

Discovery of Conserved Sequence Patterns Using a Stochastic Dictionary Model

Mayetri GUPTA and Jun S. LIU

Detection of unknown patterns from a randomly generated sequence of observations is a problem arising in fields ranging from signal processing to computational biology. Here we focus on the discovery of short recurring patterns (called *motifs*) in DNA sequences that represent binding sites for certain proteins in the process of gene regulation. What makes this a difficult problem is that these patterns can vary stochastically. We describe a novel data augmentation strategy for detecting such patterns in biological sequences based on an extension of a “dictionary” model. In this approach, we treat conserved patterns and individual nucleotides as stochastic words generated according to probability weight matrices and the observed sequences generated by concatenations of these words. By using a missing-data approach to find these patterns, we also address other related problems, including determining widths of patterns, finding multiple motifs, handling low-complexity regions, and finding patterns with insertions and deletions. The issue of selecting appropriate models is also discussed. However, the flexibility of this model is also accompanied by a high degree of computational complexity. We demonstrate how dynamic programming-like recursions can be used to improve computational efficiency.

KEY WORDS: Data augmentation; Gene regulation; Missing data; Transcription factor binding site.

1. INTRODUCTION

Genome sequencing projects have led to a rapid growth of publicly available databases of genome sequences for DNA, RNA, and proteins. A major challenge in the “post genome” era is to develop an understanding of gene regulatory networks within the cells of organisms. One of the first important steps in this process is to identify short repetitive patterns (about 7–30 nucleotides long), called sequence *motifs*, that occur in the vicinity of certain genes and may play a pivotal role in their regulation.

In *transcriptional* regulation, sequence signals upstream of each gene provide a target (the promoter region) for an enzyme complex, RNA polymerase, to bind and initiate the transcription of the gene into *messenger* RNA. Simultaneously, certain proteins called *transcription factors* (TFs) can bind to the promoter regions, either interfering with the action of RNA polymerase and inhibiting gene expression, or enhancing gene expression. TFs recognize sequence sites that give a favorable binding energy, which often translates into a sequence-specific pattern; binding sites thus tend to be relatively well-conserved in composition. For example, a crucial TF in *Escherichia coli*, the cyclic AMP receptor protein (CRP), recognizes a pattern of the form TGTGANNNNNT-CACA (with “N” denoting that any one of the four nucleotides may be present). But a substantial deviation from this pattern may sometimes be tolerated (see Fig. 4, Sec. 5.2). Laboratory assays, such as electrophoretic mobility shift and nuclease protection, have been developed to precisely locate TF-binding sites on gene-by-gene and site-by-site bases. *DNA footprinting* is a technique used to identify the location of binding sites by carrying out limited hydrolyses of the DNA with or without the protein and comparing the products. If the site is masked by the TF, then the pattern of fragments generated is different, and it is possible to work out the exact location of the site by a series of such experiments. The effectiveness of these approaches is limited, however, especially as the amount

of sequence to be analyzed increases. Computational methods that assume no prior knowledge of the pattern of the binding sites then become a necessary tool for aiding in their discovery.

Motifs in the vicinity of genes in DNA sequences often correspond to TF binding sites. The challenge of the problem is to simultaneously estimate the parameters of a model describing the position-specific nucleotide type preference for the TF (or TFs) and identify the locations of these binding sites, based only on a set of DNA sequences expected to contain multiple motif sites. Early statistical methods for finding motifs in biological sequences include a heuristic progressive alignment procedure (Stormo and Hartzell 1989), an EM algorithm (Dempster, Laird, and Rubin 1977) based on a missing-data formulation (Lawrence and Reilly 1990), and a Gibbs sampling algorithm (Lawrence et al. 1993). In both the EM and Gibbs approaches, starting positions of true motif sites were treated as “missing” components of the observed sequence data. Under the assumption that there was exactly one motif site per sequence, an iterative procedure was used to alternately refine the motif description (parameters) and sample sites in the sequences that could represent instances of the motif. Later generalizations that allow for a variable number of motif sites per sequence were the Gibbs motif sampler (Liu, Neuwald, and Lawrence 1995; Neuwald, Liu, and Lawrence 1995) and an EM algorithm using finite mixture models (Bailey and Elkan 1994).

Recently, Bussemaker, Li, and Siggia (2000) proposed a new motif-finding method, MobyDick, which treats the motifs as “words” used by nature and attempts to discover part of nature’s dictionary using a statistical method. In order to account for variations in the motif sites, we extend their dictionary model to allow for “stochastic” words, and introduce a data augmentation (DA) (Tanner and Wong 1987) procedure for finding such words. The stochastic dictionary framework allows us to find nonexact patterns of multiple types (having

Mayetri Gupta is a doctoral candidate and Jun S. Liu is Professor, Department of Statistics, Harvard University, Cambridge, MA 02138 (E-mail: gupta@stat.harvard.edu; jliu@stat.harvard.edu). The authors would like to thank Hao Li and Xiao-Li Meng, two anonymous referees, and the associate editor for helpful comments. This research is partly supported by the National Science Foundation grants DMS-0094613 and DMS-0204674.

one or more mismatches), patterns with insertions or deletions, and patterns of unknown width. We demonstrate how recursion methods can be used for greater efficiency in data augmentation.

