Algorithms for Promoter Prediction in DNA Sequences *

Jih-Wei Huang, Chang-Biau Yang, and Kuo-Tsung Tseng
Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
cbyang@cse.nsysu.edu.tw

Yow-Ling Shiue
Institute of Bio-Medicine Science
National Sun Yat-sen University, Kaohsiung, Taiwan

Abstract

Nowadays, the prediction of promoters has attracted many researchers' attention. Unfortunately, most previous prediction algorithms did not provide high enough sensitivity and specificity. The goal of this paper is to develop an efficient prediction algorithm that can increase the detection power (power = 1 - false negative). We present two methods that use the computer power to calculate all possible patterns which are the possible features of promoters. The first method we present FTSS (Fixed Transcriptional Start Site) uses the known TSS positions of promoter sequences to train the score file that helps us in promoter prediction. The other method is NTSS (Non-fixed TSS). The TSS positions of promoter sequences used in NTSS are assumed to be unknown, and NTSS will not take the absolute positions of TSSs into consideration. By the experimental results, our prediction has higher correct rate than other previous methods.

Key words: DNA, promoter prediction, transcriptional element, TATA-box, CpG island

1 Introduction

The biological technology becomes popular science in these years. Biologists try to investigate the secrets of life by going into gene sequences. However the gene sequence data grow too huge recently. Though some mathematicians have presented mathematical or statistical methods to discover features of gene sequences, it is still time consuming and inefficient if we study gene sequences by human power only. Thus many computer scientists get into the biological technology, and give some methods which take advantages of computer power to see into gene sequences.

The *promoter* plays an important role in DNA transcription. It is defined as the sequence in the region of the upstream of the *transcriptional start site* (TSS) and responsible for the transcription from DNA to RNA. Through the study on promoters, we can find out which DNA sequence will be transcribed into RNA, and we can even transcribe any DNA sequence which we intend to study into RNA. The related position of the promoter in a DNA sequence is illustrated in Figure 1.

A promoter is required for a DNA sequence to be transcribed. In a DNA sequence transcription, there must be a promoter in the sequence. When the promoter sequence is bound with the RNA Polymerase II enzyme, the DNA sequence can be transcribed into mRNA sequence. The central dogma of molecular biology is shown in Figure 2.

Because the gene sequence data are growing fast recently, it is important to maintain and annotate such data. However, traditional biological experiments is not enough. How to design good computer algorithms and softwares to analyze and annotate gene sequences becomes one of the most important issues today.

Since the promoter is located around the upstream of TSS in a DNA sequence, and the RNA Polymerase II is always binding in that region. The transcription starts from the end of 5' of the DNA sequence, the 5' UTR (upstream of TSS) contains promoter sites (such as TATA-box), and the 3' UTR (downstream of TSS) contains stop codon. The translation stops when the

*This research work was partially supported by the National Science Council of the Republic of China under contract NSC-92-2213-E-110-005.

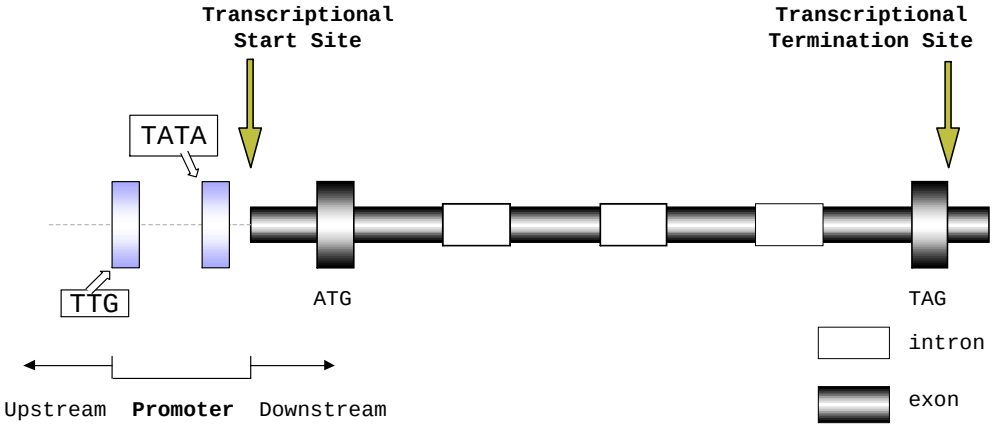

Figure 1: The promoter region in a DNA sequence.

