Molecular Therapy
Nucleic Acids
Original Article

iPromoter-2L2.0: Identifying Promoters and Their Types by Combining Smoothing Cutting Window Algorithm and Sequence-Based Features

Bin Liu[1,2] and Kai Li[3]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; [2]Advanced Research Institute of Multidisciplinary Science, Beijing Institute of Technology, Beijing 100081, China; [3]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Guangdong 518055, China

Promoters are short regions at specific locations of DNA sequences, which are playing key roles in directing gene transcription. They can be grouped into six types (σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54}, σ^{70}). Recently, a predictor called "iPromoter-2L" was constructed to predict the promoters and their six types, which is the first approach to predict all the six types of promoters. However, its predictive quality still needs to be further improved for real-world application requirement. In this study, we proposed the smoothing cutting window algorithm to find the window fragments of the DNA sequences based on the conservation scores to capture the sequence patterns of promoters. For each window fragment, the discriminative features were extracted by using kmer and PseKNC. Combined with support vector machines (SVMs), different predictors were constructed and then clustered into several groups based on their distances. Finally, a new predictor called iPromoter-2L2.0 was constructed to identify the promoters and their six types, which was developed by ensemble learning based on the key predictors selected from the cluster groups. The results showed that iPromoter-2L2.0 outperformed other existing methods for both promoter prediction and identification of their six types, indicating that iPromoter-2L2.0 will be helpful for genomics analysis.

INTRODUCTION

A promoter is a DNA fragment at a specific location that can be recognized and bound by RNA polymerase to initiate transcription. In bacteria, the RNA polymerase contains five subunits ($2\alpha, \beta, \beta', \omega$) and an extra σ factor.[1,2] Theσ factors can be labeled as σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54} and σ^{70} according to the molecular weights. Differentσ factors direct the RNA polymerase binding to different promoter regions, which can affect the consequent activation of genes. σ^{24} and σ^{32} participate in heat-shock response, σ^{28} participates in the flagellar gene expression during normal growth, σ^{54} participates in nitrogen metabolism, and σ^{70}, called primary σ factor, is in charge of transcription of most genes in growing cells.[2–4]

Because the wet experiments are expensive to identify the types of promoters, several predictors were developed to identify the promoters based on the DNA sequence information; for example, iPro54-PseKNC[5] based on the PseKNC[6] was constructed to identify promoters. A position-correlation scoring function (PCSF)[7] and Bayes profile[8] were proposed to identify promoter. By combining the variable window technique with the regular Z-curve method,[9–11] "variable-window Z-curve" was proposed to detect promoters. These methods were discussed in a recent study.[12]

Recently, the iPromoter-2L[12] has been proposed, which is the first predictor that is able to predict the promoters and their aforementioned six different types. This predictor employed the multi-window-based PseKNC approach to capture the sequence patterns of the promoters. However, for this predictor, it is extremely hard to find the optimized sequence windows by using the flexible-sliding-window approach to extract the discriminative features, preventing the performance improvement of this method. In order to overcome these shortcomings, in this study we proposed the smoothing cutting window (SCW) algorithm to divide the DNA sequences into fragment windows based on the conservation scores and ensemble of different predictors based on various sequence-based features to further improve the predictive performance.

RESULTS AND DISCUSSION

Comparison with Other Existing Methods

Table 1 shows the results (Equation 24) generated by iPromoter-2L2.0 via the 5-fold validation on the benchmark dataset. The corresponding rates obtained by the existing methods are also given in Table 1. For the second-layer prediction, only the iPromoter-2L and iPromoter-2L2.0 are able to predict the promoter types among the five existing methods.

Received 7 June 2019; accepted 2 August 2019;
https://doi.org/10.1016/j.omtn.2019.08.008.

Correspondence: Bin Liu, School of Computer Science and Technology, Beijing Institute of Technology, No. 5 South Street, Zhongguancun, Haidian District, Beijing 100081, China.
E-mail: bliu@bliulab.net

Table 1. A Comparison of iPromoter-2L2.0 with Other Predictors for Identifying Promoters (the First Layer) and Their Types (the Second Layer) via the 5-fold Cross-Validation on the Same Benchmark Dataset

Method	Acc (%)	MCC	Sn (%)	Sp (%)
First Layer				
PCSF[a]	74.81	0.4980	78.92	70.70
vw Z-curve[a]	80.28	0.6098	77.76	82.80
Stability[a]	78.04	0.5615	76.61	79.48
iPro54[a]	80.45	0.6100	77.76	83.15
iPromoter-2L1.0[a]	81.68	0.6343	79.20	84.16
iPromoter-2L2.0[b]	84.98	0.6998	84.13	85.84
Second Layer				
iPromoter-2L1.0[a]				
σ^{24} promoter	93.50	0.7338	72.52	96.93
σ^{28} promoter	96.82	0.5708	42.54	99.49
σ^{32} promoter	94.41	0.6524	52.58	99.14
σ^{38} promoter	94.69	0.2962	15.34	99.48
σ^{54} promoter	94.04	0.6459	53.19	99.57
σ^{70} promoter	80.66	0.6056	95.34	59.35
iPromoter-2L2.0[b]				
σ^{24} promoter	94.62	0.8053	81.82	97.22
σ^{28} promoter	97.94	0.7561	71.64	99.23
σ^{32} promoter	95.38	0.7361	71.82	98.05
σ^{38} promoter	94.58	0.2242	7.36	99.85
σ^{54} promoter	98.11	0.6714	59.57	99.42
σ^{70} promoter	85.94	0.7109	95.22	72.47

See Equation 1. Acc, accuracy; Sn, sensitivity; Sp, specificity.
[a]The results reported in Liu et al.[12]
[b]The predictor proposed in this study.

