Promoter Prediction in DNA Sequences

A Thesis

Submitted to the Faculty

of

Department of Computer Science and Engineering

National Sun Yat-sen University

by

Jih-Wei Huang

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

June 20, 2003

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

LIST OF TABLES

Table Page

ABSTRACT

Recently, 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 thesis is to develop an efficient prediction algorithm that can increase the detection power (power = 1 - false negative). We do not try to find more distinct features in promoters one by one, such as transcriptional elements. Our main idea is to use the computer power to calculate all possible patterns which are the possible features of promoters. Accordingly, we shall define some scoring methods for training a given set of sequences, which involve promoter sequences and non-promoter sequences. Then, we can obtain a threshold value for determining whether a testing sequence is a promoter or not. By the experimental results, our prediction has higher correct rate than other previous methods.

CHAPTER 1

Introduction

Promoter is a fragment of DNA sequence that is 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.

In the previous studies on the promoter predictions, hidden Markov model (HMM) [17], artificial neural network (NN) [7,11,14,16,19], or some data mining [15] and weight matrix [4] methods were used. Most of them tried to find the features of the promoters.

Anders Gorm Pedersen [17, 19] studied the promoter prediction problem for many years and used the hidden Markov model [17] and the neural network [19] to find out some distinct features of promoters. Takashi Matsuda [15] predicted the promoter using the data mining method, which is a graph-based induction method, and the accuracy of their methods is about 84.91% in their testing data. Prestridge [20] proposed a prediction program, PROMOTER SCAN, divides the promoters into two parts, TATA-box and non TATA-box and gives a testing sequence two scores. One is the score which we get from the weighted matrix and if a sequence contains TATA-box, it gets a higher score. Another score is decided by whether the sequence contains some obvious transcriptional elements. The program takes the higher one as the sequence's score and judges whether the sequence is promoter or not. From

these papers, we find that to find out the distinct features of promoters are thought to be useful in promoter prediction. In this thesis, we do not search for the features of promoters by observation. There may exist some more implicit features in promoter regions. If we know more features, 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. We propose prediction methods and by comparing our prediction results with others, we have a higher prediction accuracy.

The dataset for our promoter prediction in this thesis 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 thesis is as follows. In Chapter 2 we talk about the definition and some significant features of the promoter, then we discuss some previous studies about the promoter. We present our methods in the promoter prediction and material in Chapter 3. We compare our experimental results with others and give some conclusions in Chapter 4 and 5.

CHAPTER 2

Preliminaries

In this chapter, we shall first explain what the promoter is and biological function role of the promoter in more detail, then present some distinct features of promoter sequences that have been known and some methods for the promoter predictions.

2.1 Definition of the Promoter

A gene can be roughly divided into five parts : promoter, 5' UTR (untranslated region), exons, introns, 3'UTR, and polyadenylation site [25]. Exons can be translated into proteins after they are transcribed into RNA. The *promoter* plays an important role in DNA transcription. The promoter is defined as the sequence in the region of the upstream of the *transcriptionalstartsite* (TSS). The related position of the promoter in a DNA sequence is illustrated in Figure 2.1.

A promoter is required for a DNA sequence to be transcribed. In a DNA sequence transcription, there must be a promoter in the sequence. Binding with different kinds of RNA polymerases in the promoter regions, the DNA sequence will be transcribed into various RNA sequences, such as tRNA, rRNA, and mRNA. mRNA is important for protein production. After translation, an mRNA sequence will be translated into a protein (an amino acid sequence). Thus, if we know which

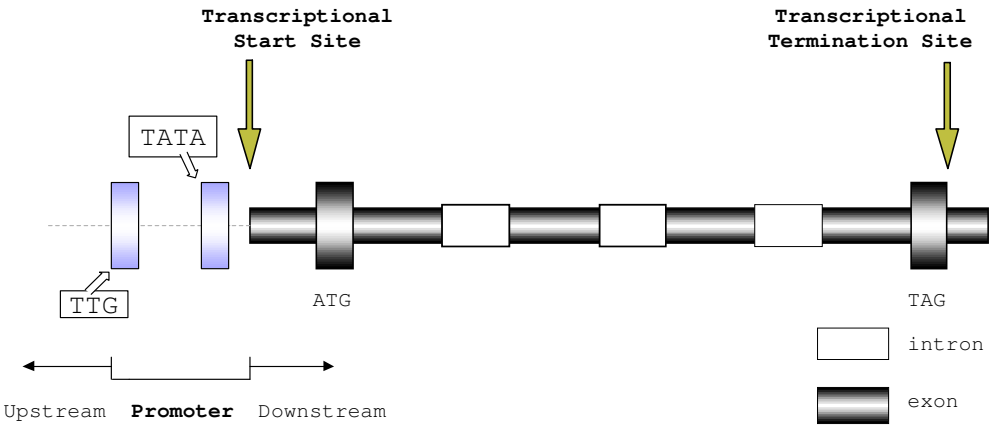

Figure 2.1 The promoter region in a DNA sequence.

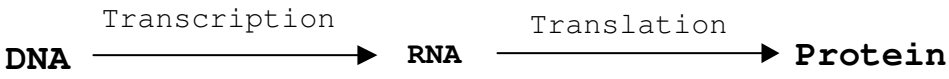

Figure 2.2 The central dogma of molecular biology.

DNA sequence can be transcribed and translated into an amino acid sequence, it will be useful for biologists to understand the gene regulation and expression. When the promoter sequence is bound with the RNA Polymerase II enzyme, which we want to predict, the DNA sequence can be transcribed into mRNA sequence. The central dogma of molecular biology is shown in Figure 2.2.