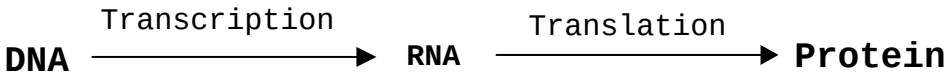

Figure 2: The central dogma of molecular biology.

stop codon is met.

However, sometimes even the upstream of TSS of a DNA sequence contains some transcriptional features, the promoter may not exist. Whether a DNA sequence transcribed or not can be verified by biological experiments, but experiments are usually time consuming and take high cost. With the promoter prediction method, we may be able to narrow down the promoter regions among massive DNA sequences. A further experiment then can be designed and tested. Therefore, much more time and cost will be saved.

In the previous studies on the promoter prediction, hidden Markov model (HMM)[15], artificial neural network (NN)[17, 7, 12, 14, 11], or some data mining[13] and weight matrix[4] methods were used. Most of them tried to find the features of the promoters. We will introduce some of these papers more detailed in Section 2.

From the above papers, we find that to find out the distinct features of promoters are thought to be useful in promoter prediction. In this paper, we do not search for the features of promoters by observation or guessing. There may exist some more implicit features in promoter regions. More features we know, to predict the promoter is more easily. We are here trying to take advantages of computers to do some operations in sequences to help us in predicting promoters. Some of promoter features will be covered after performing our operations. By comparing our prediction results with others, our methods have a higher prediction accuracy.

The dataset for our promoter prediction in this paper contains only one species, Escherichia coli (E. coli). However, the promoter regions in the homologous gene from different species may be concluded into some rules. It is believed that promoters in the homologous gene are highly similar in DNA sequences, and sometimes they even have only a little position offset across different species.

The organization of this paper is as follows. In Section 2 we present some previous studies about the promoter. We propose our methods in Sections 3 and 4. Our experimental results comparing with others and some conclusions are given in Sections 5 and 6, respectively.

2 Previous Works

In this paper, we take the *E.coli* sequences as our datasets from the UCI Machine Learning Repository [3]. The dataset contains 106 DNA sequences, including 53 sample promoter sequences and 53 non-promoter sequences. Their lengths are all 57. A DNA sequence consists of four types of nucleotides: adenine (A), guanine (G), cytosine (C) and thymine (T). The range of a promoter sequence is from -49 to +7 relative to the TSS which is defined as position +1.

Now we give some distinct features of promoter sequences that have been discovered and some methods for the promoter predictions. Some significant features

of promoter sequences which have been reported in some literatures are listed below. Some of these features are valid only in either prokaryotic or eukaryoticpromoter sequences.

- **TATA-Box and TTG-Box**

  The two identified *transcriptional elements* in promoter sequences are the -10 box and -35 box. -10 and -35 means that these elements always appear around the positions of -10 and -35 (The position of TSS is +1). The -10 box is TATA-box[17, 15, 16, 8, 2, 18] and -35 box is the pattern of TTG[17, 15].

- **CpG Islands**

  CpG islands[16, 9, 5, 1, 6] is found in eukaryotic promoter sequences. No significant CpG islands have been observed in prokaryote. So this feature can not help us in the promoter prediction with *E.coli*.

Pedersen and Engelbrecht [17] used an artificial neural network to discover new signals in the upstream of the TSS. They attempted to predict whether a given DNA sequence has a TSS or not. Their method verified some known features and they presented other possible features. For example, positions 0, -10, -22, -33, and -44 have local minima of nucleotides.