From Table 1 we can see the following: (1) for the first-layer prediction, the iPromoter-2L2.0 outperformed all the other methods in terms of all the four performance measures (cf. Equation 24); (2) for the second-layer prediction, the iPromoter-2L2.0 outperformed iPromoter-2L for the prediction of σ^{24} promoters, σ^{28} promoters, σ^{32} promoters, σ^{54} promoters, and σ^{70} promoters in terms of accuracy (Acc) and Matthew's correlation coefficient (MCC), and its performance is comparable with that of iPromoter-2L for the prediction of σ^{38} promoters. The reasons for the performance improvement of the iPromoter-2L predictor is that it is based on the SCW algorithm, which is able to more accurately extract the sequence features to discriminate the promoters and their types.

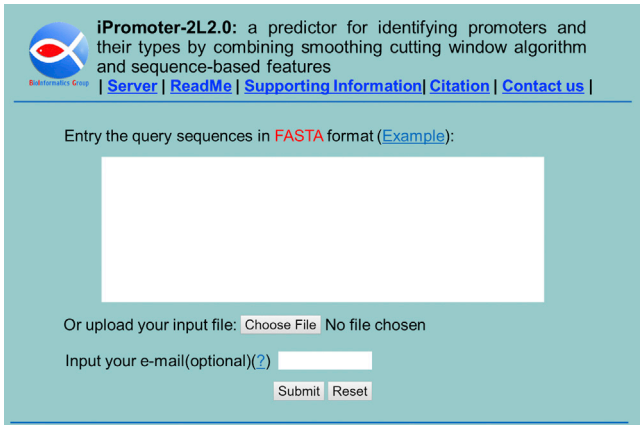

Figure 1. A Screenshot of the Homepage of the Web Server for iPromoter-2L2.0
iPromoter-2L2.0 can be accessed at http://bliulab.net/iPromoter-2L2.0/.

It can be anticipated that the proposed SCW algorithm would have many potential applications, such as enhancer prediction, DNA replication origin prediction, etc.

Web Server and Its User Guide

We established a web server for iPromoter-2L2.0 so as to help the readers to use the proposed method by following the steps below.

Step 1. Click the hyperlink http://bliulab.net/iPromoter-2L2.0/ to access the homepage as shown in Figure 1. An introduction to the web server is given in the Read Me.

Step 2. Copy/paste or type the query DNA sequences into the input box at the center of Figure 1 or upload the data by the Browse button.

Step 3. Click on the Submit button—you will see the predicted results. If using the example sequences for the prediction, you will see the following results: (1) both the first and the second query sequences are non-promoters; (2) the third query sequence is a σ^{70} promoter.

Step 4. On the results, the predictive result can be downloaded via clicking the Download button.

MATERIALS AND METHODS

Benchmark Dataset

To facilitate performance comparison of various methods, we employed the dataset $\mathbb{S}$[12] to construct the predictor and evaluate the performance of various methods, which can be formulated as[12]

$$\begin{cases} \mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^- \\ \mathbb{S}^+ = \mathbb{S}^+(\sigma^{24}) \cup \mathbb{S}^+(\sigma^{28}) \cup \mathbb{S}^+(\sigma^{32}) \cup \mathbb{S}^+(\sigma^{38}) \cup \mathbb{S}^+(\sigma^{54}) \cup \mathbb{S}^+(\sigma^{70}) \end{cases}, \quad \text{(Equation 1)}$$

where "∪" indicates the "union" in the theory; $\mathbb{S}^+$ indicates promoter samples; $\mathbb{S}^-$ indicates non-promoter samples; and $\mathbb{S}^+(\sigma^{24})$, $\mathbb{S}^+(\sigma^{28})$, $\mathbb{S}^+(\sigma^{32})$, $\mathbb{S}^+(\sigma^{38})$, $\mathbb{S}^+(\sigma^{54})$, and $\mathbb{S}^+(\sigma^{70})$ indicate six kinds of promoters. Specifically, the benchmark dataset $\mathbb{S}$ consists of 5,920 samples, half of which are promoters, and the others are non-promoters. $\mathbb{S}^+(\sigma^{24})$ contains 484 samples;$\mathbb{S}^+(\sigma^{28})$ contains 134 samples; $\mathbb{S}^+(\sigma^{32})$ contains 291 samples;$\mathbb{S}^+(\sigma^{38})$ contains 163 samples;$\mathbb{S}^+(\sigma^{54})$ contains 94 samples;$\mathbb{S}^+(\sigma^{70})$ contains 1,694 samples.

Sample Formulation

In this study, the DNA sequence samples were divided into several fragment windows by using the proposed SCW algorithm, and then for each fragment window, a sliding window approach was used to extract the sequence features by using kmer[13] and PseKNC.[6,14,15]

SCW Algorithm

Previous studies showed that the distribution of conservation scores between promoters and non-promoters are obviously different.[12] Here, we proposed the SCW algorithm to incorporate these sequence patterns into the predictor so as to improve the predictive performance.