2.2 Significance of the Promoter Prediction

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.

If we know the position of the promoter (first exon), we will know the position of the first exon (promoter). With knowing the position of the first exon, we get the starting position of the coding region, which will be translated into the protein sequence. How to find the promoter of a DNA sequence is one of the essential points in the work of gene sequence analysis.

If we know which segment of a DNA sequence is the promoter sequence, we can use the promoter sequence to regulate the speed of translation from DNA into a protein. Furthermore, the promoter is also useful in genetically modified foods [23]. With the similar method, we can also have the protein which causes disease grow more slowly, even destroy it. Through the recombination of DNA sequences with the promoter in transgenic technology, animals may get the gain-of-function or loss-of-function [24].

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 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. By 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.

2.3 The Features of Promoter Sequences

Here we will show some significant features of promoter sequences which have been reported in some literatures. Some of these features are valid only in either prokaryotic or eukaryotic promoter sequences.

2.3.1 TATA-Box and TTG-Box

The *transcriptional elements*, which have high appearance frequency in the promoter region, are guessed to play an important role in transcriptions to find transcriptional elements from the promoter sequences is the basic concept of finding the features of promoters. Experimentally, 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 [2,8,17–20] and -35 box is the pattern of TTG [17,19].

2.3.2 CpG Islands

CpG islands [1, 5, 6, 9, 18] is another feature of promoter sequences. CpG islands means that in a sequence, the G nucleotide usually appears following the C nucleotide. The p in CpG denotes the phosphodiester linkage of the DNA sequence. In fact, a DNA sequence is always methylated around the TSS, so CpG islands have high appearance frequency in the promoter in all DNA sequences. This feature 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*.

2.4 Artificial Neural Network

Pedersen and Engelbrecht [19] 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. The dataset they took is the promoter sequences of *E.coli* from the complication by Lisser and Margalit [13].

In their study, the neural network has three layers of neurons. Input data are the DNA sequences in binary representation. They represent each nucleotide as 4 binary digits: A = 0001, C = 0010, G = 0100, T = 1000. The output layer shows whether the nucleotide at a given position is a transcription start site or not, the range of the output value is from 0 to 1. If the given position is TSS, the output value is 1.0. If the position is not TSS, the output value is 0.0. If the output value is greater than 0.5, the position is guessed as a TSS. Otherwise, the position is not a TSS.

They use two types of neural networks. Type I contains only a single input window whose size can be varied. Type II input window can cover 65 nucleotides and has 7 nucleotide holes. The positions of the 7 nucleotide holes can be varied during the training period. The neural network of type I is shown in Figure 2.3 and type II in Figure 2.4.

The example shown in Figure 2.3 has input window with five nucleotides. In the neural network of type I, the range of input window is varied from 1 to 51 nucleotides. In Figure 2.4, the input window of neural network can cover 65 nucleotides in which only 7 nucleotide holes are used at one time. The positions of the nucleotide holes can be varied. The encoding scheme is the same as type I, so the encoded results are not shown in the figure.

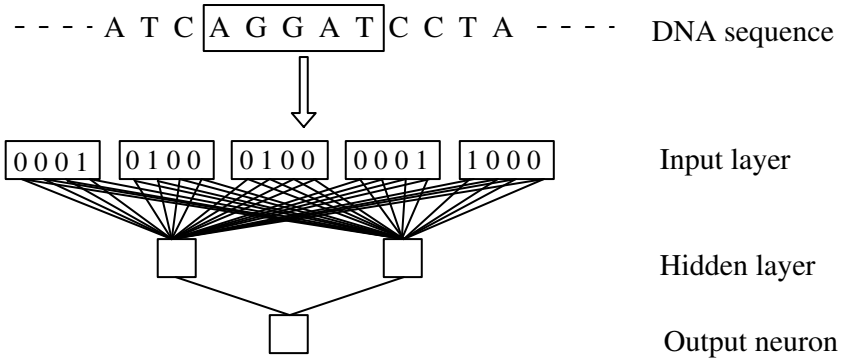

Figure 2.3 Type I neural network.

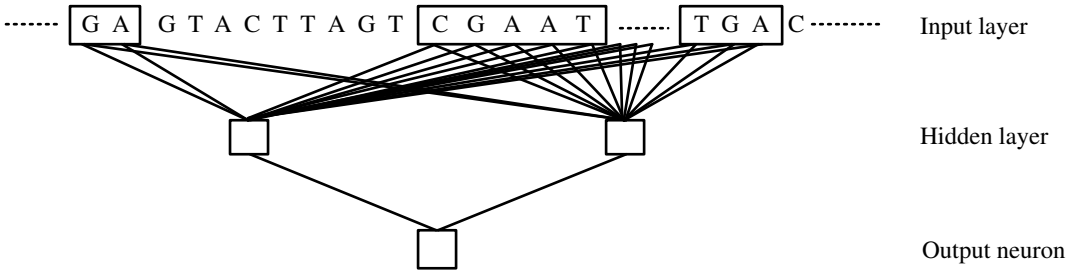

Figure 2.4 Type II neural network.

