Digital Signal Processing 117 (2021) 103202

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

DeepTIS: Improved translation initiation site prediction in genomic sequence via a two-stage deep learning model

Chao Wei, Junying Zhang *, Xiguo Yuan

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

ARTICLE INFO

Article history:
Available online 12 August 2021

Keywords:
Deep learning
Bioinformatics
Translation initiation site prediction
Features fusion
Label dependency

ABSTRACT

Translation initiation site (TIS) prediction is one of the most crucial subtasks for gene annotation. Many computational methods have been proposed and achieved acceptable accuracy in transcripts (e.g., cDNA, mRNA). However, the prediction of TIS at the genome level is far more challenging and the computational methods for TIS prediction in genomic sequences so far reach modest performance. Recently, we proposed a method that improves the prediction of TIS in mRNA sequences and demonstrated the significance of explicitly modeling coding features. In this paper, we extend the same results to genomic sequence and present a two-stage deep learning model for TIS prediction in genomic sequence: the first stage to extract coding contrast features around TIS by a hybrid Convolutional Neural Network-Bidirectional Recurrent Neural Network architecture (Content-RCNN), and the second stage to integrate coding contrast features around TIS with TIS sequence encoded by one-hot encoding to jointly predict TIS by a CNN (Integrated-CNN). Four-fold cross validation tests on genome-wide human and mouse datasets demonstrate that the proposed model yields an improved prediction performance of TIS over existing state-of-the-art methods. The source code and the dataset used in the paper are publicly available at: https://github.com/xdcwei/DeepTIS/.

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 exponential increase of sequence data. Many efforts have been dedicated to the identification of genomic mutations by using NGS datasets [43,42,41,40] in the past few years, it is urgent to find reliable genome annotation techniques for predicting genes [4]. Traditional experimental methods for gene prediction are costly and time-consuming. Hence, various computational methods have recently been focused on automatic annotation of uncharacterized DNA sequences.

In eukaryotic genomes, accurate prediction of translation initiation site (TIS) is crucial to the determination of true gene structure. TIS is the position in gene to start constructing proteins. The majority of TIS is conserved with tri-nucleotide ATG, while a few exceptions are reported in eukaryotes [13]. Identification of TIS from uncharacterized biological sequence is a challenging task. This is because (1) the conserved tri-nucleotide AUG is not sufficient to determine the true TISs due to presence of a large number of AUG in genes, which induces considerable false positives; (2) unlike highly-conserved splicing signals, TISs are surrounded by relatively poorly conserved sequences and harder to predict [1]. The situation becomes more complex for genomic sequence where coding region is interrupted by introns [29] (e.g., coding feature is more difficult to capture, scanning model [18] is not applicable).

Many existing computational methods [24,30,21,13,27,35,5,39] have been proposed for TIS prediction in transcripts during the past decades. However, works of predicting TIS at the genomic level are relatively rare, mainly including Support Vector Machine (SVM) [44,25,11], Artificial Neural Networks (ANN) [45,16]. In one of works, [44] studies Support Vector Machine (SVM) using different kinds of kernel functions for TIS prediction. They believe that carefully designing kernel functions are useful for achieving higher TIS prediction accuracy. [29] presents several simple prediction methods to identify TIS on a genomic scale. They show that simple models based on Kozak sequence [17] and the scanning model [18] can be highly effective. [25] improve the performance of the method [44] for TIS and stop codon recognition by dividing the negative class into four different groups and training one classifier for each type of negative class. Recently, [45] develop a method called TISRover, which directly learn biological

* Corresponding author.
E-mail address: jyzhang@mail.xidian.edu.cn (J. Zhang).