A DNA sample is represented as

$$\mathbf{D} = N_1 N_2 \cdots N_i \cdots N_{81}, \quad \text{(Equation 2)}$$

where N_i denotes the i-th nucleotide at the sequence position i. It can be one of the following four nucleotides, i.e.,

$$N_i \in \{\, \mathrm{A(adenine)} \quad \mathrm{C(cytosine)} \quad \mathrm{G(guanine)} \quad \mathrm{T\ (thymine)} \,\}, \quad \text{(Equation 3)}$$

where $\in$ refers to "member of," a symbol in set theory.

To reflect the conservation score distribution patterns along $\mathbf{D}$, it was split into S+1 fragments $\rho([1,\tau_1-1],[\tau_1,\tau_2-1],\cdots,[\tau_s,L])$by the cutting points $\tau_j(j=1,\ 2,\cdots,S)$ (S is the total number of cutting points), which can be represented as

$$\begin{cases} \rho_1 = N_1 N_2 \cdots N_{\tau_1 - 1} \\ \rho_2 = N_{\tau_1} N_{\tau_1 + 1} \cdots N_{\tau_2 - 1} \\ \cdots \\ \rho_{S+1} = N_{\tau_s} N_{\tau_s + 1} \cdots N_L \end{cases}. \quad \text{(Equation 4)}$$

The cutting point τ_j is defined as follows:

$$\tau_j = \begin{cases} \varphi_1, \ \text{if } \varphi_1 > \alpha \text{ and } \varphi_2 - \varphi_1 > \alpha \\ \varphi_m, \quad \text{if } 1 < m < Z \text{ and } \varphi_m - \varphi_{m-1} > \alpha \text{ and } \varphi_{m+1} - \varphi_m > \alpha \\ \varphi_Z, \text{if } L - \varphi_Z > \alpha \text{ and } \varphi_Z - \varphi_{Z-1} > \alpha \\ \text{is not a cutting point, otherwise} \end{cases}, \quad \text{(Equation 5)}$$

where$\alpha$is a distance threshold, which was set as 8 in this study, $\varphi$is the candidate cutting point, and Z is the total number ofφ.For a given sequence position i, φ is defined as

$$\varphi_m = \begin{cases} i, & \text{if } \mathrm{SSD}_i < \mathrm{SSD}_{i-1} \text{ and } \mathrm{SSD}_i < \mathrm{SSD}_{i+1} \text{ and } 1 < i < L \\ & 1, \text{if } \mathrm{SSD}_i < \mathrm{SSD}_{i+1} \text{ and } i = 1 \\ & Z, \text{if } \mathrm{SSD}_i < \mathrm{SSD}_{i-1} \text{ and } i = L \\ & \text{is not a candidate cutting point, otherwise} \end{cases}, \quad \text{(Equation 6)}$$

where SSD_i represents the smooth standard deviation of the average conservation score (CS) of sequence position i, which can be calculated by

$$\mathrm{SSD}_i = \begin{cases} \frac{1}{5} \sum_{k=i-2}^{i+2} \mathrm{SD}_k, \ 2 < i < L-1 \\ \frac{1}{i+2} \sum_{k=1}^{i+2} \mathrm{SD}_k, \ i = 1, 2 \\ \frac{1}{L-i+3} \sum_{k=i-2}^{L} \mathrm{SD}_k, \ i = L-1, L \end{cases}, \quad \text{(Equation 7)}$$

where k is the sequence position and SD_k is the standard deviation of the average CS at the k-th sequence position, which can be calculated by

$$\mathrm{SD}_k = \sqrt{\frac{1}{Y} \sum_{y=1}^{Y} \left(\varepsilon_k^y - \mu\right)^2}, \quad \text{(Equation 8)}$$

where Y represents number of labels, which is equal to 2 for the first layer and 6 for the second layer. ε_k^y denotes the y-th class samples' average CS at the k-th sequence position, which can be calculated by the approach introduced in Schneider and Stephens.[16] μ is the average CS of all labels at the k-th position.

The conservation profiles and the standard deviations of promoters and non-promoters are shown in Figure 2A, and the conservation profile and the standard deviation of each promoter type are shown in Figure 3A. The smooth standard deviation curves are shown in Figures 2B and 3B. The DNA sequences were divided into several fragments by SCW as shown in Figures 2C and 3C. The pseudo-code of SCW algorithm is shown in Box 1.

After the process shown in Box 1, each DNA sequence in $\mathbb{S}$ (cf. Equation 1) was divided into four fragments ([1, 28], [29, 44], [45, 56], [57, 81]), and each DNA sequence in $\mathbb{S}^+$ (cf. Equation 1) was divided into four fragments ([1, 17], [18, 41], [42, 56], [57, 81]). Then for each fragment, the sliding-window approach was used to extract the features.