Pedersen et al.[15] took the HMM (*hidden Markov model*) to characterize the prokaryotic and eukaryotic promoters. They used promoters from two species to train the HMM and found that HMMs after training can be used to help to classify the unknown promoters in prokaryotic. Human genes could be modelled by the signals which we have already known.

The GBI (Graph-based induction) method [13] is one kind of data mining methods, brought up by Takashi et al. The method of GBI is illustrated in Figure 3. The authors took the same datasets as us from the UCI Machine Learning Repository. The original GBI is applied to minimize the size of graph by replacing the identical pattern and assigning a new node. In the GBI method for promoter prediction, they first transform the promoter sequences and non-promoter sequences into two different groups in a directed graph, then use the GBI method to extract some obvious patterns. If the pattern in the directed graph has the frequency threshold greater than 4%, then this pattern is replaced with another new node in the graph. Repeat this procedure until no pattern can be replaced. Finally extract patterns as the rules to classify the promoter sequences and non-promoter sequences.

We will compare the accuracy of promoter prediction of previous methods with ours in Section 5.

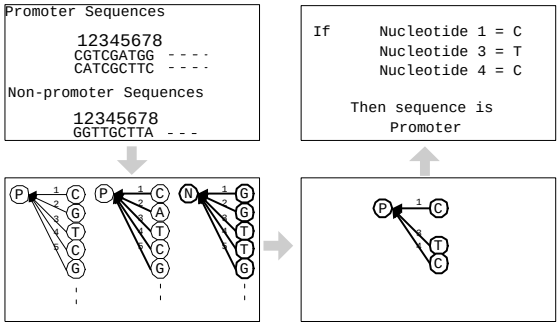

Figure 3: GBI method.

3 Method for Fixed Transcriptional Start Site

In this method, we calculate the occurrence frequency of each nucleotide in each position by summing all promoter and non-promoter sequences in our dataset, and then decide the difference value of frequencies between these two groups. Our method is described as follows:

Algorithm FTSS (Fixed Transcriptional Start Site)

Training phase:

Input: A set of DNA sequences of the same length that we have already known which are promoters and which are not, as the training dataset. TSS positions are known if they are promoter sequences.

Output: The score file which contains the frequency differences of A, G, C and T, between promoter and non-promoter sequences at each position.

Step 1: Divide the training dataset into two groups, one containing the promoter sequences and the other containing the non-promoter sequences.

Step 2: Align all sequences with the position of TSS. For each of A, G, C and T, calculate the frequency of the sequences in the same group at each position. (Non-promoter sequences have no TSSs. Since all sequences are of the same length, we still can align them.)

Step 3: Subtract each corresponding nucleotide frequency of promoter sequences from that of non-promoter sequences at each position. Then we will get the file that contains four scores for each corresponding nucleotides at all positions.

Testing phase:

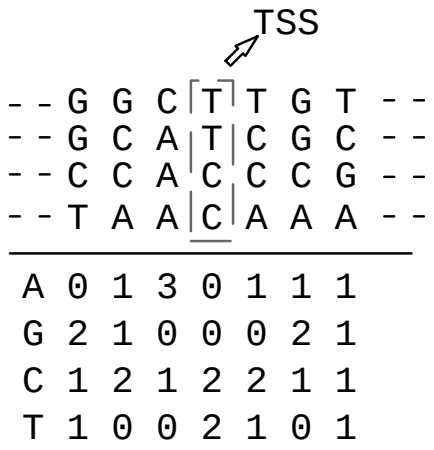

Figure 4: Example of FTSS

Input: A DNA sequence of the same length as the training dataset, and the score file which contains the frequency differences of A, G, C and T, between promoter and non-promoter sequences at each position.

Output: Answering YES if it is predicted to be a promoter; NO, if otherwise.