Section 2 introduces the missing-data framework that forms the basis of the stochastic dictionary-based data augmentation (SDDA) algorithm. Section 3 describes extensions to the general model, tracking patterns with random insertions, and a MCMC approach for finding motifs when the width of the pattern is unknown. A connection is drawn to the Gibbs motif sampler, highlighting how operational distinctions may render the dictionary-based segmentation approach more efficient. Section 4 discusses the performance of several Bayesian model selection criteria in judging the appropriateness of the proposed motif alignment model. Section 5, presents empirical comparisons to illustrate the performance of the SDDA algorithm. For convenience, multiple instances of the same pattern in the data are referred to as motif sites, whereas different patterns are termed motifs.

2. A STOCHASTIC DICTIONARY FRAMEWORK FOR DATA AUGMENTATION

2.1 A Dictionary Model

Mathematically, the motif-finding problem can be stated as follows. Suppose that we have a set of n DNA sequences (composed of four letters, A, C, G, and T). We want to find a way to distinguish repetitive “signals” from the “background,” that is, to find distinct patterns that occur an excessive number of times in the set of sequences.

An important aspect of the dictionary model is to treat the observed sequence as being generated by concatenating “words” independently drawn from a dictionary according to a vector of “usage preference” probabilities. Indeed, if the single letters A, C, G, and T are valid words of the dictionary, then any sequence can be constructed in this way. Thus the motif-finding problem can be recast as discovering non-trivial words in the dictionary and their usage frequencies. However, even when all of the words in the dictionary are known, estimating the word usage frequencies by exhaustive enumeration is infeasible. For example, consider the unsegmented sentence “ofallthewordsinthisunsegmentedphrasetherearesomehidden.” It is easy for someone who knows English to pick out the most probable segmentation that breaks up this sequence into a meaningful set of patterns. But were a computer to do this, it would have to parse out the sentence into all possible combinations of patterns, then choose the one that satisfies a certain criterion. Two more complications arise in analyzing biological sequences: (1) we do not know nature’s “dictionary,” and (2) instances of the same “word” occurring at different places may not be exact replicas of the same pattern.

To handle the first difficulty, Bussemaker et al. (2000) adopted an iterative approach to build up a dictionary of words. Starting with a dictionary consisting of the single-letter words, they tested the concatenation of each pair of words for overrepresentation with respect to the background frequencies. The dictionary was then enlarged by adding the overrepresented words, and the process was repeated for the new set of

words in the dictionary. At each stage, maximum likelihood estimates of word usage frequencies were calculated using a Newton–Raphson procedure. This approach becomes difficult to generalize to the case of nonexact words, however. Also, even in the exact case, the assumption that longer words are made up of overrepresented shorter words may not generally be true.

2.2 The Stochastic Dictionary Model

To counter the second difficulty, we introduce the idea of a *stochastic* dictionary, which consists of a collection of “stochastic words” represented by the probabilistic word matrices (PWM). Each column of a PWM (Θ) gives the probabilities of finding each letter in that position of the corresponding stochastic word. For example, ACAGG and GCAGA may be two realizations, with probabilities .4328 and .0072, of the stochastic word with the PWM

$$\Theta = \begin{bmatrix} A & .85 & .07 & .80 & .02 & .12 \\ C & .05 & .78 & .07 & .01 & .01 \\ G & .10 & .05 & .12 & .96 & .85 \\ T & 0 & .10 & .01 & .01 & .02 \end{bmatrix}.$$

Suppose that the sequence data, \mathcal{S} , are generated by the concatenation of words from a dictionary of size D , $\mathcal{D} = \{M_1, M_2, \dots, M_D\}$ (including the single letters), sampled randomly according to a probability vector $\rho = (\rho(M_1), \dots, \rho(M_D))$. The likelihood of the data $\mathcal{S} = \{x_1 x_2 \dots x_n\}$ is

$$P(\mathcal{S} | \rho) = \sum_{\Pi} \prod_{i=1}^{N(\Pi)} \rho(\mathcal{S}[P_i]) = \sum_{\Pi} \prod_{j=1}^D \rho(M_j)^{N_{M_j}(\Pi)}, \quad (1)$$

where $\Pi = (P_1, \dots, P_k)$ is a partition of the sequence so that each part P_i corresponds to a word in the dictionary, $N(\Pi)$ is the total number of partitions in Π , and $N_{M_j}(\Pi)$ is the number of occurrences of word type M_j in the partition.

Evaluating the foregoing summation over all partitions Π by brute force would involve a prohibitive amount of computation, increasing exponentially with the size of the dataset. Instead, the summation can be achieved recursively. Let $L_i(\rho) = P(x_1 \dots x_i | \rho)$ be the likelihood for the partial sequence $\mathcal{S}_{[1:i]}$. Then

$$L_i(\rho) = \sum_{j=1}^D P(\mathcal{S}_{[i-w_j+1:i]} | \rho) L_{i-w_j}(\rho), \quad (2)$$

where w_j ($j = 1, \dots, D$) denotes the word lengths.

