Statistical Feature Selection from Chaos Game Representation for Promoter Recognition

Orawan Tinnungwattana and Chidchanok Lursinsap

Advanced Virtual and Intelligent Computing (AVIC) Center
Department of Mathematics, Chulalongkorn University
Bangkok 10330, Thailand
orakik@yahoo.com, lchidcha@chula.ac.th

Abstract. The accuracy of promoter recognition depends upon not only the appropriate representation of the promoter sequence but also the essential features of the sequence. These two important issues are addressed in this paper. Firstly, a promoter sequence is captured in form of a Chaos Game Representation (CGR). Then, based on the concept of Mahalanobis distance, a new statistical feature extraction is introduced to select a set of the most significant pixels from the CGR. The recognition is performed by a supervised neural network. This proposed technique achieved 100% accuracy when it is tested with the E.coli promoter sequences using a leave-one-out method. Our approach also outperforms other techniques.

1 Introduction

Currently, promoter prediction is one of the most challenging problems in bioinformatics studied by many researchers. A promoter is the region of DNA sequence which is usually located on the upstream of a transcription start site (TSS). If the position of promoter is known then the starting position of the coding region, which will be translated into the protein sequence, can be correctly located. In this paper we focus on recognition of E.coli promoter.

E.coli promoter contains two binding sites are the -35 and -10 hexamer boxes upstream of the TSS (position +1). The consensus sequences, i.e. the prototype sequences are composed of the most frequently occurring nucleotides at each position, for the -35 box is TTGACA and -10 box is TATAAT. However, very few of existing E.coli promoters exactly contains the two consensus sequences [1]. The region between two binding sites is called the *spacer*. Figure 1 shows the structure of promoter with two binding sites and two spacers. The distance of spacer between the -35 box and the -10 box varies from 15 to 21 bases. Another spacer between the -10 box and the transcription start site varies from 3 to 11 bases. Since the distances and positions of two binding sites are uncertain, recognizing the correct location of the promoter is rather difficult. Therefore, these features are not appropriate for promoter prediction. In this paper we do not try to find these regions.

Several algorithms have been proposed for E.coli promoter recognition, including artificial neural network (ANN) [2, 3, 4, 5, 6], hidden markov models (HMM)

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 838–845, 2006.
© Springer-Verlag Berlin Heidelberg 2006

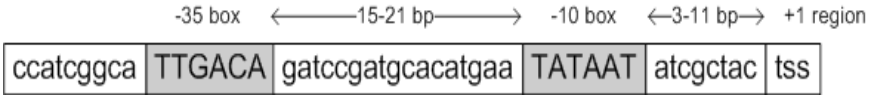


Fig. 1. The structure of E.coli promoter with two binding sites are known as the -35 and the -10 hexamer boxes and two spacers

[7], weight matrices [8], as well as graph-based induction method [9]. Many algorithms used hybrid systems which integrated neural network (NN) and other statistical decision techniques. Ma et al. [1] combined expectation maximization (EM) algorithms with NN. The EM algorithm is used for locating the -35 and -10 binding sites. Then, the features in each training promoter are chosen according to their information content and fed to an ANN for promoter recognition. Matsuyama and Kawamura [10] proposed Independent Component Analysis (ICA) algorithm including a position-dependent conversion based on symbol frequencies. Huang and Wang [11] proposed a hybrid learning system to calculate the distribution of oligo-nucleotides statistics as position weight matrices and fed as inputs to the KBANN, SVM and multilayer feed-forward neural network for discriminate promoters and non-promoters. Their result is better than other promoter prediction methods [3, 9, 11, 20, 21] with 97.2% accuracy.

The accuracy of promoter prediction is based on two factors, i.e. the representation of the given DNA sequence and the essential features of the sequence. The goal of this paper concerns both issues of above solutions that can provide a distinct classification between the promoter and non-promoter sequences. A chaos game representation (CGR) is adopted for transforming a DNA sequence having promoters and non-promoters into an image. The essential features of the CGR is selected by applying the concept of Mahalanobis distance. Then, the recognition is performed by a supervised neural network. Our method compared with [11] using same dataset, the result shown 100% accuracy. The rest of the paper is organized as follows. Section 2 presents our proposed methods that are composed of CGR for representing DNA sequences, feature selection and structure of neural network used in this problem. Section 3 shows the dataset that used in this paper, explain the evaluation methods and the results. The conclusion is in Section 4.