Step 1: Use the score file to calculate its corresponding score in each position. Sum the scores at all positions as the final score.

Step 2: If the final score is greater than zero, answer YES; NO, if otherwise.

Figure 4 shows the way we calculate the frequency of each nucleotide in the sequences of one group (either promoter sequences, or non-promoter sequences) at each position. In the position of TSS, there are T, T, C and C, so the scores of position TSS are A = 0, G = 0, C = 2 and T = 2.

Figure 5 shows the score file obtained in the training phase of FTSS. In Figure 5, the *position* means the position of each nucleotide in sequences and the *score* means the frequency difference of each nucleotide in two groups. In Figure 5, we can see some promoter features, such as TTG-box in -35, which is an obvious feature found by biology scientists.

4 Method for Nonfixed Transcriptional Start Site

We find that in some of promoter sequences we get from the UCI Machine Learning Repository, the TSS positions are slightly different from some other databases, such as PromEC[10]. In FTSS, the corresponding positions in each promoter are important. Even if the TSS positions of some promoters have only slight shifts, the frequencies of nucleotides in each position will become noise in our score file.

TSS position of a DNA sequence can be found by experiments, but it can not be sure that the TSS of a promoter we find is exactly correct. Thus we want to find another method to help us to predict promoters and we hope this method will not take the absolute position of TSS into consideration.

In the second method, we want to find out all transcriptional elements which appear in promoters and may have some influence in transcription. Our idea is to create all possible transcriptional elements and to check if these possible transcriptional elements appear in promoter sequences.

We only create all possible transcriptional elements up to four fixed nucleotides. If we create all possible transcriptional elements with more than four fixed nucleotides, the transcriptional elements file will become very large and this will lead to our prediction time too long to be accepted. Besides, we take the length of training sequences as our maximum frame length. By some promoter features we have already known, such as TATA-box or TTG-box, the promoter features may not be too long. So NTSS should be practical with shorter frame constraints. In this way, we can create all possible transcriptional elements up to six fixed nucleotides. And by some testing, we find that the frame with maximum length 15 has the better results. We take the frame length 15 in NTSS and this length is shorter than the testing sequence length.

We define sotme symbols first. Σ denotes the set of alphabets in our sequences, which is {A, G, C, T} here. $\sigma_1\#\sigma_2$ represents the transcriptional element type, where σ_1 and σ_2 are fixed alphabets in Σ, # represents the number of arbitrary alphabets between σ_1 and σ_2, $\# \in N \bigcup \{0\}$. For example, σ_1 = A, σ_2 = T, and # = 1, then $\underline{\mathbf{A}}1\underline{\mathbf{T}}$ = {$\underline{\mathbf{A}}$A$\underline{\mathbf{T}}$, $\underline{\mathbf{A}}$G$\underline{\mathbf{T}}$, $\underline{\mathbf{A}}$C$\underline{\mathbf{T}}$, $\underline{\mathbf{A}}$T$\underline{\mathbf{T}}$}. As another example, σ_1 = G, σ_2 = C, and # = 2, then $\underline{\mathbf{G}}$AC$\underline{\mathbf{C}}$, $\underline{\mathbf{G}}$TT$\underline{\mathbf{C}}$, $\underline{\mathbf{G}}$GA$\underline{\mathbf{C}}$ $\in$ $\underline{\mathbf{G}}$2$\underline{\mathbf{C}}$ and the size of G2C is $|\underline{\mathbf{G}}2\underline{\mathbf{C}}| = |\Sigma|^2 = 4^2 = 16$. Our second method is shown as follows.

Algorithm NTSS (Nonfixed Transcriptional Start Site)

Training phase:

Input: A set of DNA sequences of the same length that we have already known which are promoters and which are not, as the training dataset.

