

Promoter Recognition for E. coli DNA Segments by Independent Component Analysis

Yasuo MATSUYAMA
Department of Computer Science,
Waseda University, Tokyo, 169-8555 Japan
yasuo2@waseda.jp

Ryo KAWAMURA
Sony-Kihara Research Center Inc.,
Tokyo, 141-0022 Japan
ryo@asagi.waseda.jp

Abstract

A new method for E. coli DNA segment classification on promoters and non-promoters is presented. The algorithm is based on the Independent Component Analysis (ICA). Since the DNA segments are composed of discrete symbols, this paper contains two major steps: (1) Position-dependent transformation of DNA segments to real number sequences, and (2) Applications of the ICA to the E. coli promoter recognition. These steps are related to each other. Therefore, algorithmic explanations are given in detail while referring mutually. The automatic precision of 93.7% is obtained. Since the presented method allows threshold adjustments, twilight-zone data can be further cross-checked individually so that false negatives are reduced.

1. Introduction

DNA sequences contain portions of special functions [1], [2]. The promoter is one of such an important structure which works as a polymerase binding site. Recognition of promoter patterns keeps its importance because of the relationship to the transcription [3]. In this paper, the recognition of E. coli promoter segments is addressed. Existing recognition methods use artificial neural networks [4] and their combination with the expectation maximization algorithm [5]. The new method in this paper, however, is based on the Independent Component Analysis (ICA). On the ICA, we will use our own generalized algorithm [6], [7] derived by the minimization of the convex divergence which is the ultimately general version of the entropy.

The ICA is a statistical learning algorithm for numerical data. On the other hand, DNA sequences are composed of four symbols $\{A, T, G, C\} \stackrel{\text{def}}{=} \mathcal{D}$. Thus, consistent conversions from symbol sequences to real number series are necessary. Therefore, this paper includes a position-dependent conversion based on symbol frequencies. By the ICA with such numerical conversions, the resulting automatic precision of 93.7% is obtained. Since the presented method al-

lows threshold adjustments, twilight-zone data can be further cross-checked individually so that false negatives are reduced.

2. E. coli Promoter Recognition

2.1. Structure and Function of E. coli Promoters

Figure 1 is a conceptual illustration explaining the structure of the E. coli promoter. There are specific sub-

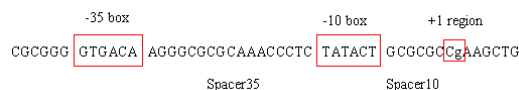


Figure 1. Conserved regions in the E. coli promoter.

configurations in the E. coli promoter. They are called the -35 box, the -10 box, and the +1 region. The transcription starts from the +1 region. The region between the -35 box and the -10 box is called the Spacer35. The region between the -10 box and the +1 region is called the Spacer10. The -35 box and the -10 box have the fixed length of 6 nt¹. But, their contents may vary. The length of the Spacer35 may vary from 15 to 21 nt. The Spacer10 may vary from 3 to 11 nt length. Thus, there are various promoter patterns for E. coli sequences. Therefore, symbolic pattern matching is not quite appropriate for the promoter analysis, but efficient pattern recognition methods including probabilistic or soft decisions are required.

2.2. Procedure of E. coli Promoter Recognition

Every pattern recognition method contains an off-line training phase (learning phase) and an on-line test phase

¹ "nt" stands for nucleotide.

(execution phase). This paper's training procedure for the E. coli promoter recognition is previewed as follows.

[Training Steps: Off-line]

[TR 1] A set of length-adjusted E. coli promoter segments is prepared.

[TR 2.1] The set of -35 boxes is changed to a real valued matrix.

[TR 2.3] A random matrix for the -35 box is generated and juxtaposed to the -35 box matrix.

[TR 2.3] From the total -35 box matrix, the feature of the -35 box is learned by the ICA.

[TR 3.1] The set of -10 boxes is changed to a numerical matrix.

