and y_{i-1}). In the following subsection, we introduce a CNN-BRNN architecture to effectively estimate conditional probability $p(y_i|\boldsymbol{s}_{i,r}, \boldsymbol{y}_{\neg i,u})$.

3.2 CNN-BRNN for protein coding regions prediction

3.2.1 Hybrid encoding

The basic problem of machine learning methods for protein coding regions prediction is designing an effective encoding scheme for a biological sequence. Considering the demerits of the sequential model and discrete model, we propose a hybrid encoding scheme that combines the sequential model and discrete model, which could exploit the joint merits of each model. Given a subsequence $\boldsymbol{s}_{i,r}$, the hybrid encoding can be represented as:

$$\boldsymbol{E} = [E_1; E_2] \tag{5}$$

we here adopt a sequential model such as C2 Arniker *et al.* (2011) and a discrete model like non-overlapping kmer representation, respectively. Hence, E_1 and E_2 can be respectively formulated as:

$$E_1 = [C2(s_{i-r}), C2(s_{i-r+1}), ..., C2(s_{i+r})] \tag{6}$$

and

$$E_2 = [f(AAA), f(AAC), ..., f(TTT)] \tag{7}$$

where $C2$ converts a nucleotide into 2-bit binary (e.g., A-[0,0], C-[1,1],G-[1,0],T-[0,1]) and $f(AAA)$ counts the occurrence frequency of non-overlapping trinucleotides AAA in biological sequence. For example, $\boldsymbol{s}_{i,r} = AAAAACCC$ would have two codons $f(AAA) = 1$ and $f(AAC) = 1$.

Note that C4 encoding Arniker *et al.* (2011) is also a sequential model that preserves the global sequence order information, however, it is more computationally expensive than C2 encoding, hence we here adopt C2 encoding to substitute for C4 encoding. Meanwhile, the sliding window size is relevant to prediction performance, the use of a smaller size can increase the accuracy on the border of exon but leads to of higher rate of false positive prediction Hatzigeorgiou *et al.* (1996). They Hatzigeorgiou *et al.* (1996); Shuo and Yi-sheng (2009) use 91 whereas we adopt 90 in practice for convenience of counting the number of codons. The slight difference has almost no effect on the prediction performance.

3.2.2 CNN-BRNN

Given hybrid encoding for each sliding window $\boldsymbol{s}_{i,r}$, another problem is how to build an effective machine learning method to integrate global sequence order information, non-overlapping kmer features, and label dependencies. We present a hybrid CNN-BRNN architecture to achieve this goal. The graphical illustration is shown in Figure 1.

The hybrid CNN-RNN architecture has been successfully applied to many applications including image segmentation Wang *et al.* (2016), speech emotion recognition Yao *et al.* (2020). It provides a very natural way for feature extraction and statistical dependency modeling. As one part of CNN-RNN, CNN Lecun *et al.* (2015) is a specialized feedforward neural network, which is characterized by the presence of convolutional layers that use a stack of convolutional kernels to detect local patterns. Typically, a CNN consists of an input layer, multiple pairs of convolutional-pooling layers, a flatten layer, one or more fully connected layers, and the last softmax layer. The convolutional layer is the most crucial part of CNN. The output of a layer comes from its previous layer convolved with a set of filters, that is

$$H^{(k)} = \sigma(\boldsymbol{W}^{(k-1)} \otimes H^{(k-1)} + b^{(k)}) \tag{8}$$

where $H^{(k)}, W^{(k)}$, and $b^{(k)}$ respectively denote the feature map, convolutional filter, and biases of k-th layer, $H^{(0)} = E_1$, σ denotes an activation function that is usually employed to guarantee the non-linearity of neural network. The most popular activation function is the rectified linear unit (ReLU) defined as $ReLU(x) = \max(0, x)$. In contrast, as shown in Figure 1, we employ a CNN architecture that receives two inputs, which separates two kinds of features (e.g., C2 encoding and kmer) by feeding them into additional univariate networks summed at the flatten layer of CNN, and then for a fixed window $\boldsymbol{s}_{i,r}$, the flatten layer of CNN can be formulated as:

$$O_i = H^{(l)} \oplus E_2 \tag{9}$$

where $\oplus$ is the concatenation operator. l denotes the flatten layer of CNN. We adopt CNN to incorporate global sequence information and non-overlapping kmer features in viewing of its capabilities of modeling non-linearities and capturing local patterns such as codon. It is worth noting that this network architecture is very common in recent works Ghafoorian (2017); Zhehuan *et al.* (2016) which claim that a better performance can be obtained when domain knowledge is incorporated into CNN.