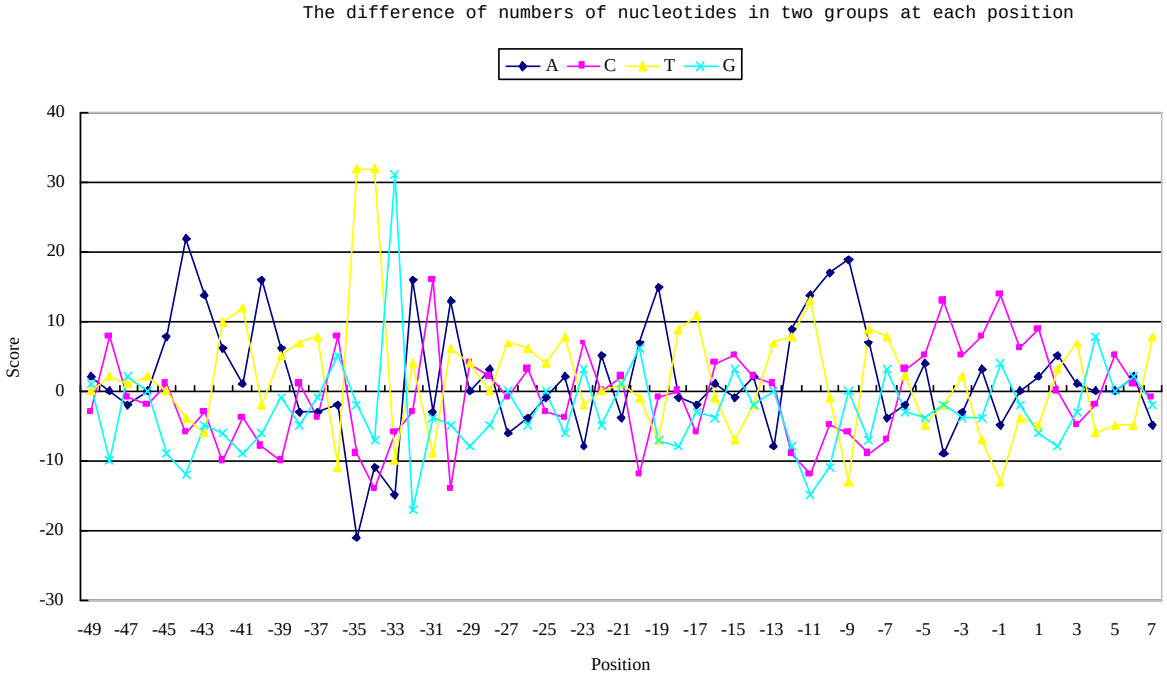

Figure 5: The distribution of each nucleotide.

Output: A threshold and the score file which contains the numbers of occurrences of each transcriptional element in the sequences.

Step 1: Create $\sigma, \sigma\#\sigma, \sigma\#\sigma\#\sigma, \sigma\#\sigma\#\sigma\#\sigma, \sigma\#\sigma\#\sigma\#\sigma\#\sigma, \sigma\#\sigma\#\sigma\#\sigma\#\sigma\#\sigma$, $\sigma \in \sum$, all possible kinds of transcriptional element types with at most 6 fixed nucleotides. The length of each transcriptional element is limited to at most 15.

Step 2: Calculate the score of each possible transcriptional element. If one transcriptional element have ever appeared in one of sample promoter sequence, no matter how many times it appears, we add one point to its score. Note the initial scores of all possible transcriptional elements are zero.

Step 3: Store the corresponding score of each possible transcriptional element in the transcriptional element file.

Step 4: Take all sequences as our input testing sequences (including promoters and non-promoters). By the transcriptional element file with scores, we calculate each sequence a gain score in the testing. If the sequence contains some transcriptional elements, we add the scores of those transcriptional elements to the gain score of the sequence. Each sequence has its own gain score that the initial score is zero.

Step 5: Sort the gain scores of all sequences and find one proper score as the threshold. The threshold is the gain score which is less than most promoter sequences's gain scores and greater than most non-promoter sequences's gain scores.