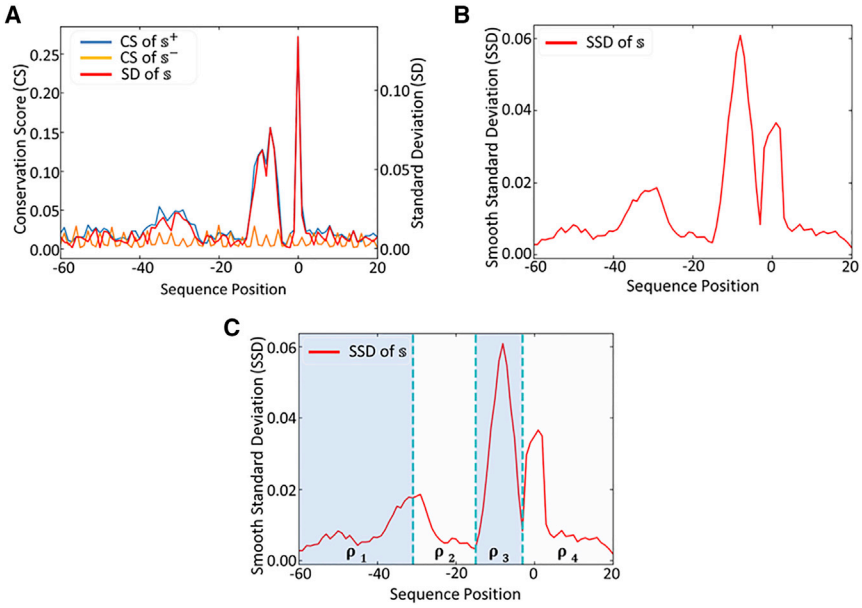

Figure 2. A Flowchart Shows the Steps of the Proposed Smoothing Cutting Window Algorithm for the First-Layer Prediction

The standard deviations shown in (A) are converted into the smooth standard deviations as shown in (B), based on which the DNA sequences are divided into several fragments, as shown in (C).

A sliding window can be expressed by $[\xi, \delta]$, where ξ is the width of the window and δ is the step of sliding window. For each fragment obtained, the number of the segments produced by $[\xi, \delta]$ along the fragment sequence is given by[12]

$$\eta = \mathrm{INT}\left[\frac{|\rho_i| - \xi + \delta}{\delta}\right], \qquad \text{(Equation 9)}$$

where "INT" is an "integer-cutting operator." $|\rho_i|$ denotes the length of the i-th fragment. For example, assuming $|\rho_i| = 29$, $\xi = 6$, and $\delta = 1$ in Equation 9, we obtain $\eta = 24$. For example, we can obtain 24 DNA segments with the sliding window of $[6, 1]$ on the i-th fragment of length 29.

kmer

kmer[13] is a simple and effective method to extract the information in the DNA sequence. By using kmer, the DNA sequence fragment ρ (cf. Equation 4) can be represented as

$$\boldsymbol{\rho} = \left[f_1^{\mathrm{kmer}} \quad f_2^{\mathrm{kmer}} \quad \cdots \quad f_i^{kmer} \quad \cdots \quad f_{4^k}^{\mathrm{kmer}}\right]^{\mathbf{T}}, \qquad \text{(Equation 10)}$$

where f_i^{kmer} $(i = 1, 2, \cdots, 4^k)$ is the frequencies of k neighboring nucleotides in the fragment ρ,and $\mathbf{T}$ represents transpose operator. For example, Equation 10 is a 4-mer vector when $k = 4$.

$$\begin{aligned}\boldsymbol{\rho} &= [f(\mathrm{AAAA}) \quad f(\mathrm{AAAC}) \quad f(\mathrm{AAAT}) \quad \cdots \quad f(\mathrm{TTTT})]^{\mathbf{T}} \\ &= \left[f_1^{4\mathrm{mer}} \quad f_2^{4\mathrm{mer}} \quad f_3^{4\mathrm{mer}} \quad \cdots \quad f_{256}^{4\mathrm{mer}}\right]^{\mathbf{T}}.\end{aligned} \qquad \text{(Equation 11)}$$

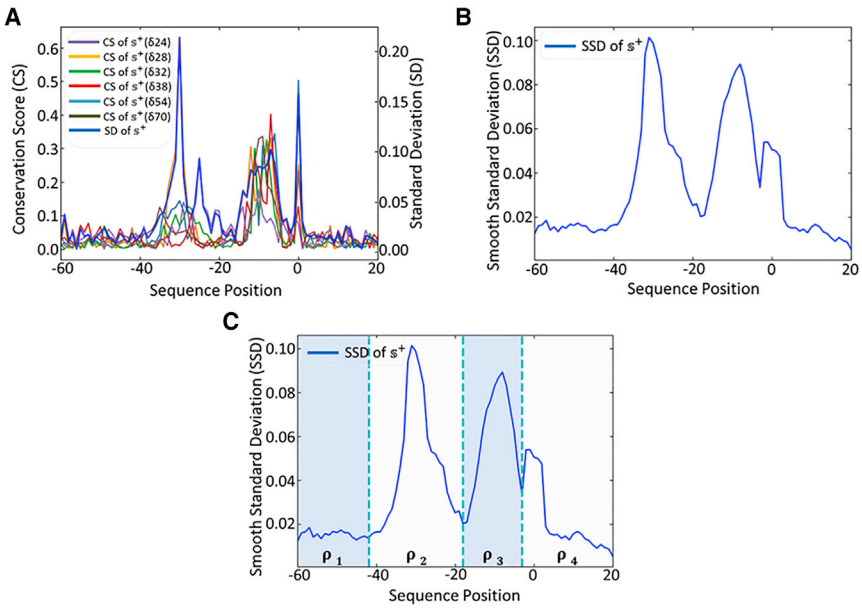

Figure 3. A Flowchart Shows the Process of the Proposed Smoothing Cutting Window Algorithm for the Second-Layer Prediction

The SDs shown in (A) are converted into the smooth SDs as shown in (B), based on which the DNA sequences are divided into several fragments, as shown in (C).