Ander also takes Kullback Leibler's distance [12] (information measure) as the relative entropy between TSS with other positions. The formula of Kullback Leibler's distance is given as follows

$$D(i) = \sum_{N} P_i^N \log \frac{P_i^N}{Q_i^N}$$

P_i^N and Q_i^N denote the probabilities of occurrence for one of particular nucleotide N at position i, $N \in \{A, C, T, G\}$, where P_i^N is taken by the relative to TSS and Q_i^N is taken the relative to all other positions in the promoter sequences. The value of $D(i)$ has range from 0 to ∞. If $D(i)$ is large, then it means the frequency of a nucleotide, A, C, T, or G, is different from the average at position i. If it is zero, it means that the two distributions are all equal at position i.

The result of the neural network type I by Kullback Leibler distances shows that around 0, -10, and -35 have the peaks. This indicates that the frequencies of nucleotides in these three positions are not equal to the average, which are the features of promoters we knew before. Nucleotides C and A at around position 0 have higher frequency than other positions. Position -10 has TATA-box and position -35 has TTG-box. In the result of neural network type II, it can be found that positions 0, -10, -22, -33, and -44 have local minima of nucleotides. These new signals in analysis are spaced evenly along the promoter region with a period of 10 or 11 base pairs.

2.5 Hidden Markov Model

Pedersen et al. [17] took the HMM (*hidden Markov model*) to characterize the prokaryotic and eukaryotic promoters. They use promoters from two species to train the HMM. One is for prokaryotic promoters, using *E.coli* promoters which is presented by Lisser and Margalit [13]; the other is for eukaryotic promoters, using human promoters.

The HMM contains a set $\mathbf{S}$ of states, an alphabet set A of m symbols, a probability transition matrix $\mathbf{T} = (t_{ij})$, and a probability emission matrix $\mathbf{E} = (e_{iX})$. The state system changes state randomly but the emitting symbols change from the alphabet. In the system, the probability from state i to state j is t_{ij}. The probability e_{iX} is for state i moving to emitting symbol X.

In promoter sequences, the alphabet size m is defined as 4 for four nucleotides. There are five kinds of states in HMM, which are $start, end, main, deletion$, and $insertion$ states. For example S = $start, m_1, \ldots, m_N, i_1, \ldots, i_{N+1}, d_1, \ldots, d_N, end$.

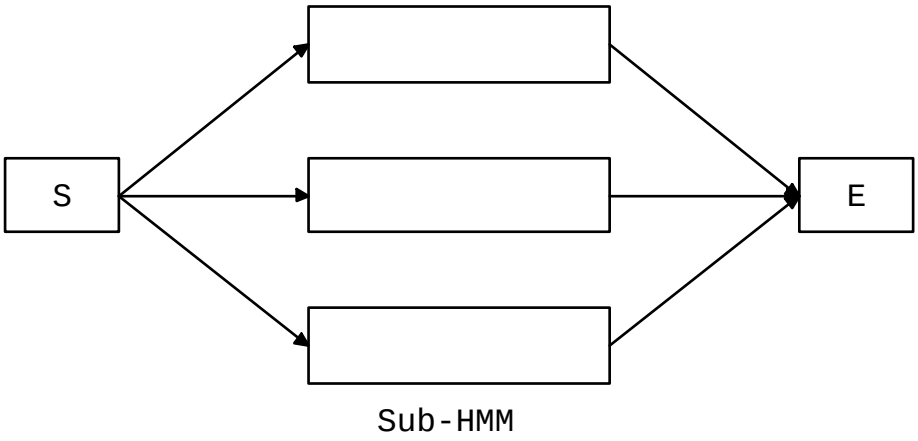

Figure 2.5 Example of the super-HMM.

The symbols of *m, d, i* stand for $main, delete$, and *insert* states. The average length of the model is N. They use the *supper-HMM*, consisting of several sub-HMMs in parallel, to classify the subclass of promoters. Each sub-HMM represents one sub-class. The Viterbi path of training sequences will be computed first if it is represented for the super-HMM. Each Viterbi path goes through only one of the sub-HMM and updates the parameters of this HMM during training. By this way, they can increase the likelihood of the corresponding sequence. The model of the super-HMM is shown in Figure 2.5. They also use the Kullback Leibler distance [12] for measurement.

They found that HMMs after training can be used to help to classify the unknown promoters in prokaryotic. They divide *E.coli* promoters into three classes, $\alpha 54, \alpha 70$, and the rest. Actually in their result, the features of promoter sequences are not so clear after measurement, but in emission probability of the main states the features of promoter sequences are obvious, such as TTG-box or TATA-box. Human genes after HMMs could be modelled by the signals which we have already known. In the result, they present the TATA-box around -10.

2.6 Graph-based Induction Method

Here we introduce the method of GBI(Graph-based induction) [15]. The GBI method is one kind of data mining methods, brought up by Takashi et al. The method of GBI is illustrated in Figure 2.6. The authors took the same datasets as that from the UCI Machine Learning Repository. The original GBI is applied to minimize the size of graph by replacing the identical pattern and assign a new node. The GBI method for promoter prediction is as follow:

Step 1: Transform the promoter sequences and non-promoter sequences into two different groups in a directed graph.

Step 2: Use the GBI method to extract some obvious patterns. If the pattern in the directed graph has the frequency threshold greater than 4%, then replace this pattern with another new node in the graph. Repeat this step until no pattern can be replaced.

Step 3: Extract patterns as the rules to classify the promoter sequences and non-promoter sequences.

We will compare the accuracy of promoter prediction of their method and ours in Chapter 4.