2.3 A Data Augmentation Approach

For operational convenience, we assume that the first q ($q < D$) words in the dictionary are the single-letter ones (for DNA sequences, $q = 4$). Let $\rho = (\rho_1, \dots, \rho_D)$ denote the word usage probabilities for the set of all words (letters and non-single-letter words) in the dictionary. If the partition $\Pi = (P_1, \dots, P_k)$ of the sequence into words were known, then the resulting distribution of counts of words $\mathbf{N} = (N_1, \dots, N_D)^T$ would be multinomial, characterized by the probability vector ρ . For a model with $D - q$ motifs of

widths w_{q+1}, \dots, w_D , we denote the $D - q$ motif matrices by $\{\Theta_{q+1}, \dots, \Theta_D\} = \Theta^{(D)}$. If the k th word is of width w , then its probability matrix is represented as $\Theta_k = (\theta_{1k}, \dots, \theta_{wk})$. Thus, when multiple occurrences of word k are aligned, the letter counts in the j th aligned column, $\mathbf{c}_{jk} = (c_{1jk}, \dots, c_{qjk})^T$, ($j = 1, \dots, w$), are characterized by the probability vectors, $\theta_{jk} = (\theta_{1jk}, \dots, \theta_{qjk})^T$, of a product multinomial model. The count matrices corresponding to the word probability matrices are denoted by $\{\mathbf{C}_{q+1}, \dots, \mathbf{C}_D\} = \mathbf{C}$.

The partition variable Π can be equivalently expressed by the motif site indicators, denoted by $\mathbf{A} = \{A_{ik}, i = 1, \dots, n, k = q + 1, \dots, D\}$, where

$$A_{ik} = \begin{cases} 1, & \text{if } i \text{ is the start of a site corresponding to} \\ & \text{motif type } k \\ 0, & \text{otherwise.} \end{cases}$$

From here on, we use the unique notation \mathbf{A} to interchangeably denote the site indicator vector \mathbf{A} or the equivalent partition vector Π . The complete data likelihood is

$$L(\mathbf{N}, \mathbf{C}, \mathbf{A} \mid \Theta^{(D)}, \boldsymbol{\rho}) \propto \prod_{l=1}^D \rho_l^{N_l} \prod_{k=q+1}^D \prod_{j=1}^{w_k} \prod_{i=1}^q \theta_{ijk}^{c_{ijk}}.$$

We assume a Dirichlet prior distribution for $\boldsymbol{\rho}, \boldsymbol{\rho} \sim \text{Dirichlet}(\boldsymbol{\beta}_0)$, $\boldsymbol{\beta}_0 = (\beta_{01}, \dots, \beta_{0D})$, and a corresponding product Dirichlet prior (i.e., independent priors over the columns) PD(\mathbf{B}) for $\Theta_k = (\theta_{1k}, \dots, \theta_{wk})$, ($k = q + 1, \dots, D$), where $\mathbf{B} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_{w_k})$ is a $q \times w_k$ matrix with $\boldsymbol{\beta}_j = (\beta_{1j}, \dots, \beta_{qj})^T$. The conditional posterior distribution of $\Theta_k \mid \mathbf{N}, \mathbf{C} \propto \prod_{j=1}^{w_k} \prod_{i=1}^q \theta_{ijk}^{c_{ijk} + \beta_{ij}}$, is product Dirichlet PD($\mathbf{B} + \mathbf{C}_k$), with the pseudocount parameters \mathbf{B} updated by the column counts of the k th word, $\mathbf{C}_k = (\mathbf{c}_{1k}, \dots, \mathbf{c}_{w_k k})$. The conditional posterior of $\boldsymbol{\rho} \mid \mathbf{N}, \mathbf{C}$ is Dirichlet($\mathbf{N} + \boldsymbol{\beta}_0$) $\propto \prod_{l=1}^D \rho_l^{N_l + \beta_{0l}}$.

Bayes estimates of θ may be obtained through a DA procedure using the full conditional distributions $P(\mathbf{A} \mid \theta, \mathcal{S})$ and $P(\theta \mid \mathcal{S}, \mathbf{A})$ and using techniques of recursion. The procedure for finding multiple motifs is described later, starting with a single-word matrix (i.e., $D = q + 1$) and expanding the model progressively at the end of each step.

2.4 The Algorithm

1. Partitioning: Sample for words (partitions) given the current value of the stochastic word matrix and word usage probabilities, that is, drawing $[\mathbf{A} \mid \Theta, \boldsymbol{\rho}, \mathcal{S}]$.

- Do a recursive summation of probabilities as shown in (3) to evaluate the partial likelihood up to every point in the sequence, that is,

$$L_i(\theta) = \sum_{k=1}^D P(\mathcal{S}_{[i-w_k+1:i]} \mid \theta) L_{i-w_k}(\theta). \quad (3)$$

$L_n(\theta)$ then represents the observed data likelihood, $P(\mathcal{S} \mid \theta) = P(x_1 \dots x_n \mid \theta)$.