Box 1 Algorithm: Smoothing Cutting Window

Parameters: sequence length L, number of label Y

Input: DNA sequence in Equation 1

Output: **cutting** points $\tau_1, \tau_2, \ldots \tau_s$

For $y = 1$ to Y **do**

For $i = 1$ to L **do**

Calculate **conservation** score ε_i^y

End for

End for

For $i = 1$ to L **do**

Calculate $\mathbf{SSD_i}$ by Equation 7

End for

Calculate **cutting** points $\tau_1, \tau_2, \ldots \tau_s$ by Equations 5 and 6 and SSD

Return $\tau_1, \tau_2, \ldots \tau_s$

PseKNC

The PseKNC[6] incorporates the short-range sequence information, the long-range sequence information, and the physicochemical properties of the dinucleotides,[6] which can formulate the DNA sequence fragment ρ of Equation 4 as

$$\boldsymbol{\rho} = \left[f_1^{\text{PseKNC}} \quad f_2^{\text{PseKNC}} \quad \cdots \quad f_{4^k}^{\text{PseKNC}} \quad f_{4^k+1}^{\text{PseKNC}} \quad \cdots \quad f_{4^k+\lambda}^{\text{PseKNC}} \right]^{\mathrm{T}}. \tag{Equation 12}$$

PseKNC[6] has three parameters: k, λ (the number of sequence correlations considered[17]), and w (the weight factor). Each of the parameters has been clearly defined in a paper[6] and a comprehensive review.[18]

The kmer and PseKNC can be easily generated by some exiting tools, such as Pse-in-One[19] and PseKNC-General.[14]

Operation Engine

Support vector machines (SVMs) were successfully applied in several bioinformatics problems (B.L., C. L., and K. Yan, unpublished data).[20–24] In this study, we employed SVMs to build the predictor. We used the SVM with radial basis function (RBF) kernel in the Scikit-learn package.[25] The SVM has two parameters: C (regularization) and γ (kernel width).

Accordingly, when combining sliding-window approach and SVM based on kmer or PseKNC, there are a total of $(2 + 2 + 1) = 5$, or $(2 + 2 + 3) = 7$ parameters, respectively. The values of C and γ will be given later.

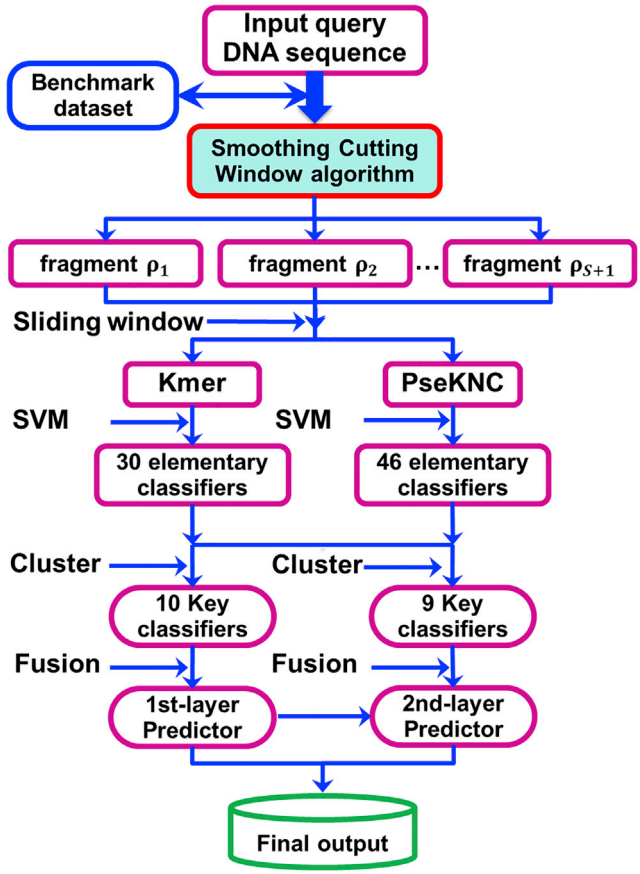

Figure 4. A Flowchart Shows How iPromoter-2L2.0 Is Working

For the sliding-window with,

$$\begin{cases} 5 \le \xi \le 9 & \text{with step gap } \triangle = 1 \\ 1 \le \delta \le 2 & \text{with step gap } \triangle = 1 \end{cases}. \tag{Equation 13}$$

For the kmer approach with

$$k = 1, 2, 3, \tag{Equation 14}$$

30 elementary classifiers can be developed, as denoted by

$$\mathcal{C}(i), \quad (i = 1, 2, \cdots, 30). \tag{Equation 15}$$

For the PseKNC approach with

$$\begin{cases} 1 \le k \le 4 & \text{with step gap } \triangle = 1 \\ 2 \le \lambda \le \xi - k & \text{with step gap } \triangle = 3, \\ & w = 0.5 \end{cases} \tag{Equation 16}$$

46 elementary classifiers can be developed, denoted by

$$\mathcal{C}(i), (i = 31, 32, \cdots, 76). \tag{Equation 17}$$

Therefore, we have a total of 30 + 46 = 76 elementary classifiers.