2.7 Predicting Pol II Promoter Sequences Using Transcription Factor Binding Sites

Prestridge released one promoter prediction program, PROMOTER SCAN [20]. The dataset took 167 primate Pol II promoter sequences from Eukaryotic

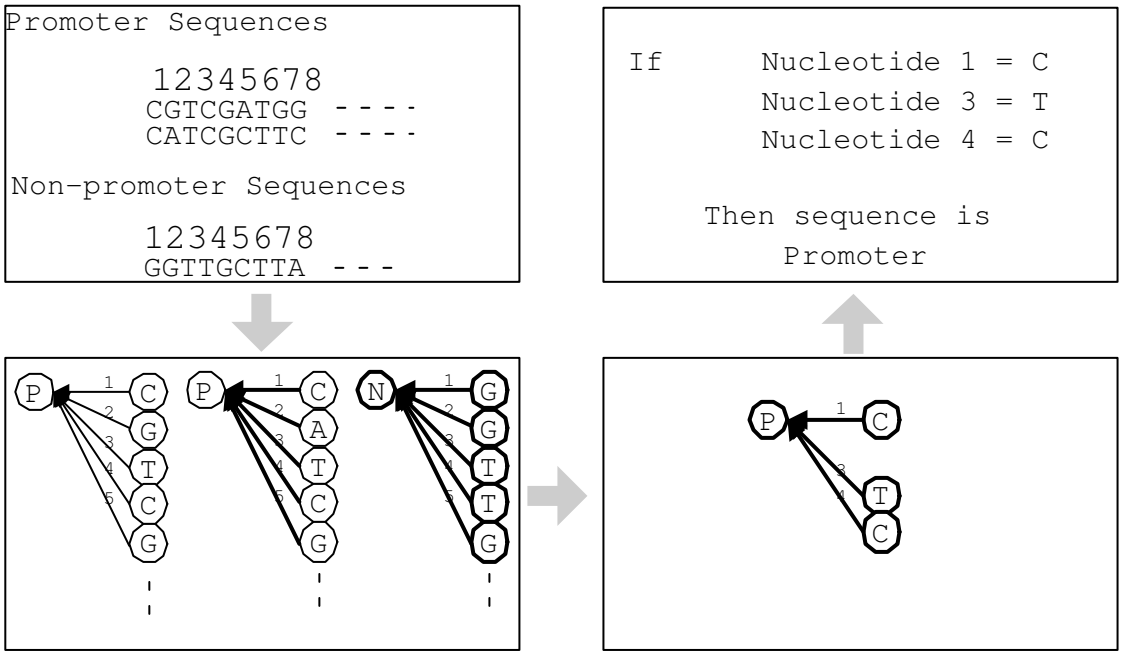

Figure 2.6 GBI method.

Promoter Database (EPD), 999 primate non-promoter sequences from GenBank (NCBI, USA), and 1438 unique mammalian transcription factor binding site Ghosh Transcription Factor Database (TFD).

The PROMOTER SCAN utilizes the Bucher TATA weight matrix [4] as weighted matrix for scoring a TATA box. The sequence with higher score will be adopted from the promoter recognition profile or from the Bucher TATA weight matrix. The flow chart of PROMOTER SCAN is shown in Figure 2.7.

Here we shortly explain the operations of the flow chart. The input of PROMOTER SCAN include the promoter recognition profile, the Bucher TATA weight matrix, and the primary DNA sequence signal file of the unknown sequences. The promoter recognition profile contains two parts. One is the recognition of transcription factor binding site elements defined in the TFD, and the other is the density of

transcriptional elements, which is the ratio of the transcriptional elements in promoter sequences. Putative transcriptional elements in promoter and non-promoter sequences are recognized by modified version of the SIGNAL SCAN program. This transcriptional element recognition step produces a 'signal file' containing a list of putative transcriptional elements for each promoter sequence and each non-promoter sequence.

The formula of SIGNAL SCAN : $D_E = \frac{\sum_{i=1}^{N} E_{il}}{n \cdot l}$. D_E denotes the number of occurrences of the transcriptional elements per bp, n denotes the total number of sequences, l denotes the length of the sequence window (the number of bp 5 to the TSS in promoter sequence), and E_{ij} denotes the total number of that transcriptional elements found in length l of sequence i.