[TR 3.2] A random matrix for the -10 box is generated and juxtaposed to the -10 box matrix.

[TR 3.3] From the total -10 box matrix, the feature of the -10 box is learned by the ICA.

[TR 4.1] The set of the length-adjusted promoters are changed to a real number matrix.

[TR 4.2] A random matrix for the promoters is generated using the ICA results of TR 2.3 and TR 3.3. This random matrix is juxtaposed to the promoter matrix.

[TR 4.3] From the total data matrix, the promoter structure is learned by the ICA.

It is important to note here that the ICA is used three times; on the -35 box, on the -10 box, and on the total segment.

The test phase is previewed as follows.

[Test Steps: On-line]

[TS 1] A segment to be tested is given. This may be a set of segments.

[TS 2] The given segment is length-adjusted² by using the ICA matrices for the boxes obtained in the training steps.

[TS 3] The length-adjusted segment is transformed to a real number sequence.

[TS 4] Using the ICA matrices, the segment is judged to contain an E. coli promoter or not. Estimated boxes are obtained here.

In the above training and testing steps, there are novel features distinctive to this paper.

- (a) To all aspects of the training and test steps, the Independent Component Analysis (ICA) is related.
- (b) On the conversion of symbols to real numbers, position-dependent base frequencies are used. This is not a naive transformation of symbols to unit vectors.

² The length adjustment in this paper has a different purpose from ClustalW, BLAST, and so on.

- (c) In the training data for boxes and total segments, random segments are juxtaposed to pure data. This is related to the data augmentation or the bootstrap method.
- (d) The length-adjustment uses learned ICA matrices.

3. Independent Component Analysis

3.1. Mixture of Independent Components

In the problem of the Independent Component Analysis (ICA), observed or given data are assumed to be an unknown mixture of unknown data. That is, the observed data $\mathbf{x}(t) = [x_1(t), \dots, x_N(t)]^T$ is generated from unknown source $\mathbf{s}(t) = [s_1(t), \dots, s_N(t)]^T$ by

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = [\mathbf{a}_1, \dots, \mathbf{a}_N]\mathbf{s}(t) = \sum_{i=1}^N \mathbf{a}_i s_i(t), \quad (1)$$

as is illustrated in Figure 2. We want to estimate $\mathbf{s}(t)$ and \mathbf{A} by observing only $\mathbf{x}(t)$. The ICA algorithm gives an estimation of \mathbf{A}^{-1} as a de-mixing matrix \mathbf{W} . The vector $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ has de-mixed, or independent components. In this problem setting, we are allowed to assume that the components of $\mathbf{s}(t)$ are independent each other.

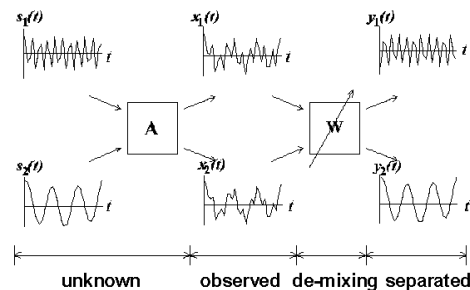


Figure 2. ICA structure ($N = 2$).

In our application of ICA to the promoter recognition, the data $\mathbf{x}(t)$ come from -35 boxes, -10 boxes, and total segments of the E. coli DNA. Therefore, three de-mixing matrices, say \mathbf{W}_{-35} , \mathbf{W}_{-10} , and $\mathbf{W}_{\text{total}}$ will be learned from given training sets.

To be precise, available information on the mixing structure is only the following: (a) The components s_i and s_j are independent each other if $i \neq j$. (b) The components $s_i(t)$, ($i = 1, \dots, N$), are non-Gaussian except for at most one i .

