Eukaryotic promoter prediction based on relative entropy and positional information

Shuanhu Wu[*]
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
and School of Computer Science and Technology, Yantai University, Yantai 264005, China

Xudong Xie
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
and Department of Electronic Engineering, Tsinghua University, Beijing, China

Alan Wee-Chung Liew[†]
School of Information & Communication Technology, Griffith University, Queensland, Australia

Hong Yan[‡]
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
and School of Electronic and Information Engineering, University of Sydney, NSW, 2006, Australia
(Received 14 July 2006; revised manuscript received 14 December 2006; published 12 April 2007)

The eukaryotic promoter prediction is one of the most important problems in DNA sequence analysis, but also a very difficult one. Although a number of algorithms have been proposed, their performances are still limited by low sensitivities and high false positives. We present a method for improving the performance of promoter regions prediction. We focus on the selection of most effective features for different functional regions in DNA sequences. Our feature selection algorithm is based on relative entropy or Kullback-Leibler divergence, and a system combined with position-specific information for promoter regions prediction is developed. The results of testing on large genomic sequences and comparisons with the PromoterInspector and Dragon Promoter Finder show that our algorithm is efficient with higher sensitivity and specificity in predicting promoter regions.

DOI: 10.1103/PhysRevE.75.041908 PACS number(s): 87.14.Gg

I. INTRODUCTION

With the completion of the human genome draft [1,2], a challenging task today is to find the genes and their regulatory network. It is possible to use the prediction of promoter sequences and transcriptional start point as a signal; i.e., by knowing the position of a promoter, one can deduce at least the approximate start of the transcript, thus delineating one end of the gene. A number of systems for promoter prediction have been developed. However, as indicated by Fickett and Hatzigeorgiou [3] and Prestridge [4], recognition of eukaryotic promoters is still a difficult task. The general problem is that the level of false positive predictions appears to be unacceptably high.

PromoterInspector [5] is able to reduce false positive predictions substantially while maintaining relatively high true positives (sensitivity). The method is based on content analysis of promoter sequence represented by word groups rather than specific transcription elements and predicts regions containing a promoter, where each word group is uniquely defined by a set of oligonucleotides and a number of undefined base-pairs (wildcards, "N"). After the appearance of PromoterInspector, several other advanced systems for promoter prediction have also been developed so that the false positive predictions are lowered [6–10]. The strategies used by these systems are different. For instance, Dragon Promoter Finder [8] is based on similar idea as PromoterInspector but predicts the actual transcription start site (TSS). Eponine [9] also aims to predict TSS but based on analyzing the well-known TATA-box (a DNA sequence with consensus TATAAA at about 25 nucleotides upstream of the transcription start site) and its flanking regions of C-G enrichment. The second group of systems such as CpG-Promoter [6] and CpGProd [10] make predictions of a region by searching for CpG islands that should be in proximity with the TSS while other systems such as FirstEF [7] is based on quadratic discriminant analysis of promoters, first exons, and the first donor site, using CpG islands. Compared with PromoterInspector, all of these systems with the exception of CpG-Promoter [6] claimed better overall performance on the data sets used. But a comparative study [11] for these algorithms showed that it is difficult to assess their performance, as this depends very much on the individual problems to be solved. It was suggested that to obtain an initial annotation of the whole genomes, PromoterInspector and Dragon Promoter Finder should be the first choice because these two algorithms do not depend on special signals (such as CpG islands), which is not common to all promoters.

The underlying principle of the existing algorithms for promoter region recognition is that the properties of the promoter regions are different from the properties of other functional regions. Existing algorithms can be subdivided into three main categories: (1) search by signal, (2) search by content, and (3) search by CpG island. "Search by signals"