- Let $\mathcal{A}_i = \{(A_{ik}, \dots, A_{nk}) : k \in q + 1, \dots, D\}$ denote the motif site indicators for motifs M_k , ($k = q + 1, \dots, D$) from position i to n . Words are sampled sequentially

backward, starting from the end of the sequence. We sample for a word starting at position i , according to the conditional probability,

$$\begin{aligned} P(A_{ik} = 1 \mid \mathcal{A}_{i+w_k}, \theta) &= \frac{P(\mathcal{S}_{[i:i+w_k-1]} \mid \Theta_k, \boldsymbol{\rho}) L_{i-1}(\theta)}{\sum_{l \in \mathcal{D}} P(\mathcal{S}_{[i+w_k-w_l:i+w_k-1]} \mid \Theta_l, \boldsymbol{\rho}) L_{i+w_k-w_l-1}(\theta)} \\ &= \frac{P(\mathcal{S}_{[i:i+w_k-1]} \mid \Theta_k, \boldsymbol{\rho}) L_{i-1}(\theta)}{L_{i+w_k-1}(\theta)}, \quad k = q + 1, \dots, D. \end{aligned}$$

If at position i none of the longer “words” is selected, then the appropriate single-letter word is assumed and k is decremented by 1.

2. Parameter update. Update the word stochastic matrix Θ_D and word probabilities $\boldsymbol{\rho}$, given the partition \mathbf{A} , by sampling columns j of word stochastic matrix Θ_D , ($j = 1, \dots, w_D$) from $\theta_{jD} \mid \mathbf{N}, \mathbf{C}, \boldsymbol{\rho} \sim \text{Dir}(\mathbf{c}_{jD} + \boldsymbol{\beta}_j)$. Next, sample the word frequency vector, from $\boldsymbol{\rho} \mid \mathbf{N}, \mathbf{C}, \Theta^{(D)} \sim \text{Dir}(\mathbf{N} + \boldsymbol{\beta}_0)$.

3. Repeat steps 1 and 2 until convergence.

4. Increase dictionary size; $D = D + 1$. Repeat again from step 1, now treating Θ_{D-1} as a *known* word matrix.

For initialization of the algorithm, we experimented starting either with an arbitrary number of random start sites to construct the initial “count” matrix \mathbf{C} or drawing columns of the initial word stochastic matrix Θ independently from a multinomial distribution with frequencies proportional to the background nucleotide distribution. In either case, the initial values did not appear to make a significant difference in terms of computational time. At each stage, we want to judge whether the new words added to the dictionary are “true” motifs or are chance patterns generated randomly. The maximum *a posteriori* (MAP) score at the optimal motif alignment (described in Sec. 4) can be used as a terminating criterion; stop adding words to the dictionary if the MAP score ceases to increase.

2.5 Phase Shift via Metropolis Steps

Let $\mathbf{a} = (a_1, a_2, \dots, a_m)$ be the set of starting positions for m occurrences of a motif (with PWM Θ_k) of width w_k in \mathcal{S} . Then $\mathbf{a} + \delta = (a_1 + \delta, a_2 + \delta, \dots, a_m + \delta)$ (for a small integer δ) are local modes of the distribution differing by a shift, because $w_k - \delta$ positions are still aligned. A Metropolis “phase-shift” update can be used to help the sampler shift back to the right phase if it is trapped in such a “shift” mode. By marginalizing (integrating) out parameters Θ , it is possible to compute the Metropolis acceptance ratio for the shift $= \min\{1, \frac{P(\mathbf{a} + \delta \mid \mathcal{S})}{P(\mathbf{a} \mid \mathcal{S})}\}$ as a ratio of gamma functions, which amounts to comparing the “information” of the new left column with the old right column or vice versa. In other words, we adopt the following procedure:

- Choose $\delta = \pm 1$ with probability 1/2 each.
- Update $\mathbf{a} \rightarrow \mathbf{a} + \delta$ with probability $\min(1, \eta)$, where

$$\eta = \frac{P(\mathbf{a} + \delta \mid \mathcal{S})}{P(\mathbf{a} \mid \mathcal{S})} = \begin{cases} \frac{\prod_{i=1}^q \Gamma(c_{i,w+1,k} + \beta_i)}{\prod_{i=1}^q \Gamma(c_{i,1,k} + \beta_i)} & \text{if } \delta = 1 \\ \frac{\prod_{i=1}^q \Gamma(c_{i,0,k} + \beta_i)}{\prod_{i=1}^q \Gamma(c_{i,w,k} + \beta_i)} & \text{if } \delta = -1. \end{cases}$$

The update is used only if the “shifted” motif does not overlap with any of the previously sampled ones.

Stirling’s approximation can be used instead of explicitly evaluating the gamma functions to increase computational efficiency. This Metropolis updating step is used between steps (1) and (2) of the algorithm.

3. EXTENSIONS OF THE STOCHASTIC DICTIONARY MODEL

3.1 Patterns With Variable Insertions and Deletions

Insertions and deletions of nucleotides may occur in biological sequences in functionally less important blocks in otherwise well-conserved motif patterns. We can generalize the stochastic dictionary model to the case where in one or more of the patterns, not all of the w columns of the “true” word matrix are present contiguously.

