Evelyn M. Crowley
Epidemiology Data Center, University of Pittsburgh, Pittsburgh, PA 15261

Received 11 October 1999; accepted 13 June 2000

A Bayesian Method for Finding Regulatory Segments in DNA

Abstract: *A goal of the human genome project is to determine the entire sequence of DNA (3×10^9 base pairs) found in chromosomes. The massive amounts of data produced by this project require interpretation. A Bayesian model is developed for locating regulatory regions in a DNA sequence. Regulatory regions are areas of DNA to which specific proteins bind and control whether or not a gene is transcribed to produce templates for protein synthesis. Each human cell contains the same DNA sequence. Thus the particular function of different cells is determined by the genes that are transcribed in that cell. A Hidden Markov chain is used to model whether a small interval of the DNA is in a regulatory region or not. This can be regarded as a changepoint problem where the changepoints are the start of a regulatory or nonregulatory region. The data consists of protein-binding elements, which are short subsequences, or "words," in the DNA sequence. Although these words can occur anywhere in the sequence, a larger number are expected in regulatory regions. Therefore, regulatory regions are detected by locating clusters of words. For a particular DNA sequence, the model automatically selects those words that best predict regions of interest. Markov chain Monte Carlo methods are used to explore the posterior distribution of the Hidden Markov chain. The model is tested by means of simulations, and applied to several DNA sequences.*
 Biopoly 58: 165–174, 2001

Keywords: *Gibbs sampling; hidden Markov chain; Markov chain Monte Carlo; Metropolis-Hastings; regulatory region*

INTRODUCTION

The problem of detecting regulatory regions in the genome is considered. Every cell in an individual contains the same DNA. However, the different cells have very different functions. For example, blood cells produce different proteins than liver cells. Regulatory regions are the control regions that switch genes on and off, determining what proteins are produced by a cell.

DNA consists of a sequence of chemical structures (nucleotides), represented by the letters A, C, G, and T. It occurs naturally as a double helix, consisting of two strands, where A pairs with T and G pairs with C. Because of this complementarity, the second strand may be considered to be redundant, and the DNA may be represented by a single strand running in one direction, known as the "5′ to 3′ " direction. Regulatory proteins bind to the DNA, causing genes to be switched on or off.

Regulatory regions represent the DNA segments that contribute to the control of gene expression. Regulation is a very complex process involving proteins that bind to specific control elements in the DNA. In addition, control involves various types of regulatory regions. Consider, for example, the protein encoding

Correspondence to: Evelyn M. Crowley; email: crowley@edc.gsph.pitt.edu
Contract grant sponsor: National Science Foundation
Contract grant number: DMS-9303556

genes. These genes are transcribed by RNA polymerase II. The initiation of transcription is controlled by a DNA segment (basal promoter), which is in the immediate upstream of the transcription initiation site. Some basal promoters contain a sequence known as a TATA element but there are many genes whose control does not require this sequence (see, for example, Pugh and Tjian[1]). A second regulatory region (proximal promoter) is often present in the immediate upstream of the basal sequence. The genome also includes regulatory segments that are found far upstream or far downstream of a gene. These segments may include regions known as locus control regions. Some regulatory regions occur within a gene, often in noncoding sequences known as introns. These regulatory regions may include segments that are known as transcriptional enhancers. Thus, how to locate regulatory regions in DNA is a complex problem.

Most previous work has concentrated on trying to locate promoters. Prestridge[2] uses the density of words computed in promoter and nonpromoter regions for primates to come up with a promoter recognition profile for a sequence of interest. This is combined with the TATA box method of Bucher[3] to locate promoters in the sequence. Kondrakhin et al.[4] compute a measure of similarity between the sequence of interest and a sample of 472 promoters. If this measure is greater than a threshold value, it suggests a potential promoter.

The focus of the studies described here is on the regulatory regions that function through interactions with proteins that bind to DNA sequences known as control elements. These elements occur in regions that could correspond to proximal promoters, locus control regions, and enhancers. The model presented in this report does not address how to locate basal promoters because these sequences are considered to be a different type of regulatory region. The piece of the DNA to which the regulatory proteins bind is called a protein binding element. It consists of a short (5–20 base pairs) series of nucleotides, or *word*, in the DNA, e.g., AGAACA. These words occur frequently in the genome but are expected to occur more often in regulatory regions.[5] Hence, the idea is to look for clusters of words in the DNA. This is done by means of a Hidden Markov model. The parameters of the model are estimated using Markov chain Monte Carlo (MCMC).

To locate the regulatory regions, a finite collection of words or catalogue is needed. In studies of regulatory elements, researchers often use the transcription factor database (TFD) constructed by Ghosh.[6] While this database might be a good resource for finding potential protein binding sites in DNA, it is not suitable for statistical work. The problems are extensive, including redundancy. In addition, TFD includes a concoction of sequences: from alternating purine (R = A or G) pyrimidine (Y = T or C) found in Z-DNA, to binding sites used for transcription by RNA polymerase III. The problem here is that RY repeats represent a structural motif, and not necessarily a specific control element in DNA. Control sites for RNA polymerase III are also not relevant for regulation of transcription of protein-encoding genes since these genes are exclusively transcribed by RNA polymerase II. To resolve the problem of redundancy, Prestridge[2] extracted only TFD site numbers rather than transcription factor name. However, a later study noted that this approach does not resolve the problem of redundancy.[7] In addition, sequences that are extracted from TFD do not necessarily define specific control elements (words) in DNA.