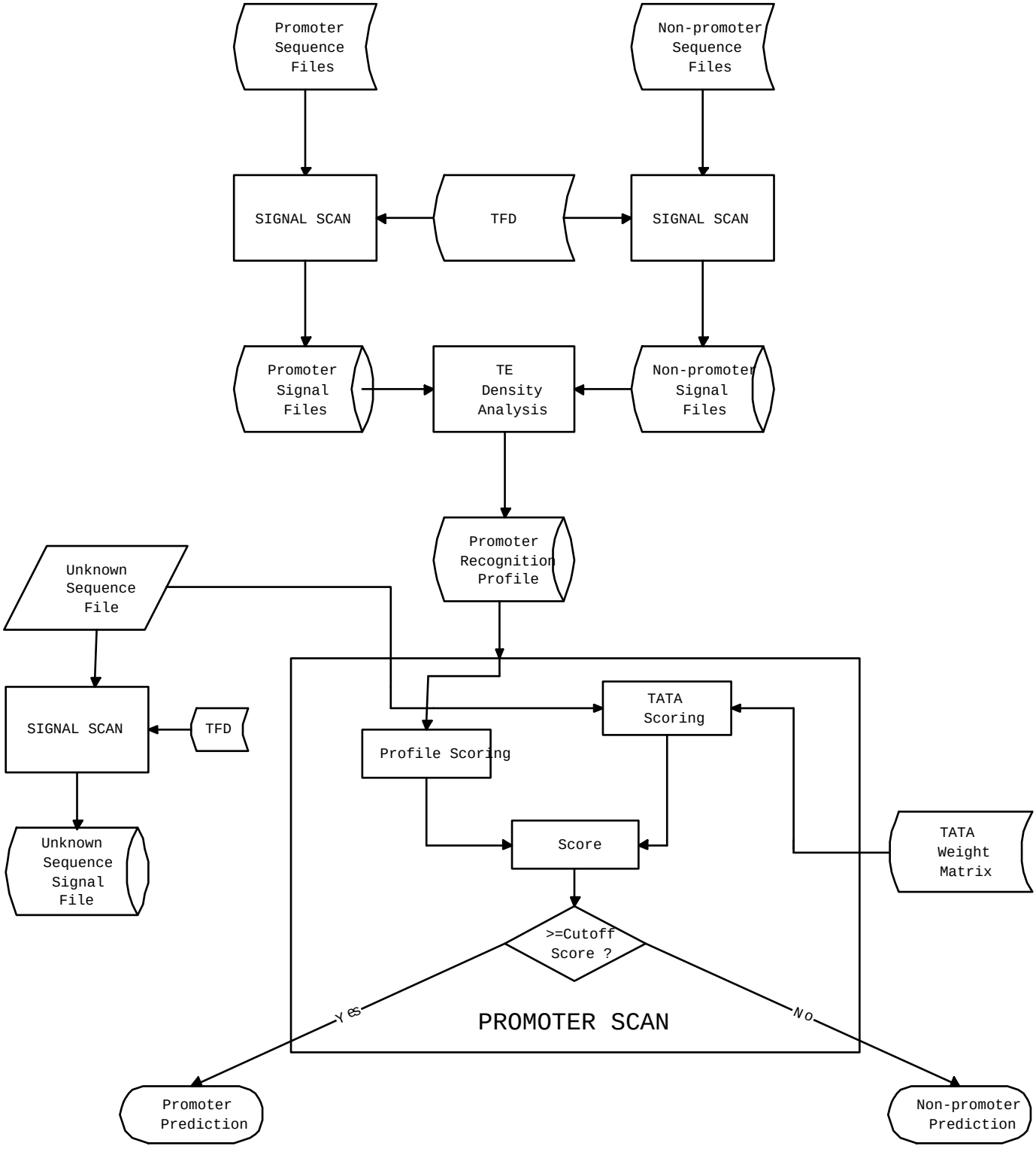

Figure 2.7 The flow chart of PROMOTER SCAN.

CHAPTER 3

Material and Methods

In this chapter, we shall propose our prediction methods, and explain how we get our datasets. Our first training method uses the frequencies of A, G, C and T at each position in sequences to obtain a score function for prediction. However, the TSS (transcriptional start site) positions in training sequences may not be the same, thus the method may not work well. In our second method, we assume that TSS positions are not fixed. By the experimental results, we find that our second method has better performance that our first method. We will compare our results with the graph-based induction in Chapter 4.

3.1 Datasets

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.

Most promoter prediction programs try to use many kinds of statistical methods to find significant features of promoters, and then check if these features are

useful to classify the difference between the promoter and the non-promoter sequences. With this ways, we may easily miss some implicit features. It is believed that the accumulation of many these implicit features will have quite improvement in the accuracy of promoter prediction program. So in our method, we do not try to find new obvious features, but try to find a possible scoring method by preforming some operations in sample sequences.

3.2 Method 1

In this method, we do not inspect the feature of promoters carefully. 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 Method 1

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.

Output: The score file which contains the 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 point of TSS. For each of A, G, C and T, calculate the frequency of the sequences in the same group at each position.

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 to each corresponding nucleotides at all positions.

Testing phase:

Input: A DNA sequence of the same length as the training dataset, and the score file which contains the 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 promoter; NO, if otherwise.

Step 1: Using 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 3.1 shows the way we add 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 3.2 shows the score file obtained in the training phase of our method 1. In Figure 3.2, the *position* means the position of each nucleotide in sequences and the *score* means the difference of the frequencies of each nucleotide in two groups. In Figure 3.2, we can see some promoter features, such as TTG-box in -35, which is an obvious feature found by biology scientists.

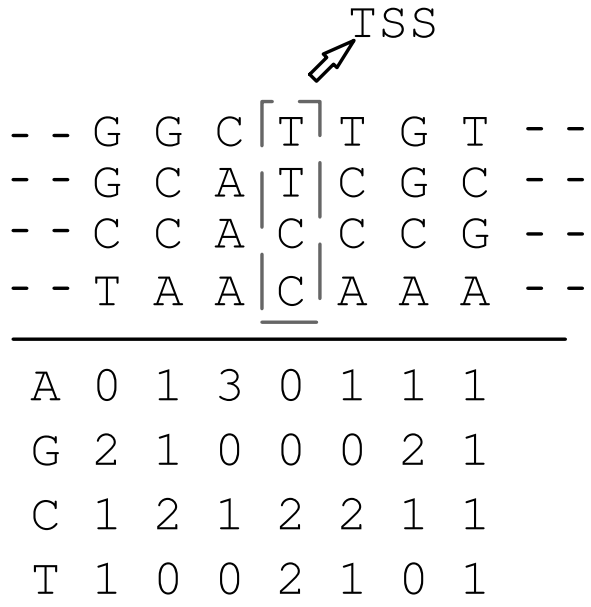

Figure 3.1 Example of method 1

3.3 Method 2

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 our method 1, 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.