Testing phase :

Input : A DNA sequence of the same length as the training dataset.

The score file which contains the number of occurrences of each transcriptional element in sequences.

The threshold score obtained in the training phase.

Output : Answering YES if it is predicted to be a promoter; NO, if otherwise.

Step 1: Using the score file to calculate its corresponding score in each position. If the input sequence contains one transcriptional element, we add the score of that transcriptional element.

Step 2: If the score is greater than or equal to the threshold score, answer YES; NO, if otherwise.

```
Possible TE      Score
A 0 A 0 A 0 A      19
A 0 A 0 A 0 C      24
A 0 A 0 A 0 T      11
A 0 A 0 A 0 G      10
A 0 A 0 A 1 A      31
A 0 A 0 A 1 C      11
A 0 A 0 A 1 T      19
        ⋮
```

Figure 6: The score file of NTSS.

The result of the possible transcriptional elements with scores in NTSS is shown in Figure 6. In Figure 6, **A**0**A**0**A**0**A** = {**AAAA**} and this transcriptional element appears in 19 sequences of the training dataset. **A**0**A**0**A**1**C** = {**AAA**A**C**, **AAA**G**C**, **AAA**C**C**, **AAA**T**C**} and they appear in 11 sequences of the training dataset.

The result will also be discussed in Section 5.

5 Experimental Results and Accuracy Analysis

In our experiments, the error rate of FTSS is 9/106. And the prediction results of NTSS are shown in Table 1. It is clear that the prediction accuracy of FTSS may be better than NTSS. Table 2 shows the detailed FP (false positive), miscarriaging a non-promoter sequence as a promoter sequence, and FN (false negative), miscarriaging a promoter sequence as a non-promoter sequence, and error rate of all our methods in the promoter prediction.

In NTSS, we can not find the appropriate threshold which divides the promoter and non-promoter sequences by the type of one or two fixed nucleotides. Besides, we find that the result of four fixed nucleotides type is better than three. The longer transcriptional element type may get the better result than three and four.

In the same type of fixed nucleotides, NTSS without frame constraints has the better result, but in NTSS with frame constraints, the number of nucleotides can grow up to six. This is a trade off. We find that it is worth for us to make the frame constraints in NTSS. In our experimental results, NTSS with frame length 15 has the best accuracy. We compare our results with the results of ID3[19], C4.5[20] and GBI[13]. All of these methods are use the same dataset as ours. Figure 7 shows the promoter prediction accuracy by comparing

Table 2: The result of our methods. The unit of each rate is %.

	FP rate	FN rate	Error rate
FTSS	0	16.98	8.49
NTSS (σ#σ#σ)	15.09	28.31	21.70
NTSS (σ#σ#σ#σ)	16.98	11.32	14.15
NTSS (σ#σ#σ#σ#σ)	13.21	3.77	8.49
NTSS (σ#σ#σ#σ#σ#σ)	7.55	5.66	6.60

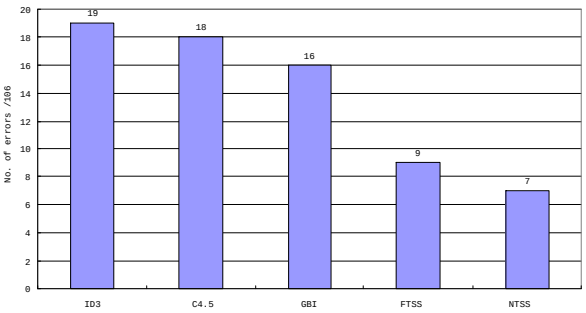

Figure 7: The promoter prediction accuracy comparison of our methods with others. NTSS here is with frame of length 15.

our methods and these methods. All these results are obtained by only inside test, which means that the testing sequences are the same as the training sequences. We can get only the inside test results of these previous methods. Latter we will build a testing model which allows our methods to do outside test.