To avoid these problems, Bina (unpublished data) compiled a different catalogue, and this is the one used in the method described here. A control element is defined as a binding site for specific or related transcription factors for genes transcribed by RNA polymerase II (see, for example, Crowley, Roeder, and Bina[8]). Experimental data have shown that the control elements are usually short (5–12 base pairs). Somewhat longer elements provide sites for proteins that have more than one DNA binding domain (i.e., the PAX family). Note that TFD includes a large number of relatively long sequences (see for example, Figure 3 in the article by Prestridge[2]). These sequences do not define a specific word. They often represent regions protected in DNase I footprinting experiments. These footprints are often longer than the actual size of a site and in some cases they may also include more than one protein-binding element.

To resolve the general problem of redundancy, when possible, the control elements are defined and classified according to the structure of the DNA binding domain of the proteins.[8] In some cases, a consensus core is defined to represent the binding site of proteins that interact with related sequences. Often, these proteins correspond to products of evolutionarily related genes, and thus the similarity of their binding sites may contribute to the redundancy problem. In addition, the entries of the catalogue are closely checked against the actual experimental data for accuracy and update. Experimental data represent the results of published work including electrophoretic mobility-shift assays, methylation interference assays, and mutational analysis. Updating is also needed since new data often correct and improve published work. In this context, a database is never as definitive as it should be, and thus it should be con-

sidered as an evolving process. However, for a given set of analyses, and in particular those of this paper, the catalogue does not change.

This paper describes the statistical aspects of the method for finding regulatory regions in a DNA sequence. Applications of the method may be found in Crowley, Roeder, and Bina.[8] The layout of the paper is as follows. First, the Hidden Markov model and the model used for the distribution of the words are described. The priors used are also discussed. Implementation of the MCMC algorithm is next described. This is followed by a simulation study and then the method is applied to some DNA sequences. Finally, there is a discussion of the method.

THE MODEL

The DNA sequence of interest is divided into I intervals of length Δ, where Δ is the length of the longest word that occurs in the sequence. A word is in an interval if its center (which is defined to be the middle nucleotide if the word has an odd number of nucleotides, and the upper middle nucleotide if the word has an even number of nucleotides) is in the interval. Hence, if a word overlaps two intervals, it is chosen to be in only one of the intervals, depending on where its center lies. As the model is looking for clusters of words, it makes little difference which of the two intervals the word is in.

Let Y_i represent the state at the ith interval, with $Y_i = 1$ if i is in a regulatory region and $Y_i = 0$ otherwise. Assume that $Y_1, Y_2, \ldots, Y_I$ is a Hidden Markov chain, with transition probabilities

$$Pr(Y_{i+1} = s|Y_i = r) = \Omega_{rs},\ r \in \{0,1\},\ s \in \{0,1\}$$

with

$$\Omega = \begin{bmatrix} 1-\lambda & \lambda \\ \tau & 1-\tau \end{bmatrix}$$

The lengths of regulatory and nonregulatory regions have a Geometric distribution with expected values $1/\tau$ and $1/\lambda$, respectively. The Hidden Markov model gives a reasonable approximation to the behaviour of the DNA. As will be show later, it works well for this problem. It is already a computationally intensive model. The model works well enough that a further increase in computational burden from a more complicated model seems unjustified.

The data consist of the J words in the DNA sequence under consideration. Let

$$X_{ij} = \begin{cases} 1, & \text{if a word of the } j\text{th type occurs in the } i\text{th interval} \\ 0, & \text{otherwise} \end{cases}$$

Note that there is a possibility that a word could occur twice in an interval. This does not happen often and by this definition is treated the same as if it had occurred once. As this can only happen with the shorter words, which are more common, it has little effect on the results.

The model is first described, assuming that all words in the catalogue are good predictors of being in a regulatory region for every dataset. Let θ_r be the probability that any of the words occurs when $Y_i = r$. It is assumed that words occur more frequently in regulatory regions than in nonregulatory regions so that $\theta_0 < \theta_1$. Some words occur more often than others. Given that a word occurs, let c_{jr} be the probability that it is of type j when $Y_i = r$. Let

$$X_{ij}|Y_i = r \sim \text{Bernoulli}\ (c_{jr} \cdot \theta_r) \tag{1}$$

where $i = 1, \ldots, I$, $j = 1, \ldots, J$, and $r = 0, 1$. The X_{ij}'s are assumed to be conditionally independent, given the Y_i's.

Note that if a word occurs in an interval, it can occur at Δ different positions. Some of these positions may not allow the other words to occur in the interval, thus violating the assumption of conditional independence above. However, the more restrictive words are the longer ones and they occur less often. So, the assumption of conditional independence should hold approximately.

The c_{jr}'s are treated as constants. Let $n_j = \sum_i X_{ij}$ be the number of words of type j and let $w_j = n_j/\sum_{j'} n_{j'}$ be the relative proportion of words that are of type j, $j = 1, \ldots, J$. In nonregulatory regions, let $c_{j0} = w_j$. In regulatory regions, rare words are expected to have increased probability. So c_{j1} needs to be chosen so that c_{j1}/c_{j0} is a decreasing function of w_j. To achieve this, take $c_{j1} = w_j \ln(1/w_j)$. It would also be possible to treat the c_{jr}'s as parameters and estimate them.

The priors for the parameters λ, τ, θ_0, and θ_1 are now described. Beta priors are put on λ and τ. In the model, high prior probability is given to regulatory regions of length 200–600 base pairs occuring 0–4 times every 5000 base pairs, as typical regulatory regions fall in this range. The prior parameters are chosen to approximately achieve this. For example, when Δ is equal to 13 base pairs, a Beta(1.3,100) prior is used for λ and a Beta(5.5,100) prior is used for τ. A

uniform prior is put on θ_0 and θ_1, with the restriction $\theta_0 < \theta_1$.