We can find the TSS position of a DNA sequence 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.

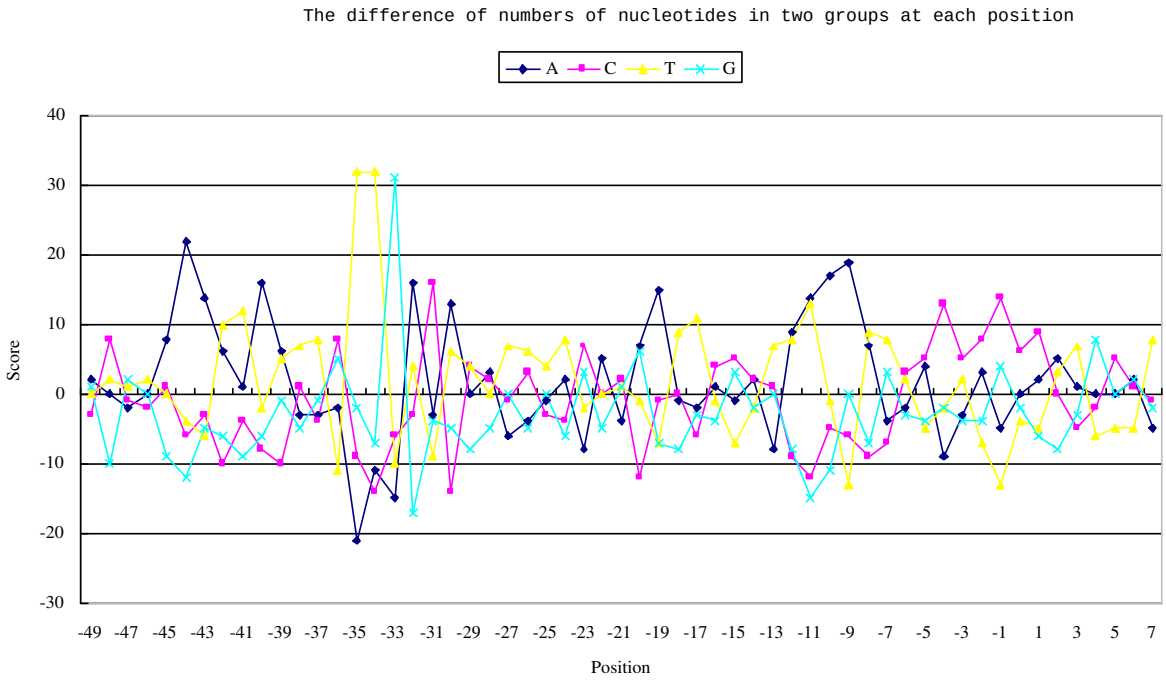

Figure 3.2 The distribution of each nucleotide.

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. Here we define some 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 of Σ, # represents the number of arbitrary alphabets between σ_1 and σ_2, $\# \in N \bigcup \{0\}$. For example, $\sigma_1 = $ A, $\sigma_2 = $ T, and $\# = 1$, then $\underline{\mathbf{A}}1\underline{\mathbf{T}} = \{\underline{\mathbf{A}}\mathrm{A}\underline{\mathbf{T}}, \underline{\mathbf{A}}\mathrm{G}\underline{\mathbf{T}}, \underline{\mathbf{A}}\mathrm{C}\underline{\mathbf{T}}, \underline{\mathbf{A}}\mathrm{T}\underline{\mathbf{T}}\}$. As another example, $\sigma_1 = $ G, $\sigma_2 = $ C, and $\# = 2$, the $\underline{\mathbf{G}}\mathrm{AC}\underline{\mathbf{C}}$, $\underline{\mathbf{G}}\mathrm{TT}\underline{\mathbf{C}}$, $\underline{\mathbf{G}}\mathrm{GA}\underline{\mathbf{C}} \in \underline{\mathbf{G}}2\underline{\mathbf{C}}$ 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 Method 2

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.

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

Step 1: Create $\sigma, \sigma\#\sigma, \sigma\#\sigma\#\sigma, \sigma\#\sigma\#\sigma\#\sigma$, $\sigma \in \sum$,all possible kinds of transcriptional element types.

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. The initial scores of all possible transcriptional elements are zero.

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

Step 4: We 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 of 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 score and the initial score is zero.

Step 5: We 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 promoter; NO, if otherwise.

Step 1: Using the score file to calculate its corresponding score in each position. If the input sequence contain 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.

The result of the possible transcriptional elements with scores in our method 2 is shown in Figure 3.3. In Figure 3.3, $\underline{\mathbf{A}}0\underline{\mathbf{A}}0\underline{\mathbf{A}}0\underline{\mathbf{A}} = \{\underline{\mathbf{AAAA}}\}$ and this transcriptional element appears in 19 sequences of the training dataset. $\underline{\mathbf{A}}0\underline{\mathbf{A}}0\underline{\mathbf{A}}1\underline{\mathbf{C}} = \{\underline{\mathbf{AAA}}\mathrm{A}\underline{\mathbf{C}}, \underline{\mathbf{AAA}}\mathrm{G}\underline{\mathbf{C}}, \underline{\mathbf{AAA}}\mathrm{C}\underline{\mathbf{C}}, \underline{\mathbf{AAA}}\mathrm{T}\underline{\mathbf{C}}\}$ and they appear in 11 sequences of the training dataset.

