Journal of Biomedical Informatics 61 (2016) 55–62

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

SD-MSAEs: Promoter recognition in human genome based on deep feature extraction

Wenxuan Xu, Li Zhang *, Yaping Lu

School of Computer Science and Technology & Joint International Research Laboratory of Machine Learning and Neuromorphic Computing, Soochow University, Suzhou 215006, Jiangsu, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000, Jiangsu, China

ARTICLE INFO

Article history:
Received 28 August 2015
Revised 22 March 2016
Accepted 23 March 2016
Available online 24 March 2016

Keywords:
Context features
Promoter recognition
Sparse autoencoder
Statistical divergence
Support vector machine

ABSTRACT

The prediction and recognition of promoter in human genome play an important role in DNA sequence analysis. Entropy, in Shannon sense, of information theory is a multiple utility in bioinformatic details analysis. The relative entropy estimator methods based on statistical divergence (SD) are used to extract meaningful features to distinguish different regions of DNA sequences. In this paper, we choose context feature and use a set of methods of SD to select the most effective *n*-mers distinguishing promoter regions from other DNA regions in human genome. Extracted from the total possible combinations of *n*-mers, we can get four sparse distributions based on promoter and non-promoters training samples. The informative *n*-mers are selected by optimizing the differentiating extents of these distributions. Specially, we combine the advantage of statistical divergence and multiple sparse auto-encoders (MSAEs) in deep learning to extract deep feature for promoter recognition. And then we apply multiple SVMs and a decision model to construct a human promoter recognition method called SD-MSAEs. Framework is flexible that it can integrate new feature extraction or new classification models freely. Experimental results show that our method has high sensitivity and specificity.

1. Introduction

In genetics, promoter is a region of DNA that initiates transcription of a particular gene. It contains gene transcription start sites and controls biological activities of genes [1]. Promoter determines the direction, speed and accuracy of DNA transcription. The promoter recognition plays an important role in studying the regulation of human gene expression. Thus, it is a very important task that how to quickly and accurately recognize human promoter at present.

Since Fickett and Hatzigeorgiou published the first review paper on promoter recognition algorithms in 1997 [2], the recognition technologies of human promoter have been developed rapidly. More and more researchers use the knowledge of bioinformatics to predict and recognize the promoters with the help of computer technology to get more reliable results.

One of the key problems in promoter recognition is how to extract the most informative and discriminative features to differentiate the categories of promoters from non-promoters. Signal, context and structure features are the three types of features which can be used to recognize core-promoter regions essentially. CpG-islands are widely used in many recognition algorithms as one of the signal features [3–5]. The statistical features based on the unit DNA words called *n*-mers are also used to predict and recognize promoters which belong to the context features [6] since DNA sequences are always taken as the collections of documents. Specially, *n*-mers may reduce the false positive rates while maintaining a relatively high sensitivity in promoter recognition due to the biological significance in *n*-mers' distribution [7,8]. Thus, we choose *n*-mers extracted from the datasets as the informative and discriminative features for classification.

Context features are extracted from the total possible combinations of *n*-mers with large search space at each site of sequences. Many promoter recognition algorithms adopt information theory to simplify the extraction of *n*-mers. For example, the well-known Kullback–Leibler (KL) divergence in conditional entropy of statistical divergence (SD) [9] is widely applied. Zeng et al. [10] constructed two class models based on the maximum relative entropy and used the KL divergence as the weight to evaluate the discriminative ability of each *n*-mer. The two class models respectively contain a group of words for promoters and non-promoters. The observed *n*-mers are conditional independent, so

* Corresponding author.
E-mail addresses: rifflexiansen@qq.com (W. Xu), zhangliml@suda.edu.cn (L. Zhang), 20134227010@stu.suda.edu.cn (Y. Lu).

http://dx.doi.org/10.1016/j.jbi.2016.03.018
1532-0464/

that the class-conditional probability can be rewritten as a product of probabilities of individual *n*-mers at all sites. In addition, the maximum entropy hidden Markov model [14] without the independence assumption was also used to select signal features based on the frequency, such as TATA box, GC box, and CAAT box. This method has good performance on limited datasets, but the sparse data increases the difficulty of recognition. Naive Bayes rules are also used to significantly reduce the *n*-mer search space in consideration of the positional information and rewrite the class-conditional probability as a product of probabilities of individual *n*-mers at all sites according to the naive Bayes rules [4]. In addition, the symmetrized divergence (also called the J divergence) and the Jensen–Shannon (JS) divergence are also the meaningful statistical measures of SD [11], which can measure the distance between two probability distributions. The two divergences are based on the KL divergence, with some notable and useful differences, including that JS divergence is symmetric and always a finite value [12]. Here, we apply these SD methods to select the most informative *n*-mers as the features based on promoter and non-promoters training samples sparse distributions. Besides feature extraction, another important task is to select appropriate classifiers to differentiate categories of promoters from non-promoters based on selected features. A lot of models in machine learning can be applied to promoter recognition, such as support vector machine (SVM) [13], Markov model [14], relevance vector machine [15], linear and quadratic discriminant analysis [16] and neural network [17].

Here, we take SVM as the classifier for promoter recognition. SVM is supervised classification method and has been proved to be an effective promoter recognition algorithm. SVM deals with the large number of high-dimensional and complex data much better than other statistical or machine learning methods in most promoter recognition tasks.

To improve the promoter recognition performance, this paper introduces sparse auto-encoder (SAE) to transform feature further. SAE is a neural network model for unsupervised feature learning [18]. As an autoencoder, SAE is imposed with a sparseness constraint on hidden units and tries to learn an approximation to the identity function which minimizes an average distortion measure between inputs and outputs [19]. Typically, SAE is used for learning a representation of the raw data. For example, SAE is combined with a softmax classifier as a whole deep network which is called stacked autoencoder [20].

By using SAE, we propose a human promoter recognition method based on statistical divergence and multiple sparse auto-encoders (SD-MSAEs). The promoter, the coding exons and the introns of DNA sequences are considered at the same time. In our method, *n*-mers statistical features are first extracted from four kinds of gene sequences in human genome. Second, we apply a set of methods of SD to select the most informative and discriminative *n*-mers to identify the promoters and non-promoters in large genomic sequences. And then, MSAEs are combined with SD to extract the deep features so as to get a meaningful representation. Finally, three support vector machines (SVMs) are independently adopted to classify these processed features. And a decision module (MV) with voting algorithm is used to combine the results of three classifiers.