Let $\mathcal{G} = (\mathcal{G}_I, \mathcal{G}_D)$ denote a set of indicator variables representing the positions of insertions and deletions within all motif occurrences (collectively termed “gaps”). For deletions within a motif, the convention taken is $\mathcal{G}_{Di} = 1$ if there is a deletion before letter x_i in the sequence $x_1 x_2 \dots x_n$. Note that now, the observed data likelihood $P(\mathcal{S} | \theta)$ must be obtained by *two* summations over possible partitions, one partition corresponding to the partition of the sequence into words, the other for all possible gap positions within motifs. In other words, $P(\mathcal{S} | \theta) = \sum_{\mathbf{A}} \sum_{\Lambda_{\mathbf{A}}} P(\mathcal{S}, \mathbf{A}, \Lambda_{\mathbf{A}} | \theta)$, where $\Lambda_{\mathbf{A}}$ denotes the set of all possible partitions of the pattern segments into site and nonsite (i.e., gap) letters under the partition \mathbf{A} of \mathcal{S} . The additional prior parameters in the gapped model are

- λ_m , probability of a *match* between a nucleotide (of a sequence segment) and stochastic word matrix
- λ_{Io} , *insertion-opening* (starting) probability
- λ_{Ie} , *insertion-extension* probability
- λ_{Do} , *deletion opening* probability
- λ_{De} , *deletion-extension* probability.

From Figure 1, it can be seen that $\lambda_m + \lambda_{Io} + \lambda_{Do} = 1$.

In alignment of biological sequences, the gap extension penalty is typically considered lower than the gap-opening penalty, because fewer but longer random insertions or deletions are more likely to occur in a sequence than a series of short ones. In the case of short motif patterns, such an assumption may not be necessary, and we may simplify the model by setting $\lambda_{Io} = \lambda_{Ie}$ and $\lambda_{Do} = \lambda_{De}$. Similarly, an insertions-only model may be obtained by setting the set of deletion parameters, $\lambda_{Do} = \lambda_{De} = 0$. We currently keep these parameters fixed at values based on empirical biological evidence (.01 – .05 for λ_{Io} and .2 – .4 for λ_{Ie} were used in the CRP data example, Sec. 5.2), but this may be later extended to a more general hierarchical model under the Bayesian framework.

Iterative Update Procedure. As previously, we let w denote the width of a *single* motif and let g and d denote the

total allowable lengths of insertions and deletions in the motif pattern:

1. Partition the sequence into “gapped” words; sample from $P(\mathbf{A} | \theta, \mathcal{S})$. Computing the probability of each segment $P(\mathcal{S}_{[i-j:i]} | \theta)$, ($j = w - d, \dots, w + g$) in the exact likelihood computation (3) now involves an additional recursive sum, involving the set of equations (4).

2. Sample gap positions within words; draw from $P(\mathcal{G} | \mathbf{A}, \theta, \mathcal{S})$. Consider an individual segment, say \mathbf{x} , of size $w + h$. Finding likely gap positions within this segment corresponds to “aligning” this segment with a $q \times w$ word stochastic matrix (under the restriction that there is a maximum of g insertions and d deletions). Insertions at the ends of segments are treated as continuations of the background. This step is schematically represented in Figure 1 and described later.

3. Update parameters, given the sampled partition of the sequence; draw from $P(\theta | \mathbf{A}, \mathcal{G}, \mathcal{S})$.

Assume that the segment $\mathbf{x} = (x_1, \dots, x_{w+h})^T$ is aligned exactly with the word stochastic matrix at the end, where $h = g - d, \dots, g$. Let M_j denote the number of matches before position j . Let $P_p(x)$ denote the probability of nucleotide x being generated from the background model as an insertion in the motif. Also, let $P_\theta(x, i)$ denote the probability of x being realized from the i th column of the motif model. Denote the probabilities of the k th position being the j th match, belonging to an insertion after the j th match, or coming after a deletion after the j th match as

$$F_M(j, k) = P(x_k \in \mathcal{G}^c | M_k = j - 1),$$

$$F_I(j, k) = P(x_k \in \mathcal{G}_I | M_k = j),$$

and

$$F_D(j, k) = P(x_k \in \mathcal{G}_D | M_{k-1} = j).$$

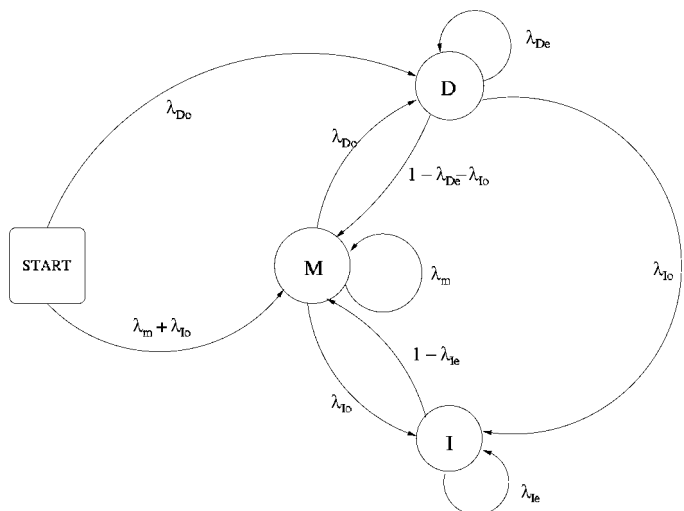


Figure 1. Schematic Diagram of Alignment of the “Gapped” Motif With the Stochastic Word Matrix. M, I, and D represent match, insertion, and deletion states.