In practice, it is not known which words in the catalogue are good predictors for a particular dataset. The following approach automatically chooses the words that best predict for a particular sequence. Words that predict well are called "predictive" words. Let

$$z_j = \begin{cases} 1, & \text{if the } j\text{th word is predictive} \\ 0, & \text{if the } j\text{th word is nonpredictive} \end{cases}$$

Equation (1) is replaced by

$$X_{ij}|Y_i = r,\ z_j \sim \begin{cases} \text{Bernoulli}(c_{jr} \cdot \theta_r), & z_j = 1 \\ \text{Bernoulli}(c_{j0} \cdot \theta_0), & z_j = 0 \end{cases}$$

That is, for nonpredictive words, regulatory regions are treated the same as nonregulatory regions. Let z_j have a Bernoulli(0.5) prior distribution. This method both adjusts for the nonpredictive words and gives us information about which words are good predictors for a particular sequence.

IMPLEMENTATION

Let $X_i = (X_{i1}, X_{i2}, \ldots, X_{iJ})'$ and let $\mathbf{X} = (X_1, X_2, \ldots, X_I)$. Also, let $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_I)$ and $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_J)$. The posterior probability that $Y_i = 1$, $i = 1, \ldots, I$ and that $Z_j = 1$, $j = 1, \ldots, J$ need to be estimated. This is done using Gibbs sampling and the Metropolis–Hastings Algorithm.[9,10] Start with some initial estimate of the parameters in the model: $\mathbf{Y}^0$, λ^0, τ^0, θ_0^0, θ_1^0, $\mathbf{Z}^0$. First, the general steps in the Markov chain Monte Carlo algorithm are described and then more details are given below. For $k = 1, \ldots, N$:

1. Sample $Y_i^k|Y_1^k, \ldots, Y_{i-1}^k, Y_{i+1}^{k-1}, \ldots, Y_I^{k-1}, \mathbf{X}, \lambda^{k-1}, \tau^{k-1}, \theta_0^{k-1}, \theta_1^{k-1}, \mathbf{Z}^{k-1}$ for $i = 1, \ldots, I$, using Gibbs sampling.
2. Sample $\lambda^k, \tau^k|\mathbf{Y}^k$ from their posteriors.
3. Sample $\theta_0^k, \theta_1^k|\mathbf{Y}^k, \mathbf{X}, \mathbf{Z}^{k-1}$ using Metropolis–Hastings.
4. Sample $Z_j^k|\mathbf{Y}^k, \mathbf{X}, \theta_0^k, \theta_1^k$, for $j = 1, \ldots, J$, from their posteriors.

Ignoring the first n_0 runs, the posterior probability that $Y_i = 1$ is estimated by $1/(N - n_0)\sum_{k=n_0+1}^{N} Y_i^k$ and that $Z_j = 1$ by $1/(N - n_0)\sum_{k=n_0+1}^{N} Z_j^k$.

Step 1. The posterior distribution of each Y_i needs to be computed. This is denoted by $\pi(Y_i|Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_I, \mathbf{X}, \lambda, \tau, \theta_0, \theta_1, \mathbf{Z})$. This is done using an algorithm for hidden Markov models. This algorithm is described in the context of DNA sequences by Churchill[11] and by Dupuis.[12] For convenience, some of the parameters are suppressed. Let $X^i = (X_1, X_2, \ldots, X_i)$. The hidden Markov chain algorithm gives $\pi(y_i|X^{i-1})$, $\pi(y_i|X^i)$ and $\pi(y_i|\mathbf{X})$. Hence,

$$\begin{aligned} &\pi(y_i|y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_I, \mathbf{X}) \\ &\quad = \pi(y_i|y_{i-1}, y_{i+1}, \mathbf{X}) \\ &\quad = \frac{\pi(y_i|\mathbf{X})\,\pi(y_{i-1}|y_i, \mathbf{X})\,\pi(y_{i+1}|y_i, \mathbf{X})}{\pi(y_{i-1}|\mathbf{X})\,\pi(y_{i+1}|y_{i-1}, \mathbf{X})} \end{aligned}$$

The first term in the numerator and the denominator are given directly by the hidden Markov chain algorithm, and the other conditional distributions are easily computed. For example,

$$\pi(y_{i-1}|y_i, \mathbf{X}) = \frac{\pi(y_{i-1}|X^{i-1})\,\pi(y_i|y_{i-1})}{\pi(y_i|X^{i-1})}$$

Step 2. Suppose that the prior for λ is Beta(a_0, b_0) and that the prior for τ is Beta(a_1, b_1). Let $S_{rt} = \sum_{i=1}^{I-1}\{Y_i = r, Y_{i+1} = t\}$, $r = 0, 1$, $t = 0, 1$. Then the posterior distribution of λ is approximately Beta($a_0 + S_{01}$, $b_0 + S_{00}$) and the posterior distribution of τ is approximately Beta($a_1 + S_{10}$, $b_1 + S_{11}$).