The contribution of this paper is to combine the advantage of statistical divergence and multiple sparse autoencoders to extract deep feature of *n*-mers for promoter recognition, significantly reducing the *n*-mer search space. In addition, a classification framework of multiple SVMs based on these features is presented and a decision model is used to integrate the prediction. The rest of this paper is organized as follows. Section 2 introduces deep feature extraction. Section 3 proposes classifier ensemble recognition method based on the multiple SVMs and majority voting, called SD-MSAEs. We show experimental results in Section 4 and conclude this paper in Section 5.

2. SD-MSAEs: deep feature extraction

In this paper, we focus on differentiating [−200,+50] bps around the transcription start point (TSS) which are defined by the DBTSS database [21] from other genomic regions, and other alternative TSSs related to tissue specific gene expression are not considered. Fig. 1 shows a schematic representation of the locations of the promoter region, transcription start point, exons, introns and 3′UTR. The present development trend of promoter recognition is considering the promoter, the coding exons and the introns of genomic regions at the same time for the reason that the properties of promoter regions are considerably different from those of other genomic regions, such as exons, introns, 3′UTRs and intragenic regions [9,10].

Fig. 3 shows the framework of our promoter recognition method, which considers the promoter, the coding exons and the introns of genomic regions at the same time. There are two stages in our framework. The first stage extracts deep features for promoter recognition. The other stage is to learn these deep features by the ensemble way.

This section introduces the first stage about deep feature extraction. Feature extraction is a critical step for good recognition performance, which depends on the properties of promoters. In the following, we introduce feature extraction methods which would be used in our learning framework. First, we introduce context feature which could be extracted by using three statistical divergence techniques. Then, sparse auto-encoders are discussed to extract deep features for promoter recognition.

2.1. Context feature

Since DNA sequences are always taken as the collections of documents, the statistical features based on *n*-mers are also used to predict and recognize promoters which represent context features [6]. Some experiments in the study [4,7,8] has shown that the statistic of *n*-mers may have biological significance and even may reveal fine details of unknown promoter characteristics. Specially, *n*-mers may help reduce the false positive rates while maintaining

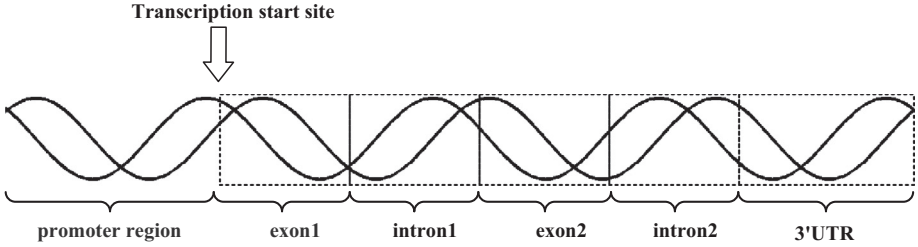

Fig. 1. Schematic representation of the locations of the promoter region, transcription start sequence (TSS), exons, introns, and 3′untranslated region (3′UTR).

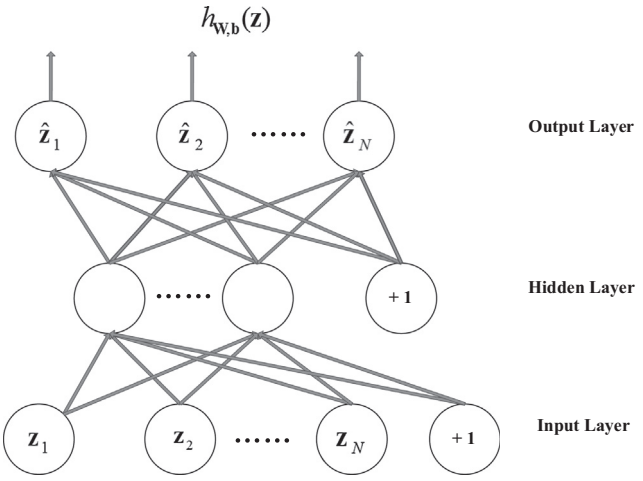

Fig. 2. A sparse auto-encoder.

a relatively high sensitivity in promoter recognition because of the biological significance in their distribution [4,7,8,10].

Let M be the set of total possible combinations of n-mers, where $|M| = 4^n$. We get n-mers probability distribution of M by counting the frequency of n-mers at each site of the nucleotide sequence. Note that the tractable search space has a size $4^{n^{L-n+1}}$, where L is the length of sequence. Thus, the search space becomes more and more complex when n and L increase. In order to get a balance between the size of the tractable search space and the discriminant performance, in our framework, we extract 5-mers from the sequences as the informative and discriminative features for the following processing. Let $F_{n\text{-mer}} = \{\mathbf{f}_p, \mathbf{f}^1_{np}, \mathbf{f}^2_{np}, \mathbf{f}^3_{np}\}$ be the set of the probability density of n-mers, where $\mathbf{f}_p \in R^{4^n}$ is the probability density on promoters, $\mathbf{f}^r_{np} \in R^{4^n}$, $r = 1, 2, 3$, are on three kinds of non-promoters, r = 1, 2, and 3 represents exon, intron and 3′-UTR, respectively.

2.2. Statistical divergence

The concepts of bio-entropy and bioenergetics stem from the classical notions of thermodynamic entropy and weaved in the web of information theory (due to Shannon and Weaver). Entropy, in Shannon sense, of information theory is a multiple utility in bioinformatics details analysis [12]. The relative entropy estimator methods based on statistical divergence are used to extract meaningful features to distinguish different regions of DNA sequences. Many promoter recognition algorithms make simplifications in the n-mer feature extraction based on the knowledge of information theory [4,7,8].

We apply the KL divergence, the J divergence, the JS divergence as the set of methods of SD to select the most informative n-mers as the features based on the probability distributions of promoter and non-promoters training samples.