Given such assumptions, we want to estimate a de-mixing matrix $\mathbf{W} = \Lambda\Pi\mathbf{A}^{-1}$ so that the components $y_i(t)$, ($i = 1, \dots, N$), of

$$\mathbf{W}\mathbf{x}(t) \stackrel{\text{def}}{=} \mathbf{y}(t) = [y_1(t), \dots, y_N(t)]^T \quad (2)$$

are independent each other. Here, Λ is a nonsingular diagonal matrix which decides each component's scale, and Π is a permutation matrix. These matrices are unknown too. This property is called the indeterminacy.

3.2. ICA Bases

Column vectors \mathbf{u}_i of $\mathbf{U} \stackrel{\text{def}}{=} \mathbf{W}^{-1}$ are interpreted as ICA bases. This is because the observed data \mathbf{x} is expressed by the weighted summation of the de-mixed components:

$$\mathbf{x}(t) = \mathbf{U}\mathbf{y}(t) = [\mathbf{u}_1, \dots, \mathbf{u}_N]\mathbf{y}(t) = \sum_{i=1}^N \mathbf{u}_i y_i(t). \quad (3)$$

The terminologies “ICA bases” and “DNA bases” should not be confused. They are totally different concepts. The ICA basis is the very one which represents the promoter structure. This will be illustrated as an experimental result in Section 6.2 (see Figure 3).

3.3. Derivation of the ICA Algorithm

Let $p(\mathbf{y}) = p(y_1, \dots, y_N)$ be a joint probability density, and $q(\mathbf{y}) = \prod_{i=1}^N q_i(y_i)$ be a product probability density. Then, the independence is obtained by the minimization of the following cost function.

$$\begin{aligned} I_f(\bigwedge_{i=1}^N Y_i) &\stackrel{\text{def}}{=} D_f\left(p(y_1, \dots, y_N) \parallel \prod_{i=1}^N q_i(y_i)\right) \\ &\stackrel{\text{def}}{=} D_f(p(\mathbf{y}) \parallel q(\mathbf{y})) = D_g(q(\mathbf{y}) \parallel p(\mathbf{y})) \\ &= \int_{\mathcal{X}} p(\mathbf{x}) g\left(\frac{\det(\mathbf{W})q(\mathbf{y})}{p(\mathbf{x})}\right) d\mathbf{x} \end{aligned} \quad (4)$$

Here, $D_f(p \parallel q)$ is the convex divergence [8] between p and q in terms of a twice differentiable convex function $f(r)$ with $f(1) = 0$. The dual function g is defined by $g(r) = rf(1/r)$. Note that a special case of the convex divergence is the Kullback-Leiber divergence or the average mutual information. Further special case is the Shannon’s entropy.

By computing the negative gradient of $I_f(\bigwedge_{i=1}^N Y_i)$, the increment $\tilde{\Delta}_f \mathbf{W}$ of the following ICA learning equation can be obtained [6], [7]:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \tilde{\Delta}_f \mathbf{W}, \quad (5)$$

where t is the iteration index for learning³, and

$$\begin{aligned} \tilde{\Delta}_f \mathbf{W} &= -\rho \frac{\partial I_f}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} \\ &= \rho \left[\left\{ \mathbf{I} - E_{p(\mathbf{y})}[\tilde{\varphi}(\mathbf{y})\mathbf{y}^T] \right\} \mathbf{W} \right. \\ &\quad \left. + \mu \left\{ \mathbf{I} - E_{q(\mathbf{y})}[\tilde{\varphi}(\mathbf{y})\mathbf{y}^T] \right\} \mathbf{W} \right]. \end{aligned} \quad (6)$$

Here, $\tilde{\varphi}(\mathbf{y})$ is a nonlinear vector function such as $\varphi_i(\mathbf{y}) = y_i^3$ or $\varphi_i(\mathbf{y}) = \tanh(y_i)$. The coefficient ρ is a small positive number called the learning rate. μ is a non-negative number for the effect of the acceleration on the learning. $E_{p(\mathbf{y})}[\cdot]$ and $E_{q(\mathbf{y})}[\cdot]$ are expectations with respect to the suffixd probability densities. Both are computed from given data. Since $q(\mathbf{y})$ is the unknown target, this quantity is regarded as a time-shifted version of $p(\mathbf{y})$ in programming.