2 Proposed Methods

A DNA sequence are composed of four different nucleotides, adenine (A), thymine (T), cytosine (C) and guanine (G). The important features of each sequence must be extracted in order to classify it as in the promoter class of in non-promoter class. Here, the classification is achieved by using a supervised neural network. An important issue in applying neural networks to classify promoter and non-promoter sequences is how to extract the important features representing a given DNA sequence [12]. The key concept is based on the observation that all features of promoter sequences must be distinguishable from all features of

non-promoter sequences. This implies that each DNA sequence must be uniquely captured. In this paper, we used a chaos game representation (CGR) to capture a given DNA sequence. Some quadrants of the CGR are systematically selected as important features of the DNA sequence using the concept of Mahalanobis distance. The details of CGR and feature selection are discussed in the next two following subsections.

2.1 Chaos Game Representation (CGR)

Chaos Game Representation (CGR) is a method for uniquely representing DNA sequence patterns in forms of an image [13, 14, 15, 16]. The value of each pixel in the image is based on iterated function systems of fractal theory mapping a discrete sequence of symbols onto a continuous space. Initially, the image is divided into four quadrants which each of them represents one of the four possible nucleotides (A, C, G and T). CGR position is calculated by moving a pointer to a half way between the previous point and the corner of current symbol in the sequence. Let $S = (s_1 s_2 \dots s_n)$ be a DNA sequence, either having promoters or having no promoters, where $s_i \in \{A, C, G, T\}$ and l denotes the length of S. Initially, quadrants whose corners are at coordinates $(0, 0)$, $(0, 1)$, $(1, 1)$, and $(1, 0)$ are assigned to nucleotides A, C, G, and T, respectively. Each s_i is mapped to the appropriate quadrant corresponding to its nucleotide symbol. The position of s_i, denoted by p_i, is defined by the following steps.

$$\begin{aligned} p_0 &= (0.5, 0.5) \\ p_i &= p_{i-1} + 0.5(x_i - p_{i-1}), \quad i = 1, 2, \dots, n \end{aligned} \tag{1}$$

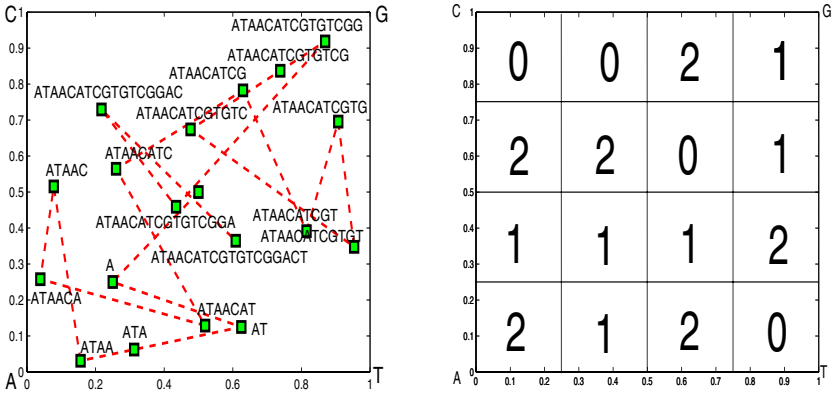
x_i is the coordinate variable of nucleotide s_i. The value of x_i is set as follows. Position p_i is denoted by a dot. The corresponding CGR image can be viewed as

s_i	x_i (coordinate)
A	$(0, 0)$
C	$(0, 1)$
G	$(1, 1)$
T	$(1, 0)$

an image of distributed dots. This image is, then, partitioned by grids into a set of square entries of equal size. The number of square entries is equal to $2^n \times 2^n$. For example, suppose a considered DNA sequence is the following. The value l for this sequence is equal to 18.

ATAACATCGTGTCGGACT

This sequence is captured by a CGR as shown in Figure 2(a) and the CGR is partitioned into a set of square entries of equal size. Suppose n is set to 2. The number of entries is equal to $2^2 \times 2^2$ or 16 and the partitioned CGR is shown un Figure 2(b).



(a) (b)

Fig. 2. The CGR of sequence GATCCGATGCACATGAA. (a) The CGR after mapping each mucleotide to its corresponding position. (b) The CGR after being partitioned into 16 equal square entries.