Table 2. The Six Key Classifiers for the First-Layer Prediction

Key Classifier	Feature Vector	Dimension
$\mathcal{C}^1(1)$	kmer[a]	768
$\mathcal{C}^1(2)$	kmer[b]	396
$\mathcal{C}^1(3)$	kmer[c]	2,880
$\mathcal{C}^1(4)$	kmer[d]	624
$\mathcal{C}^1(5)$	PseKNC[e]	1,080
$\mathcal{C}^1(6)$	PseKNC[f]	11,880
$\mathcal{C}^1(7)$	PseKNC[g]	46,440
$\mathcal{C}^1(8)$	PseKNC[h]	1,566
$\mathcal{C}^1(9)$	PseKNC[i]	2,808
$\mathcal{C}^1(10)$	PseKNC[j]	729

[a]The parameters used: $\xi = 5$, $\delta = 1$, $k = 1$, $C = 2^3$, $\gamma = 2^{-6}$.
[b]The parameters used: $\xi = 5$, $k = 1$, $C = 2$, $\gamma = 2^{-4}$.
[c]The parameters used: $\xi = 6$, $\delta = 1$, $k = 2$, $C = 2$, $\gamma = 2^{-4}$.
[d]The parameters used: $\xi = 8$, $\delta = 1$, $k = 1$, $C = 2^3$, $\gamma = 2^{-6}$.
[e]The parameters used: $\xi = 6$, $\delta = 1$, $k = 1$, $\lambda = 2$, $w = 0.5$, $C = 2^3, \gamma = 2^{-4}$.
[f]The parameters used: $\xi = 6$, $\delta = 1$, $k = 3$, $\lambda = 2$, $w = 0.5$, $C = 2^3, \gamma = 2^{-4}$.
[g]The parameters used: $\xi = 6$, $\delta = 1$, $k = 4$, $\lambda = 2$, $w = 0.5$, $C = 2, \gamma = 2^{-4}$.
[h]The parameters used: $\xi = 7$, $\delta = 2$, $k = 2$, $\lambda = 2$, $w = 0.5$, $C = 2, \gamma = 2^{-2}$.
[i]The parameters used: $\xi = 8$, $\delta = 1$, $k = 2$, $\lambda = 2$, $w = 0.5$, $C = 2^3, \gamma = 2^{-4}$.
[j]The parameters used: $\xi = 8$, $\delta = 2$, $k = 1$, $\lambda = 5$, $w = 0.5$, $C = 2, \gamma = 2^{-2}$.

Table 3. The 10 Key Classifiers for the Second-Layer Prediction

Key Classifier	Feature Vector	Dimension
$\mathcal{C}^2(1)$	kmer[a]	1,584
$\mathcal{C}^2(2)$	kmer[b]	2,688
$\mathcal{C}^2(3)$	PseKNC[c]	11,880
$\mathcal{C}^2(4)$	PseKNC[d]	1,008
$\mathcal{C}^2(5)$	PseKNC[e]	3,528
$\mathcal{C}^2(6)$	PseKNC[f]	1,566
$\mathcal{C}^2(7)$	PseKNC[g]	2,808
$\mathcal{C}^2(8)$	PseKNC[h]	729
$\mathcal{C}^2(9)$	PseKNC[i]	1,296

[a]The parameters used: $\xi = 5$, $\delta = 2$, $k = 2$, $C = 2^4$, $\gamma = 2^{-4}$.
[b]The parameters used: $\xi = 7$, $\delta = 1$, $k = 2$, $C = 2^4$, $\gamma = 2^{-4}$.
[c]The parameters used: $\xi = 6$, $\delta = 1$, $k = 3$, $\lambda = 2$, $w = 0.5$, $C = 2^4, \gamma = 2^{-4}$.
[d]The parameters used: $\xi = 7$, $\delta = 1$, $k = 1$, $\lambda = 2$, $w = 0.5$, $C = 2^4, \gamma = 2^{-1}$.
[e]The parameters used: $\xi = 7$, $\delta = 1$, $k = 2$, $\lambda = 5$, $w = 0.5$, $C = 2, \gamma = 2^{-1}$.
[f]The parameters used: $\xi = 7$, $\delta = 2$, $k = 2$, $\lambda = 2$, $w = 0.5$, $C = 2^4, \gamma = 2^{-1}$.
[g]The parameters used: $\xi = 8$, $\delta = 1$, $k = 2$, $\lambda = 2$, $w = 0.5$, $C = 2^4, \gamma = 2^{-1}$.
[h]The parameters used: $\xi = 8$, $\delta = 2$, $k = 1$, $\lambda = 5$, $w = 0.5$, $C = 2^4, \gamma = 2^{-1}$.
[i]The parameters used were as follows: $\xi = 9$, $\delta = 1$, $k = 1$, $\lambda = 5$, $w = 0.5$, $C = 2^4$, $\gamma = 2^{-1}$.

Ensemble Learning

Inspired by the previous studies,[13,26–32] by using a voting system, a series of individual predictors can develop an ensemble predictor with better prediction quality.

When developing an ensemble learning model, there are two fundamental issues: the selection of the individual classifiers with low correlation from the elementary classifiers and the construction of an ensemble classifier by fusing the selected classifiers. In this study, we employed the affinity propagation (AP) clustering algorithm[33] to cluster the elementary classifiers based on the distance among classifiers. For each cluster, one key classifier was selected.