Clearly, both FTSS and NTSS have prediction accuracy improvement comparing to other previous methods.

In an outside test, the testing sequence is not contained in the training set. Our outside testing model is as follows:

Step 1: Randomly select 6 sequences in the dataset as our testing sequences.

Step 2: Use the remaining 100 sequences (excluding the above 6 selected sequences) as the training sequences to get the score file with FTSS (or NTSS).

Step 3: Test the 6 selected sequences with the score file we get from the training sequences.

Table 1: The promoter prediction accuracy using NTSS. NTSS in this table is with frame length 15.

Method	NTSS (σ#σ#σ)	NTSS (σ#σ#σ#σ)	NTSS (σ#σ#σ#σ#σ)	NTSS (σ#σ#σ#σ#σ#σ)
No. of error /106	23	15	9	7

Table 3: The experimental of outside tests in results FTSS. P means promoter sequences and NP meas non-promoter sequences.

	Inside Testing		Outside Testing	
Sequence	P	NP	P	NP
Total No.	1000	1000	60	60
Error No.	3	178	3	25
Error rate(%)	FN=0.30	FP=17.80	FN=5.00	FP=41.67

Table 4: The experimental of outside tests in results NTSS. P means promoter sequences and NP meas non-promoter sequences.

	Inside Testing		Outside Testing	
Sequence	P	NP	P	NP
Total No.	1000	1000	60	60
Error No.	99	35	6	3
Error rate(%)	FN=9.90	FP=3.50	FN=10.00	FP=5.00

In FTSS, we repeat the above procedure twenty times. Totally we select 60 promoter sequences and 60 non-promoter sequences for testing. The testing results are shown in Table 3. We find that the result for testing sequences selected from training data is the same as the result in Table 2. The prediction error rates of testing data are a little higher, but we can see that the error rates still gets low FN rate.

For the outside test in NTSS, we also randomly select 60 promoters and 60 non-promoter sequences for testing totally in 20 experiments. The result is shown in Table 4. We find that in outside testing, the prediction accuracy of promoter is a little higher and the non-promoter is a little lower. We think the little higher FN and lower FP than prior is because the prior result is just one kinds of the case. In our testing model, NTSS runs each random selected case more times and the result should be more correct. In this result, we think that NTSS is useful not only for our training set but also for the testing set.

In our result we can find that NTSS has the better prediction accuracy in our whole data (promoter and non-promoter sequences). If we consider the FN rate only, the result of FTSS is better than NTSS. So when we want to predict promoter, we can take both the results of FTSS and NTSS into consideration and these results can help us to eliminate many sequences which should not be promoter sequences.

6 Conclusion

In this paper, we propose new methods for solving the promoter prediction problem. The experimental results indicate our methods perform better than some other previous prediction methods with respect to the recognition rate. Our main idea is to find all possible patterns (transcriptional elements) which are the possible features of promoters. We do not consider some well-known obvious features of promoters, such as TATA-box, which were discovered by researchers previously.

Though a set of DNA sequences of the same length is requested in training phase of our FTSS, FTSS in fact can be applied if the given DNA sequences are with variable lengths. The only condition is that the position of TSS of each sequence is known. If the set of DNA sequences with variable lengths is given, the error rate of FTSS may increase.

In NTSS, the frame of each possible transcriptional element is of length at most 15 and it contains at most six fixed nucleotides. We find that if the fixed nucleotides contained in a frame is greater than six, the required computing time becomes very much, but the accuracy does not increase.

The experiments we do in this paper are only on one species, *E.coli*, which is a prokaryotic cell. In fact, our methods can be applied to any single species provided that some promoter sequences of the species have been found.

Our result may be helpful for finding the binding sites in the promoter. A *binding site* is a segment of a promoter at which a *transcriptional factor* (a protein) can bind to the promoter. We guess that a frame with high score has a high potential to be a binding site.