The other part of CNN-RNN is an RNN which has been successfully applied to bioinformatics. Sequence data usually exhibit statistical dependencies and consider these dependencies usually yield performance benefits. There exists a considerable number of works that exert efforts to exploit statistical dependencies in DNA sequences, such as quantifying the function of DNA sequences Daniel and Xie (2016), subcellular protein localization Snderby *et al.* (2015), protein secondary structure prediction Spencer *et al.* (2015), segmentation of DNA sequences Cheng *et al.* (2012). In our previous work Wei *et al.* (2020), we demonstrate that label dependencies among coding labels play an important role in protein coding regions prediction for transcript sequences. In this work, we extend this characteristic to genomic sequences and adopt the same BRNN architecture in Wei *et al.* (2020). Instead of estimating the conditional probability $p(y_i|\boldsymbol{s}_{i,r}, \boldsymbol{y}_{\neg i,u})$ that has high-order dependencies among coding labels, the BRNN architecture in Wei *et al.* (2020) consider two-order dependencies among coding labels, and hence more computationally efficient, it reduces the problem of Eq. 4 to the following formula:

$$y_i^* = \arg\max_{y_i} p(y_i|\boldsymbol{s}_{i,r}, y_{i-v}, y_{i+v}) \tag{10}$$

where v is a step interval that defines how far that two positions correlate. As shown in Figure 1, after obtaining the output of CNN for three subsequences, the forward and backward pass of BRNN can be formulated as:

$$I_{i-v} = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_f I_{i-2v} + \boldsymbol{W}_1 O_{i-v})))) \tag{11}$$

$$J_{i+v} = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_b J_{i+2v} + \boldsymbol{W}_1 O_{i+v})))) \tag{12}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_b, \boldsymbol{W}_f$ respectively denote the weight matrices in the first hidden layer, second hidden layer, forward recurrent layer, backward recurrent layer of BRNN. σ_1 and σ_2 denote sigmoid and softmax activation function, respectively. I_i and J_i respectively denote the forward and backward passing message in a position i of a sequence. I_0 and J_0 denote the initial states with constant zero entries. From Eq. 11, we can see that the forward passing message I_i in a position i is composed of the forward passing message in a position $i-v$, and the output of CNN in the position i. Similarly, the backward passing message J_i in the position i is composed of the backward passing message in a position $i + v$, and the output of CNN in the position i. Actually, I_i and J_i is the estimations of y_i, and the difference lies in that I_i determines y_i by the current and past information whereas J_i determines y_i by the current and future information. Finally, the prediction for the sample $\boldsymbol{s}_{i,r}$ can be formulated as:

$$\hat{y}_i = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_f I_{i-v} + \boldsymbol{W}_1 O_i + \boldsymbol{W}_b J_{i+v})))) \tag{13}$$

where $\hat{y}_i$ indicates how likely is it that the nucleotide in the center of the sliding window is coding. From Eq. 13, we can see that the prediction of a sample $\boldsymbol{s}_{i,r}$ is dependent on the feedbacks from its neighboring positions $i - v$ and $i + v$, and the output O_i of CNN in the position i.