[*] Electronic address: wushuanhu@gmail.com
[†] Electronic address: a.liew@griffith.edu.au
[‡] Electronic address: h.yan@cityu.edu.hk

techniques are based on the identification of putative transcriptional patterns such as TATA-box and CAAT-box [a DNA sequence with consensus GG(T/C)CAATCT at about 75 nucleotides upstream of the transcription start site]. The CAAT box signals the binding site for the RNA transcription factor) in DNA sequences, but these patterns cannot be the only determinants of the promoter function. For instance, in one study it was found that applying a Buchers TATA-box weight matrix to a set of mammalian nonpromoter DNA sequences resulted in an average of one predicted TATA-box every 120 bp [12]. That means that the application of some known transcriptional motifs to the prediction of promoters introduces many false positives. "Search by content" techniques are often based on the difference in the local base and local word composition between regulatory and nonregulatory DNA regions. This class of algorithms assumes that the difference is caused by the presence of transcriptional signals, such as the binding motifs for transcriptional regulators in the promoter regions. This concept was explored by analyzing the most frequent hexamers (differential hexamer frequency) [13], other variant-length motifs, and short words [5,8]. "Search by CpG island" techniques are based on the fact that most human promoters are correlated with CpG islands and many genes are recognized and validated successfully by using CpG islands as gene markers [1,2]. CpG sites are regions of DNA where a cytosine nucleotide occurs next to a guanine nucleotide in the linear sequence of bases. "CpG" stands for cytosine and guanine separated by a phosphodiester link, which links the two nucleosides together in the DNA sequence. CpG islands are found around gene starts in approximately half mammalian promoters and are estimated to be associated with ~60% of human promoters [14]. Therefore, it is a good indicator for the presence of promoters. Algorithms such as the CpG-promoter [6], CpGProd [10], and FirstEF [7] make use of the information of CpG islands. Nevertheless, we must bear in mind that not all the human promoters are related to CpG islands, and from this point of view, there are at least ~40% false predictions and correct predictions are limited and cannot exceed 60%, if the prediction is based on CpG islands alone.

In this paper, we introduce a method for human promoter regions recognition. The proposed algorithm is word based, and it belongs to the category of "search by content." Our focus is on how to select the most effective features and use them to improve the prediction efficiency. We choose words from different functional regions in DNA sequences based on relative entropy or the Kullback-Leibler (KL) divergence [15]. This feature selection method is combined with position-specific information encoded in the position weight matrix for promoter regions prediction. We have tested our method on large genomic sequences and compared the results with those from PromoterInspector and Dragon Promoter Finder. The experiments show that our algorithm is efficient with higher sensitivity and specificity in predicting promoter regions. In Sec. II of the paper, we introduce our system model and architecture. Section III describes the feature selection strategy and promoter prediction algorithm. Experimental results and comparisons are provided in Sec. IV. The paper is concluded in Sec. V.

II. SYSTEM MODEL AND ARCHITECTURE

The underlying principle for promoter recognition is based on the fact that the properties of the promoter regions are different from other functional regions in DNA sequences. Many features may be associated with promoter sequences and functions. They include core promoter elements such as TATA-boxes, CAAT-boxes, and transcription initiation sites (INR), CpG islands, secondary structure elements like the HIV-1 TAR regions [16] (HIV-1: human immunodeficiency virus of type-1; TAR: transactivation-responsive region), cruciform DNA structures [17], and three-dimensional structures such as curved DNA [18]. Although most of these elements can be detected by means of computer-assisted sequence analysis, none of them are really promoter specific and can be found frequently outside promoters. Therefore, it is important to combine them to distinguish promoters from other DNA sequences, such as exon, intron, and 3′UTR (untranslated region in the 3′ end).

The number and distribution of words of length k or k words ($k>3$) in a DNA sequence can have biological significance. Some particularly important k words with $k \geqslant 4$ are useful for analyzing particular genomic subsequences. For example, four-word frequencies can be used to quantify the differences between E.coli promoter sequences and "average" genomic DNA [19]; coding and noncoding DNA can be distinguishable in terms of their pentamer (five-word) and hexamer (six-word) distributions [20]. Here, we wish to find k words that distinguish promoter sequence regions from other DNA genomic sequence regions. Therefore, we attempt to select the most effective k words that are overrepresented within the promoter regions compared with other DNA sequence regions and can help identify the DNA "signals" required for promoter functions. Thus, the focus of our work is on the following issues: (i) the relationship between word length and discriminability of promoter regions and other regions in the DNA sequences, (ii) how to select the words of fixed length with the highest discriminability, (iii) how to use the selected features to build a classifier to predict promoter regions. We tackle the former two issues using the KL divergence [15] and the last issue by the words' position-specific distribution score.