- Kullback–Leibler divergence

Such as in conditional entropy aspects of SD, the well-known Kullback–Leibler measure are widely applied [10]. The Kullback–Leibler divergence is defined as follows:

$$D_r(\mathbf{f}_p \| \mathbf{f}^r_{np}) = \sum_{i=1}^{4^n} f_p(i) \ln \frac{f_p(i)}{f^r_{np}(i)} = \sum_{i=1}^{4^n} d(f_p(i), f^r_{np}(i)) = \sum_{i=1}^{4^n} d^r_i \quad (1)$$

where $d^r_i = d(f_p(i), f^r_{np}(i))$, $\mathbf{f}_p = [f_p(1), \ldots, f_p(4^n)]$ and $\mathbf{f}^r_{np} = [f^r_{np}(1), \ldots, f^r_{np}(4^n)]$.

- J divergence

The KL divergence is often intuited as a metric or distance, but it is not a true metric because it is not symmetric. The KL divergence from P to Q is not generally the same as the one from Q to P. To obtain a symmetric measure, the symmetrized divergence (J divergence) can be used, which is defined as:

$$\begin{aligned} JD_r(\mathbf{f}_p \| \mathbf{f}^r_{np}) &= \frac{1}{2} D_r(\mathbf{f}_p \| \mathbf{f}^r_{np}) + \frac{1}{2} D_r(\mathbf{f}^r_{np} \| \mathbf{f}_p) \\ &= \frac{1}{2} \sum_{i=1}^{4^n} (d(f_p(i), f^r_{np}(i)) + d(f^r_{np}(i), f_p(i))) = \sum_{i=1}^{4^n} d^r_i \end{aligned} \quad (2)$$

where both $D_r(\mathbf{f}_p \| \mathbf{f}^r_{np})$ and $D_r(\mathbf{f}^r_{np} \| \mathbf{f}_p)$ are the KL divergence, and $d^r_i = \frac{1}{2}(d(f_p(i), f^r_{np}(i)) + d(f^r_{np}(i), f_p(i)))$.

- Jensen–Shannon (JS) divergence

The Jensen–Shannon (JS) divergence is a meaningful statistical measure of SD, which can measure the distance between two probability distributions. The JS divergence is based on the KL divergence with some notable and useful differences, such as the JS divergence is symmetric and always has a finite value [13].

The JS divergence is defined as follows:

$$\begin{aligned} JSD_r(\mathbf{f}_p \| \mathbf{f}^r_{np}) &= \frac{1}{2} D_r(\mathbf{f}_p \| \bar{\mathbf{f}}^r) + \frac{1}{2} D_r(\mathbf{f}^r_{np} \| \bar{\mathbf{f}}^{\mathbf{r}}) \\ &= \frac{1}{2} \sum_{i=1}^{4^n} (d(f_p(i), \bar{f}^r(i)) + d(\bar{f}^r(i), f^r_{np}(i))) = \sum_{i=1}^{4^n} d^r_i \end{aligned} \quad (3)$$

where $\bar{\mathbf{f}}^r = \frac{1}{2}\mathbf{f}_p + \frac{1}{2}\mathbf{f}^r_{np}$ and

$$d^r_i = \frac{1}{2}(d(f_p(i), \bar{f}^r(i)) + d(\bar{f}^r(i), f^r_{np}(i)))$$

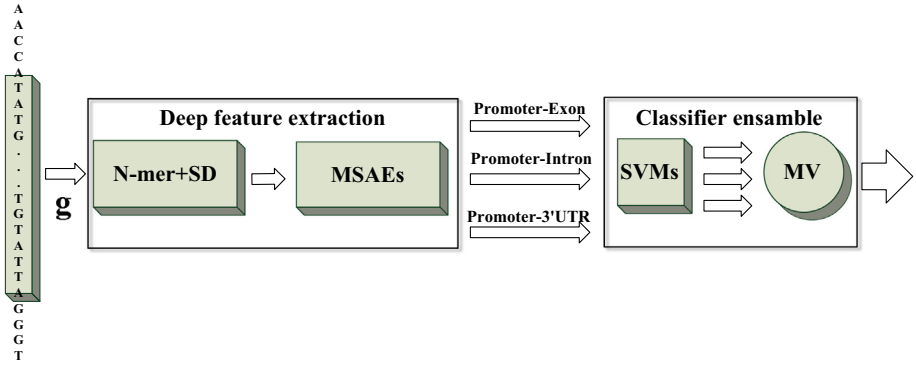

Fig. 3. Framework of SD-MSAEs.

Now, we can select informative n-mers based on these SD methods (1), (2) or (3). First, we simply sort d_i^r, $i = 1, \ldots, 4^5$ in descending order and form a new vector $\mathbf{d}^r = [d_1^r, \ldots, d_{4^n}^r]^T \in R^{4^n}$. The following optimization problem is defined to obtain the informative n-mers:

$$\min_{m^r} \quad \frac{\sum_{i=1}^{m^r} d_i^r}{\sum_{i=1}^{4^n} d_i^r} - \theta \tag{4}$$

where m^r is the number of the informative n-mers, and $\theta > 0$ is a threshold, say 0.98. Let G^r be the set of the first m^r n-mers.

We apply the KL divergence, the J divergence and the JS divergence as SD to select the most informative n-mers as the features based on promoter and non-promoters training samples probability distributions on the DBTSS database. Table 1 shows the statistical divergence between probability distributions of promoter and non-promoter sequences. The context features are extracted from the total possible combinations of n-mers with a large search space at each site of sequences, totally, the size of the search space is 4^n. And when n increases, the extraction of total possible combinations of n-mers is overwhelmingly large. However, the computation can be simplified for maintaining the most information as show in Table 2.

Generally given a gene $\mathbf{g} \in R^L$, we can extract the context feature for it according to G^r, $r = 1, 2, 3$. Let the context features of $\mathbf{g}$ be $\mathbf{z}^r$, where $\mathbf{z}^r \in Z^r \subset R^{m^r}$ and Z^r is the set of context features. The context feature $\mathbf{z}^r$ is the probability density of G^r which is used to recognize the promoter and the non-promoter.