Note that the step interval v must be a multiple of three for the reason that in practice the coding label sequence $\boldsymbol{y}$ is actually not always 1 in open reading frame, but shows a periodicity of three nucleotides (e.g., [1,0,0,1,0,0,...]). Meanwhile, as confirmed by experiments in Wei *et al.* (2020), the setting of v is significantly relevant to the prediction performance. Theoretically, the most neighboring positions (v=3) contribute the most to position i, while the reverse is true in our situation. If v is set small, $\boldsymbol{s}_i$ is almost the same as $\boldsymbol{s}_{i-v}$ and $\boldsymbol{s}_{i+v}$, in which case I_{i-v} and J_{i+v} provide almost no additional information for the prediction $\hat{y}_i$. Hence, the setting of v is optimal when it equals the sliding window size, in which case $\boldsymbol{s}_i$, $\boldsymbol{s}_{i-v}$ and $\boldsymbol{s}_{i+v}$ is completely different so that I_{i-v} and J_{i+v} provide the most information for the prediction $\hat{y}_i$.

4 Result

In this section, we conduct three experiments on four gene datasets. The first is to prove the significance of hybrid encoding. In the second experiment, we make a comparison of the proposed method to existing state-of-art methods such as C4+MLP Hatzigeorgiou *et al.* (1996), C4+SVM Shuo and Yi-sheng (2009), Z curve+LDA Zhang *et al.* (1998), kmer+MLP Guigó (1997); Hatzigeorgiou (2002), kmer+SVM Tzanis *et al.* (2012), kmer+BRNN Wei *et al.* (2020). The goal of the last experiment is to evaluate the time cost of the proposed method. The network architectures are shown in Table 1.

4.1 Datasets

Four datasets are collected for performance comparison, including two genomic datasets, e.g., H10873, BGHM975, and two transcript datasets, e.g., H24842 and M19900. H10873 is built by extracting human chromosome 1-22 from Refseq build 37.3. A total number of 10873 sequences are obtained after removing repeated entries. H24842 and M19900 come from our previous work Wei *et al.* (2020). We adopt the hold-out strategy, randomly selecting 8000, 20000, 15000 sequences as training set to test the remaining 2873, 4842, and 4900 sequences, for H10873, H24842, and M19900 datasets, respectively. We refer to Bernal

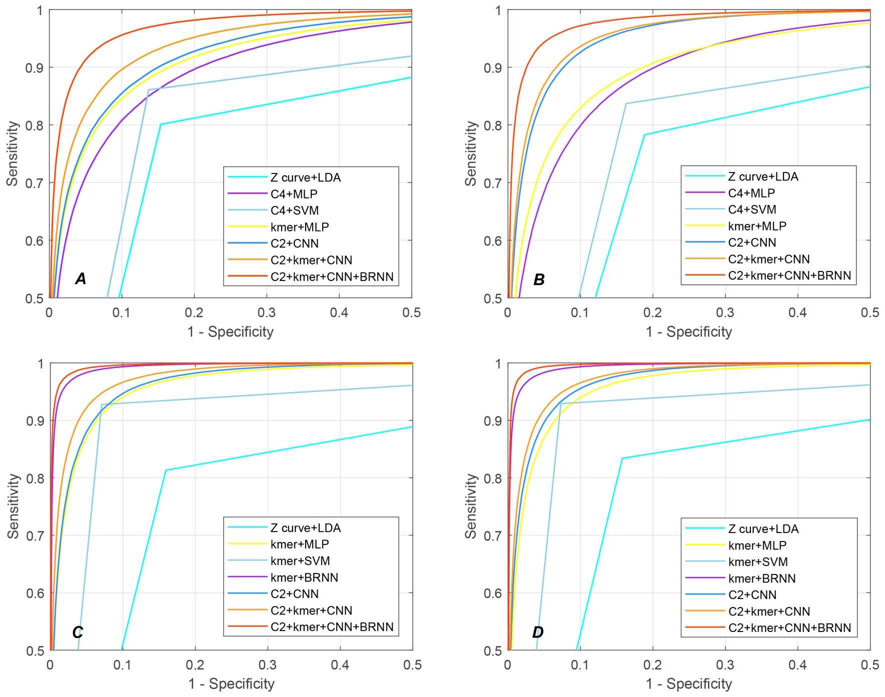

Fig. 2. Performance comparison of the proposed method with C4+MLP, C4+SVM, Z curve+LDA, kmer+MLP, kmer+SVM and kmer+BRNN. (A) the ROC curves on BGHM975 dataset; (B) the ROC curves on H10873 dataset; (C) the ROC curves on H24842 dataset; (D) the ROC curves on M19900 dataset.

et al. (2007) and select 1284 human genomic sequences as training set for BGHM975. To avoid the imbalanced data problem, negative examples are chosen such that their number equals that of the positive examples.