Step 3. The posterior distribution of θ_0 and θ_1, $\pi(\theta_0, \theta_1|\mathbf{Y}, \mathbf{X}, \mathbf{Z})$, $0 \le \theta_0 < \theta_1 \le 1$, needs to be sampled from. Let $\delta_r = \log(\theta_r/(1 - \theta_r))$, $r = 0,1$. Then,

$$\begin{aligned} \pi(\delta_0, \delta_1|\mathbf{Y}, \mathbf{X}, \mathbf{Z}) &\propto \pi_0(\delta_0|\mathbf{Y}, \mathbf{X}, \mathbf{Z}) \\ &\quad \times \pi_1(\delta_1|\mathbf{Y}, \mathbf{X}, \mathbf{Z}),\ -\infty < \delta_0 < \delta_1 < \infty \end{aligned}$$

Metropolis Hastings is performed on δ_0 and δ_1, instead of on θ_0 and θ_1. The candidate generating densities are $g_0(\delta_0) = N(\delta_0, \sigma_0^2)$ and $g_1(\delta_1) = N(\delta_1, \sigma_1^2)$, $\delta_1 > \delta_0$. The probability of a move from a to b is

$$\alpha_r(a, b) = \begin{cases} \min\left(1, \dfrac{\pi_r(b|\mathbf{Y}, \mathbf{X}, \mathbf{Z})}{\pi_r(a|\mathbf{Y}, \mathbf{X}, \mathbf{Z})}\right) & \text{if } \pi_r(a|\mathbf{Y}, \mathbf{X}, \mathbf{Z}) > 0 \\ 1 & \text{otherwise} \end{cases}$$

For $r = 0, 1$, start with δ_r^0. The kth step is

- Generate δ_r^* from $g_r(\delta_r)$ and U_r from $U(0,1)$
- Let

$$\delta_r^k = \begin{cases} \delta_r^* & \text{if } U_r \leq \alpha_r(\delta_r^{k-1}, \delta_r^*) \\ \delta_r^{k-1} & \text{otherwise} \end{cases}$$

Transform to get θ_r^k.

Step 4. $P(Z_j = r|\mathbf{Y}, \mathbf{X}, \theta_0, \theta_1)$, $r = 0,1$ needs to be computed. This is straightforward.

Run lengths of N equal to 10,000 or 50,000 are used, and diagnostic plots are used to assess convergence. For δ_0 and δ_1, δ_0^k and δ_1^k are plotted against k, for $k = 1, \ldots, N$. To check convergence for the Y_i's, $1/100 \sum_{1=w+1}^{w+100} Y_i^k$ is plotted against k. This is done for $w = 250\ j$ where $j = 0, 1, 2, \ldots,$ int$(I/250)$.

SIMULATIONS

A simulation study is performed to examine how well the method picks out regulatory regions and detects nonpredictive words. In each of the four scenarios considered, the sequence is of length approximately 30,000 base pairs. There are 40 words, of which 5 are nonpredictive. The 40 words consist of 10 low frequency words, 20 medium frequency words and 10 high frequency words. Low, medium, and high frequency mean that approximately 5, 50, and 200 of these words, respectively, would be expected in the sequence. This is similar to some of the DNA sequences considered in terms of the number of low, medium, and high frequency words seen. For each scenario, the posterior probability that each interval is a regulatory region is plotted against its location in the sequence. It would be optimal if this probability is close to one in a regulatory region and close to zero in a nonregulatory region. The posterior probability that a word is predictive, and its standard error, are also computed, for each of the 40 words. The standard errors are computed using the method of batch means. This was done for batchlengths of 100, 150, 200, 300, and 600. For each word, the standard error associated with the smallest batchlength that gave a serial correlation between -0.1 and 0.1 is given. For each scenario, the results for one simulation are shown. Other simulations follow a similar pattern. The diagnostic plots showed that a run length of $N = 10{,}000$ was sufficient for convergence in the simulations.

Data Generated from the Exact Model

Here, the five nonpredictive words are medium frequency words and $\theta_0 = 0.1$ and $\theta_1 = 0.9$. The first scenario considered is one that is very probable under the prior. There is one regulatory region every 6000 base pairs. Each of the (five) regulatory regions is of length 300 base pairs. The method correctly detected the five regulatory regions (the probabilities were all close to one)—see Figure 1(a). The posterior probability that a word is predictive, for each of the 40 words, is contained in Table I. The nonpredictive words tend to have a low probability of being predictive, while the predictive words have a high probability of being predictive. The probability of being predictive is higher for the medium and high frequency words than for the low frequency words. Note that the standard errors for these posterior probabilities are very small for all of the words.

Next, the effect of varying the length of the regulatory regions is considered. The above scenario is changed by considering regulatory regions of length 120 and 600 base pairs, instead of 300 base pairs. The shorter and longer regulatory regions are clearly detected—see Figure 1(b) and 1(c), respectively. Also looked at is what happens when the number of regulatory regions per 6000 base pairs is increased (and also this number differs across the sequence). In this scenario, there were five regulatory regions in the first 6000 base pairs and none in the rest of the sequence. The method clearly picks out these five regulatory regions—see Figure 1(d). Note that the probability that a word is predictive follows the same pattern for the last three scenarios as it does for the first.

Nonpredictive Words

The effect of varying the types of nonpredictive words under the four scenarios above is examined. The posterior probability of a regulatory region for the low frequency and high frequency words for each of the four scenarios looks almost identical to that for the medium frequency words given in Figure 1. However, there is a difference in the posterior probability that a word is predictive. For nonpredictive words, this probability decreases and for predictive words, this probability increases as the nonpredictive words occur with higher frequency. So the method is better at finding and adjusting for the higher frequency nonpredictive words than for the low frequency ones.