³ This should not be mistaken for the sample data index.

3.4. Sample Data and Pre-processing

In actual data processing, we are given samples of random vectors $\mathbf{x}(t)$, ($t = 1, \dots, M$). We write down the set of samples in matrix forms: $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(M)]$, $\mathbf{S} = [\mathbf{s}(1), \dots, \mathbf{s}(M)]$, $\mathbf{Y} = [\mathbf{y}(1), \dots, \mathbf{y}(M)]$. Thus, the data generation model (1) is expressed by $\mathbf{X} = \mathbf{A}\mathbf{S}$. Also, the de-mixed result is expressed by $\mathbf{Y} = \mathbf{W}\mathbf{X}$. Then, the expectation $E_{p(\mathbf{x})}[\cdot]$ is replaced by $\sum_{j=1}^M [\cdot]$.

Usually, observed data \mathbf{X} is pre-processed so that the separability into independent components is increased. The first pre-processing is to make the data to be zero-mean. Another important pre-processing is the whitening using the covariance matrix. On these standard pre-processing strategies, readers are requested to refer to ICA literatures, e.g., [7].

4. Parts of Training Steps

4.1. Learning on De-mixing Matrices for Boxes

4.1.1. Numerical Expression of Training Data for Boxes

First, the training set of -35 boxes are changed to a numerical matrix by the following way.

- For the training steps, length-adjusted promoters are given. In the later experiment, the total number is $N_p = 106$ drawn from [4]. Each -35 box is 6 nt length.
- For each position of the -35 box, normalized $\{A, T, G, C\}$ -frequencies are count. This position-dependent count generates a 4×6 matrix, which we call the Table_{-35} .
- From the training data of -35 boxes, we have an $(N_p - k_{-35}) \times 6$ matrix. Here, $N_p - k_{-35}$ means the removal of k_{-35} identical patterns. In the training set, $N_p - k_{-35} = 72$. Then, each entry of the -35 boxes is changed to a numeral by using Table_{-35} . The resulting matrix is called \mathbf{B}_{-35} . The matrix \mathbf{B}_{-35} will be used as a core part for obtaining the ICA de-mixing matrix \mathbf{W}_{-35} .

Next, the same procedure is tried for -10 boxes to obtain the ICA de-mixing matrix \mathbf{W}_{-10} .

4.1.2. Random Matrix Juxtaposition for Learning on \mathbf{W}_{-35} and \mathbf{W}_{-10} For obtaining the de-mixing matrix \mathbf{W}_{-35} , an artificially generated random matrix is juxtaposed to the data matrix \mathbf{B}_{-35} . This is a kind of data augmentation.

- A random data matrix of $(N_p - k_{-35}) \times 6$ is prepared. Here, each entry is drawn from $\{A, T, G, C\} = \mathcal{D}$.
- Using the Table_{-35} , each entry is changed to numerals. This matrix is called \mathbf{C}_{-35} .
- By juxtaposing \mathbf{B}_{-35}^T and \mathbf{C}_{-35}^T , we have a data matrix of size $6 \times 2(N_p - k_{-35})$. This is called \mathbf{X}_{-35} .

- (d) The matrix \mathbf{X}_{-35} is preprocessed for the ICA to be the zero mean and the unit covariance. The resulting matrix is renamed \mathbf{X}_{-35} .
- (e) Using the data matrix \mathbf{X}_{-35} , the ICA learning is carried out. Then, the de-mixing matrices \mathbf{W}_{-35} and the de-mixed data matrix \mathbf{Y}_{-35} are obtained.

For \mathbf{W}_{-10} and \mathbf{Y}_{-10} , the same procedure using \mathbf{B}_{-10}^T and \mathbf{C}_{-10}^T is executed.