4.2 Performance measurements

In order to evaluate the performance of the proposed method for coding regions prediction, the analysis in this paper employs three evaluation criteria in terms of Sensitivity (Sn), Specificities (Sp) and Area Under the Receiver Operating Characteristic curve (auROC). All these criteria are based on the notions of TP, FP, TN, and FN, which correspond to number of true positives, false positives, true negatives, and false negatives. In a ROC, one typically plots the true positive rate (TPR=TP/(TP+FN)) as a function of the false negative rate (FNR=FN/(FN+TN)). The auROC can be calculated by using the trapezoidal areas created between each ROC points. The detailed definition can be found in Mitchell *et al.* (1997); Davis and Goadrich (2006).

4.3 Significance of hybird encoding

In order to prove the effectiveness of hybrid encoding, we conduct an ablation study to separate the hybrid encoding and observe the prediction performance for each single encoding. To be specific, hybrid encoding is separated into C2 and kmer encoding and then fed into CNN and MLP (e.g., C2+CNN and kmer+MLP). As it can be seen from Table 2-3, C2+kmer+CNN significantly outperform C2+CNN and kmer+MLP, both on genomic and transcript sequences. Moreover, it is worth noting that C2 encoding achieves better performance than kmer+MLP, especially in genomic sequences, which prove the effectiveness of global sequence order information, but at the cost of computational complexity. All the prediction performance decreases from transcript sequences to genomic sequences for the fact that coding regions are continuous on transcript sequences but interrupted by introns on genomic sequences. In brief, we can conclude from the result that there exists a complementation relationship between C2 encoding and kmer features, integration of which can facilitate the machine learning method to fully capture coding features.

4.4 Performance comparison with existing state-of-the art methods on genomic and transcript sequences

We compare the performance of the proposed method with that of existing methods such as C4+MLP, C4+SVM, Z curve+LDA, kmer+MLP, kmer+SVM, kmer+BRNN. All the methods are trained and evaluated with the same dataset for a fair comparison. From Table 2-3, it is observed that our proposed method performs the best among the existing methods and achieves the highest Sn, Sp and auROC scores on all the four datasets, the average Sn, Sp, auROC scores reach .929, .9498, .9843 on genomic sequences, and .9837, .977, .9976 on transcript sequences, respectively. We also plot the ROC curve on genomic and transcript sequences. As it can be seen in Figure 2, given a fixed false positive rate of .05, our proposed method respectively achieves a sensitivity of .913, .9433, on BGHM975, H10873, an improvement of .1518, .2084 over the second best method, kmer+MLP. Meanwhile, given a fixed false positive rate of .025, the proposed method respectively achieves a sensitivity of .9791, .984 on H24842 and M19900, an improvement of .014, .0185 over the second best method, kmer+BRNN. All the results demonstrate that our proposed method is a high-accuracy protein coding regions prediction method, especially in transcript sequences.

4.5 Time cost of the proposed method

We further briefly analyze the computational cost of the proposed method. All the experiments are conducted on an Intel Core i5-10400 CPU 2.90 GHz PC with 16 GB RAM. The proposed method is implemented mainly in Tensorflow and partly in Matlab. Table 4 gives the time cost of the proposed method on four datasets. It can be observed that a larger dataset requires more additional training time. The training and test process is performed efficiently on all four datasets.

Table 1. Different network architectures in experiments.