With initial conditions

$$\begin{aligned} F_I(0, 0) &= 1, \text{ and} \\ F_M(1, k) &= (\lambda_m + \lambda_{I_o})F_I(0, k-1)P_{\Theta}(x_k, 1), \\ &1 \leq k \leq \min\{g+d+1, w+g-1\}, \end{aligned}$$

we have

$$\begin{aligned} F_I(0, k) &= F_I(0, k-1)P_{\rho}(x_k), \quad 1 \leq k \leq g+d; \\ F_I(j, k) &= [\lambda_{I_e}F_I(j, k-1) + \lambda_{I_o}\{F_D(j, k-1) \\ &\quad + F_M(j, k-1)\}]P_{\rho}(x_k) \\ &j < k \leq \min\{g+d+j, w+g-1\}, \\ &1 \leq j \leq w-1; \\ F_M(j, k) &= [(1-\lambda_{I_e})F_I(j, k-1) + \lambda_m F_M(j, k-1) \\ &\quad + (1-\lambda_{D_e}-\lambda_{I_o})F_D(j, k-1)]P_{\Theta}(x_k, j) \\ &j \leq k \leq \min\{g+d+j, w+g-1\}, \\ &1 < j \leq w-1; \end{aligned}$$

and

$$\begin{aligned} F_D(j, k) &= \lambda_{D_o}F_M(j, k-1) + \lambda_{D_e}F_D(j, k-1) \\ &j \leq k \leq g+d+j-1, \quad 1 \leq j \leq w-1. \quad (4) \end{aligned}$$

In the foregoing, note that deletions after insertions are not allowed, to avoid redundancy. Then, for the segment \mathbf{x} ,

$$\begin{aligned} P(\mathbf{x}) &= F_M(w, w+h) \\ &= [(1-\lambda_{I_e})F_I(w-1, w+h-1) + \lambda_m F_M(w-1, w+h-1) \\ &\quad + (1-\lambda_{D_e}-\lambda_{I_o})F_D(w-1, w+h-1)] \times P_{\Theta}(x_{w+h}, w). \end{aligned}$$

Motif site and gap positions are then sampled sequentially starting from the end of the segment and progressing backward. If x_k is the sampled j th match (i.e., $M_k = j-1$, $j = w-1, w-2, \dots, 1$), then conditionally sample $x_{k-1} \in \mathcal{G}^c \mid M_k = j-1$ with probability

$$\frac{F_M(k-1, j-1)}{F_I(k-1, j-1) + F_M(k-1, j-1) + F_D(k-1, j-1)}.$$

The motif matrix is updated by sampling from the posterior Dirichlet distribution $P(\Theta \mid \mathbf{A}, \mathcal{G}, \mathcal{S})$ for the selected nongap positions of the motif.

3.2 Patterns of Unknown Width

The Bayesian framework of the stochastic dictionary model can be exploited to determine the likely pattern width if it is unspecified. A Poisson prior is used for the motif width w , with mean w_0 that reflects our approximate idea of the likely width. Some additional notation is introduced for a clearer presentation. For vectors $\mathbf{v} = (v_1, \dots, v_p)^T$, define $|\mathbf{v}| = |v_1| + \dots + |v_p|$, and $\Gamma(\mathbf{v}) = \Gamma(v_1) \cdots \Gamma(v_p)$, so that the normalizing constant for a p -dimensional Dirichlet distribution can be denoted as $\Gamma(|\boldsymbol{\alpha}|)/\Gamma(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)^T$. Additionally, define $\boldsymbol{\alpha}_{[1:k]} = (\alpha_1, \dots, \alpha_k)^T$, the first k components of $\boldsymbol{\alpha}$. For notational simplicity, assume here that all the columns of the matrix have the same prior distribution, that is, $\beta_{il} = \gamma_l$ for $i = 1 \dots w$, $l = 1 \dots 4$.