Our system consists of three classifiers: Promoter-Exon classifier, Promoter-Intron classifier, and Promoter-3′-UTR classifier. Each classifier is specialized to differentiate between promoter regions and one of the three nonpromoter regions: exon, intron, and 3′-UTR. The choice of these three nonpromoter regions follows that of PromoterInspector [5]. The prediction system assigns a sequence to the promoter class only if all three classifiers decide that the sequence belongs to this class.

III. WORD SELECTION BASED ON RELATIVE ENTROPY

Relative entropy can be interpreted as a distance, using what is called the KL divergence. Let p_{promoter}^{k} and $p_{\text{nonpromoter}}^{k}$ be the probability density functions of words in promoter sequences and in nonpromoter sequences for fixed word length k that has 4^k words totally. Where $p_{\text{nonpromoter}}^{k}$ comes

TABLE I. Maximum KL distance between the promoter words and nonpromoter words for different word lengths.

	KL distance		
Word length	Promoter vs Exon	Promoter vs Intron	Promoter vs $3'$UTR
$k=4$	0.290781	0.776157	0.583311
$k=5$	0.349264	0.929367	0.686333
$k=6$	0.406076	1.098631	0.786895
$k=7$	0.470600	1.437328	0.928344

TABLE II. The number of selected words by fixing 98% of maximum KL distance for three classifiers.

Word length	Promoter vs Exon	Promoter vs Intron	Promoter vs $3'$UTR
$k=5$	257	305	303
$k=6$	1019	1238	1189
$k=7$	4291	4965	4738

from one of the three nonpromoter regions: exon, intron, and $3'$-UTR. The KL divergence is defined as follows:

$$\delta(p^k_{\text{promoter}},p^k_{\text{nonpromoter}})=\sum_{i=1}^{4^k}p^k_{\text{promoter}}(i)\ln\frac{p^k_{\text{promoter}}(i)}{p^k_{\text{nonpromoter}}(i)}. \tag{1}$$

The KL divergence can be considered as a distance between the two probability densities, because it is always nonnegative, and zero if, and only if, the two distributions are identical. Our aim is to select an effective subgroup of the 4^k words that can maximally distinguish promoter sequences and nonpromoter sequences. This subgroup of words can be obtained by maximizing the following criterion function:

$$S=\underset{\{i|i\in\{1,2,\ldots,4^k\}\}}{\arg}\{\max\delta(p^k_{\text{promoter}},p^k_{\text{nonpromoter}})\}, \tag{2}$$

where S represents the set of subscripts of all the words in the subgroup that are selected. The maximization in Eq. (2) can be carried out by simply sorting $\{p^k_{\text{promoter}}(i)\ln\frac{p^k_{\text{promoter}}(i)}{p^k_{\text{nonpromoter}}(i)}, i\in S\}$ in descending order and then selecting the desirable number of words that forms the subgroup. This way, we can guarantee the selected words are of most discriminability relative to other selections. We remark that the idea of word selection has also been used in [21] and that the KL divergence has been employed in many signal processing applications [22].

Our promoter prediction system consists of three classifiers, so it needs three different groups of words to distinguish promoter regions from exons, introns and $3'$-UTR regions. These three groups of k words can be obtained by available training sets, and we will discuss it in detail in the experimental section.

IV. PROMOTER PREDICTION BASED ON POSITION INFORMATION

By using the proposed method described above, three different groups of k words, which are of most discriminablility between promoter and exon regions, promoter and intron regions, and promoter and $3'$-UTR regions, respectively, can be obtained according to the KL divergence. It is well known that certain motifs or word patterns appear more frequently in promoter regions than in nonpromoter regions and further the frequency of occurrence of words in the motifs is different for different position in the promoter region, so we represented this using a positional weight matrix (PWM). The PWM describes the patterns of word occurrence within a substring, and a PWM trained using the promoter sequences

TABLE III. Description of the large genomic sequences in the evaluation set.