2.3. MSAEs

As mentioned before, SAE is a neural network model for unsupervised feature learning and used for learning a representation of the raw data. Fig. 2 shows the basic structure of a SAE. Here, we think that meaningful deep features could be obtained by applying SAE to n-mers, which is expected to improve the separability of promoter and non-promoters.

In order to extract deep features of n-mers, we use three SAEs $h^r_{\mathbf{W},\mathbf{b}}(\cdot)$ to process the three context feature sets Z^r, respectively, where $r = 1, 2, 3$, $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias vector. The number of neurons in the output layer is equal to that in the input layer and each SAE has three hidden layers.

In the following, we discuss a general situation for SAE. Assume $\{\mathbf{z}_i\}_{i=1}^N$ be the inputs of the SAE model, where $\mathbf{z} \in R^m$ and N is the number of samples. SAE aims to find a hypothesis function $h_{\mathbf{W},\mathbf{b}}(\mathbf{z})$ and $h_{\mathbf{W},\mathbf{b}}(\mathbf{z}) = 1/(1 + e^{(-\mathbf{W}\cdot\mathbf{z}-\mathbf{b})})$. To achieve this, SAE minimizes the cost function $J(\mathbf{W}, \mathbf{b})$ after randomly initializing $\mathbf{W}$ and $\mathbf{b}$. Namely,

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N} \|h_{\mathbf{W},\mathbf{b}}(\mathbf{z}_i) - \mathbf{z}_i\|_2^2 + \lambda\|\mathbf{W}\|_2^2 + \beta\sum_{j=1}^{l} S\left(\frac{1}{N}\sum_{i=1}^{N} a_j^{(i)}\right) \tag{5}$$

where $\lambda > 0$ is the weight attenuation coefficient, $\|\cdot\|_2$ denotes the l_2 norm, $\beta \geqslant 0$ is a control factor, l denotes the number of hidden units, $a_j^{(i)}$ represents the output of the j th hidden unit of $\mathbf{z}_i$, and the sparsity penalty function $S(\cdot)$ penalizes the activations of hidden units so as to learn the sparse representations for input observations, which could be the form:

$$S(t) = \rho \log\frac{\rho}{t} + (1-\rho)\log\frac{1-\rho}{1-t} \tag{6}$$

where $0 < \rho < 1$ is the sparseness parameter.

In (5), the first term tries to minimize the difference between the output and the input, the second term is a weight decay term aiming to avoid over-fitting, and the last term is a sparse penalty term.

By separately training three SAEs on the obtained context feature sample sets $\{\mathbf{z}_i^r\}_{i=1}^{N^r}$ with $r = 1, 2, 3$ and $N^r = |Z^r|$, we can extract the deep features and have three new sample sets $\{\mathbf{x}_i^r\}_{i=1}^{N^r}$ where $\mathbf{x}_i^r \in R^{l^r}$, $\mathbf{x}_i^r$ is the output of the last hidden layer for the rth SAE, l^r is the number of the hidden unit in the last hidden layer for the rth SAE, and $r = 1, 2, 3$.

Table 2
The number of n-mer features from the optimization problem.

	m^r		
KL/JD/JSD	Promoter &Exon	Promoter &Intron	Promoter &3′-UTR
3-mer	60/60/63	61/61/63	59/59/63
4-mer	214/215/245	215/216/244	213/214/244
5-mer	695/677/950	702/692/943	700/696/942

3. SD-MSAEs: classifier ensemble

As mentioned before, our framework has two stages. In the first stage, we use SD to select the most informative n-mers to extract the context features and apply MSAEs to extract the deep features to get a meaningful input for the subsequent classifiers. In this section, we discuss the classifier ensemble.

In this paper, we apply three support vector machines (SVMs) as the classifiers to classify promoters from non-promoters based on processed feature sets $\{\mathbf{x}_i^r\}_{i=1}^{N^r}$, $r = 1, 2, 3$. Finally, we combine the outputs of three classifiers using the majority voting algorithm.

3.1. Support vector machines

SVM is a universal learner based on statistical learning theory proposed by Vapnic et al. [22] and has been proved to be a good algorithm for promoter recognition [13]. SVM can implement the structural risk minimization rule to achieve good generalization performance. In SVM, kernel functions are used to map the original samples into a high-dimensional feature space, in which the original sample could be linearly separable.

Here we design three SVMs as the appropriate classifiers to distinguish promoters from exons, introns and 3′UTRs, respectively. Given three set of training samples $\{(\mathbf{x}_i^r, y_i^r)\}_{i=1}^{N^r}$, where $\mathbf{x}_i^r \in R^{l^r}$, l^r is the number of features, $y_i^r \in \{-1, +1\}$ is the class label of $\mathbf{x}_i^r$,

Table 1
Maximum SD between two probability distributions of promoter and nonpromoter sequences.

	n-mer		
KL/JD/JSD	Promoter & Exon	Promoter & Intron	Promoter & 3′-UTR
2-mer	0.2931/0.2981/0.0703	0.2612/0.2643/0.0627	0.2960/0.3046/0.0723
3-mer	0.3671/0.3698/0.1512	0.3751/0.3788/0.1409	0.3772/0.3784/0.1530
4-mer	0.1823/0.1838/0.3385	0.2033/0.1992/0.3311	0.1876/0.1867/0.3401
5-mer	0.0387/0.0387/0.5412	0.0411/0.0411/0.5383	0.0385/0.385/0.5438

the goal of the rth SVM is to separate promoter with exons ($r=1$), introns ($r=2$) and 3′UTRs ($r=3$). The dual programming of the rth SVM can be described as:

$$\max \sum_{i=1}^{N^r}\alpha_i^r - \frac{1}{2}\sum_{i=1}^{N^r}\sum_{j=1}^{N^r}\alpha_i^r\alpha_j^r y_i^r y_j^r k(\mathbf{x}_i^r,\mathbf{x}_j^r) \tag{7}$$

$$\text{subject to } \sum_{i=1}^{N^r}\alpha_i^r y_i^r = 0, \quad 0\leqslant\alpha_i^r\leqslant C, \quad i=1,\ldots,N^r$$