The conditional posterior distribution of the unknown parameters given the partition indicator \mathbf{A} ($\equiv \mathbf{N}, \mathcal{C}$) is

$$\begin{aligned} P(W = w, \Theta \mid \mathbf{A}) &= \\ &\propto \left[\prod_{l=1}^D \rho_l^{N_l} \prod_{j=1}^w \prod_{i=1}^q \theta_{ij}^{c_{ij}} \right] \left[\frac{e^{-w_0} w_0^w}{w!} \prod_{l=1}^D \rho_l^{\beta_l-1} \frac{\Gamma(|\boldsymbol{\beta}_0|)}{\Gamma(\boldsymbol{\beta}_0)} \right. \\ &\quad \left. \times \prod_{j=1}^w \left\{ \prod_{i=1}^q \theta_{ij}^{\beta_{ij}-1} \frac{\Gamma(|\boldsymbol{\beta}_j|)}{\Gamma(\boldsymbol{\beta}_j)} \right\} \right]. \end{aligned}$$

Suppose that we want to decide whether the true width of the pattern is w or $w + \delta$, ($\delta = \pm 1$), with the count vector for the column in question denoted by $\mathbf{H} = (h_1, \dots, h_q)^T$. Given the current partition \mathbf{A} , a Metropolis-like update for the width can be implemented as follows:

1. Choose $\delta = \pm 1$ with probability 1/2 each; that is, choose to either increment or reduce the motif width by 1.
2. Update the width of the motif from w to $w + \delta$ with probability $\min(1, B_{\delta 0})$, where

$$\begin{aligned} B_{\delta 0} &= \frac{\int_{\Theta} p(W = w + \delta, \Theta \mid \mathbf{A}) d\Theta}{\int_{\Theta} p(W = w, \Theta \mid \mathbf{A}) d\Theta} \\ &= \frac{\Gamma(\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H})}{\Gamma(|\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H}| + N_D + \beta_{0D})} \\ &\quad \times \frac{\Gamma(|\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]}|)}{\Gamma(\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]})} \\ &\quad \times \left\{ \frac{w_0}{(w + \frac{\delta+1}{2})} \frac{\Gamma(\mathbf{H} + \boldsymbol{\gamma})}{\Gamma(|\mathbf{H} + \boldsymbol{\gamma}|)} \frac{\Gamma(|\boldsymbol{\gamma}|)}{\Gamma(\boldsymbol{\gamma})} \right\}^{\delta}. \quad (5) \end{aligned}$$

Computational details are given in the Appendix.

3.3 Motif Detection in the Presence of “Low-Complexity” Regions

A distinct characteristic of biological sequence data (especially in the genomes of higher organisms) is the presence of long stretches of a single nucleotide (e.g., AAAA ...), or dimer and more complex repeats (e.g., CGCGCG ...). These “low-complexity” regions are traps for pattern-finding algorithms, and one often needs to preprocess the data using a heuristic method, such as a “masking” procedure, which screens DNA sequences against a library of repetitive elements and returns a sequence with apparent long repeats deleted. However, judging whether a part of the sequence consists of repeats based on some threshold could potentially lead to a loss of information if the threshold is set either too high or too low. It is desirable to develop a more principled approach for recognizing and filtering out the low-complexity repeats. The progressive updating in the stochastic dictionary model is expected to provide an effective control, by treating the low-complexity “repeats” as series of adjacent words of the same pattern. Simulation studies are discussed (Sec. 5.1) that compare the performance of the stochastic dictionary model with methods that uses either a masking procedure or a higher-order Markov background model.

3.4 Stochastic Dictionary Versus Gibbs Motif Sampler

It is worthwhile to make a connection here between the DA approach used by the stochastic dictionary model and the Gibbs motif sampler (GMS) of Liu et al. (1995). For simplicity, we consider the case with only one nontrivial stochastic word, with PWM Θ , in the dictionary.

Let \mathbf{A} again be the motif indicator and let $\mathbf{A}_{[-r]} = \{A_1, \dots, A_{r-1}, A_{r+w-1}, \dots, A_n\}$. Then the GMS iteratively draws from the conditional distribution, $P(A_r | \mathbf{A}_{[-r]}, \Theta)$. That is, the GMS iteratively visits each sequence position r , updating its motif indicator conditional on the indicators for other positions. This approach tends to be “sticky” when the motif sites are abundant. For example, once we have set $A_r = 1$, we will not be able to allow segment $x_{[r+1:r+w]}$ to be a motif site.

In contrast, the new DA approach samples \mathbf{A} jointly according to the conditional distribution

$$P(\mathbf{A} | \Theta) = P(A_n | \Theta) \prod_{r=1}^{n-1} P(A_r | A_{r+1}, \dots, A_n, \Theta).$$

At a position r , the current knowledge of motif positions is updated using the conditional probability $P(A_r | A_{r+1} \dots A_n, \Theta)$ (backward sampling), with $A_{r-1} \dots A_1$ marginalized out (forward summation).

Theoretically, it has been shown that *grouping* or *collapsing* variables in a Gibbs sampler is likely to improve its convergence rate (Liu, Wong, and Kong 1994). The DA method based on the stochastic dictionary model corresponds to a grouping scheme (with \mathbf{A} sampled together), whereas the GMS corresponds to a collapsing approach (with Θ integrated out). Although both approaches are better than the standard Gibbs sampler, it is not obvious how to compare the GMS with the DA method. Intuitively and empirically, the DA approach is more efficient, because the dependencies among the A_i 's has a greater impact on the convergence than does the sampling of θ . This advantage of the DA is more prominent as the motif sites become more crowded and multiple motif types are involved.