The promoter sequences in different species have some distinct features. For example, the CpG islands look obvious in eukaryotic promoter regions, but these can not be applied in prokaryotic promoter regions. We should analyze the features of promoter sequences

for each organisms. With the collection of the features of promoters in different species, we may find out the relationship between different species.

References

[1] F. Antequera and A. Bird, "Number of CpG islands and genes in human and mouse," *Proc. Natl. Acad. Sci., USA*, Vol. 90, pp. 11995–11999, 1993.

[2] S. Audic and J. M. Claverie, "Visualizing the competitive recognition of TATA-boxes in vertebrate promoters," *Trends Genet*, Vol. 14, pp. 10–11, 1998.

[3] C. Blake and C. Merz, "http://www.ics.uci.edu/∼mlearn/mlrepository.html," *UCI Repository of machine learning databases*, 1998.

[4] P. Bucher, "Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences," *J. Mol.Biol.*, Vol. 212, pp. 563–578, 1990.

[5] J. M. Craig and W. A. Bickmore, "The distribution of CpG islands in mammalian chromosomes," *Nature Genetics*, Vol. 7, pp. 376–382, 1994.

[6] S. H. Cross and A. Bird, "CpG islands and genes," *Current Opinion in Genetics & Development*, Vol. 5, pp. 309–314, 1995.

[7] B. Demeler and G. W. Zhou, "Neural network optimization for E. coli promoter prediction," *Nucleic Acids Research*, Vol. 19, pp. 1593–1599, 1991.

[8] G. Gill and R. Tjian, "Eukaryotic coactivators associated with the TATA box binding protein," *Current Opinion in Genetics & Development*, Vol. 2, pp. 236–242, 1992.

[9] S. Hannenhalli and S. Levy, "Promoter prediction in the human genome," *Bioinformatics*, Vol. 17, pp. 90–96, 2001.

[10] R. Hershberg, G. Bejerano, A. Santos-Zavaleta, and H. Margalit, "Promec: An updated database of Escherichia coli mRNA promoters with experimentally identified transcriptional start sites," *Nucleic Acids Research*, Vol. 29, p. 277, 2001.

[11] P. B. Horton and M. Kanehisa, "An assessment of neural network and statistical approaches for prediction of E. coli promoter sites," *Nucleic Acids Research*, Vol. 20, pp. 4331–4338, 1992.

[12] I. Mahadevan and I. Ghosh, "Analysis of E. coli promoter structures using neural networks," *Nucleic Acids Research*, Vol. 22, pp. 2158–2165, 1994.

[13] T. Matsuda, H. Motoda, and T. Washio, "Graph-based induction and its applications," *Advanced Engineering Informatics*, Vol. 16, pp. 135–143, 2002.

[14] M. C. O'Neill, "Escherichia coli promoters: neural networks develop distinct descriptions in learning to search for promoters of different spacing classes," *Nucleic Acids Research*, Vol. 20, pp. 3471–3477, 1992.

[15] A. G. Pedersen, P. Baldi, S. Brunak, and Y. Chauvin, "Characterization of prokaryotic and eukaryotic promoters using hidden Markov models," *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB98)*, 1998.

[16] A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak, "The biology of eukaryotic promoter prediction - a review," *ComputerChemistry*, Vol. 23, pp. 191–207, 1999.

[17] A. G. Pedersen and J. Engelbrecht, "Investigations of Escherichia coli promoter sequences with artifical neural network: New signals discovered upstream of the transcriptional startpoint," *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB95)*.

[18] D. S. Prestridge, "Predicting pol II promoter sequences using transcription factor binding sites," *J. Mol.Biol.*, 1995.

[19] J. Quinlan, "Induction of decision trees," *Machine Learning*, Vol. 1, pp. 81–106, 1986.

[20] J. Quinlan, *C4.5: programs for machine learning*. Los Altos: CA:Morgan(Kaufmann), 1993.