Robustness and Sensitivity

Now consider what happens when the data is generated without the assumption that rare words have increased probability in regulatory regions. Let c_{j1}

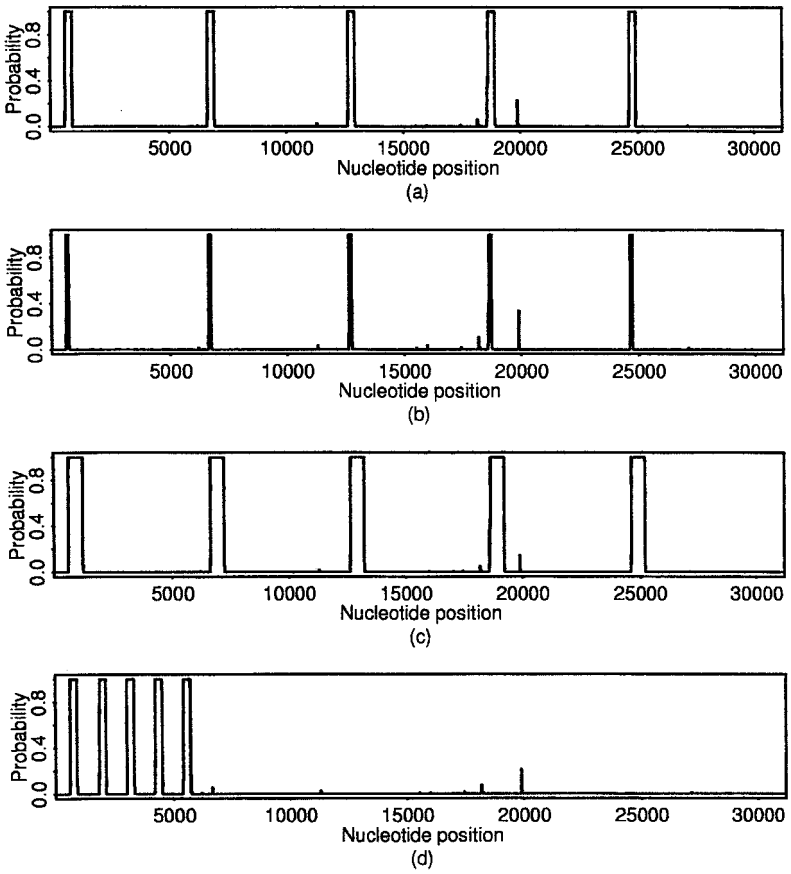

FIGURE 1 Posterior probability of a regulatory region for data generated from the model with $\theta_0 = 0.1$, $\theta_1 = 0.9$, and five medium frequency nonpredictive words. (a) One regulatory region of length 300 base pairs every 6000 base pairs; (b) one regulatory region of length 120 base pairs every 6000 base pairs; (c) one regulatory region of length 600 base pairs every 6000 base pairs; (d) five regulatory regions of length 300 in the first 6000 base pairs and none in the rest of the sequence.

$= w_j$ [instead of $w_j \ln(1/w_j)$]. The same four scenarios are used and there are five medium frequency nonpredictive words with $\theta_0 = 0.1$ and $\theta_1 = 0.9$.

The posterior probabilities of a regulatory region are given in Figure 2. This looks very similar to Figure 1, except for the case of short regulatory re-

Table I The Posterior Probability that a Word is Predictive for Data Generated from the Model[a]

Word	PWP	Word	PWP	Word	PWP	Word	PWP
1	.997 (1)	11	.005 (1)	21	1.000 (0)	31	1.000 (0)
2	—	12	.012 (1)	22	1.000 (0)	32	1.000 (0)
3	.967 (2)	13	.045 (2)	23	1.000 (0)	33	1.000 (0)
4	1.000 (0)	14	.023 (1)	24	.998 (1)	34	1.000 (0)
5	—	15	.140 (4)	25	1.000 (0)	35	1.000 (0)
6	.945 (2)	16	1.000 (0)	26	1.000 (0)	36	1.000 (0)
7	.992 (1)	17	1.000 (0)	27	1.000 (0)	37	1.000 (0)
8	—	18	1.000 (0)	28	1.000 (0)	38	1.000 (0)
9	—	19	1.000 (0)	29	1.000 (0)	39	1.000 (0)
10	.229 (4)	20	1.000 (0)	30	1.000 (0)	40	1.000 (0)

[a] $\theta_0 = 0.1$ and $\theta_1 = 0.9$. Words 1–10 are low frequency, words 11–30 are medium frequency, and words 31–40 are high frequency. Words 11–15 are nonpredictive. PWP is the posterior probability that a word is predictive. Words 2, 5, 8, and 9 did not occur in this simulation. Values in parentheses multiplied by 10^{-3} are the standard errors of the posterior probabilities.