Layer	C4+MLP	kmer+MLP	C2+CNN	kmer+BRNN	C2+kmer+CNN		C2+kmer+CNN+BRNN	
input layer	360	64	90 × 2	3 × 64	64	90 × 2	3 × 64	3 × 90 × 2
conv layer	-	-	80 × [7,2]	-	-	80 × [7,2]	-	80 × [7,2]
maxpool layer	-	-	[2,1]	-	-	[2,1]	-	[2,1]
dropout layer	-	-	0.5	-	-	0.5	-	0.5
conv layer	-	-	80 × [3,1]	-	-	80 × [3,1]	-	80 × [3,1]
maxpool layer	-	-	[2,1]	-	-	[2,1]	-	[2,1]
dropout layer	-	-	0.5	-	-	0.5	-	0.5
dense layer	100	6	20	5		20		20
dropout layer	0.5	-	0.5	-		0.5		-
recurrent layer	-	-	-	$\boldsymbol{W}_f\boldsymbol{W}_b : 2 \times 5$		-		$\boldsymbol{W}_f\boldsymbol{W}_b : 2 \times 20$
softmax layer	2	2	2	2		2		2

Table 2. Performance comparison of DeepCoding with the other state-of-the-art methods on genomic datasets.

Method	BGHM957			H10873		
	Sn	Sp	auROC	Sn	Sp	auROC
Z curve+LDA Zhang et al. *(1998)*	.8010	.8462	-	.7829	.8114	-
C4+MLP Hatzigeorgiou et al. *(1996)*	.8564	.8558	.9342	.8423	.8671	.9328
C4+SVM Shuo and Yi-sheng (2009)	.8609	.8629	-	.8369	.8369	-
kmer+MLP Guigó (1997)	.8648	.8815	.9451	.8685	.8606	.9378
C2+CNN	.8739	.8814	.9508	.8940	.9275	.9717
C2+kmer+CNN	.8760	.9181	.9632	.9104	.9258	.9745
C2+kmer+CNN+BRNN	**.9215**	**.9439**	**.9816**	**.9365**	**.9558**	**.9870**

Table 3. Performance comparison of DeepCoding with the other state-of-the-art methods on transcript datasets.

Method	H24842			M19900		
	Sn	Sp	auROC	Sn	Sp	auROC
Z curve+LDA Zhang et al. *(1998)*	.8131	.8407	-	.8339	.8422	-
kmer+MLP Hatzigeorgiou (2002)	.9243	.9177	.9746	.9321	.9086	.9746
kmer+SVM Tzanis et al. *(2012)*	.9276	.9292	-	.9291	.9271	-
kmer+BRNN Wei et al. *(2020)*	.9682	.9713	.9951	.9689	.9716	.9954
C2+CNN	.9209	.9261	.9775	.9446	.9146	.9804
C2+kmer+CNN	.9504	.9258	.9844	.9464	.9281	.9838
C2+kmer+CNN+BRNN	**.9837**	**.9744**	**.9976**	**.9837**	**.9799**	**.9977**

Table 4. Brief description of time cost on four datasets with regard to the proposed method.

Datasets	Type	Sequence Number (train/test)	Coding Number (train/test)	Time Cost (s)
BGHM957	Genomic	1283/957	1,023,874/497,110	760
H10873	Genomic	8000/2873	1,778,488/632,030	1886
H24842	Transcript	20000/4842	1,828,898/438,710	1972
M19900	Transcript	15000/4900	1,751,904/578,198	1856

5 Conclusions

In summary, protein coding regions prediction is a very important but overlooked subtask for tasks such as prediction of complete gene structure, coding/non-coding RNA. However, it is still a lack of effective computational methods to learn coding features from genomic and transcript sequences. Indeed, coding features in biological sequences usually exhibit heterogeneity (e.g., global sequence order information, frequency domain of features like kmer, statistical dependencies among coding labels) and are difficult to capture by using a single encoding scheme and machine learning method. In this paper, we present a deep learning framework with hybrid encoding for protein coding regions prediction, which effectively incorporates the three kinds of features into a hybrid CNN-BRNN architecture and achieves a remarkable prediction performance when compared with existing state-of-the-art methods on genomic and transcript sequences.

Funding

This work is supported by the Natural Science Foundation of China under Grants 11674352, and 91853123.

References

Al-Turaiki, I. M., Mathkour, H., Touir, A., and Hammami, S. (2011). Computational approaches for gene prediction: a comparative survey. In *International Conference on Informatics Engineering and Information Science*, pages 14–25. Springer.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, **33**(8), 831–838.