https://doi.org/10.1016/j.dsp.2021.103202
1051-2004/

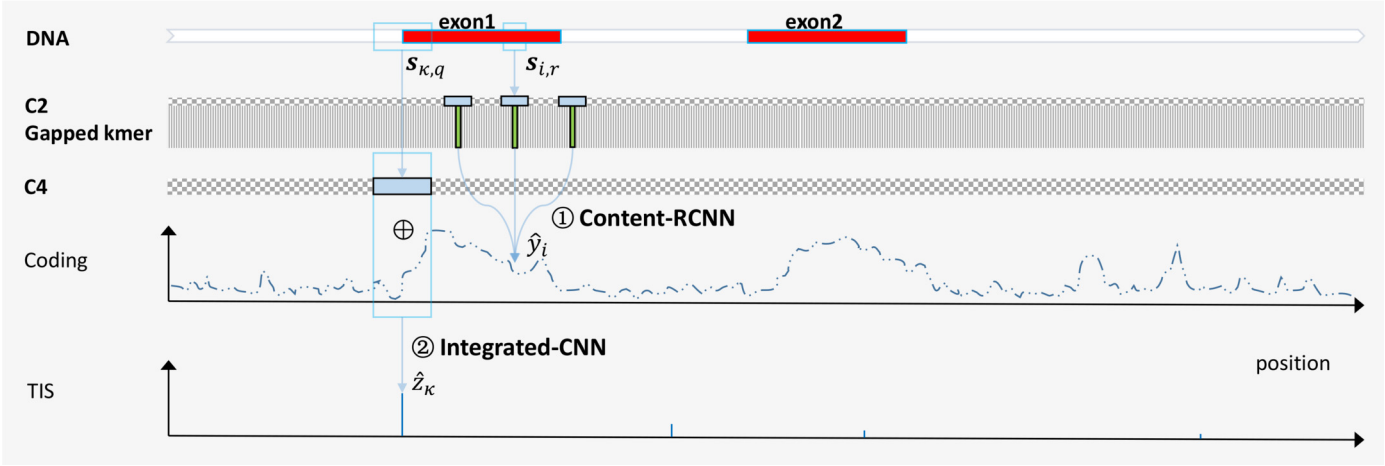

Fig. 1. Pipeline of DeepTIS. It consists of two stages: Content-RCNN and Integrated-CNN. At the first stage, Content-RCNN infers each coding score by using the current and its two neighboring subsequences encoded by C2 and gapped kmer (gkm), and at the second stage, Integrated-CNN concatenates the corresponding coding scores with TIS sequence encoded by C4 to infer TIS.

knowledge from DNA sequences with a Convolutional Neural Network (CNN) [20]. Owing to capabilities of modeling non-linearities and capturing local patterns, CNN finds the most notable features such as the Kozak sequence, the reading frame characteristics, the influence of stop and start codons in the sequence, and the presence of donor splice site patterns. To our best knowledge, this is currently the best work for TIS prediction in genomic sequence.

Despite the effectiveness of convolutional neural networks-based method for TIS prediction, it cannot fully capture coding contrast feature (the transition from a non-coding region to a coding region in the first reading frame around TIS) which is the most relevant feature for TISs prediction [29], e.g., ignorance of coding labels information, the high-order distant interactions among coding features [39]. Our recent work [39] proposed a method for TIS prediction in mRNA sequence and alleviated this problem by explicitly modeling coding contrast feature. This work gives us a strong intuition that we can extend the same conclusion from transcript to genomic sequence. However, direct extension of the method that is meant for mRNA sequence to genomic sequence is not suitable for the fact that there is severe performance reduction for coding regions prediction using codon usage measure [39,38] in genomic sequence where exons are interrupted by introns. Fortunately, our recent work [38] promotes the prediction of coding regions in genomic sequence by integrating global sequence order information, gapped kmer (gkm) [10], and coding labels dependencies.

Inspired by our previous two works [39,38], we explore how to enhance the prediction of TIS in genomic sequence in this paper. We present a two-stage deep learning model for TIS prediction in genomic sequence: the first stage to extract coding contrast features around TIS by a hybrid Convolutional Neural Network-Bidirectional Recurrent Neural Network architecture (Content-RCNN) [38], and the second stage to integrate coding contrast features around TIS with TIS sequence to jointly predict TIS by a CNN (Integrated-CNN). Experimental tests on genome-wide human and mouse datasets demonstrate that the proposed model yields an improved prediction performance of TIS over existing state-of-the-art methods.

2. Method

In this section, definition of problem, the two stages of DeepTIS are introduced. Pipeline of DeepTIS is shown in Fig. 1.

2.1. Preliminaries

In what follows, $\boldsymbol{s} = s_1 s_2 ... s_n$ is a genomic sequence and $\boldsymbol{z} = z_1 z_2 ... z_n$ is the label sequence of $\boldsymbol{s}$, where $s_i \in \{A, C, T, G\}$ and $z_i \in \{1, 0\}$. Let $\boldsymbol{s}_{p,q}$ indicate a subsequence of $\boldsymbol{s}$ centered at position p with a fixed length window $2 \times q + 1$, and then the TIS prediction is equivalent to solve the following *maximum a posteriori* (MAP) estimation problem

$$z_\kappa^* = \arg\max_{z_\kappa} p(z_\kappa | \boldsymbol{s}_{\kappa,q}) \tag{1}$$

where q denotes the half-length of TIS sequence, without loss of generality, we assume that κ denotes the position of k-th trinucleotide ATG in the sequence $\boldsymbol{s}$, and z_κ denotes whether the position κ in $\boldsymbol{s}$ is true TIS ($z_\kappa = 1$) or not ($z_\kappa = 0$).

Note that almost all machine learning based methods regard TIS prediction as an independent binary classification problem and learn the conditional probability Eq. (1) to infer TIS. However, they ignore potential dependency among coding labels. NeuroTIS [39] *explicitly* model local label dependency between coding region and TIS using a natural dependency network representation [14]. In this paper, we follow the representation of NeuroTIS and extend the results from transcript to genomic sequence. The difference lies in that we cannot exploit coding features around stop codon in genomic sequence for interrupted coding structure. Hence, we here consider the following MAP problem,

$$(z_\kappa^*, \boldsymbol{y}_{\kappa,q}^*) = \arg\max_{z_\kappa, \boldsymbol{y}_{\kappa,q}} p(z_\kappa, \boldsymbol{y}_{\kappa,q} | \boldsymbol{s}_{\kappa,q}) \tag{2}$$

where $\boldsymbol{y} = y_1 y_2 ... y_n$ is the label sequence of $\boldsymbol{s}$ and $y_i \in \{1, 0\}$ denotes whether the position i in $\boldsymbol{s}$ is coding ($y_i = 1$) or not ($y_i = 0$).