4.2. Segment Length Adjustment

On the -35 box and -10 box, length adjustments were not needed since their lengths are fixed to 6 nt. But, on the processing of the total promoter region, appearing later in Section 4.3, a length adjustment will become necessary. This is because the Spacer35 and the Spacer10 are variable-length. The algorithm is based on [4]. But, our method uses ICA results of \mathbf{W}_{-35} , \mathbf{Y}_{-35} , \mathbf{W}_{-10} and \mathbf{Y}_{-10} , which are obtained in advance.

The following steps generate length-adjusted segments.

- (a) A segment to be length-adjusted is given with an identified start point.
- (b) Looking back from the starting point, find the best ending point of the -35 box in the region [15...21]. The best position is identified by the maximum inner product using the column vectors $\mathbf{y}_{-35}(k)$, $k = 1, \dots, 7$. Here, 7 appears as the zone length of the possible ending point. The mechanism of the maximum inner product will be explained in detail in 4.3.2.
- (c) The best ending point of the -10 box is identified in the same way using the column vectors $\mathbf{y}_{-10}(k)$ in the region [3...11].
- (d) Gaps are inserted so that the total length becomes 65 nt [4].

4.3. Feature Extraction for Total Promoter Structure by the ICA

4.3.1. Numerical Expression for Promoters First, the training set of promoters is changed to a numerical matrix by the following way: (a) There are $N_p = 106$ training promoters with 65 nt length. (b) For each column, the normalized frequencies on each position in the 65 nt length are obtained. This generates a table of the size 5×65 since the promoter sequence contains $\{A, T, G, C, -\} \stackrel{\text{def}}{=} \mathcal{D}^+$. This frequency table is called the $\text{Table}_{\text{promoter}}$. (c) By using the $\text{Table}_{\text{promoter}}$, a numerical matrix of size $N_p \times 65$ is obtained. This is called $\mathbf{B}_{\text{promoter}}$.

4.3.2. Random Matrix Generation for Promoter Learning Similar to the ICA learning on the -35 box and -10 box, a random matrix, say $\mathbf{C}_{\text{promoter}}$, is generated. This matrix will be juxtaposed to the data matrix $\mathbf{B}_{\text{promoter}}$ later.

- (a) Prepare N_p random segments of length 50 nt whose elements contain $\{A, T\}$ and $\{G, C\}$ to be 60% and 40%.
- (b) The first A or G from the end is regarded as the starting point in this random sequence [4].
- (c) A putative -35 box in each segment is found as follows: (1) Prepare 7 sliding segments of length 6 in the region [15...21]. Make a 6×7 matrix. Change each element to numerals using Table_{-35} . This matrix is called \mathbf{Z}_{-35} . (2) Compute $\mathbf{W}_{-35}\mathbf{Z}_{-35}$. This de-mixed matrix is called $\mathbf{Y}_{-35,C}$, whose column vectors are called $\mathbf{y}_{-35,C}(j)$, $j = 1, \dots, 7$. (3) Using the column vectors $\mathbf{y}_{-35,C}(k)$, of $\mathbf{Y}_{-35,C}^T$, compute the summations of the inner products by $q(j) = \sum_{k=1}^{58} \mathbf{y}_{-35,C}^T(j) \mathbf{y}_{-35,C}(k)$. (4) Set the putative end position of the -35 box to be $J_{-35} = \arg \max_{1 \leq j \leq 7} q(j)$.
- (d) Find a putative -10 box in the segment in the same way as the -35 box.
- (e) Perform the length adjustment to arrange the length to be 65 nt. This generates a random matrix of the size 106×65 .
- (f) Change the entries of this random matrix to numerical numbers by using $\text{Table}_{\text{promoter}}$. The resulting matrix is named $\mathbf{C}_{\text{promoter}}$.

4.3.3. ICA on the Total Promoter Structure Since the matrices $\mathbf{B}_{\text{promoter}}$ and $\mathbf{C}_{\text{promoter}}$ are prepared, the ICA learning for the total promoter region can be carried out.