3.4 Method 2 with Frame Constraints

In our method 2, we only create all possible transcriptional elements up to four fixed nucleotides. We find when we create all possible transcriptional elements

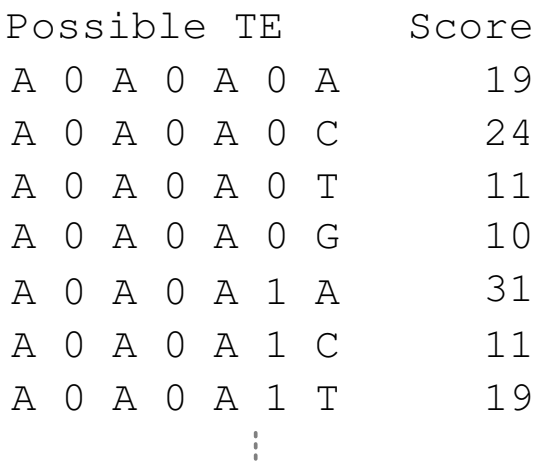

Figure 3.3 Score file of method 2.

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, we think the promoter features should not be too long, so we may use method 2 practically 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 have the better results. We take the frame length 15 in our method 2 and this length is shorter than the testing sequences length.

The result will also be discussed in Chapter 4.

CHAPTER 4

Experimental Results and Accuracy Analysis

The testing results of our method 1 and method 2 are shown in Table 4.1. The prediction accuracy of method 1 is better than method 2. The prediction results of method 2 with frame constraints are shown in Table 4.2 and method 2 with frame constraints gets better results than method 1. Table 4.3 shows the detailed FP (false positive), miscarriaging a non-promoter sequence as a promoter sequence, and FN (false negative), miscarriage a promoter sequence as a non-promoter sequence, and error rate of all our methods in the promoter prediction.

In method 2, 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 method 2, if the fixed nucleotides grows up to five, the transcriptional element score file becomes very large and the prediction time is too long. So we think we can knock out some transcriptional elements of low frequency by adding the frame constraints in method 2.

In the same type of fixed nucleotides, method 2 without frame constraints has the better result, but in method 2 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 method 2. In our experimental results, method 2 with frame

Table 4.1 The promoter prediction accuracy comparison

Method	Method 1	Method 2 ($\sigma\#\sigma\#\sigma$)	Method 2 ($\sigma\#\sigma\#\sigma\#\sigma$)
No. of error /106	9	15	12

Table 4.2 The promoter prediction accuracy using method 2 with frame constraints. Method 2 in this table means method 2 with frame constraints. The frame length is 15 in this table.

Method	Method 2 ($\sigma\#\sigma\#\sigma$)	Method 2 ($\sigma\#\sigma\#\sigma\#\sigma$)	Method 2 ($\sigma\#\sigma\#\sigma\#\sigma\#\sigma$)	Method 2 ($\sigma\#\sigma\#\sigma\#\sigma\#\sigma\#\sigma$)
No. of error /106	23	15	9	7

constraints of length 15 has the best accuracy. We compare our results with the results of ID3 [21], C4.5 [22] and GBI [15]. All of these methods are use the same dataset as ours. Figure 4.1 shows the promoter prediction accuracy by comparing our method2 with frame constrains 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 do outside test.

Clearly, both our method 1 and method 2 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.