Arniker, S. B., Kwan, H. K., Law, N.-F., and Lun, D. P.-K. (2011). Dna numerical representation and neural network based human promoter prediction system. In *2011 Annual IEEE India Conference*, pages 1–4. IEEE.

Axelson-Fisk, M. (2010). Comparative gene finding: Models, algorithms and implementation.

Bernal, A., Crammer, K., Hatzigeorgiou, A., and Pereira, F. (2007). Global discriminative learning for higher-accuracy computational gene prediction. *PLoS* *computational biology*, **3**(3), e54.

Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic dna. *Journal of molecular biology*, **268**(1), 78–94.

Catherine, M., Marie-France, S., Thomas, S., and Pierre, R. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research*, **30**(19), 4103–17.

Chen, W., Feng, P.-M., Deng, E.-Z., Lin, H., and Chou, K.-C. (2014). itis-psetnc: a sequence-based predictor for identifying translation initiation site in human genes using pseudo trinucleotide composition. *Analytical biochemistry*, **462**, 76–83.

Cheng, W. C., Huang, J. C., and Liou, C. Y. (2012). Segmentation of dna using simple recurrent neural network. *Knowledge-Based Systems*, **26**(none), 271–280.

Choong, A. C. H. and Lee, N. K. (2017). Evaluation of convolutionary neural networks modeling of dna sequences using ordinal versus one-hot encoding method. In *2017 International Conference on Computer and Drone Applications (IConDA)*, pages 60–65. IEEE.

Chun-Ting, Z. and Ren, Z. (1991). Analysis of distribution of bases in the coding sequences by a digrammatic technique. *Nuclc Acids Research*, (22), 6313.

Claverie, J. M., Sauvaget, I., and Bougueleret, L. (1990). K-tuple frequency analysis: From intron/exon discrimination to t-cell epitope mapping. *Methods Enzymol*, **183**, 237–252.

Daniel, Q. and Xie, X. (2016). Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic Acids Research*, (11), e107–e107.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Du, X., Yao, Y., Diao, Y., Zhu, H., Zhang, Y., and Li, S. (2018). Deepss: Exploring splice site motif through convolutional neural network directly from dna sequence. *IEEE Access*, **6**, 32958–32978.

Fickett, J. W. and Tung, C. S. (1992). Assessment of protein coding measures. *Nucleic Acids Research*, **20**(24), 6441–50.

Fu, L., Peng, Q., and Chai, L. (2019). Predicting dna methylation states with hybrid information based deep-learning model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Ghafoorian, M., K. N. H. T. v. U. I. W. S. C. I. L. G. P. B. (2017). Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *Scientific Reports*, **7**(1), 1–12.

Guigó, R. (1997). Dna composition, codon usage and exon prediction. *Genetic Databases*, pages 53–80.

Hatzigeorgiou, A., Mache, N., and Reczko, M. (1996). Functional site prediction on the dna sequence by artificial neural networks. In *Proceedings IEEE International Joint Symposia on Intelligence and Systems*, pages 12–17. IEEE.

Hatzigeorgiou, A. G. (2002). Translation initiation start prediction in human cdnas with high accuracy. *Bioinformatics*, **18**(18), 343–350.

Kalkatawi, M., Magana-Mora, A., Jankovic, B., and Bajic, V. B. (2019). Deepgsr: an optimized deep-learning structure for the recognition of genomic signals and regions. *Bioinformatics*, **35**(7), 1125–1132.

Lafferty, J., Mccallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436.

Li, A., Zhang, J., and Zhou, Z. (2014). Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme. *BMC bioinformatics*, **15**(1), 311.

Li, J., Liu, H., Wong, L., and Yap, R. H. (2004). Techniques for recognition of translation initiation sites. In *The Practical Bioinformatician*, pages 71–89. World Scientific.

Min, X., Zeng, W., Chen, S., Chen, N., Chen, T., and Jiang, R. (2017). Predicting enhancers with deep convolutional neural networks. *BMC bioinformatics*, **18**(13), 478.

Mitchell, T. M., Carbonell, J. G., and Michalski, R. S. (1997). *Machine Learning*.