2.2 Feature Selection of DNA Sequences

The number of dots in each square entry is counted and used as a feature. Thus, there are $2^n \times 2^n$ features for each DNA sequence of length l. This number can make a neural training non-convergent. To overcome this problem, only relevant square entries or features must be selected from the CGR image and used as the training input. Feature selection is an effective technique in dealing with dimensionality reduction [17]. Our selection is based on the concept of Mahalanobis distance. All DNA sequences having promoters are assigned to class 1 denoted by set S_1 and those having no promoters are assigned to class 0 denoted by set S_0. All sequences in both S_1 and S_0 are represented by a set of CGR images. Let $f^{(c)}_{i,j,k}$, $c \in \{0, 1\}$, be the value of square entry (i, j) of the k^{th} CGR image for class c. A distance measure between promoter class and non-promoter class at entry (i, j) is defined as follows.

$$D_{i,j} = \frac{(m_1^2 - m_0^2)}{\delta_1^2 - \delta_0^2} \tag{2}$$

$$m_0 = \frac{\sum_{k \in S_0} f^{(0)}_{i,j,k}}{|S_0|} \tag{3}$$

$$m_1 = \frac{\sum_{k \in S_1} f^{1}_{i,j,k}}{|S_{(1)}|} \tag{4}$$

$$\delta_0 = \sqrt{\frac{\sum_{k \in S_0} (f^{(0)}_{i,j,k} - m_0)^2}{|S_0| - 1}} \tag{5}$$

$$\delta_1 = \sqrt{\frac{\sum_{k \in S_1} (f_{i,j,k}^{(1)} - m_1)^2}{|S_1| - 1}} \tag{6}$$

The high value of $D_{i,j}$ implies that the features of promoter and non-promoter at this entry can be highly distinguished. The first d square entries with highest values of $D_{i,j}$'s are selected as the essential features for training. The value of d in our experiment is set to 16

2.3 Artificial Neural Network (ANN) Classifier

From the proposed method, a feed-forward backpropagation neural network with a single hidden layer was used. There are one output unit for two target classes, class 1 for promoter and class 0 for non-promoter. Figure 3 shows the network architecture composed of input, hidden and output layers. In this paper, the ANN implementation was achieved using Stuttgart Neural Network Simulator (SNNS) 4.2 which is freely downloadable at http://www-ra.informatik.uni-tuebingen.de/SNNS/. At the beginning of each simulation, the weights were initialized with random values. The training of the network was carried out using error back-propagation with a sum square error function. In this study, there are 16 input units, 10 hidden units and one output unit.

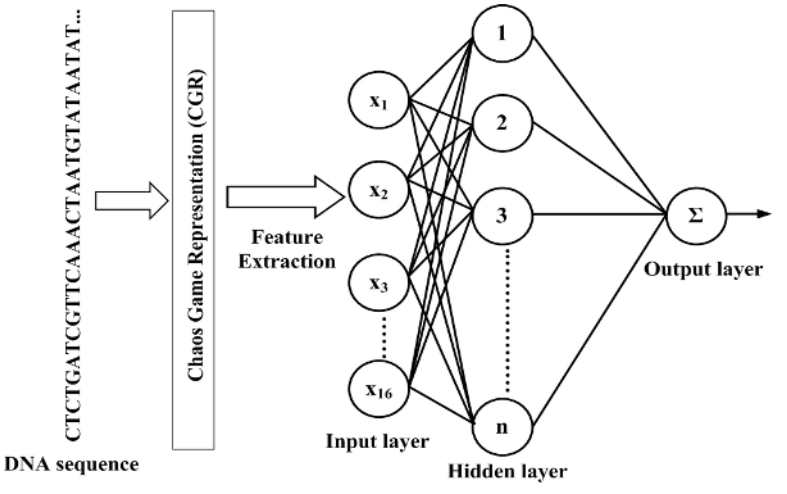


Fig. 3. The structure of neural network used in this problem

3 Experiments and Results

3.1 Data

In this paper, the data sets were taken from the UCI Machine Learning Repository [18]. It consists of 106 sequences including 53 promoter sequences and 53 non-promoter sequences. All sequences are 57 bp long. Each promoter sequence contains 49 bp upstream and 7 bp downstream of the TSS.

3.2 Performance Evaluation

Prediction performance was determined by measuring the precision, specificity (sp) and sensitivity (sn). These are defined as Eqs. (7)-(9), respectively

$$Precision = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Specificity = \frac{TP}{TP + FP} \tag{8}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{9}$$

where TP, TN, FP, FN are the number of true positives, true negatives, false positives and false negatives, respectively. A *true positive* is a promoter sequence that was also classified as a promoter sequence. A *false positive* is a non-promoter sequence that was misclassified as a promoter sequence.

3.3 Results

In this paper, our method was compared with Huang and Wang [11]. They used leave-one-out method for evaluating the performance and comparing the result with several other promoter prediction systems. We used the same data sets and the same evaluation method as theirs. Table 1 shows the number of errors compared with other methods including our method. The first number denotes the number of error patterns and the second one is the total number of testing patterns. Table 2 we compared our method with [11] by using precision, specificity and sensitivity criteria. It is clear that our method performs better than the others.