In order to measure the complementarity of different elementary classifiers, the distance between any two elementary classifiers $\mathcal{C}(i)$ and $\mathcal{C}(j)$ was measured by the following equation:

$$\text{Distance}(\mathcal{C}(i),\mathcal{C}(j)) = \sqrt{\frac{1}{m}\sum_{k=1}^{m}\left(d_{ik}\Delta d_{jk}\right)}, \quad \text{(Equation 18)}$$

where m is the training sample number, d_{ik} is the classification probability of classifier $\mathcal{C}(i)$ on the k-th sample, and $d_{ik}\Delta d_{jk}$ is calculated by

$$d_{ik}\Delta d_{jk} = \begin{cases} \frac{1}{Y}\sum_{y=1}^{Y}\left(d_{iky} - d_{jky}\right)^2, \text{if } C(i) \text{ and } C(j) \text{have different prediction on the } k-\text{th sample} \\ 0, \quad \text{otherwise} \end{cases}, \quad \text{(Equation 19)}$$

where Y represents number of labels. Y was set as 2 and 6 for promoter identification and their type prediction, respectively. d_{iky} represents the probability of $\mathcal{C}(i)$ predicting k-th sample as category y. By using Equations 18 and 19, the distance between any elementary classifiers can be accurately measured. The range of Distance$(\mathcal{C}(i),\mathcal{C}(j))$ is from 0 to 1, where 1 indicates the predictive results of two classifiers are completely complementary and 0 means that their results are identical. The elementary classifiers were then grouped into different clusters by using the AP clustering algorithm.[33]

The flowchart of the proposed iPromoter-2L2.0 predictor is shown in Figure 4.

For the first layer, 10 key classifiers were obtained (Table 2) as formulated by

$$\mathcal{C}^1(i),\ (i = 1, 2, \cdots, 10). \quad \text{(Equation 20)}$$

For the second layer, nine key classifiers were obtained (Table 3) as formulated by

$$\mathcal{C}^2(i),\ (i = 1, 2, \cdots, 9). \quad \text{(Equation 21)}$$

By fusing the 10 key classifiers (cf. Equation 20) following this study,[13] we can obtain the first-layer ensemble predictor as given by

$$\mathcal{C}^{\mathrm{E1}} = \mathcal{C}^1(1) \forall \mathcal{C}^1(2) \forall \cdots \forall \mathcal{C}^1(10) = \forall_{i=1}^{10} \mathcal{C}^1(i). \quad \text{(Equation 22)}$$

By fusing the nine key classifiers (cf. Equation 21), we can obtain the second-layer ensemble predictor given by

$$\mathcal{C}^{\mathrm{E2}} = \mathcal{C}^2(1) \forall \mathcal{C}^2(2) \forall \cdots \forall \mathcal{C}^2(9) = \forall_{i=1}^{9} \mathcal{C}^2(i), \quad \text{(Equation 23)}$$

where the symbol $\forall$ in Equations 22 and 23 means that linear combination of the key individual classifiers. The weight factors were optimized by the genetic algorithm,[34] and the parameters (population size, evolutional generations) of genetic algorithm were set as 200 and 2,000, respectively, for the first and second layers.

Cross-Validation and Performance Measures

The performance of various predictors was evaluated by using 5-fold cross-validation with the following performance measures:[12]

$$\begin{cases} \mathrm{Sn}(i) = 1 - \dfrac{N_-^+(i)}{N^+(i)} & 0 \le \mathrm{Sn} \le 1 \\ \mathrm{Sp}(i) = 1 - \dfrac{N_+^-(i)}{N^-(i)} & 0 \le \mathrm{Sp} \le 1 \\ \mathrm{Acc}(i) = 1 - \dfrac{N_-^+(i) + N_+^-(i)}{N^+(i) + N^-(i)} & 0 \le \mathrm{Acc} \le 1 \\ \mathrm{MCC}(i) = \dfrac{1 - \left(\dfrac{N_-^+(i)}{N^+(i)} + \dfrac{N_+^-(i)}{N^-(i)}\right)}{\sqrt{\left(1 + \dfrac{N_+^-(i) - N_-^+(i)}{N^+(i)}\right)\left(1 + \dfrac{N_-^+(i) - N_+^-(i)}{N^-(i)}\right)}} & -1 \le \mathrm{MCC} \le 1 \end{cases}, \quad \text{(Equation 24)}$$

where $i = 1, 2, \cdots, Y$, and Y is the number of classes of this system. i is the i-th class or type. For the first-layer prediction, the value of Y is 2, and the value of i represents the promoter ($i = 1$) or non-promoter ($i = 2$). Similarly, for the second-layer prediction, the value of Y is 6 and the value of i is 1, 2, 3, 4, 5, or 6 for σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54}, or σ^{70} promoters, respectively. For the detail of these performance measures, please refer to a recent study.[12]

AUTHOR CONTRIBUTIONS

B.L. provided the main idea of the manuscript and wrote the manuscript. K.L. did the experiments and wrote the manuscript.

CONFLICTS OF INTEREST

The authors declare no competing interests.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61672184 and 61822306), the Fok Ying-Tung Education Foundation for Young Teachers in the Higher Education Institutions of China (161063), and the Scientific Research Foundation in Shenzhen (JCYJ20180306172207178).