Accession number	Description	Length (bp)	Number of TSS
AC002397	Complete sequence of mouse chromosome 6 BAC-284H12	227538	17
L44140	Homosapiens chromosome X region from filamin (FLN) gene to glucose-6-phosphate dehydrogenase (G6PD) gene. There are 13 known and six candidate genes in the sequence	219447	11
D87675	Homosapiens DNA for amyloid precursor protein	301692	1
AF017257	Homosapiens chromosome 21-derived BAC containing erythroblastosis virus oncogene homolog 2 protein (ets-2) gene	101569	1
AF146793	Mouse protein B, Clock, PFT27 and HSAR gene	204625	4
AC002368	Homosapiens Xq28 BAC PAC and cosmid clones containing FMR2 gene	324816	1
Total		1.38 Mb	35

TABLE IV. Results of large genomic sequence analysis.

Accession number	Method	TP[a]	FP[b]	% TP total predictions	% Coverage[c]
AC002397	PrometerInspector	4	1	80	23.5
	DPF ($s=0.45$)	6	4	60	35.2
	Eponine ($t=0.995$)	8	1	88.8	47
	FirstEF ($p=0.98$)	7	3	70	41.1
	Our System ($k=5$)	6	1	85.7	35.2
	($k=6$)	6	1	85.7	35.2
	($k=7$)	6	1	85.7	35.2
L44140	PromoterInspector	6	14	30	54.5
	DPF ($s=0.45$)	6	14	30	54.5
	Eponine ($t=0.995$)	6	12	33.3	54.5
	FirstEF ($p=0.98$)	6	11	35.2	54.5
	Our System ($k=5$)	8	15	38	72.7
	($k=6$)	8	13	40	72.7
	($k=7$)	7	13	40	63.6
D87675	PromoterInspector	1	2	33.3	100
	DPF ($s=0.45$)	1	3	25	100
	Eponine ($t=0.995$)	1	1	50	100
	FirstEF ($p=0.98$)	1	0	100	100
	Our System ($k=5$)	1	0	100	100
	($k=6$)	1	0	100	100
	($k=7$)	1	1	50	100
AF017257	PromoterInspector	1	0	100	100
	DPF ($s=0.45$)	1	0	100	100
	Eponine ($t=0.995$)	1	3	25	100
	FirstEF ($p=0.98$)	1	0	100	100
	Our System ($k=5$)	1	1	50	100
	($k=6$)	1	0	100	100
	($k=7$)	1	1	50	100
AF146793	PromoterInspector	1	2	33.3	25
	DPF ($s=0.45$)	1	4	20	25
	Eponine ($t=0.995$)	1	3	25	25
	FirstEF ($p=0.98$)	1	3	25	25
	Our System ($k=5$)	1	2	33.3	25
	($k=6$)	1	2	33.3	25
	($k=7$)	1	2	33.3	25
AC002368	PromoterInspector	1	1	50	100
	DPF ($s=0.45$)	1	3	25	100
	Eponine ($t=0.995$)	1	0	100	100
	FirstEF ($p=0.98$)	1	1	50	100
	Our System ($k=5$)	1	0	100	100
	($k=6$)	1	0	100	100
	($k=7$)	1	1	50	100

[a] TP: true positive.

[b] FP: false positive.

[c] Coverage: the percentage of true promoters in a sequence.

would give preference for highly probable word patterns that are associated with promoter sequences. Two PWMs are computed from the training set for each of the three classifiers, each corresponding to a different training set. For instance, for a promoter-exon classifier, two PWMs are generated in terms of a promoter training set and an exon training set, respectively. A total of six PWMs are calculated for our prediction system. Each element in PWM represents the probability distribution of some selected word at a corresponding position in the training sequence. For example, for the training set with sequence length 250, word length of 6, and the number of words chosen to be 1024, a PWM of order 1024×245 can be generated.

For each classifier in our prediction system, we can obtain one group of selected k words, W_k, and two corresponding position weight matrixes, $\text{PWM}_{\text{promoter}}$, computed from a promoter training set, and $\text{PWM}_{\text{nonpromoter}}$, computed from some nonpromoter training set (one of exon, intron and 3-UTR training sets). When an unknown sequence is input into a prediction system, two scores can be calculated by the corresponding two PWMs and a classification result is determined by these two scores. The prediction system assigns a sequence to the promoter class only if all three classifiers decide that the sequence belongs to this class. Let $p_{i,j}$ be an element of PWM at the ith row and the jth column that represents the probability of the ith selected k word at position j estimated from the training dataset and $S = c_1 c_2, \ldots, c_{L-k+2} c_{L-k+1}$ be the unknown input words sequence (L is the training-unknown input DNA sequence, and k is the word length). Then a score can be calculated by