where $C>0$ is the penalty factor, $k(\cdot,\cdot)$ is a Mercer kernel function, and α_i^r is the Lagrange multiplier. When, the corresponding sample $\mathbf{x}_i^r$ is called the non-bounded support vector. If $\alpha_i^r=C$, then the corresponding $\mathbf{x}_i^r$ is called the bounded support vector. Once we solve the programming (7), we can make a decision for an unseen sample $\mathbf{x}^r$, or

$$f^r(\mathbf{x}^r)=\operatorname{sgn}\left(\sum_{i=1}^{N^r}\alpha_i^r y_i^r k(\mathbf{x}_i^r,\mathbf{x}^r)+\tau^r\right) \tag{8}$$

where $\operatorname{sgn}(\cdot)$ denotes the sign function, and τ^r is the threshold of the rth SVM, which can be computed by

$$A.\ \tau^r = y_{sv}^r - \sum_{i=1}^{N^r}\alpha_i^r y_i^r k(\mathbf{x}_i^r,\mathbf{x}_{sv}^r) \tag{9}$$

where $(\mathbf{x}_{sv}^r, y_{sv}^r)$ is a non-bounded support vector. From the decision model (9), we can see that the function $f^r(\mathbf{x}^r)$ is constructed by limited training samples.

3.2. Majority voting

An unseen gene $\mathbf{g}$ can be first described as three context features $\mathbf{z}^r$ according to G^r, and then be respectively represented as the deep features $\mathbf{x}^r$ according to $h_{\mathbf{W},\mathbf{b}}^r(\mathbf{z}^r)$, $r=1,2,3$. By using the functions (8), we have three outputs $f^r(\mathbf{x}^r)$, $r=1,2,3$. In order to determine whether $\mathbf{g}$ is the promoter or not, we need to combine these outputs and use the majority voting rule. Simply, the majority voting rule can be described as

$$\widehat{y}=\begin{cases}+1, & \text{if } \sum_{r=1}^{3}\frac{(f^r(\mathbf{x}^r)+1)}{2}\geqslant 2\\ -1, & \text{Otherwise}\end{cases} \tag{10}$$

where $\widehat{y}$ is the estimated value for $\mathbf{g}$. If $\widehat{y}=+1$, then $\mathbf{g}$ is a promoter. Otherwise $\mathbf{g}$ is a non-promoter.

4. Experiments and results

To implement SVM, we use the libsvm-2.89 toolbox written by Chih-Jen Lin (http://www.csie.ntu.edu.tw/~cjlin). We choose the radial basis function (RBF) kernel $k(\mathbf{x}_i,\mathbf{x})=exp(-\gamma\|\mathbf{x}_i-\mathbf{x}\|^2)$ with a kernel parameter $\gamma>0$. There are two parameters C and γ in SVMs. To obtain optimal parameters, we use grid search algorithm and 10-fold cross-validation. First, we set a range for C or γ, and then perform 10-fold cross-validation based on all the combinations of C and γ under given ranges. Finally, we select the combination of C and γ corresponding to the highest recognition rate. There are many different combinations of C and γ corresponding to highest accuracy rate. Although the accuracy of cross-validation can be improved by a high penalty parameter value, the high penalty parameter value would lead to over-fitting. Thus, we just choose the combination with the minimum C value.

4.1. Datasets

An experiment of a recognition algorithm using statistical pattern recognition methods requires a large number of the promoters and the non-promoters with accurate annotation. In this paper, we focus on differentiating short [−200,+50] ps promoter and non-promoter sequences in the same length around the transcription start point (TSS) which are defined by the DBTSS database [21] from other genomic regions, and other alternative TSSs related to tissue specific gene expression are not considered.

We use 30,964 promoter sequences [−200,+50] bps around the TSSs from the DBTSS dataset to be the training and test sets because DBTSS provides the best combination of coverage and quality at present. In order to accurately estimate n-mer frequency, we construct non-promoter sets by randomly extracting 10,000 exons and 10,000 introns with 251 bps in length from the EID database [23], and 10,000 3′-UTR sequences with 251 bps in length from the UTRdb database [24].

We randomly select 8000 samples from the promoter, exon, intron and 3′-UTR sets, respectively. Among 8000 samples, 4000 samples are considered as the training ones and the rest are tests ones for each class. Thus, we have 12,000 training and 12,000 test samples, respectively. In both training and test sets, the ratio of promoter, exon, intron and 3′-UTR is 1:1:1:1. The sampling process is repeated 10 times. In this paper, the promoter is taken as the class +1 and the non-promoter is the class −1. Thus, totally the positive and negative samples are unbalanced.

In our experiments that we use the SD methods to select the most informative 5-mers. Let $\theta=0.98$ in (4). Then we get the values of m^r in the interval [695, 942] shown in the last row of Table 2. All experiments are performed on the personal computer with a 2.5 GHz Intel(R) Core(TM) i5-2450M CPU and 4G bytes of memory. This computer runs on Windows7, with MATLAB R2013a compiler installed.

4.2. Performance evaluation

Some evaluation measures proposed by Bajic [25] are used to evaluate our method, which are the sensitivity S_n, the specificity S_p and the averaged conditional probability ACP. These measures are defined as follows:

$$S_n=\frac{TP}{TP+FN} \tag{11}$$

$$S_p=\frac{TN}{TN+FP} \tag{12}$$

$$ACP=\frac{1}{4}\left(\frac{TP}{TP+FN}+\frac{TP}{TP+FP}+\frac{TN}{TN+FP}+\frac{TN}{TN+FN}\right) \tag{13}$$

where TP denotes the number of positive sample identified correctly, TN denotes the number of negative sample identified correctly, FP is the number of negative sample which is identified as positive samples and FN denotes the number of positive samples which are not to be identified correctly.