Using a greedy inference like NeuroTIS, Eq. (2) can be reduced to the following two-stage problems: (1) protein coding regions prediction by exploiting neighboring coding labels and subsequence:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \prod_{i=1}^{n} p(y_i | \boldsymbol{s}_{i,r}, y_{i-v}, y_{i+v}) \tag{3}$$

and (2) TIS prediction by integrating coding scores and TIS sequence:

$$\hat{z}_\kappa = \arg\max_{z_\kappa} p(z_\kappa | \boldsymbol{s}_{\kappa,q}, \hat{\boldsymbol{y}}_{\kappa,q}) \tag{4}$$

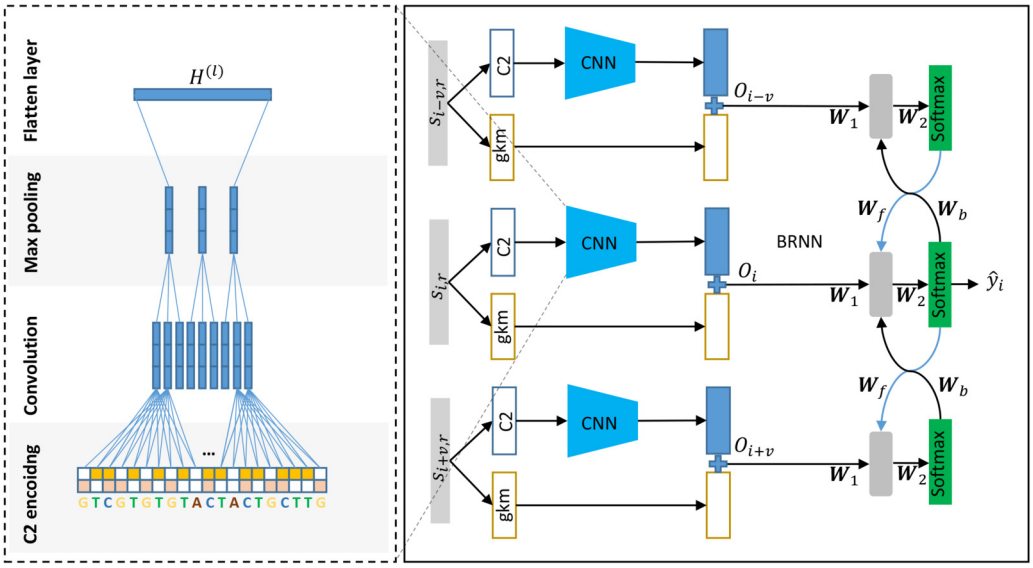

Fig. 2. A graphical illustration of Content-RCNN, a hybrid deep learning architecture for protein coding regions prediction. For each position in a biological sequence, the current subsequence and its neighboring subsequences are firstly encoded into C2 and gapped kmer features, then C2 encoding into a CNN and merges with gapped kmer, which are finally fed into a BRNN for protein coding regions prediction (cited from [38]).

where v defines how far that two positions correlate, its detailed description can be found in [39]. In the following section, we introduce two deep learning models to respectively estimate conditional probabilities $p(y_i|\boldsymbol{s}_{i,r}, y_{i-v}, y_{i+v})$ and $p(z_\kappa|\boldsymbol{s}_{\kappa,q}, \hat{\boldsymbol{y}}_{\kappa,q})$, and then predict protein coding regions and TIS.

2.2. Content-RCNN

This is the first stage of DeepTIS. We adopt the Content-RCNN model [38] to estimate the conditional probability $p(y_i|\boldsymbol{s}_{i,r}, y_{i-v}, y_{i+v})$ and address the Eq. (3). The graphical illustration is shown in Fig. 2.

As one part of Content-RCNN, CNN [20] is a specialized feed-forward 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

$$H^{(k)} = \sigma(\boldsymbol{W}^{(k-1)} \otimes H^{(k-1)} + b^{(k)}) \tag{5}$$

where $H^{(k)}, W^{(k)}$, and $b^{(k)}$ respectively denote the feature map, convolutional filter, and biases of k-th layer, 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 $ReLU(x) = \max(0, x)$.

As shown in Fig. 2, Content-RCNN employs a CNN architecture that receives two inputs at the input layer and flatten layer, respectively. It separates two kinds of features (e.g., C2 encoding and gkm) by feeding them into additional univariate networks concatenated at the flatten layer of CNN, and then for a fixed window $\boldsymbol{s}_{i,r}$, the input layer and the flatten layer of CNN can be respectively formulated as:

$$H^{(0)} = C2(\boldsymbol{s}_{i,r}) \tag{6}$$

and

$$O_i = H^{(l)} \oplus gkm(\boldsymbol{s}_{i,r}) \tag{7}$$

where $\oplus$ is the concatenation operator. l denotes the flatten layer of CNN. $C2(\cdot)$ converts each nucleotide of subsequence into 2-bit binary (e.g., A-[0,0], C-[1,1], G-[1,0], T-[0,1]) and $gkm(\cdot)$ convert a subsequence into a feature vector that counts the occurrence frequency of non-overlapping gapped kmer (640 dimensional here). We adopt CNN to incorporate global sequence order information and non-overlapping kmer features in viewing of its capabilities of modeling non-linearities and capturing local patterns such as codon.

