CHAPTER IV

PROPOSED METHODS FOR PROMOTER RECOGNITION

4.1 Problem Formulation

The problem can be defined as binary classification of DNA sequences, which is to classify a set of unlabeled data into two or more classes. Here, I focus on classifying DNA sequences into two categories: promoters and non-promoters.

Problem: Given an unlabeled DNA sequence S, we want to determine whether or not S is in promoter.

Input: $S_1, S_2, ..., S_N$ where N denote a number of DNA sequences and l denote length of S.

A labeled sequence $S = s_1 s_2 ... s_{l-1} s_l$, where $s_i \in \{A, T, C, G\}$ is a nucleotide.

Output: $promoter\ or\ non-promoter$

4.2 Feature extraction and selection from DNA sequences

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. 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. This dissertation uses 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 feature selection. Details of CGR and feature selection are discussed in the following subsections.

4.2.1 Chaos Game Representation (CGR)

Chaos Game Representation (CGR) was proposed as a method to identify patterns in DNA sequences [23, 24, 25, 26]. The algorithm is based on iterated function systems of fractal theory and maps a discrete sequence of symbols onto a continuous space. It assigns each symbol of a sequence alphabets to a corner of a hypercube and represents the sequence by successively going half the distance to the corner corresponding to the following symbol in the sequence. Initially, the image is divided into four quadrants in which each of them represents one of the four possible nucleotides (A, C, G and T). Each quadrant is subsequently divided into four subquadrants, each containing sequences ending with a given dinucleotide such that the sequences differing only in the first letter are in adjacent subquadrants (Figure 4.1).

The Algorithm

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 (Figure 4.2). Let $S = (s_1 s_2 \dots s_l)$ be a DNA sequence where $s_i \in \{A, C, G, T\}$ and l denotes to the length of S. Initially, quadrants whose corners 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

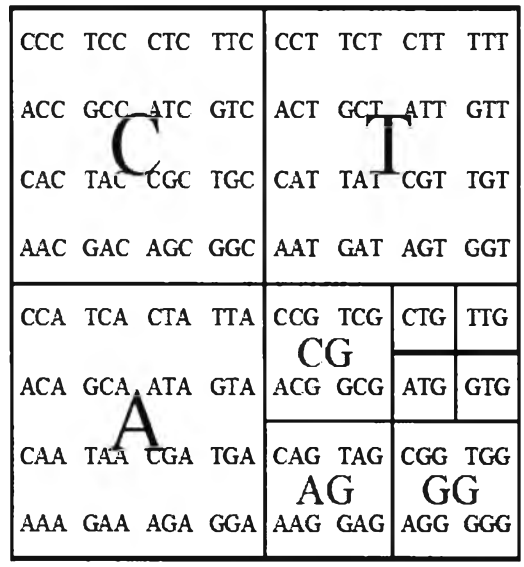


Figure 4.1: Chaos Game Representation (CGR) suffix property. Sequences ending in a specific sub-string are in the square labeled with that suffix.

the following steps.

$$\begin{aligned} p_0 &= (0.5, 0.5) \\ p_i &= p_{i-1} + 0.5(x_i - p_{i-1}), \quad i = 1, 2, \ldots, l \end{aligned} \tag{4.1}$$

x_i is the coordinate variable of nucleotide s_i. The value of x_i is set as follows. Position

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

p_i is denoted by a dot. The corresponding CGR image can be viewed as an image of distributed dots. This image is, then, partitioned by grids into a set of square entries of equal size.