- (a) Juxtapose the matrices $\mathbf{B}_{\text{promoter}}^T$ and $\mathbf{C}_{\text{promoter}}^T$. The resulting 65×212 matrix is called $\mathbf{X}_{\text{promoter}}$.
- (b) Preprocessing for the zero-mean and the whitening is executed on $\mathbf{X}_{\text{promoter}}$. The resulting matrix is renamed $\mathbf{X}_{\text{promoter}}$.
- (c) Using the data matrix $\mathbf{X}_{\text{promoter}}$, the de-mixing matrix $\mathbf{W}_{\text{promoter}}$ and the de-mixed matrix $\mathbf{Y}_{\text{promoter}}$ are obtained by the ICA algorithm. This completes the whole training phase.

5. Testing on Given Segments

5.1. Preparation of Test Data

Each test segment is processed in the following way: (a) A test segment with a given starting point is given. (b) The sequence is adjusted to be the length of 65 nt by using \mathbf{W}_{-35} , \mathbf{Y}_{-35} , \mathbf{W}_{-10} and \mathbf{Y}_{-10} . On the estimation of -35 box and -10 box, Table_{-35} and Table_{-10} are used for

the numerical conversion. The mean values and the whitening matrices obtained in the learning phase are also applied. (c) The resulting segment is changed to a 65-row numerical vector using the $\text{Table}_{\text{promoter}}$. Then, the mean value adjustment and the whitening are carried out by using the results obtained in the training phase. This is called \mathbf{x}_{test} .

5.2. Promoter Recognition

Given a vector \mathbf{x}_{test} to be tested, the following judgment is carried out:

(a) Compute $\mathbf{y}_{\text{test}} = \mathbf{W}_{\text{promoter}}\mathbf{x}_{\text{test}}$. (b) If the first element “ $y_{\text{test}}(1)$ ” is positive, the tested sequence is judged to contain a promoter. Otherwise, it is regarded as a non-promoter. For the positive sequence, gaps are removed to identify the estimated boxes. This completes the testing phase.

6. Experiments on Training and Testing

6.1. Data Preparation

A set of training data of length-adjusted segments were obtained from [4]. From this set, the matrices \mathbf{B}_{-35} , \mathbf{B}_{-10} , and $\mathbf{B}_{\text{promoter}}$ were generated. Then, they were changed to numerical matrices by the aforementioned methods which reflect position-dependent symbol frequencies.

Next, random matrices \mathbf{C}_{-35} and \mathbf{C}_{-10} were generated and changed to numerical matrices. Then, by the juxtaposition, \mathbf{X}_{-35} and \mathbf{X}_{-10} were generated. Then, by the ICA algorithm, \mathbf{W}_{-35} , \mathbf{Y}_{-35} , \mathbf{W}_{-10} , and \mathbf{Y}_{-10} were obtained.

Next, the random matrix $\mathbf{C}_{\text{promoter}}$ was generated. Then, it was changed to a numerical matrix by the aforementioned method. By the juxtaposition of $\mathbf{C}_{\text{promoter}}^T$ to $\mathbf{B}_{\text{promoter}}^T$ the training matrix $\mathbf{X}_{\text{promoter}}$ was generated.

6.2. Execution of the ICA Algorithm

By the execution of the ICA on $\mathbf{X}_{\text{promoter}}$, the de-mixing matrix $\mathbf{W}_{\text{promoter}}$ was obtained. As was explained in Section 3.2, each column vector of $\mathbf{W}_{\text{promoter}}^{-1}$ works as an ICA basis. The first one, say \mathbf{u}_1 , represents the major property of the promoter. Figure 3 illustrates the resulting ICA basis. We can observe that there are humps around the positions 27 and 52. They correspond to the -35 box and the -10 box, respectively.