REFERENCES

1. Borukhov, S., and Nudler, E. (2008). RNA polymerase: the vehicle of transcription. Trends Microbiol. *16*, 126–134.
2. Silva, S.D.A.E., and Echeverrigaray, S. (2012). Bacterial Promoter Features Description and Their Application on E. coli In Silico Prediction and Recognition Approaches (Intech).
3. Janga, S.C., and Collado-Vides, J. (2007). Structure and evolution of gene regulatory networks in microbial genomes. Res. Microbiol. *158*, 787–794.
4. Potvin, E., Sanschagrin, F., and Levesque, R.C. (2008). Sigma factors in Pseudomonas aeruginosa. FEMS Microbiol. Rev. *32*, 38–55.
5. Lin, H., Deng, E.Z., Ding, H., Chen, W., and Chou, K.C. (2014). iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. Nucleic Acids Res. *42*, 12961–12972.
6. Chen, W., Lei, T.-Y., Jin, D.-C., Lin, H., and Chou, K.-C. (2014). PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition. Anal. Biochem. *456*, 53–60.
7. Li, Q.Z., and Lin, H. (2006). The recognition and prediction of sigma70 promoters in Escherichia coli K-12. J. Theor. Biol. *242*, 135–141.
8. He, W., Jia, C., Duan, Y., and Zou, Q. (2018). 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. BMC Syst. Biol. *12* (*Suppl 4*), 44.
9. Zhang, C.T. (1997). A symmetrical theory of DNA sequences and its applications. J. Theor. Biol. *187*, 297–306.
10. Zhang, C.T., Zhang, R., and Ou, H.Y. (2003). The Z curve database: a graphic representation of genome sequences. Bioinformatics *19*, 593–599.
11. Song, K. (2012). Recognition of prokaryotic promoters based on a novel variable-window Z-curve method. Nucleic Acids Res. *40*, 963–971.
12. Liu, B., Yang, F., Huang, D.S., and Chou, K.-C. (2018). iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. Bioinformatics *34*, 33–40.
13. Liu, B., Long, R., and Chou, K.-C. (2016). iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. Bioinformatics *32*, 2411–2418.
14. Chen, W., Zhang, X., Brooker, J., Lin, H., Zhang, L., and Chou, K.-C. (2015). PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. Bioinformatics *31*, 119–120.

15. Liu, B., Liu, F., Fang, L., Wang, X., and Chou, K.-C. (2015). repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. Bioinformatics *31*, 1307–1309.

16. Schneider, T.D., and Stephens, R.M. (1990). Sequence logos: a new way to display consensus sequences. Nucleic Acids Res. *18*, 6097–6100.

17. Chou, K.-C. (2005). Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics *21*, 10–19.

18. Liu, B. (2017). BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. Brief. Bioinform. Published online December 19, 2017. https://doi.org/10.1093/bib/bbx165.

19. Liu, B., Liu, F., Wang, X., Chen, J., Fang, L., and Chou, K.-C. (2015). Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res. *43* (W1), W65–W71.

20. Li, D., Ju, Y., and Zou, Q. (2016). Protein Folds Prediction with Hierarchical Structured SVM. Curr. Proteomics *13*, 79–85.

21. Zhang, N., Sa, Y., Guo, Y., Lin, W., Wang, P., and Feng, Y. (2018). Discriminating Ramos and Jurkat Cells with Image Textures from Diffraction Imaging Flow Cytometry Based on a Support Vector Machine. Curr. Bioinform. *13*, 50–56.

22. Wang, S.P., Zhang, Q., Lu, J., and Cai, Y.D. (2018). Analysis and Prediction of Nitrated Tyrosine Sites with the mRMR Method and Support Vector Machine Algorithm. Curr. Bioinform. *13*, 3–13.

23. Chen, W., Lv, H., Nie, F., and Lin, H. (2019). i6mA-Pred: Identifying DNA N6-methyladenine sites in the rice genome. Bioinformatics *35*, 2796–2800.

24. Feng, P.M., Chen, W., Lin, H., and Chou, K.C. (2013). iHSP-PseRAAAC: Identifying the heat shock protein families using pseudo reduced amino acid alphabet composition. Anal. Biochem. *442*, 118–125.

25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2012). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

26. Liu, B., Wang, S., Long, R., and Chou, K.-C. (2017). iRSpot-EL: identify recombination spots with an ensemble learning approach. Bioinformatics *33*, 35–41.

27. Liu, B., Yang, F., and Chou, K.-C. (2017). 2L-piRNA: A two-layer ensemble classifier for identifying piwi-interacting RNAs and their function. Mol. Ther. Nucleic Acids *7*, 267–277.

28. Lin, C., Chen, W., Qiu, C., Wu, Y., Krishnan, S., and Zou, Q. (2014). LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy. Neurocomputing *123*, 424–435.

29. Zou, Q., Guo, J., Ju, Y., Wu, M., Zeng, X., and Hong, Z. (2015). Improving tRNAscan-SE Annotation Results via Ensemble Classifiers. Mol. Inform. *34*, 761–770.

30. Zou, Q., Wang, Z., Guan, X., Liu, B., Wu, Y., and Lin, Z. (2013). An approach for identifying cytokines based on a novel ensemble classifier. BioMed Res. Int. *2013*, 686090.

31. Yan, K., Fang, X., Xu, Y., and Liu, B. (2019). Protein Fold Recognition based on Multi-view Modeling. Bioinformatics. Published online January 21, 2019. https://doi.org/10.1093/bioinformatics/btz040.

32. Liu, B., and Zhu, Y. (2019). ProtDec-LTR3.0: protein remote homology detection by incorporating profile-based features into Learning to Rank. IEEE Access. Published online July 18, 2019. https://doi.org/10.1109/ACCESS.2019.2929363.

33. Frey, B.J., and Dueck, D. (2007). Clustering by passing messages between data points. Science *315*, 972–976.

34. Mitchell, M. (1998). An Introduction to Genetic Algorithms (MIT Press).