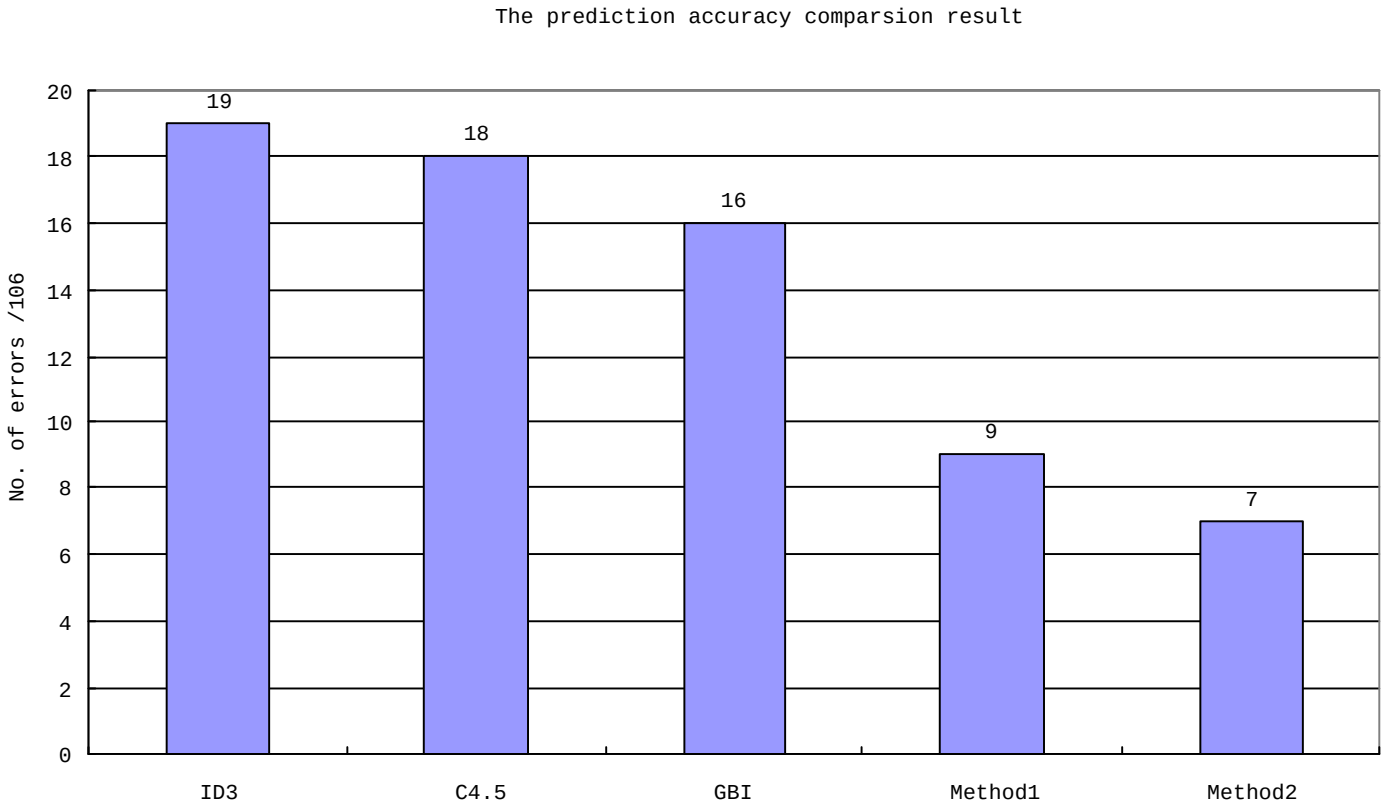

Figure 4.1 The promoter prediction accuracy comparison of our methods with others. Method 2 here is with frame constraint of length 15.

Table 4.3 The result of our methods

	FP rate (%)	FN rate (%)	Error rate (%)
Method 1	0	16.98	8.49
Method 2($\sigma\#\sigma\#\sigma$)	18.87	9.43	14.15
Method 2($\sigma\#\sigma\#\sigma\#\sigma$)	11.32	11.32	11.32
Method 2 with frame constraints ($\sigma\#\sigma\#\sigma$)	15.09	28.31	21.70
Method 2 with frame constraints ($\sigma\#\sigma\#\sigma\#\sigma$)	16.98	11.32	14.15
Method 2 with frame constraints ($\sigma\#\sigma\#\sigma\#\sigma\#\sigma$)	13.21	3.77	8.49
Method 2 with frame constraints ($\sigma\#\sigma\#\sigma\#\sigma\#\sigma\#\sigma$)	7.55	5.66	6.60

Step 2: Use the remaining 100 sequences (excluding the above 6 selected sequences) as the training sequences to get the score file with method 1 (or method 2).

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

In method 1, 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 4.4. We find that the result for testing sequences selected from training data is consistent with the result in Table 4.3. 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 method 2 with frame constraint $\sigma\#\sigma\#\sigma\#\sigma\#\sigma\#\sigma$ of length at most 15, we also randomly select 60 promoters and 60 non-promoter

Table 4.4 The experimental of outside tests in results method 1. P means promoter sequences and NP meas non-promoter sequences.

	Training Data		Testing Data		(Training + Testing)Data		
Sequence	P	NP	P	NP	P	NP	ALL
Total No.	1000	1000	60	60	1060	1060	2120
Error No.	3	178	3	25	5	203	208
Error rate	FN	FP	FN	FP	FN	FP	
(%)	0.30	17.80	5.00	41.67	0.47	19.15	9.81

sequences for testing totally in 20 experiments. The result is shown in Table 4.5. We find that in testing data, the prediction accuracy of promoter is a little higher and the non-promoter is a little lower comparing with Table 4.3. We think the little higher FN and lower FP than Table 4.3 is because the result in Table 4.3 is only one testing case. In our testing model of method 2 with frame constraints, each randomly selected case more times and the result should be more correct. In this result, we think that method 2 is accurate not only for the training set but also the testing set.

In our result we can find that method 2 have the better prediction accuracy in our whole data (promoter and non-promoter sequences). If we consider the FN rate only, the result of method 1 is better than method 2. So when we want to predict promoter, we can take both the results of method 1 and method 2 into consideration and these results can help us to eliminate many sequences which should not be promoter sequences.

Table 4.5 The experimental of outside tests in results method 2 with frame constraint $\sigma\#\sigma\#\sigma\#\sigma\#\sigma\#\sigma$ of length at most 15. P means promoter sequences and NP meas non-promoter sequences.

	Training Data		Testing Data		(Training + Testing)Data		
Sequence	P	NP	P	NP	P	NP	ALL
Total No.	1000	1000	60	60	1060	1060	2120
Error No.	99	35	6	3	105	38	143
Error rate	FN	FP	FN	FP	FN	FP	
(%)	9.90	3.50	10.00	5.00	9.91	3.59	6.75

CHAPTER 5

Conclusion

In this thesis, 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 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.

In our second method, 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 thesis 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 that 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.

BIBLIOGRAPHY

BIBLIOGRAPHY

[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] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, 1951.

[13] S. Lisser and H. Margalit, "Compilation of E. coli mRNA promoter sequences," *Nucleic Acids Research*, Vol. 21, pp. 1507–1516, 1993.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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).*

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

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

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

[23] T. Reichhardt, "Will souped up salmon sink or swim?," *Nature*, Vol. 406, pp. 10–12, 2000.

[24] B. A. L. T. R. J. R. G. W. K. W. K. M. M. B. V. A. W. C. S. Rorth P, Szabo K, "Systematic gain-of-function genetics in Drosophila," *Development*, Vol. 125, pp. 1049–1057, 1998.

[25] C. Starr and R. Taggart, *Biology: The Unity and Diversity of Life.* Boston, USA: Wadsworth Publish Company, five ed., 1995.