$$S_{seq} = \sum_{\substack{c_j \in W_k \\ j=1,2,\ldots,L-k+1}} p_{i_{c_j},j}, \tag{3}$$

where i_{c_j} represents the row number of word c_j in the PWM. Since the words in each selected word group are dominant in the promoter region, we can expect that larger scores would be obtained when an unknown input sequence belongs to a promoter region and a smaller score would be obtained when an unknown input sequence belongs to a nonpromoter region. Based on this motivation, an unknown input sequence is predicated as the promoter in one of three classifiers if the following two conditions are satisfied simultaneously:

$$\frac{S_{\text{promoter}}}{S_{\text{nonpromoter}}} > T_1, \quad S_{\text{promoter}} > T_2, \tag{4}$$

where S_{promoter} and $S_{\text{nonpromoter}}$ are calculated by the selected words and the corresponding position weight matrices, $\text{PWM}_{\text{promoter}}$ and $\text{PWM}_{\text{nonpromoter}}$ according to Eq. (3), and T_1 and T_2 are two thresholds that can be optimized and tuned using the training sequence sets as follows. We first fixed a predefined sensitivity parameter—i.e., $s=50\%$. The initial T_1 and T_2 are then set to $T_1 = x(\text{avg\_promoter})/x(\text{avg\_nonpromoter})$ and $T_2 = [s(\text{avg\_promoter}) + s(\text{avg\_nonpromoter})]/2$, respectively, where the quantities avg_promoter and avg_nonpromoter are the average score of the randomly selected promoter training set on the $\text{PWM}_{\text{promoter}}$ and $\text{PWM}_{\text{nonpromoter}}$, respectively. Then T_1 and T_2 are adjusted in small increment such that s is close to the preset value while maximizing the selectivity. We add the second inequality in Eq. (4) based on biological consideration; i.e., a transcriptional factor needs some amount of binding sites and other conditions to initialize the transcription. Different from other promoter prediction systems that adopt more complex classifiers—for example, artificial neural networks [8], relevance vector machines [9], quadratic discriminant analyses [7], etc.—our classifier is intuitive and efficient. Traditional classifiers built for promoter recognition often tend to find compromised solutions that may results in too many false positives.

V. RESULTS

A. Training sequence sets

Our vertebrate promoter sequence set comes from the database of transcription start sites (DBTSS) [23], and we only take human promoter sequences as training set. For each sequence, a section is taken from 200 bp upstream to 50 bp downstream of the TSS. Vertebrate exon and intron sequences are extracted from the Exon/Intron database that can be downloaded from [24]. Vertebrate $3'$-UTR sequences are extracted from the UTR database [25]. All the training sequences are of length 250 bp and nonoverlapping. Redundant sequences are cleared by the program CleanUp [26] which results in the training sets consisting of 10513 promoter sequences, 6500 exon sequences, 8000 intron sequences, and 7500 $3'$-UTR sequences. For a fair evaluation of the prediction performance of our system, we removed from the training set any promoter sequences that appeared in the testing set, in which there are 313 promoters in the DBTSS that belong to the human chromosome 22, 11 to the chromosome X, and 1 to the chromosome 21, respectively.

B. Word length and discriminability analysis

In this section, we first illustrate the relationship between word length and discriminability by experiments and then a selection strategy is proposed. In our experiments, the word length k ranges from 4 to 7. We do not choose longer words since the number of words is very large and is not practical in real application. First we calculate the word's probability distributions for different word lengths ($k=4-7$) and the corresponding training sequence set, and then the maximum distance between the promoter words and nonpromoter words are estimated according to the KL divergence described in Sec. III. The computational results are provided in Table I. We can conclude from Table I that the longer the word, the larger the discriminability, but the number of words would increase exponentially. For example, the total number of words for $k=7$ is $4^7=16\,384$ and for $k=6$ is $4^6=4096$. This means that the number of selected words will also increase exponentially and that will increase the computational burden. In addition, with the increase in word length, most words will not appear simultaneously in a training-unknown input sequence since these sequences always take limited length; for example, there are at most 244 words in a 250 bp sequence length for word length $k=7$. To reduce the computational burden, we fix the percentage of the maximal KL divergence at 98% and calculate the number of words needed. In doing so, we ensure that each classifier is built on the same divergence. Table II shows the number of selected words for $k=5-7$. For the selected words, two PWMs are calculated for each promoter sensor through corresponding

TABLE V. Comparison of five prediction systems on the testing set shown in Table IV.