6.3. Promoters and Non-Promoters

For the testing and performance evaluation, we prepared 126 promoter segments and 1,000 non-promoter segments. The set of 126 promoter segments were drawn from [4]. But, all gaps were removed in advance since our length-adjustment uses \mathbf{W}_{-35} , \mathbf{Y}_{-35} , \mathbf{W}_{-10} and \mathbf{Y}_{-10} . Therefore, the length of each segment varies at first.

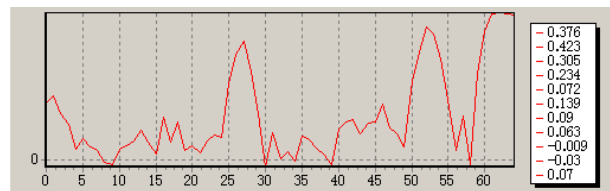


Figure 3. Obtained first ICA basis (\mathbf{u}_1).

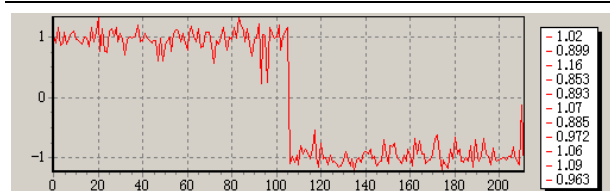


Figure 4. Decision by a threshold.

Generated non-promoter segments contain $\{A, T\}$ and $\{G, C\}$ by the ratio of 60% and 40%. The length of each segment is 50 nt. The first A or G from the end was regarded as the starting point.

On such 1,126 segments, the following test procedure was carried out.

- Using \mathbf{W}_{-35} , \mathbf{Y}_{-35} , \mathbf{W}_{-10} and \mathbf{Y}_{-10} , the length-adjustment was made.
- Each entry was changed to a numerical value by using $\text{Table}_{\text{promoter}}$. Pre-processing using the learned data was executed.
- The resulting matrix \mathbf{X}_{test} is of 65×1126 in the size.
- The de-mixed matrix was computed by $\mathbf{Y}_{\text{test}} = \mathbf{W}_{\text{promoter}}\mathbf{X}_{\text{test}}$.
- The first row of \mathbf{Y}_{test} , say “ $\mathbf{y}_{\text{test}}(1)$,” was taken out. Each element corresponds to each tested segment. The 1,126 elements in “ $\mathbf{y}_{\text{test}}(1)$ ” were judged if they are positive or not. If the k -th element is positive, then a promoter exists in the k -th segment, $k = 1, \dots, 1126$. Otherwise, the segment was judged to be a non-promoter. Figure 4 illustrates the resulting “ $\mathbf{y}_{\text{test}}(1)$.” As can be observed, there is a clear separation by the threshold at zero.

6.4. Performance Evaluation

On the recognition of promoters, the performance measures {precision, specificity, sensitivity} were computed. In order to compare the performances with existing studies [4], [5], the performance measures are defined in the usual way.

- The precision is computed by $\mathcal{P} = C/N_{\text{total}} \times 100\%$. Here, C is the number of correct judgments, and N_{total} is the total number of tested segments.

- (b) The specificity is defined by $\mathcal{S}_p = (1 - N_{fp}/N_{np}) \times 100\%$. Here, N_{fp} is the number of false positives. N_{np} is the number of tested non-promoter segments.
- (c) The sensitivity is defined by $\mathcal{S}_n = N_{tp}/N_p \times 100\%$. Here, N_{tp} is the number of true positives. $N_p = N_{total} - N_{np}$ is the number of tested promoter segments.