The other part of Content-RCNN is a Recurrent Neural Network (RNN) which has been successfully applied to bioinformatics [6,31, 7,37]. As shown in Fig. 2, after obtaining the output of CNN for three subsequences, the forward and backward pass of BRNN can be formulated as:

$$I_{i-v} = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_f I_{i-2v} + \boldsymbol{W}_1 O_{i-v}))) \tag{8}$$

$$J_{i+v} = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_b J_{i+2v} + \boldsymbol{W}_1 O_{i+v}))) \tag{9}$$

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, and 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. Finally, the prediction for the sample $\boldsymbol{s}_{i,r}$ can be formulated as:

$$\hat{y}_i = \sigma_2(\boldsymbol{W}_2(\sigma_1(\boldsymbol{W}_f I_{i-v} + \boldsymbol{W}_1 O_i + \boldsymbol{W}_b J_{i+v}))) \tag{10}$$

where $\hat{y}_i$ indicates how likely is it that the nucleotide in the center of the sliding window is coding. From Eq. (10), 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.

Neural networks-based methods are widely applicable to predict protein coding regions for eukaryotic genes [2,12,36]. Most methods employ either genomic sequence or coding measures which calculate for a sliding window of genomic sequence as input into neural networks. However, as mentioned by [38], coding feature in biological sequences usually exhibits heterogeneity (e.g., global sequence order information, frequency domain of features

Table 1
Different network architectures in experiments.

ID	Network architecture	Frame/Frame-shift	Content-RCNN		Integrated-CNN
0	input layer	400×1	3×640	$3 \times 90 \times 2$	400×5
1	conv layer	$200 \times (8, 1)$	–	$250 \times (7, 2)$	$200 \times (8, 5)$
2	maxpool layer	(3,1)	–	(2,1)	(3,1)
3	dropout layer	0.3	–	0.3	0.3
4	conv layer	$200 \times (3, 1)$	–	$200 \times (3, 1)$	$200 \times (4, 1)$
5	maxpool layer	(2,1)	–	(2,1)	(2,1)
6	dropout layer	0.3	–	0.3	0.3
7	conv layer	$200 \times (3, 1)$	–		$200 \times (3, 1)$
8	maxpool layer	(2,1)	–		(2,1)
9	dropout layer	0.3	–		0.3
10	dense layer	100	30		100
11	dropout layer	0.3	–		0.3
12	recurrent layer	–	$\boldsymbol{W}_f \boldsymbol{W}_b : 2 \times 30$		–
13	softmax layer	2	2		2

Note: - means empty.

like kmer, statistical dependencies among coding labels) and is difficult to capture by using a single encoding scheme and machine learning method. Hence, we here adopt the Content-RCNN model for its remarkable protein coding regions prediction performance in genomic sequence.

2.3. Integrated-CNN

This is the second stage of DeepTIS. The coding score $\hat{\boldsymbol{y}}$ is calculated for a genomic sequence at the first stage. At this stage, we employ a CNN-based method (Integrated-CNN) to integrate coding scores with TIS sequence to estimate the conditional probability $p(z_\kappa | \boldsymbol{s}_{\kappa,q}, \hat{\boldsymbol{y}}_{\kappa,q})$ and address the Eq. (4).

How to incorporate two kinds of features into a classifier for TIS prediction? A common architecture is directly incorporating domain knowledge into fully connected layer of CNN just like [9,39,38]. However, we find that a simple concatenation of two features can achieve a little performance improvement in our application. To be specific, $\boldsymbol{s}_{\kappa,q}$ is encoded into one-hot encoding, and concatenated with $\hat{\boldsymbol{y}}_{\kappa,q}$ to form a matrix with $2 \times q + 1$ rows and 5 columns, which is finally fed into Integrated-CNN for final TIS prediction. The input layer of Integrated-CNN can be formulated as:

$$H^{(0)} = \hat{\boldsymbol{y}}_{\kappa,q} \oplus C4(\boldsymbol{s}_{i,q}) \tag{11}$$

where $C4(\boldsymbol{\cdot})$ converts each nucleotide of subsequence into 4-bit binary (also called one-hot encoding, e.g., A-[1,0,0,0], C-[0,1,0,0], T-[0,0,1,0], G-[0,0,0,1]). The pipeline and architecture of Integrated-CNN is shown in Fig. 1 and Table 1, respectively.

Note that we call this stage of DeepTIS as Integrated-CNN because it resembles another kind of gene prediction methods called integrated methods [3,33,1], which exploit multiple evidences from content and signal sensors to predict entire gene structure. These methods regard gene prediction as a sequence labeling problem and employ the structure learning frameworks (e.g., HMM [15], CRF [19]) for learning and inference. Since the various parts of a gene should influence each other in a non-local manner, these methods usually show remarkable results. Similarly, Integrated-CNN makes a simple feature concatenation between coding features and TIS sequence, which enables extracting coding features and consensus motifs simultaneously for TIS prediction.

3. Experiments

In this section, we conduct four experiments on benchmark gene datasets. The first one is to compare two network architectures that integrate coding contrast features. The second experiment is to check the first frame sensitivity of CNN. In the third experiment, we make a comparison of DeepTIS to existing state-of-art methods such as SVM [44], DeepGSR [16] and TISRover [45]. The goal of last experiment is to evaluate the time cost of DeepTIS.