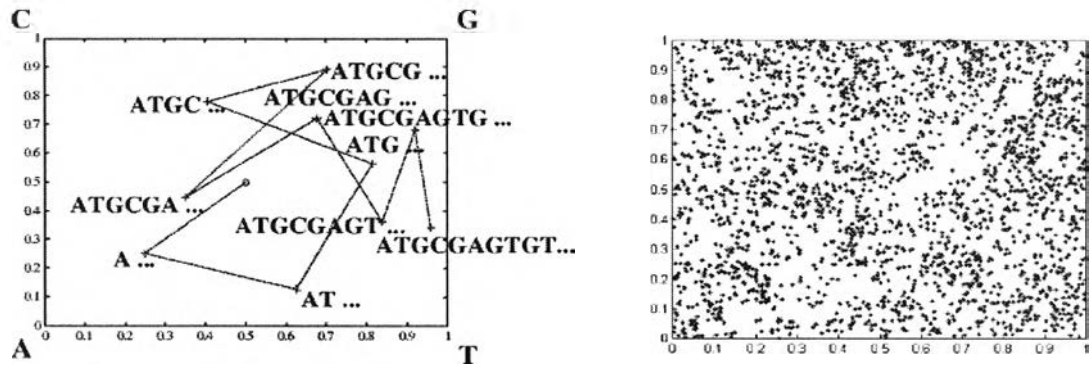


Figure 4.2: (a) Chaos Game Representation (CGR) of the first 10 nucleotides of *E.coli* gene *thrA*: ATGCGAGTGT. (b) CGR of the full *thrA* sequence, totaling 2463 pairs of bases.

The set of words or patterns length n denote W_n can be extracted from CGR that has k elements.

$$W_n = \{W_{n,1}, W_{n,2}, ..., W_{n,k}\} \tag{4.2}$$

$$k = 2^n \times 2^n \tag{4.3}$$

The identification of word frequencies (f_n) in the sequence S can then be object of counting occurrence in each quadrant.

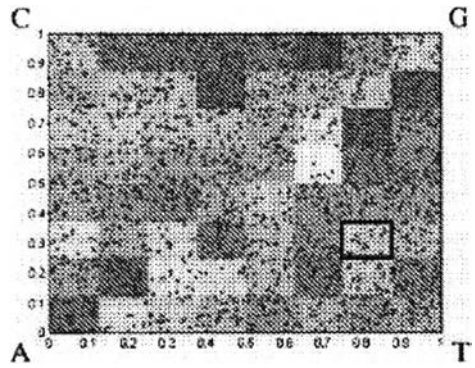
$$f_n = \{f_{n,1}, f_{n,2}, ..., f_{n,k}\} \tag{4.4}$$

For example, for the sequence $S = ATATAC$ where $l = 6$ the frequencies of all trinucleotides would be:

$$W_3 = \{ATA, TAT, TAC, AAA...\} \tag{4.5}$$

$$f_3 = (2, 1, 1, 0, ...) \tag{4.6}$$

The vectors f_3 have length $k = 2^3 \times 2^3 = 64$ and the zero coordinates correspond to missing words in sequences S, in this case absent trinucleotides. Figure 4.3 shows the example of trinucleotide frequency matrix ($n = 3$) obtained for the *thrA* sequence.



C G

27	56	60	64	57	76	46	28
36	24	28	68	33	33	30	68
24	22	29	29	37	19	72	46
37	30	41	40	38	8	55	48
36	43	50	37	23	45	43	43
15	38	20	54	29	46	25	31
39	64	14	17	30	55	28	50
58	15	22	33	43	33	43	36

A T

Figure 4.3: The CGR coordinates for the 2463 base pairs are plotted with the relative frequencies for each 8×8 quadrant represented as a grayscale(left). The distribution of counts in listed in the table(right).

4.3 Statistical 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 length l, but some of these are not essential and 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. Our selection is based on the concept of statistical methods.

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 $f_{i,j,k}^{(c)}$, $c \in \{0, 1\}$, be the value of square entry (i, j) of the k^{th} CGR image for class c and $D_{i,j}$ be the distance values. In this dissertation, distance measurement between promoter class and non-promoter class at entry (i, j) is proposed in the following subsections.

4.3.1 Statistical Feature Method

In this dissertation, I use mahalonobis distance method for selecting features of *E.coli* data sets. The method is based on t-Test that determines whether a significant difference exists between the means of two classes. The formula of t-Test is the ratio of the difference between group means and the variability of groups.

$$D_{i,j} = \frac{(m_1^2 - m_0^2)}{\delta_1^2 + \delta_0^2} \tag{4.7}$$

$$m_0 = \frac{\sum_{k \in S_0} f_{i,j,k}^{(0)}}{|S_0|} \tag{4.8}$$

$$m_1 = \frac{\sum_{k \in S_1} f_{i,j,k}^{(1)}}{|S_1|} \tag{4.9}$$

$$\delta_0 = \sqrt{\frac{\sum_{k \in S_0} (f_{i,j,k}^{(0)} - m_0)^2}{|S_0|}} \tag{4.10}$$

$$\delta_1 = \sqrt{\frac{\sum_{k \in S_1} (f_{i,j,k}^{(1)} - m_1)^2}{|S_1|}} \tag{4.11}$$

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 (i, j) square entries with highest values of $D_{i,j}$ are selected as the essential features for training.