4.3. Effectiveness evaluation of deep feature extraction

In order to get meaningful inputs, we apply a set of methods (KLD, JD and JSD) of SD to select the most informative 5-mers as the context feature and then three MSAEs are used to extract deep feature sets, respectively. SVMs are used to classify these features. Here, we report the effectiveness of deep feature extraction and show the results in Fig. 4. Fig. 4(a) shows the sensitivity and specificity of a sub-classifier for discriminating between promoters and exons, and Fig. 4(b) and (c) for promoters and introns, and

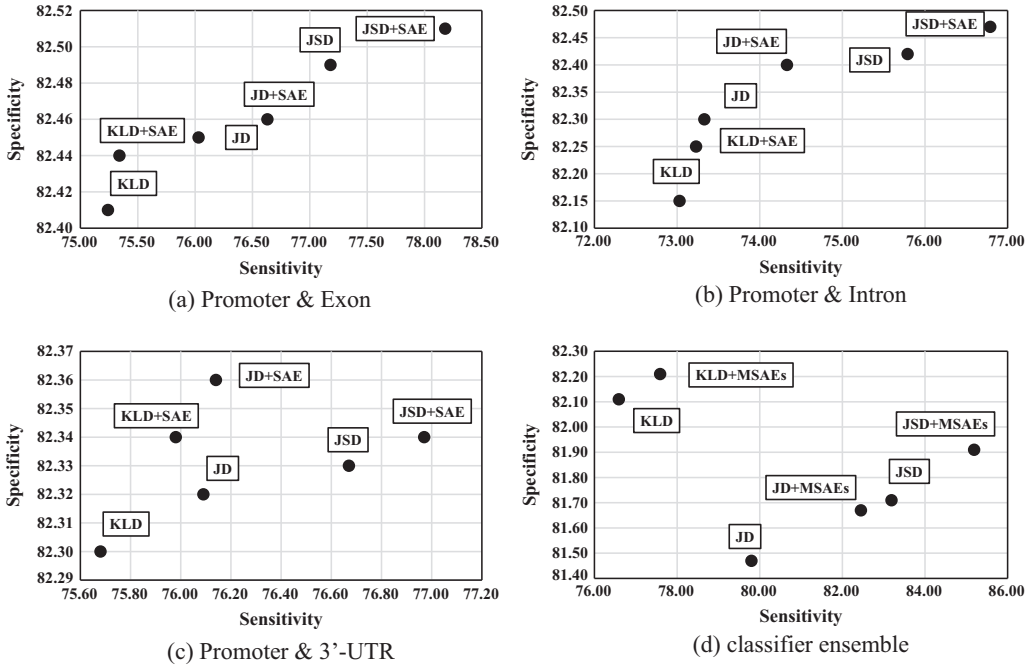

Fig. 4. Performances of promoter & exon, promoter & intron and promoter & 3′-UTR classifiers with or without deep feature on the controlled datasets in the 10-fold cross-validation test. The performance is evaluated in terms of sensitivity and specificity.

promoters and 3'UTRs, respectively. Fig. 4(d) gives sensitivity and specificity of combining three sub-classifiers.

In Fig. 4(a)–(c), KLD, JD and JSD mean the performance of a single SVM only trained on the context features extracted by the KL divergence, the J divergence, and the JS divergence, respectively. KLD + SAE, JD + SAE and JSD + SAE mean the performance of a single SVM trained on the deep features extracted by SAE. In Fig. 4(d), KLD, JD and JSD mean the performance of combining three SVMs trained on the context features extracted by the KL divergence, the J divergence, and the JS divergence, respectively. KLD + MSAEs, JD + MSAEs and JSD + MSAEs mean the performance of combining three SVM trained on the deep features extracted by MSAEs.

We can see that KLD + SAE has a better sensitivity and specificity than KLD according to Fig. 4(a)–(c), which indicates that the deep feature extracted by SAE works well in all three sub-classifiers, promoter & exon, promoter & intron and promoter & 3′-UTR. For other two SD methods, JD and JSD, we have the same conclusion.

Obviously, we also see that combining classifiers achieves much better performance on the sensitivity and the specificity, as shown in Fig. 4(d). For example, the sensitivity of three sub-classifiers on the context features extracted by JD is 76.03%, 73.33% and 76.09%, respectively. That combining the three sub-classifiers results in a higher sensitivity, 79.80%. In addition, the deep feature can further improve the ensemble performance since the sensitivity of JD + MSAEs has 82.45%.

Table 3 shows the performance comparison of sub-classifiers and their ensemble in terms of the averaged conditional probability, which combines the sensitivity and the specificity. We can have the same conclusions as described above. First, the deep feature is much better than the context feature in distinguishing promoters from non-promoters. Second, ensemble learning outperforms the single classifier. Naturally, our method consisting of the deep feature and ensemble learning achieves the best ACP in three SD methods. By the way, we can observe that JDS is the best among three SD methods.

4.4. Effectiveness of SD-MSAEs

In this section, we show that the deep feature and ensemble learning are effective by experiments. Here, we compare our method SD-MSAEs with methods proposed in [4,10,14]. Zeng et al. used a k-words (n-mers) as features and applied the KL divergence as a weight to evaluate the discriminative ability for each n-mer. Two groups of discriminative n-mers for promoters and non-promoters are selected by maximizing the relative entropy. In addition, two class models are constructed based on the maximum relative entropy. Here, we call this method "K-words". The maximum entropy hidden Markov model without the independence assumption was also used to select signal features based on the frequency, such as TATA box, GC box, and CAAT box, which is called ME-HMM. Naive Bayes classifiers (NBCs) were proposed for promoter recognition [4]. NBCs can significantly reduce the n-mer search space from $4^{n^{L-n+1}}$ to $4^n(L-n+1)$ in consideration of the positional information and rewrite the class-conditional

Table 3
Comparison of promoter & exon, promoter & intron and promoter & 3'UTR in terms of averaged conditional probability.

	Promoter & Exon	Promoter & Intron	Promoter & 3′-UTR	Classifier ensemble
KLD	72.89	72.18	73.13	73.25
KLD + MSAEs	73.09	72.28	73.23	73.40
JD	73.17	72.40	73.21	73.50
JD + MSAEs	73.47	72.77	73.25	74.00
JSD	73.65	72.80	73.30	74.64
JSD + MSAEs	73.95	73.51	73.40	75.04

probability as a product of probabilities of individual *n*-mers at all sites according to the naive Bayes rules, while retaining a good classification performance [4]. Compared to NBCs, SD-MSAEs are based on the probability distributions of all *n*-mers, and not considering the positional information. MSAEs are applied to generate more discriminant deep features.