3.1. Datasets

Most benchmark datasets used in TIS prediction are lack of coding labels and are not suitable here, and hence we download genomic sequences containing the entire gene structure for training and testing. We build two datasets by extracting human and mouse genomic sequences from Refseq [26] that provides comprehensive, non-redundant and well-annotated set of sequences. A total number of 19288 and 16473 genes are obtained after clean up procedure, for human and mouse datasets, respectively. TIS is selected from genomic sequence according to a fixed ratio of positive to negative samples (1:1 for balanced case and 1:5 for imbalanced case). All the samples are split into four parts to perform four-fold cross-validation.

3.2. Performance measurements

In order to evaluate the performance of DeepTIS for TIS prediction, the analysis in this paper employs four evaluation criteria in terms of Sensitivity (Sn), Specificities (Sp), area under the Receiver Operating Characteristic Curve (auROC) and Precision Recall Curve (auPRC). 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)), and in a PRC, one plots the precision (TP/(TP+FP)) as a function of the recall (TPR). The auROC and auPRC can be calculated by using the trapezoidal areas created between each ROC and PRC points, respectively. The detailed definition can be found in [23,8].

3.3. Performance comparison of two network architectures

We here compare the prediction performance of two network architectures that integrate coding contrast features at the fully-connected layer (corresponding to DeepTIS1) or input layer of CNN (corresponding to DeepTIS2). As it is shown in Table 2,3, DeepTIS2 outperform DeepTIS1 both on human and mouse datasets, which prove the effectiveness of direct concatenation of TIS sequence and coding contrast features. The underlying reason could be that there exist more feature interactions between coding contrast features and TIS sequence in DeepTIS2 than that in DeepTIS1 (feature interactions might occur earlier at the input layer for DeepTIS2 than DeepTIS1 at the fully-connected layer).

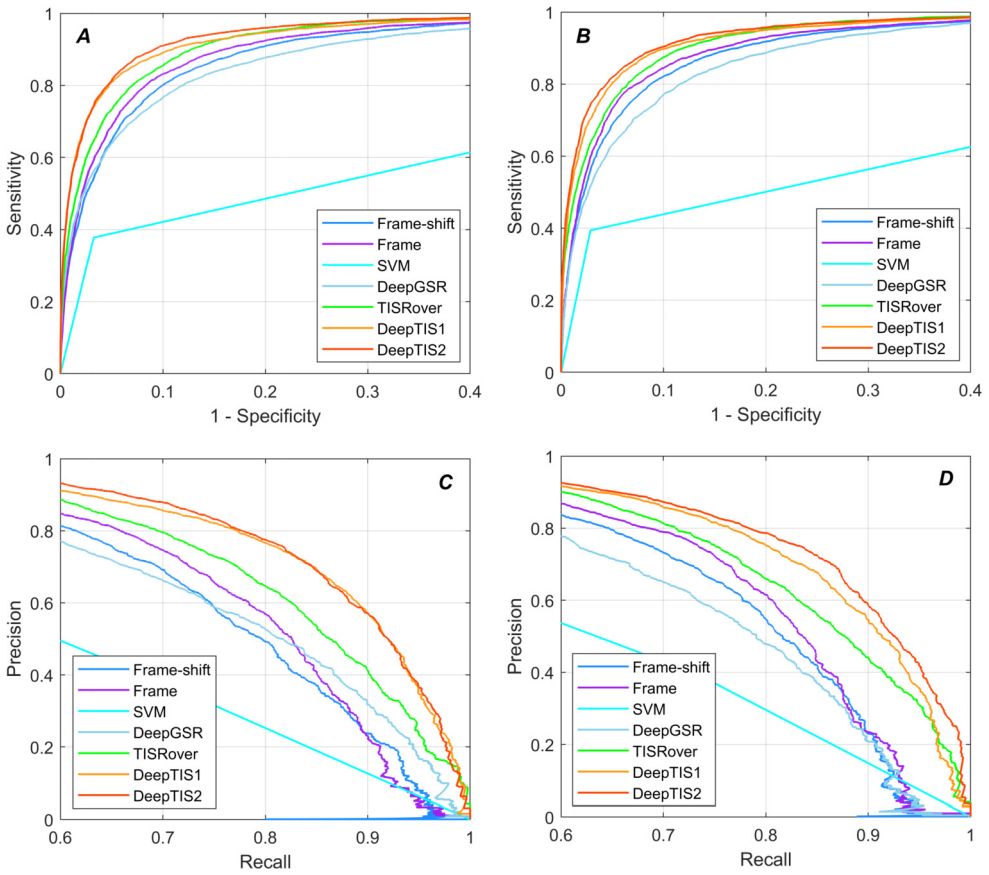

Fig. 3. Comparison results of our model, TISRover, DeepGSR, SVM, Frame and Frame-shift. (A) the ROC curve on genomic-wide Human dataset; (B) the ROC curve on genomic-wide Mouse dataset; (C) the precision-recall curve on genomic-wide Human dataset; (D) the precision-recall curve on genomic-wide Mouse dataset.

3.4. First frame sensitivity of CNN

In order to check if CNN can capture the most prominent feature of the transition from a non-coding region to a coding region in the first reading frame [28,22,29], we use the predicted coding scores only for TIS prediction. To be specific, we compare the prediction performance of two strategies, Frame and Frame-shift. The former one directly feed coding scores into CNN, whereas the latter one randomly shifts the coding scores of positive samples with 0, 1 or 2 nucleotides (note that the frame of negative samples is originally random) and feed them into CNN. As shown in Table 1, Frame/Frame-shift has the same network architecture as Integrated-CNN but with a 400×1 input layer (ID:0) and a $200 \times (8, 1)$ conv layer (ID:1). From Table 2-(3) and Fig. 3, it is observed that the prediction performance of Frame is better than that of Frame-shift, which proves that CNN is sensitive to location information and can capture the first reading frame feature.