Method	TP	FP	S_e (%)[a]	S_p (%)[b]
PromoterInspector	14	20	40.0	41.2
DPF ($s=0.45$)	16	28	45.7	36.4
Eponine ($t=0.995$)	18	20	51.4	47.3
FirstEF ($p=0.98$)	17	18	48.6	48.5
Our system ($k=5$)	18	19	51.4	48.6
($k=6$)	18	16	51.4	52.9
($k=7$)	17	19	48.6	47.2

[a] Sensitivity: S_e=TP/(TP+FN).

[b] Specificity: S_p=TP/(TP+FP). FN: false negative. TP+FN=35.

training sequence sets and two threshold T_1 and T_2 for each promoter sensor are obtained.

C. Large genomic sequence analysis and comparisons

Identification of promoter regions in large genomic sequences is performed by a sliding window approach. A window is moved over sequences and its content is classified. The window length is set as 250 bp, and it is step 1 bp in our system. A promoter region is obtained by clustering the prediction outputs with a gap tolerance 1 kb.

To verify the validity of our system, we compare the performance of our system with four other promoter prediction systems: PromoterInspector [5], Dragon Promoter Finder (DPF) [8], Eponine, and FirstEF. These four methods are selected not only because they are accessible via the Internet but also because they are currently the best four prediction systems. The evaluation set for comparison is the same as that used in PromoterInspector and DPF and is currently a standard for evaluating the performance of promoter recognition system. This set consists of six GenBank genomic sequences with a total length of 1.38 Mb and 35 known TSSs. An overview of the sequences, their length, and the number of annotated TSS in the sequence is shown in Table III (see Table 3 in [5]). We adopt the same evaluating criterion used by PromoterInspector [5]: A predicted region is counted as correct if a TSS is located within the region or if a region boundary is within 200 bp 5′ of such a TSS. The main results and comparisons are presented in Tables IV and V. Table V is obtained by summing all the TP and FP over the entire test set for each algorithm and evaluated against the actual number of promoter which equals 35. In these experiments, PromoterInspector and our system are used with default settings, and DPF is used by setting $s=0.45$. The setting $s=0.45$ is found to give a balance sensitivity and specificity result. We observed that when the s of DPF is set too high, the number of false positives will increase much more rapidly than the number of true positives. For the same reason, we set $t=0.995$ for Eponine and $p=0.98$ for FisrtEF. For $s=0.45$, DPF detects similar number of true positives as our method, while its false positives are larger than our method. By comparing the results of DPF with PromoterInspector, we can see that although DPF can predict more promoters it also results in more false positives. Comparing the predicting results of our system with DPF, Eponine, FirstEF, and PromoterInspector shows that our method has good performance in terms of both sensitivity and specificity.

TABLE VI. Results and comparisons of five prediction systems on human Chromosome 22.

Method	TP	FP	S_e (%)[a]	S_p (%)[b]
PromoterInspector	239	274	60.8	46.6
DPF ($s=0.37$)	241	482	61.3	33.3
Eponine ($t=0.9975$)	247	248	62.8	49.9
FirstEF ($p=0.98$)	242	270	61.5	47.2
Our system ($k=5$)	246	268	62.6	47.8
($k=6$)	251	248	63.8	50.3
($k=7$)	241	232	61.3	50.9

We also evaluate the performance of our system on Release 3.1 of human chromosome 22 with length 35 Mb and 393 known genes annotated by the Chromosome 22 Gene Annotation Group at the Sanger Institute. We adopt the same evaluating criterion used by Scherf with PromoterInspector [5]: all the predictions located in the range −2000 to +500 around the 5′ extremity of a known gene are considered as a true positive promoter region (TP) and other predictions outside this range are considered as false positives (FPs). The recognition results and comparisons are summarized in Table VI. Comparisons show that the predicting result of our system is better than that of PromoterInspector with lower false positives and the result predicted by DPF has much higher false positives. Compared with Eponine and FirstEF, our system also has good performance when $k=6$ and $k=7$.