4.3.2 CpG island features

I use CpG island as one feature for improve human promoter prediction. CpG island features can be used for vertebrate promoter but do not exist in non-vertebrate eukaryotes such as *D.melanogaster*. I use two CpG island features–GC content and ratio of expected to observed CG dinucleotides. Let l be the length of DNA sequence, *GC-content* and *CG-ratio* are defined as

$$GC - content = \frac{(\#G) + (\#C)}{l} \tag{4.12}$$

$$CG - ratio = \frac{(\#CG) \times l}{(\#G) + (\#C)} \tag{4.13}$$

4.4 Architecture of the prediction system

The conceptual structure of our system is depicted in Figure 4.4. From the proposed method, a feed-forward backpropagation neural network with a single hidden layer is used. There is one output unit for two target classes, i.e., class 1 for promoter and class 0 for non-promoter. The input data are composed of d hightest value from feature selection and CpG isalnd features. In this dissertation, the ANN implementation is carried out using Stuttgart Neural Network Simulator (SNNS) 4.2 which is freely distributed 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. The magnitude of the error sum in the test and training set was monitored in each cycle of the training. During testing of the network, the output of the network was compared with an arbitrarily defined cutoff value. If the output was greater than the cutoff, then the fragment was considered to be promoter, otherwise, it was considered to be a non-promoter. Identification of promoter regions in large genomic sequences is performed by partitioning an unknown sequences into windows of 300 bp long, shifted by 1 bp. For each sliding data window, compute the feature values following procedures in the previous section to be used as inputs of the neural network. A promoter region is reported if a certain number of consecutive windows are identified as members of the promoter class.

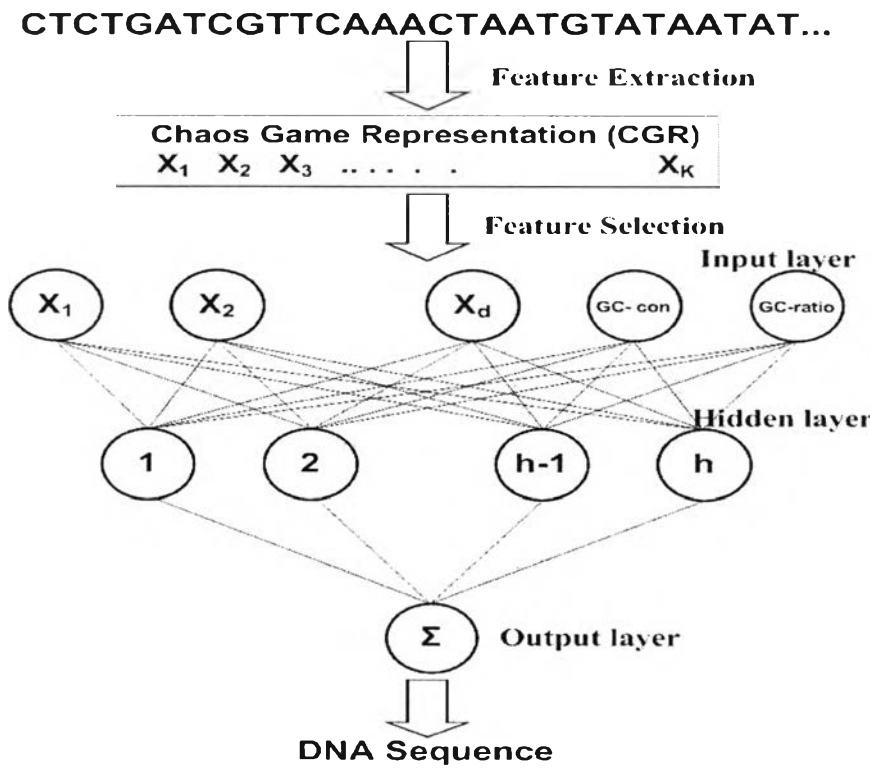


Figure 4.4: The structure of neural network used in this problem.