3.5. Performance comparison with existing state-of-the-art methods

We compare the performance of DeepTIS with that of SVM [44], DeepGSR [16] and TISRover [45]. All the methods are trained and evaluated on the same datasets for fair comparison. We implement the SVM method with simple polynomial kernel, DeepGSR with the same network architecture in its original paper, and TISRover with the same network architecture with Integrated-CNN but without coding contrast feature. Furthermore, the scale of all the neural networks is set large enough to prevent underfitting and two regularization methods (e.g., drop out [32], early stopping [34]) are used to prevent overfitting. The parameters of some network architectures are shown in Table 1.

As shown in Tables 2–3, it is observed that DeepTIS performs the best both on human and mouse datasets. In imbalanced case, DeepTIS achieves the highest scores of Sn, Sp, auROC and auPRC on human and mouse datasets, 6.25%/0.33%/0.44%/2.63% and 8.52%/0.02%/0.85%/3.61% improvement over the second-best method, TISRover on human and mouse datasets, respectively. Interestingly, as the negative samples increase, the auROC scores of all the methods increase a little. The underlying reason could be that the trained machine learning methods are more accurate when give more samples. Meanwhile, the auPRC scores of all the methods decrease a lot with the increased number of false positives. It is worth noting that TISRover resembles the method DeepGSR (both convolutional neural network-based methods that directly learn from TIS sequence) but different usages of numerical representation for TIS sequence, TISRover uses one-hot encoding while DeepGSR uses trinucleotide representation. In practice, we find that one-hot encoding is much easier to train than trinucleotide representation for the fact that trinucleotide representation is high-dimensional and very sparse (e.g., prone to overfitting, computational expensive). With respect to the SVM method, we can see that it performs not very well both on human and mouse datasets shown in Tables 2–3, this might be attributed to its limited capability of modeling non-linearity on large gene datasets. Moreover, all the methods achieve low sensitivity and high specificity in imbalanced case indicates that they all suffer a class imbalanced problem (e.g., biased decision boundaries).

We also plot the ROC and PRC curves on imbalanced human and mouse datasets. As it can be seen from Fig. 3, the curves of DeepTIS are always above than the curves of the other methods, given a fixed false positive rate of 8%, DeepTIS respectively achieves a sensitivity of 88.37% and 88.16% on human and mouse dataset,

Table 2
Performance comparison of DeepTIS with the other state-of-the-art methods on genome-wide Human dataset.

Human	Balanced				Imbalanced			
	Sn	Sp	auROC	auPRC	Sn	Sp	auROC	auPRC
Frame-shift	.8817	.8285	.9259	.9168	.6446	.9495	.9317	.7547
Frame	.8637	.8710	.9350	.9267	.7071	.9518	.9408	.7839
SVM	.7970	.7203	–	–	.3777	**.9676**	–	–
DeepGSR	.8667	.8066	.9240	.9164	.5393	.9660	.9240	.7449
TISRover	.8656	.8681	.9435	.9375	.7118	.9554	.9567	.8319
DeepTIS1	.8926	.8758	.9491	.9473	.7332	.9634	.9555	.8463
DeepTIS2	**.9016**	**.8761**	**.9555**	**.9523**	**.7747**	.9587	**.9611**	**.8582**

Table 3
Performance comparison of DeepTIS with the other state-of-the-art methods on genome-wide Mouse dataset.

Mouse	Balanced				Imbalanced			
	Sn	Sp	auROC	auPRC	Sn	Sp	auROC	auPRC
Frame-shift	.8683	.8504	.9281	.9186	.6448	.9551	.9333	.7629
Frame	.8717	.8806	.9408	.9345	.7081	.9564	.9436	.7971
SVM	.7598	.7822	–	–	.3941	**.9714**	–	–
DeepGSR	.8110	.8422	.9129	.9073	.5350	.9675	.9211	.7493
TISRover	.8597	.8736	.9398	.9361	.6530	.9683	.9536	.8300
DeepTIS1	.8935	.8738	.9496	.9490	.7198	.9673	.9552	.8494
DeepTIS2	**.8733**	**.8987**	**.9563**	**.9545**	**.7382**	.9685	**.9621**	**.8661**

Table 4
Brief description of time cost on balanced/imbalanced Human and Mouse datasets with regard to DeepTIS.

Dataset	Coding number	TISs number	Time cost (min)	
			Content-RCNN	Integrated-CNN
Human	6,809,025	38,576/115,728	217	4/9
Mouse	5,787,480	32,946/98,838	184	4/8

a 6.35%, 4.5% improvement over TISRover. As for PRC curve shown in Fig. 3, given a fixed recall of 85%, DeepTIS respectively achieves a precision of 68.77% and 72.48% on human and mouse dataset, a 15.21%, 16.62% improvement over TISRover, which proves that coding features is a good complementation to TIS prediction, and incorporating this feature explicitly into a CNN can gain better performance than TISRover that only use TIS sequence encoded with one-hot encoding directly.