Pan, X. and Shen, H. B. (2017). Rna-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *Bmc Bioinformatics*, **18**(1), 136.

Rajapakse, J. C. and Ho, L. S. (2005). Markov encoding for detecting signals in genomic sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **2**(2), 131–142.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, **45**(11), 2673–2681.

Shepherd and J., C. (1981). Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence and its possible evolutionary justification. *Proceedings of the National Academy of ences of the United States of America*, **78**(3), 1596–1600.

Shuo, G. and Yi-sheng, Z. (2009). Prediction of protein coding regions by support vector machine. In *2009 International Symposium on Intelligent Ubiquitous Computing and Education*, pages 185–188. IEEE.

Snderby, S. K., Snderby, C. K., Nielsen, H., and Winther, O. (2015). Convolutional lstm networks for subcellular localization of proteins. In *Algorithms for Computational Biology*.

Spencer, M., Eickholt, J., and Cheng, J. (2015). A deep learning network approach to ab initio protein secondary structure prediction. *Computational Biology & Bioinformatics IEEE/ACM Transactions on*, **12**(99), 103–112.

Staden, R. and McLachian, A. D. (1982). Codon preference and its use in identifying protein coding regions in long dna sequences. *Nucleic Acids Research*, **10**, 141–156.

Stanke, M., Steinkamp, R., Waack, S., and Morgenstern, B. (2004). Augustus: a web server for gene finding in eukaryotes. *Nucleic acids research*, **32**(suppl_2), W309–W312.

Tong, X. and Liu, S. (2019). Cppred: coding potential prediction based on the global description of rna sequence. *Nucleic acids research*, **47**(8), e43–e43.

Tuo, S., Liu, H., and Chen, H. (2020). Multi-population harmony search algorithm for the detection of high-order snp interactions. *Bioinformatics*.

Tzanis, G., Berberidis, C., and Vlahavas, I. (2012). Stacktis: A stacked generalization approach for effective prediction of translation initiation sites. *Computers in Biology & Medicine*, **42**(1), 61–69.

Voss, R. F. (1992). Evolution of long-range fractal correlations and 1/f noise in dna base sequences. *Physical review letters*, **68**(25), 3805.

Wang, J., Yang, Y., Mao, J., Huang, Z., and Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification.

Wei, C., Zhang, J., Yuan, X., He, Z., Liu, G., and Wu, J. (2020). Neurotis: Enhancing the prediction of translation initiation sites in mrna sequences via a hybrid dependency network and deep learning framework. *Knowledge-Based Systems*, page 106459.

Xie, X., Wu, S., Lam, K., and Yan, H. (2006). Promoterexplorer: an effective promoter identification method based on the adaboost algorithm. *Bioinformatics*, **22**(22), p. 2722–2728.

Yao, Z., Wang, Z., Liu, W., Liu, Y., and Pan, J. (2020). Speech emotion recognition using fusion of three multi-task learning-based classifiers: Hsf-dnn, ms-cnn and lld-rnn. *Speech Communication*, **120**, 11–19.

Yu, N., Li, Z., and Yu, Z. (2018). Survey on encoding schemes for genomic data representation and feature learning—from signal processing to machine learning. *Big Data Mining and Analytics*, **1**(3), 191–210.

Yuan, X., Zhang, J., Yang, L., Bai, J., and Fan, P. (2017). Detection of significant copy number variations from multiple samples in next-generation sequencing data. *IEEE transactions on nanobioscience*, **17**(1), 12–20.

Zhang, C. T., Lin, Z. S., Yan, M., and Zhang, R. (1998). A novel approach to distinguish between intron-containing and intronless genes based on the format of z curves. *Journal of Theoretical Biology*, **192**(4), 467–73.

Zhehuan, Z., Zhihao, Y., Ling, L., Hongfei, L., and Jian, W. (2016). Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, (22), btw486.

Zuallaert, J., Kim, M., Soete, A., Saeys, Y., and Neve, W. D. (2018). Tisrover: Convnets learn biologically relevant features for effective translation initiation site prediction. *International Journal of Data Mining and Bioinformatics*, **20**(3), 267–284.