VI. CONCLUSIONS

Computational prediction of eukaryotic promoters from the nucleotide sequence is one of the most important problems in sequence analysis, but it is also a very difficult one. Although a number of algorithms have been proposed, most of them suffer from low sensitivity or too many false positives. In this paper, we show how to improve this situation by focusing on the selection of the most effective words for different functional regions in DNA sequences. A feature selection strategy is based on the KL divergence and a promoter prediction system that makes use of the position-specific information is developed. Experimental results show that our method is efficient and compares favorably with PromoterInspector, Dragon Promoter Finder, Eponine, and FirstEF. In the future, we will integrate the selected words with other features and employ machine learning techniques [27] to further improve the prediction accuracy.

ACKNOWLEDGMENTS

This work is supported by a grant from City University of Hong Kong (Project No. 9610034) and a grant from the Hong Kong Research Grant Council (Project No. CityU122005).

[1] E. S. Lander *et al.*, Nature (London) **409**, 860 (2001).
[2] J. C. Venter *et al.*, Science **291** 1304 (2001).
[3] J. W. Fickett and A. G. Hatzigeorgiou, Genome Res. **7**, 1304 (1997).
[4] D. S. Prestridge, Methods Mol. Biol. **130**, 265 (2000).
[5] M. Scherf, A. Klingenhoff, and T. Werner, J. Mol. Biol. **297**, 599 (2000).
[6] I. P. Ioshikhes and M. Q. Zhang, Nat. Genet. **26**, 61 (2000).
[7] R. V. Davuluri, I. Grosse, and M. Q. Zhang, Nat. Genet. **29**, 412 (2001).
[8] V. B. Bajic, S. H. Seah, A. Chong, S. P. T. Krishnan, J. L. Y. Koh, and V. Brusic, J. Mol. Graphics Modell. **21**, 323 (2003).
[9] T. A. Down and T. J. Hubbard, Genome Res. **12**, 458 (2002).
[10] L. Ponger and D. Mouchiroud, Bioinformatics **18**, 631 (2002).
[11] T. Werner, Briefings Bioinf. **4**, 22 (2003).
[12] D. S. Prestridge and C. Burks, Hum. Mol. Genet. **2**, 1449 (1993).
[13] G. B. Hutchinson, CABIOS, Comput. Appl. Biosci. **12**, 391 (1996).
[14] S. H. Cross, V. H. Clark, and A. P. Bird, Nucleic Acids Res. **27**, 2099 (1999).
[15] T. M. Cover and J. A. Thomas, in *Elements of Information Theory*, edited by D. L. Schilling, Wiley Series in Telecommunications (Wiley, New York, 1991).
[16] P. R. Bohjanen, Y. Liu, and M. A. Garciablanco, Nucleic Acids Res. **25**, 4481 (1997).
[17] W. D. Wang *et al.*, Proc. Natl. Acad. Sci. U.S.A. **95**, 492 (1998).
[18] J. Kim *et al.*, J. Biol. Chem. **270**, 1282 (1995).
[19] R.C. Deonier, S. Tavaré, and M.S. Waterman, *Computational Genome Analysis: An Introduction* (Springer, New York, 2005).
[20] J. M. Claveria *et al.*, Methods Enzymol. **183**, 237 (1990).
[21] S. Hannenhalli and S. Levy, Bioinformatics **17**, S90 (2001).
[22] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing* (Wiley, New York, 2002).
[23] Y. Suzuki *et al.*, EMBO Rep. **2**, 388 (2001).
[24] The exon-intron database: http://hsc.utoledo.edu/bioinfo/eid/index.html
[25] G. Pesole *et al.*, Nucleic Acids Res. **30**, 335 (2002).
[26] G. Grillo *et al.*, CABIOS, Comput. Appl. Biosci. **12**, 1 (1996).
[27] X. Xie, S. Wu, K. M. Lam, and H. Yan, Bioinformatics **22**, 2722 (2006).