All the results demonstrate that DeepTIS achieves a relatively higher prediction performance when compared with existing state-of-the-art methods.

3.6. Time cost of DeepTIS

We further briefly analyze the computational cost of DeepTIS. Table 4 gives the time cost of DeepTIS on balanced/imbalanced genome-wide human and mouse datasets. All the experiments are conducted on an Intel Core i5-10400 CPU 2.9 GHz PC with 16GB RAM. Content-RCNN and Integrated-CNN are both implemented in Tensorflow. As shown in Table 4, we can see that larger dataset requires more additional training time. Moreover, the time cost of TISRover is basically equivalent to Integrated-CNN and not mentioned here, DeepTIS has one more stage compared with TISRover, and thus it requires more time to train and test Content-RCNN.

4. Conclusion

In conclusion, we present DeepTIS, a two-stage deep learning model to infer TIS in genomic sequence. DeepTIS explicitly model coding features by using a hybrid deep learning architecture and directly incorporating coding features into a CNN which is sensitive to the first reading frame for effectively predicting TIS in genomic sequence. The experiments results show that DeepTIS significantly outperforms existing state-of-the-art methods on genome-wide human and mouse datasets.

CRediT authorship contribution statement

Chao Wei: Conceptualization, Methodology, Data curation, Software, Writing Original draft preparation. **Junying Zhang:** Supervision, Writing Reviewing and Editing. **Xiguo Yuan:** Writing Reviewing and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

References

[1] A. Bernal, K. Crammer, A. Hatzigeorgiou, F. Pereira, Global discriminative learning for higher-accuracy computational gene prediction, PLoS computational biology 3 (2007) e54.
[2] S. Brunak, J. Engelbrecht, S. Knudsen, Prediction of human mrna donor and acceptor sites from the dna sequence, Journal of Molecular Biology 220 (1991) 49–65.
[3] C.B. Burge, S. Karlin, Finding the genes in genomic dna, Current opinion in structural biology 8 (1998) 346–354.
[4] M. Catherine, S. Marie-France, S. Thomas, R. Pierre, Current methods of gene prediction, their strengths and weaknesses, Nucleic Acids Research 30 (2002) 4103–4117.
[5] W. Chen, P.M. Feng, E.Z. Deng, H. Lin, K.C. Chou, Itis-psetnc: a sequence-based predictor for identifying translation initiation site in human genes using pseudo trinucleotide composition, Analytical biochemistry 462 (2014) 76–83.
[6] W.C. Cheng, J.C. Huang, C.Y. Liou, Segmentation of dna using simple recurrent neural network, Knowledge-Based Systems 26 (2012) 271–280.
[7] Q. Daniel, X. Xie, Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences, Nucleic Acids Research e107–e107 (2016).