Performances by various methods are summarized in Table 1. The first line is the performance of our method. The second line is the performance of [5] which use the unit vector expression of $\{A, T, G, C\}$, the EM algorithm, and artificial neural networks. The third line is the performance of [4] which uses the unit vector expression of $\{A, T, G, C\}$ and artificial neural networks. Thus, the presented method has the best precision of 93.7%. It is important to note that the consensus for the -35 box was TTGACA, and that of the -10 box was TATAAT.

| | \mathcal{P} | \mathcal{S}_p | \mathcal{S}_n |
|------------|---------------|-----------------|-----------------|
| This paper | 0.937 | 0.934 | 0.968 |
| Method [5] | 0.919 | 0.918 | 0.992 |
| Method [4] | 0.904 | 0.902 | 0.980 |

Table 1. Performances of various methods

The score of our ICA method in Table I is merely an automatic result via the fixed threshold of $\vartheta = 0.0$. Upon observing Figure 4, however, readers can easily find that there are negative segments having scores only slightly below 0.0 (for instance, the one around the number 210 in Figure 4). They are highly likely to be false negatives. By watching the score, our method accepts additional interactive cross-examinations to reduce the false negatives.

7. Discussions and Concluding Remarks

In this paper, a new statistical method for E. coli promoter recognition was presented. The novelties in the presented method are summarized as follows: (1) The method is based upon the independent component analysis (ICA) which is unsupervised. But, the presented method beat existing supervised learning methods in the precision. (2) The threshold can be adjusted so that false negatives are reduced. (3) The numerical expression of DNA segments reflects position-dependent symbol frequencies.

The presented method can be extended and combined with other methods for further sophistication: (a) In this paper, promoters were recognized by using identified starting points. It is known that the transcription initiation sites may be diverse and can be identified exactly only via wet biological experiments, e.g., [9]. But, posterior probability approaches looking back from the promoter patterns are possi-

ble. The ICA promoter recognition method in this paper exists in the realm of this category. Our preliminary study supports this matter. (b) Incorporation of partially supervised mechanism [10] will improve the ability of the ICA. (c) The EM algorithm [11], [12] which contains the Hidden Markov Model as its special class can be combined.

Acknowledgment

The authors are grateful to the referees of the early version for their valuable comments. This study was supported in part by the Grant-in-Aid for Scientific Research #15300077, and by the Productive ICT Academia of the 21st Century COE Program.

References

- [1] G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, vol. 15, pp. 563-577, 1999.
- [2] D.W. Mount. *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2001.
- [3] D.H. Hawley and W.R. McClure. Compilation and analysis of Escherichia coli promoter DNA sequences. *Nucleic Acids Research*, vol. 11, pp. 2237-2255, 1983.
- [4] I. Mahadevan and I. Ghosh. Analysis of E. coli promoter structures using neural networks. *Nucleic Acids Research*, vol. 22, pp. 2158-2165, 1994.
- [5] Q. Ma, T.L. Wang, D. Shasha, and C.H. Wu. 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*, vol. 31, pp. 468-475, 2001.
- [6] Y. Matsuyama, N. Katsumata, Y. Suzuki and S. Imahara. The α -ICA algorithm. *Proc. Int. Workshop on Independent Component Analysis*, pp. 297-302, 2000.
- [7] Y. Matsuyama, N. Katsumata and R. Kawamura. Independent component analysis minimizing convex divergence. *Lecture Notes in Computer Science*, No. 2714, pp. 27-34, Springer-Verlag, 2003.
- [8] I. Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Math Hungarica*, vol. 2, pp. 299-318, 1967.
- [9] Y. Suzuki, et al. (15 authors). Diverse transcriptional initiation revealed by fine, large-scale mapping of mRNA start sites. *EMBO reports*, vol. 2, pp. 388-393, 2001.
- [10] Y. Matsuyama, H. Kataoka, N. Katsumata and K. Shimoda. ICA photographic encoding gear: Image bases towards IPEG. *Proc. Int. Joint Conf. Neural Networks*, IEEE-INNS, 2004.
- [11] A.P. Dempster, N.M. Laird and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Statist. Soc., B*, vol. 39, pp. 1-38, 1977.
- [12] Y. Matsuyama. The α -EM algorithm: Surrogate likelihood maximization using α -logarithmic information measures. *IEEE Trans. on Information Theory*, vol. 49, pp. 692-706, 2003.