Table 1. Number of errors on several methods

Methods	Errors	Methods	Errors
ID3[19]	19/106	BP	8/106
C4.5[20]	18/106	NTSS[21]	7/106
GBI[9]	16/106	HM-layer	6/106
KNN	13/106	KBANN	4/106
O'Neill[3]	12/106	HSVM[11]	3/106
FTSS[21]	9/106	Our method	0/106

Table 2. Performances evaluation using precision, specificity and sensitivity

	HSVM[11]	Our method
Precision	97.2%	100%
Specificity	96.2%	100%
Sensitivity	98.1%	100%

4 Conclusion

In this paper, we proposed a new method for selecting all essential features in recognizing E.coli promoters in DNA sequences captured by Chaos Game Representation. These good features are used for training and classifying the given promoter sequences. Each selected feature is determined by using a distance based on the concept of Mahalanobis distance. We do not consider some well-known patterns around TSS, such as TATAAT-box and TTGACA-box, which were previously used by many researchers. The experimental results indicated that our proposed method perform better than the others'. The future research will focus on applying this technique to predict promoter in different species.

References

1. Ma, Q., et al.: DNA Sequence Classification via an Expectation Maximization Algorithm and Neural Networks: A Case Study. IEEE Trans. Systems, Man and Cybernetics, Part-C: Applications and Reviews. **31** (2001) 468–475
2. Mahadevan, I., Ghosh, I.: Analysis of E.coli promoter structures using neural networks. Nucleic Acids Research. **22(11)** (1994) 2158–2165
3. O'Neill, M.C.: Escherichia coli promoters: neural networks develop distinct descriptions in learning to search for promoters of different spacing classes. Nucleic Acids Research. **20** (1992) 3471–3477
4. Pedersen, A.G., Engelbrecht, J.: Investigations of Escherichia coli promoter sequences with artificial neural network: New signals discovered upstream of the transcriptional startpoint. In: Proceedings of ISMB95. (1995)
5. Horton, P.B., Kanehisa, M.: An assessment of neural network and statistical approaches for prediction of E.coli promoter sites. Nucleic Acids Research. **20** (1992) 4331–4338
6. Demeler, B., Zhou, G.W.: Neural network optimization of E.coli promoter prediction. Nucleic Acids Research. **19** (1991) 1593–1599
7. Pedersen, A.G., et.al.: Characterization of prokaryotic and eukaryotic promoters using hidden markov models. In: Proceedings of ISMB98. (1998)
8. Bucher, P.: Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. J.Mol.Biol. **212** (1990) 563–578
9. Matsuda, T., Motoda, H., Washio, T.: Graph based induction and its applications. Advanced Engineering Informatics. **16** (2002) 135–143
10. Matsuyama, Y., Kawamura, R.: Promoter Recognition for E.coli DNA Segments by Independent Component Analysis. In: Proceedings of CSB2004. (2004) 686–691
11. Huang, Y.F., Wang, C.M.: Integration of Knowledge-Discovery and Artificial-Intelligence Approaches for Promoter Recognition in DNA Sequences. In: Proceedings of ICITA'05. **1** (2005) 259–264
12. Hirsh, H. and Noordewier, M.: Using Background Knowledge to Improve Inductive Learning of DNA sequences. In: Proceeding of the Tenth Annual Conference on Artificial Intelligence for Applications, San Antonio, TX, (1994) 351–357
13. Jeffrey, H.J.: Chaos game representation of gene structure. Nucleic Acids Research. **18(8)** (1990) 2163–2170

14. Goldman, N.: Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. Nucleic Acids Research. **21(10)** (1993) 2487–2491
15. Almeida, J.S., et al.: Analysis of genomic sequences by Chaos Game Representation. Bioinformatics. **17(5)** (2001) 429–437
16. Deschavanne, P.J., et al.: Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences. Mol. Biol. Evol. **16(10)** (1999) 1391–1399
17. Dash, M. and Liu, H.: Consistency-based search in feature selection. Elsevier **151** (2003) 155–176
18. Blake, C.L., Merz, C.J.: http://www.ics.uci.edu/ mlearn/mlrepository.html. UCI Repository of machine learning databases. (1998)
19. Quinlan, J.R.: Induction of decision trees. Machine Learning. **1** (1986) 81–106
20. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann, Los Altos, CA. (1993)
21. Huang, J.-W., Yang, C.-B., Tseng, K.-T.: Promoter Prediction in DNA Sequences. In: Proceedings of National Computer Symposium, Workshop on Algorithm and Computation Theory, Taichung, Taiwan. (2003)