[8] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 233–240.
[9] M. Ghafoorian, N. Karssemeijer, T. Heskes, I.W. van Uden, C.I. Sanchez, G. Litjens, F.E. de Leeuw, B. van Ginneken, E. Marchiori, B. Platel, Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities, Scientific Reports 7 (2017) 1–12.
[10] M. Ghandi, D. Lee, M. Mohammad-Noori, M.A. Beer, Enhanced regulatory sequence prediction using gapped k-mer features, PLoS Comput Biol 10 (2014) e1003711.
[11] N. Goel, S. Singh, T.C. Aseri, Global sequence features based translation initiation site prediction in human genomic sequences, Heliyon 6 (2020) e04825.
[12] A. Hatzigeorgiou, N. Mache, M. Reczko, Functional site prediction on the dna sequence by artificial neural networks, in: Proceedings IEEE International Joint Symposia on Intelligence and Systems, IEEE, 1996, pp. 12–17.
[13] A.G. Hatzigeorgiou, Translation initiation start prediction in human cdnas with high accuracy, Bioinformatics 18 (2002) 343–350.
[14] D. Heckerman, D. Maxwell Chickering, C. Meek, R. Rounthwaite, C. Kadie, Dependency Networks for Collaborative Filtering and Data Visualization, 2013, arXiv e-prints, arXiv–1301.
[15] L.R.R.F. Ieee, A tutorial on hidden Markov models and selected applications in speech recognition, 1989.
[16] M. Kalkatawi, A. Magana-Mora, B. Jankovic, V.B. Bajic, Deepgsr: an optimized deep-learning structure for the recognition of genomic signals and regions, Bioinformatics 35 (2019) 1125–1132.
[17] M. Kozak, An analysis of 5'-noncoding sequences from 699 vertebrate messenger rnas, Nucleic acids research 15 (1987) 8125–8148.
[18] M. Kozak, The scanning model for translation: an update, The Journal of cell biology 108 (1989) 229–241.
[19] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.
[20] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[21] G. Li, T.Y. Leong, L. Zhang, Translation initiation sites prediction with mixture Gaussian models, IEEE Transactions on Knowledge & Data Engineering 17 (2005) 1152–1160.
[22] J. Li, H. Liu, L. Wong, R.H. Yap, Techniques for recognition of translation initiation sites, in: The Practical Bioinformatician, World Scientific, 2004, pp. 71–89.
[23] T.M. Mitchell, J.G. Carbonell, R.S. Michalski, Machine Learning, 1997.
[24] A.G. Pedersen, H. Nielsen, Neural network prediction of translation initiation sites in eukaryotes: perspectives for est and genome analysis, in: International Conference on Intelligent Systems for Molecular Biology, 1997.
[25] J. Pérez-Rodríguez, A.G. Arroyo-Peña, N. García-Pedrajas, Improving translation initiation site and stop codon recognition by using more than two classes, Bioinformatics 30 (2014) 2702–2708.
[26] K.D. Pruitt, T. Tatusova, D.R. Maglott, Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins, Nucleic acids research 35 (2007) D61–D65.
[27] J.C. Rajapakse, L.S. Ho, Markov encoding for detecting signals in genomic sequences, IEEE/ACM Transactions on Computational Biology and Bioinformatics 2 (2005) 131–142.
[28] Y. Saeys, Feature selection for classification of nucleic acid sequences, Ph.D. thesis, Ghent University, 2004.
[29] Y. Saeys, T. Abeel, S. Degroeve, Y. Van de Peer, Translation initiation site prediction on a genomic scale: beauty in simplicity, Bioinformatics 23 (2007), i418–i423.
[30] A.A. Salamov, Assessing protein coding region integrity in cdna sequencing projects, Bioinformatics 14 (1998) 384.
[31] S.K. Snderby, C.K. Snderby, H. Nielsen, O. Winther, Convolutional lstm networks for subcellular localization of proteins, in: Algorithms for Computational Biology, 2015.
[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.
[33] M. Stanke, S. Waack, Gene prediction with a hidden Markov model and a new intron submodel, Bioinformatics 19 (2003) ii215–ii225.
[34] N.K. Treadgold, T.D. Gedeon, Exploring constructive cascade networks, IEEE Transactions on Neural Networks 10 (1999) 1335–1350.
[35] G. Tzanis, C. Berberidis, I. Vlahavas, Stacktis: a stacked generalization approach for effective prediction of translation initiation sites, Computers in Biology & Medicine 42 (2012) 61–69.
[36] E.C. Uberbacher, R.J. Mural, Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach, Proceedings of the National Academy of Sciences of the United States of America 88 (1991) 11261–11265.
[37] Y.B. Wang, Z.H. You, S. Yang, H.C. Yi, Z.H. Chen, K. Zheng, A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network, BMC medical informatics and decision making 20 (2020) 1–9.
[38] C. Wei, J. Zhang, X. Yuan, Z. He, G. Liu, A Deep Learning Framework with Hybrid Encoding for Protein Coding Regions Prediction in Biological Sequences, 2021 bioRxiv, 2020–11.
[39] C. Wei, J. Zhang, X. Yuan, Z. He, G. Liu, J. Wu, Neurotis: enhancing the prediction of translation initiation sites in mrna sequences via a hybrid dependency network and deep learning framework, Knowledge-Based Systems 212 (2021) 106459.
[40] J. Xi, A. Li, M. Wang, A novel network regularized matrix decomposition method to detect mutated cancer genes in tumour samples with inter-patient heterogeneity, Scientific Reports 7 (2017) 2855.
[41] J. Xi, A. Li, M. Wang, A novel unsupervised learning model for detecting driver genes from pan-cancer data through matrix tri-factorization framework with pairwise similarities constraints, Neurocomputing 296 (2018) 64–73.
[42] X. Yuan, J. Bai, J. Zhang, L. Yang, J. Duan, Y. Li, M. Gao, Condel: detecting copy number variation and genotyping deletion zygosity from single tumor samples using sequence data, IEEE/ACM transactions on computational biology and bioinformatics (2018).
[43] X. Yuan, J. Zhang, L. Yang, J. Bai, P. Fan, Detection of significant copy number variations from multiple samples in next-generation sequencing data, IEEE transactions on nanobioscience 17 (2017) 12–20.
[44] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, K.R. Müller, Engineering support vector machine kernels that recognize translation initiation sites, Bioinformatics 16 (2000) 799–807.
[45] J. Zuallaert, M. Kim, A. Soete, Y. Saeys, W.D. Neve, Tisrover: convnets learn biologically relevant features for effective translation initiation site prediction, International Journal of Data Mining and Bioinformatics 20 (2018) 267–284.

Chao Wei received the B.S. degree in information and computing science from Huanggang Normal University, Huanggang, Hubei, China, in 2012 and M.E. degree in computer technology from Wuhan University, Wuhan, Hubei, China, in 2014. He is currently working towards Ph.D degree in the School of Computer Science and Technology, Xidian University, Xi'an, China. His research interests including machine learning and bioinformatics.

Junying Zhang received the Ph.D. degree in Signal and Information Processing from Xidian University, Xi'an, China, in 1998. From 2001 to 2002, she was a visiting scholar at the Department of Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC, USA, and in 2007, she was a visiting professor at the Department of Electrical Engineering and Computer Science, Virginia Polytechnic Institute and State University, USA. She is currently a Professor in the School of Computer Science and Technology, Xidian University, Xi'an, China. Her research interests focus on Intelligent Information Processing, including Machine Learning and its application to Bioinformatics.

Xiguo Yuan received the B.S. and M.S. degree in computer applications from Wuhan University of Science & Technology, Wuhan, Hubei, China, in 2005 and 2008, respectively. He received the Ph.D. degree in computer applications from Xidian University, Xi'an, China, in 2011. He is currently an associate professor in the School of Computer Science and Technology, Xidian University.