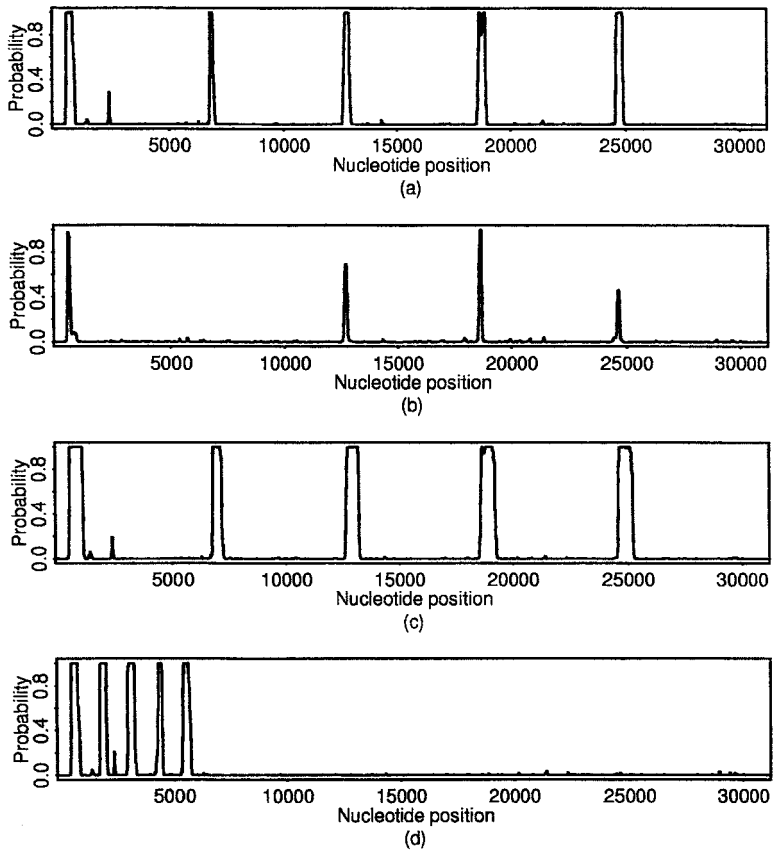

FIGURE 2 Posterior probability of a regulatory region for data generated without the assumption that rare words have increased probability in regulatory regions and with $\theta_0 = 0.1$, $\theta_1 = 0.9$, and five medium frequency nonpredictive words. (a) One regulatory region of length 300 base pairs every 6000 base pairs; (b) one regulatory region of length 120 base pairs every 6000 base pairs; (c) one regulatory region of length 600 base pairs every 6000 base pairs; (d) five regulatory regions of length 300 in the first 6000 base pairs and none in the rest of the sequence.

gions. Here, one of the regulatory regions is missed completely and two of the others have probabilities closer to 0.5 than to one. There is also a difference in the posterior probabilities that a word is predictive—see Table II. These are lower for the predictive words and higher for the nonpredictive words. The standard errors for the posterior probabilities are small here also. In summary, if the data does not agree with the assumption that rare words have increased probability in regulatory regions, the method still does well at detecting regulatory regions (except for shorter ones) but won't do as well at finding and adjusting for nonpredictive words.

Also consider what happens when θ_0 and θ_1 get closer. The same four scenarios are used and there are five medium frequency nonpredictive words with $c_{j1} = w_j \ln(1/w_j)$. First, consider $\theta_0 = 0.25$ and $\theta_1 = 0.75$. The results obtained here are very similar to the results presented in Figure 1 and Table I. Also, consider $\theta_0 = 0.4$ and $\theta_1 = 0.6$. The results obtained here are very similar to the results presented in Figure 2 and Table II. So, unless θ_0 and θ_1 are very close together, the method does well at both detecting regulatory regions and finding and adjusting for nonpredictive words. If they are very close together, it still does well at detecting regulatory regions.

EXAMPLES

Three examples are considered. The regulatory regions are known for the first two examples but not for the third. For each example, the diagnostic plots showed that a run length of $N = 50{,}000$ sufficed for convergence. First, the predictions made by the model for two datasets corresponding to sequences that contain known regulatory sequences are examined. Subsequently, a human genomic DNA whose regulatory sequences have not been experimentally defined is analyzed.

Table II The Posterior Probability that a Word is Predictive for Data Generated Without the Assumption that Rare Words Have Increased Probability in Regulatory Regions[a]

Word	PWP	Word	PWP	Word	PWP	Word	PWP
1	—	11	.172 (4)	21	1.000 (0)	31	.996 (1)
2	—	12	.038 (2)	22	.995 (1)	32	1.000 (0)
3	—	13	.499 (8)	23	.090 (3)	33	.999 (0)
4	.853 (4)	14	.053 (3)	24	.976 (2)	34	1.000 (0)
5	—	15	.068 (4)	25	.934 (3)	35	1.000 (0)
6	—	16	.743 (16)	26	1.000 (0)	36	.929 (5)
7	—	17	.328 (23)	27	.776 (5)	37	1.000 (0)
8	—	18	.505 (20)	28	.126 (3)	38	1.000 (0)
9	.348 (6)	19	.998 (0)	29	.998 (1)	39	1.000 (0)
10	—	20	.985 (3)	30	.674 (5)	40	1.000 (0)

[a] $\theta_0 = 0.1$ and $\theta_1 = 0.9$. Words 1–10 are low frequency, words 11–30 are medium frequency, and words 31–40 are high frequency. Words 11–15 are nonpredictive. PWP is the posterior probability that a word is predictive. Words 1–3, 5–8, and 10 did not occur in this simulation. Values in parentheses multiplied by 10^{-3} are the standard errors of the posterior probabilities.

The first dataset is Moloney Murine Sarcoma virus (Mo-MuSV). The regulatory signals for this virus are contained in the long terminal repeats at the two ends of the sequence. Two regulatory regions (probabilities approximately one) are predicted by the model—see Figure 3(a). These correspond to the two long terminal repeats. There is a smaller peak (probability approximately 0.25) around 1820 base pairs. In summary, there are two regulatory regions in this sequence and both are clearly detected. There is one other small peak, which does not correspond to a regulatory region.