The comparison of K-words, ME-HMM and NBCs with our method is shown in Fig. 5. From Fig. 5, we can clearly see that JSD-MSAEs can achieve the best ACP, followed by JD-MSAEs and KLD-MSAEs. JD-MSAEs and JSD-MSAEs have the sensitivity, 82.45% and 85.19%, and KLD-MSAEs is little lower than K-words but much close to ME-HMM. The sensitivity of NBCs is only about 46.97% and its specificity is highest, 82.73%. Although ME-HMM has better sensitivity than KLD-MSAEs and JD-MSAEs, its specificity is lower than KLD-MSAEs and K-words. ME-HMM is better than K-words and NBCs on ACP, but inferior to JD-MSAEs and JSD-MSAEs.

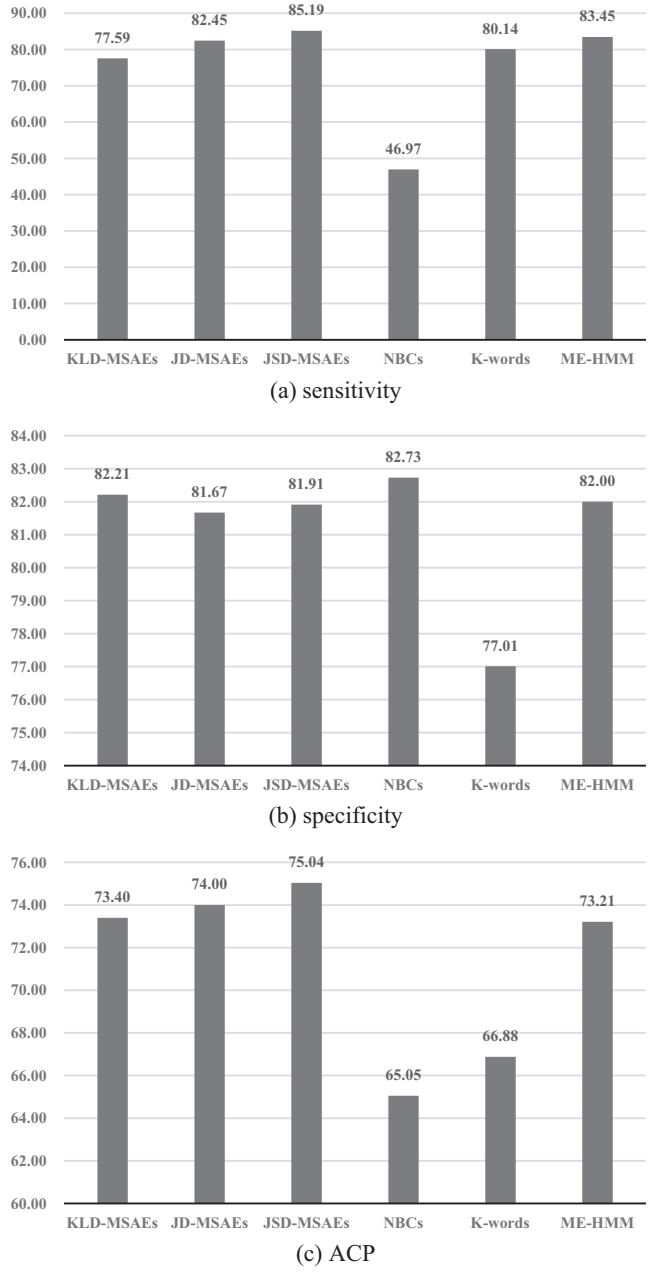

Fig. 5. Performance comparison of K-words, ME-HMM, NBCs and SD-MSAEs (KLD, JD and JSD) on the controlled datasets in the 10-fold cross-validation test. The performance is evaluated in terms of (a) sensitivity, (b) specificity and (c) averaged conditional probability.

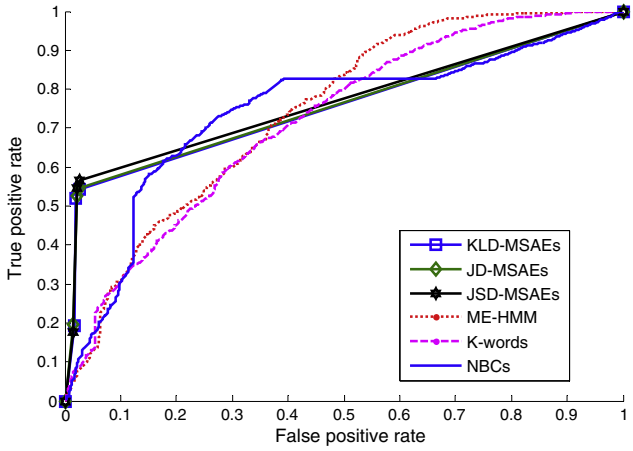

Fig. 6. Comparison of ROC curves for SD-MSAEs (KLD, JD and JSD) from K-words, ME-HMM and NBCs.

Table 4
Comparison of AUC for SD-MSAEs (KLD, JD and JSD) from K-words, ME-HMM and NBCs.

	KLD-MSAES	JD-MSAES	JSD-MSAES	NBCs	K-words	ME-HMM
AUC	75.77	75.97	76.81	74.05	71.88	75.56

We computed the ROC curves and AUC for SD-MSAEs (KLD, JD and JSD) from K-words, ME-HMM and NBCs based on average results (Fig. 6 and Table 4). When computing the ROC curves for SD-MSAEs, votes for promoter are considered as probability. Both ROC curves and AUC can also indicate that our scheme combining statistical divergence and multiple sparse auto-encoders to extract deep feature of *n*-mers for promoter recognition with multiple SVMs is highly effective.

5. Conclusion

This paper presents SD-MSAEs for promoter recognition. In SD-MSAEs, we extract the deep feature based on SD and MSAEs, and apply SVM ensemble to learn the deep feature for promoter recognition. Experimental results on the DBTSS dataset indicate that the deep feature is much better than the context feature in distinguishing promoters from non-promoters. In addition, ensemble learning outperforms the single classifier which has been proved. Thus, our method consisting of the deep feature and ensemble learning achieves the best performance in three indexes.