The second dataset is an artificial construct called pLNCX. This construct is selected as a prototype of a set of vectors designed by Miller and Rosman[13] for gene transfer experiments. These vectors are mosaic in structure. They include regulatory segments derived from viral genomes and bacterial-derived sequences with desired biological properties. This dataset is useful because it shows that the model can find regulatory regions, regardless of their location (that is, they do not have to be close to the transcription initiation site). The sequence selected for analysis (pLNCX) contains 6620 base pairs: position 145–733 corresponds to the Mo-MuSV 5′ long terminal repeat, 3666–4259 to the Moloney murine leukemia virus (Mo-MuLV) 3′ long terminal repeat, and 2800–3617 to a regulatory segment (immediate early promoter) from human cytomegalovirus (CMV). The prediction for this sequence is shown in Figure 3(b). The regu-

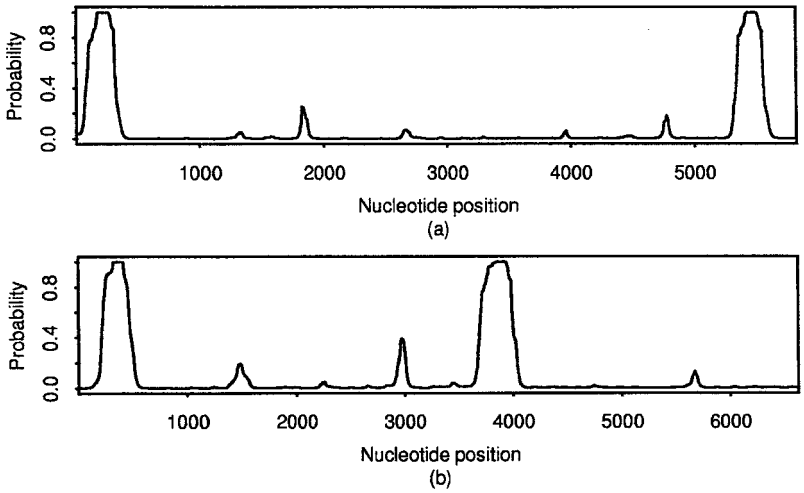

FIGURE 3 Posterior probability of a regulatory region for two DNA sequences. (a) Mo-MuSV DNA (GenBank accession no. J02266); (b) pLNCX DNA (GenBank accession no. M28247).

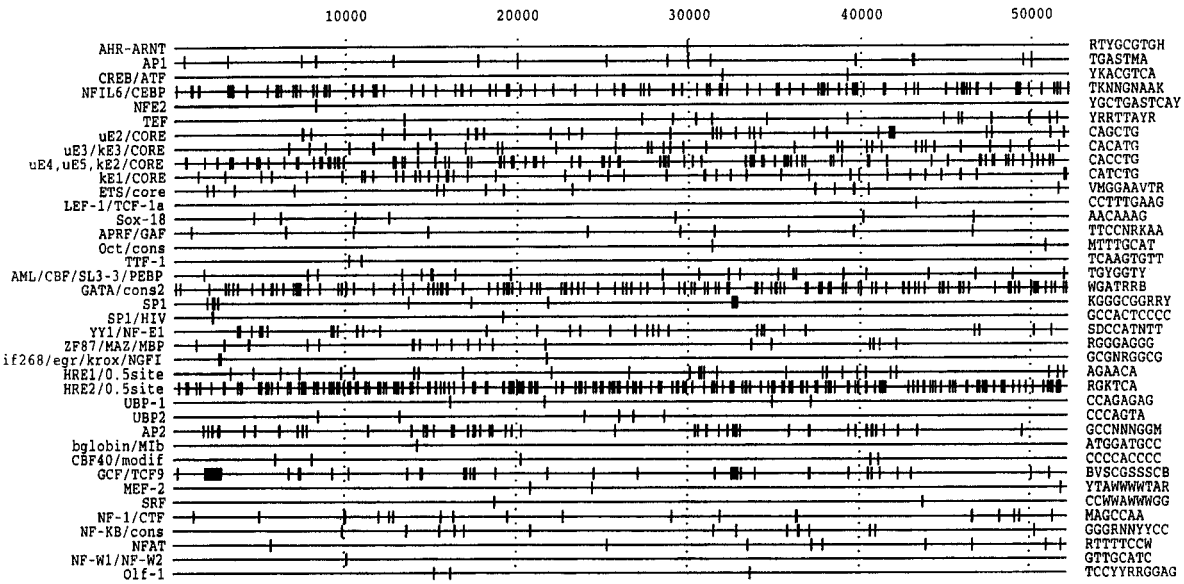

FIGURE 4 Distribution of words in HSG6PDGEN DNA (GenBank accession no. X55448 and no. Z29527). The words are shown on separate lines that define the DNA sequence. The names of the words are on the left and the corresponding sequences are provided on the right. The ambiguity codes in the sequences follow the standard notations: Y = T/C; R = A/G; W = A/T; S = G/C; K = G/T; M = A/C; B = C/G/T; D = A/G/T; H = A/C/T; V = A/C/G; N = A/G/C/T.

latory regions corresponding to the Mo-MuSV and Mo-MuLV viruses are clearly detected (probabilities approximately one). The regulatory segment derived from CMV is seen as a smaller peak (probability approximately 0.39) at around 2970 base pairs. In summary, there are three regulatory regions in this sequence and two of these are clearly found. The third regulatory region is also detected but with considerably lower posterior probability.

As an example of human genomic DNA, the nucleotide sequence of HSG6PDGEN is analyzed. This locus has previously been sequenced to examine the molecular basis of predisposition to hemolytic anemia and how this predisposition can provide partial protection against malarial infection.[14] This locus includes 52,173 base pairs. The locus contains two genes: G6PD and 2-19. Each gene is defined by transcription initiation sites, the protein coding regions of the genes and the corresponding intervening sequences. Note that the regulatory regions of this locus have not been reported. The distribution of the words in the catalogue that occur in HSG6PDGEN DNA is given in Figure 4. The name of each word and its corresponding sequence are given with the distribution. In the analysis, the site for GCF/TCF9 (which is listed in Figure 4) is not used, as there are clusters of occurrences of this word in some sequences, and these clusters obscure the predictions made by the model.

The model predicts five probable regulatory regions in the HSG6PDGEN locus (Figure 5). The first region precedes the transcription initiation site and includes exon 1 in the G6PD gene. The second regulatory region is in the immediate upstream of the transcription initiation site of the 2-19 gene. Both of these regions have probabilities approximately one. The other three probable regions have smaller probabilities and occur in the 2-19 gene. The first of these is contained in intron 2 and the second is contained in intron 3. The third region begins in intron 3 and extends to include exon 4, intron 4, exon 5 and part of intron 5 of the 2-19 gene. Note that these two exons are relatively short (124 and 59 base pairs, respectively). Currently, it is not known whether regulatory regions can extend into exons. Regardless of this possibility, the model implicates part of intron 3,

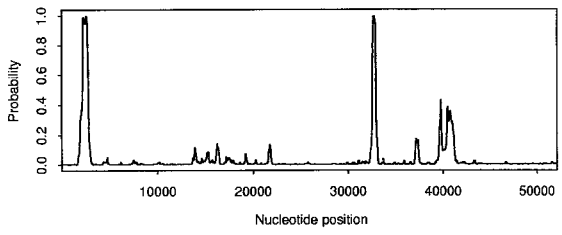

FIGURE 5 Posterior probability of a regulatory region for HSG6PDGEN DNA (GenBank accession no. X55448 and no. Z29527).

intron 4, and part of intron 5, in functioning as regulatory segments. This prediction appears intriguing since regulatory regions localized in intronic regions are difficult to identify directly by experimental strategies. Thus, the predictions made by the model locate segments that can be readily tested by experimental biologists.

DISCUSSION

The results of the simulation study show that the method works well. It detects the regulatory regions and the nonpredictive words. It is particularly good at finding and adjusting for the more high frequency nonpredictive words. When the data deviates from the assumptions, the regulatory regions are still detected, unless they are very short. For example, regulatory regions of length 120 base pairs are not detected. The method also detects the regulatory regions if the probability that any of the words occur in a regulatory region gets close to this probability in a nonregulatory region. The method is applied to DNA sequences, some of which contain known regulatory regions. It correctly identifies the known regulatory regions.

Other methods of finding regulatory regions have concentrated on finding promoters. For example, Prestridge's approach[2] relies on the TATA element localized in basal promoters and hence it might only be suitable for locating a subset of regulatory regions. Kondrakhin et al.[4] also focus on promoters. The Bayesian method described here can also find locus control regions and enhancers.

This method also differs from those of Prestridge and Kondrakhin et al. in that it looks for clusters of words, not at how similar word occurrence is to a promoter sample. It gives a probability distribution for the location of regulatory regions, and for a word being a good predictor in a particular dataset. It also has the advantage of avoiding overprediction of regulatory regions because of the way the catalogue is obtained. However, it does tend to underpredict regulatory regions. This should become less of a problem as more experimental data becomes available to update the catalogue. Future work on this method will include combining the method with methods that work well at locating basal promoters and looking at ways of speeding up the algorithm.

The software to implement the method is available from the author upon request.

This research was partially supported by National Science Foundation Grant DMS-9303556. The author is grateful to Minou Bina for suggesting this problem and obtaining the data, and for many helpful discussions. The author also thanks Kathryn Roeder for suggesting the approach.

REFERENCES

1. Pugh, B. F.; Tjian, R. Genes Dev 1991, 5, 1935–1945.
2. Prestridge, D. S. J Mol Biol 1995, 249, 923–932.
3. Bucher, P. J Mol Biol 1990, 212, 563–578.
4. Kondrakhin, Y. V.; Kel, A. E.; Kolchanov, N. A.; Romashchenko, A. G.; Milanesi, L. Comput Appl Biosci 1995, 11, 477–488.
5. Ambrose, C.; Bina, M. J Mol Biol 1990, 216, 485–490.
6. Ghosh, D. Nucleic Acids Res 1993, 21, 3117–3118.
7. Chen, Q. K.; Hertz, G. Z.; Stormo, G. D. Comput Appl Biosci 1997, 13, 29–35.
8. Crowley, E. M.; Roeder, K.; Bina, M. J Mol Biol 1997, 268, 8–14.
9. Gilks, W. R.; Clayton, D. G.; Spiegelhalter, D. J.; Best, N. G.; McNeil, A. J.; Sharples, L. D.; Kirby, A. J. J Royal Stat Soc Ser B 1993, 55, 39–52.
10. Smith, A. F. M.; Roberts, G. O. J Royal Stat Soc Ser B 1993, 55, 3–23.
11. Churchill, G. A. Bull Math Biol 1989, 41, 164–171.
12. Dupuis, J. Technical Report No. 1 1994, Stanford University, Department of Statistics.
13. Miller, A. D.; Rosman, G. J. BioTechniques 1989, 7, 980–990.
14. Chen, E. Y.; Cheng, A.; Lee, A.; Kuang, W. J.; Hillier, L.; Green, P.; Schlessinger, D.; Ciccodicola, A.; D'Urso, M. Genomics 1991, 10, 792–800.