Although SD-MSAEs are effective, they also have some disadvantage, such as high time complexity. In order to get the most meaningful deep features of each training sets, the deep feature extraction algorithm needs complex multiple parameters optimization for each hidden layer and this process costs much time in most deep-learning algorithms. The computational complexity should be taken into account in future work.

In addition, since the genetic data is very complex and high-dimensional, we only perform experiments on limited samples and limited kinds of features. Therefore, in the future research, more representative training data should be used to extracted features and more feature extraction methods should be considered.

Conflict of interest

None declared.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61402310, by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20140008, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 13KJA520001, by the Qing Lan Project, by the National College Students Innovation and entrepreneurship training program of China under Grant No. 201410285032, by the Undergraduate Science Research Foundation of Soochow University of China under Grant No. KY2015544B, and by the "3I Project" of Soochow University of China under Grant No. 29.

References

[1] V.B. Bajic, A. Chong, S.H. Seah, et al., An intelligent system for vertebrate promoter recognition, IEEE Intell. Syst. 17 (4) (2002) 64–70.
[2] J.W. Fickett, A.G. Hatzigeorgiou, Eukaryotic promoter recognition, Genome Res. 7 (September) (1997) 861–878.
[3] P. U, J.K. Dubey, R.v. K, B.S. Cherian, G. Gopalakrishnan, A.S. Nair, A novel sequence and context based method for promoter recognition, Bioinformation 10 (4) (2014) 175–179.
[4] J. Zeng, X.Y. Zhao, X.Q. Cao, H. Yan, SCS: signal, context, and structure features for genome-wide human promoter recognition, IEEE/ACM Trans. Comput. Biol. Bioinform. 7 (3) (2010) 550–562.
[5] J. Deng, H. Liang, R. Zhang, G. Ying, X. Xie, J. Yu, D. Fan, X. Hao, Methylated CpG site count of dapper homolog 1 (DACT1) promoter prediction the poor survival of gastric cancer, Am. J. Cancer Res. 4 (September) (2014) 518–527.
[6] W.L. Huang, C.W. Tung, C. Liaw, H.L. Huang, S.Y. Ho, Rule-based knowledge acquisition method for promoter prediction in human and Drosophila species, Scient. World J. 2014 (2014) 1–14.
[7] T. Werner, The state of the art of mammalian promoter recognition, Brief Bioinform. 2014 (2014).
[8] S. Wu, X. Xie, A.W.-C. Liew, H. Yan, Eukaryotic promoter prediction based on relative entropy and positional information, Phys. Rev. E 75 (2007) 041908.
[9] S. Vinga, Information theory applications for biological sequence analysis, Brief Bioinform. 15 (3) (2014) 376–389.
[10] Jia Zeng, Xiao-Qin Cao, Hong Yan, Human promoter recognition using Kullback–Leibler divergence, in: International Conference on Machine Learning and Cybernetics, 2007, vol. 6, August 2007, pp. 3319–3325.
[11] E.P. Wigner, Information-theoretic algorithms in bioinformatics and bio-/medical-imaging: a review, in: 2011 International Conference on Recent Trends in Information Technology (ICRTIT), June 2011, pp. 183–188.
[12] F. Nielsen, R. Nock, Sided and symmetrized Bregman centroids, IEEE Trans. Inf. Theory 55 (6) (2009) 2882–2904.
[13] F. Anwar, S.M. Baker, T. Jabid, M. Mehedi Hasan, M. Shoyaib, H. Khan, R. Walshe, Pol II promoter prediction using characteristic 4-mer motifs: a machine learning approach, BMC Bioinf. 9 (1) (2008) 414–418.
[14] X.Y. Zhao, J. Zhang, Y.Y. Chen, Q. Li, T. Yang, C. Pian, L.Y. Zhang, Promoter recognition based on the maximum entropy hidden Markov model, Comput. Biol. Med. 51 (August) (2014) 73–81.
[15] Y. Li, K.K. Lee, S. Walsh, C. Smith, S. Hadingham, K. Sorefan, G. Cawley, M.W. Bevan, Establishing glucose- and ABA-regulated transcription networks in Arabidopsis by microarray analysis and promoter classification using a Relevance Vector Machine, Genome Res. 16 (3) (2006) 414–427.
[16] J. Lu, L. Luo, Prediction for human transcription start site using diversity measure with quadratic discriminant, Bioinformation 2 (7) (2008) 316–321.
[17] J. Wang, L.H. Ungar, H. Tseng, S. Hannenhalli, MetaProm: a neural network based meta-predictor for alternative human promoter prediction, BMC Genom. 8 (October) (2007) 374.
[18] A. Ng, Sparse autoencoder, in: CS294A Lecture Notes for Stanford University, 2011.
[19] P. Baldi, Z. Lu, Complex-valued autoencoders, Neural Netw. 33 (September) (2012) 136–147.
[20] A. Ng, J. Ngiam, C.Y. Foo, Y. Mai, C. Suen, UFLDL tutorial: building deep networks for classification, an online tutorial, 2013.
[21] A. Suzuki, H. Wakaguri, R. Yamashita, S. Kawano, K. Tsuchihara, S. Sugano, Y. Suzuki, K. Nakai, "DBTSS as an integrative platform for transcriptome, epigenome and genome sequence variation data", Nucl. Acids Res. 43 (November) (2014) (Database issue D87-91).
[22] V. Vapnik, C. Cortes, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[23] S. Saxonov, I. Daizadeh, A. Fedorov, W. Gilbert, EID: the exon-intron database—an exhaustive database of proteincoding intron-containing genes, Nucl. Acids Res. 28 (2000) 185–190.
[24] G. Pesole, S. Liuni, G. Grillo, F. Licciulli, F. Mignone, C. Gissi, C. Saccone, UTRdb and UTRsite: specialized databases of sequences and functional elements of 5′ and 3′ untranslated regions of eukaryotic mRNAs, Nucl. Acids Res. 30 (2002) 335–340.
[25] V.B. Bajic, Comparing the success of different prediction programs in sequence analysis: a review, Brief Bioinform. 1 (3) (2000) 214–228.