4. MODEL SELECTION

One of the fundamental questions is whether the patterns “discovered” from the sequence data by our algorithm are “real.” Although the biological relevance of such findings needs to await further biological experimentation (if not already documented), we at least want to assess their statistical significance. We can formulate the question as a Bayesian model selection problem; it is of interest to assess whether the sequence data should be explained by model \mathcal{M}_1 , which assumes the existence of a nontrivial motif, or \mathcal{M}_0 , which says that the sequences are generated entirely from a background model (e.g., an iid or Markov model). The Bayes factor, which is the ratio of the marginal likelihoods under the two models, can be computed as

$$\frac{p(\mathcal{S} | \mathcal{M}_1)}{p(\mathcal{S} | \mathcal{M}_0)} = \frac{\sum_{\mathbf{A}} \int_{\Theta} p(\mathbf{A}, \mathcal{S}, \Theta | \mathcal{M}_1) d\Theta}{\int_{\Theta} p(\mathcal{S}, \Theta | \mathcal{M}_0) d\Theta} = \frac{\sum_{\mathbf{A}} p(\mathbf{A}, \mathcal{S} | \mathcal{M}_1)}{p(\mathcal{S} | \mathcal{M}_0)}. \quad (6)$$

The individual additive terms in the numerator, after integrating out Θ , consist of ratios of products of gamma functions. Evaluating this sum exhaustively over all partitions involves prohibitive amounts of computation. We thus need to resort to either Monte Carlo or some approximations.

4.1 Approximating the Bayes Factor

It can be observed that $p(\mathcal{S} | \mathcal{M}_1)$ is the normalizing constant of

$$p(\mathbf{A} | \mathcal{S}, \mathcal{M}_1) = \frac{p(\mathbf{A}, \mathcal{S} | \mathcal{M}_1)}{p(\mathcal{S} | \mathcal{M}_1)},$$

where $p(\mathbf{A}, \mathcal{S} | \mathcal{M}_1)$ is of known form. We tested several importance sampling and *bridge* sampling (Meng and Wong 1996) methods for estimating the ratio.

An obvious choice of the trial distribution for \mathbf{A} is $p(\mathbf{A} | \mathcal{S}, \Theta^*, \mathcal{M}_1)$ for a certain Θ^* (e.g., the mode), leading to an estimate of $p(\mathcal{S} | \mathcal{M}_1)$ as

$$\frac{1}{N} \sum_{k=1}^N \frac{p(\mathbf{A}^{(k)}, \mathcal{S} | \mathcal{M}_1)}{p(\mathbf{A}^{(k)} | \mathcal{S}, \Theta^*, \mathcal{M}_1)}, \quad (7)$$

where the denominator can be computed by the recursive procedure described in the previous section. However, this trial distribution is too narrowly supported and severely underestimates the Bayes factor. To counter this problem, we use a mixture distribution,

$$q(\mathbf{A}) = \frac{1}{K} \sum_{j=1}^K p(\mathbf{A} | \mathcal{S}, \Theta_j, \mathcal{M}_1), \quad (8)$$

with component parameters Θ_j corresponding to “true” and “shift” modes of the optimal Θ .

Because we can obtain MCMC draws from $p(\mathbf{A} | \mathcal{S}, \mathcal{M}_1)$, it is also possible to implement a bridge sampling method. However, because the MCMC draws are very sticky in most of the cases in which we are interested, the bridge sampler did not perform very well.

4.2 The MAP Scoring Criterion

An obvious lower bound for (6) is $p(\mathbf{A}^*, \mathcal{S} | \mathcal{M}_1) / p(\mathcal{S} | \mathcal{M}_0)$, where \mathbf{A}^* is the maximizer of the ratio. We call this lower bound the MAP(\mathbf{A}^*) score, which can be tracked along with the DA iterations. More precisely, for any indicator vector \mathbf{A} (representing the sampled motif locations), we can integrate out Θ and obtain that

$$\begin{aligned} \log \frac{p(\mathbf{A}, \mathcal{S} | \mathcal{M}_1)}{p(\mathcal{S} | \mathcal{M}_0)} &= \log \left\{ \frac{\Gamma(\mathbf{N} + \boldsymbol{\beta}_0)}{\Gamma(\mathbf{N} + \boldsymbol{\beta}_0)} \frac{\Gamma(|\boldsymbol{\beta}_0|)}{\Gamma(\boldsymbol{\beta}_0)} \right\} \\ &\quad - \log \left\{ \frac{\Gamma(\mathbf{N}_{[1:q]} + \mathbf{c} + \boldsymbol{\beta}_{0[1:q]})}{\Gamma(\mathbf{N}_{[1:q]} + \mathbf{c} + \boldsymbol{\beta}_{0[1:q]})} \frac{\Gamma(|\boldsymbol{\beta}_{0[1:q]}|)}{\Gamma(\boldsymbol{\beta}_{0[1:q]})} \right\} \\ &\quad + \sum_{j=1}^w \log \left\{ \frac{\Gamma(\mathbf{c}_j + \boldsymbol{\gamma})}{\Gamma(|\mathbf{c}_j + \boldsymbol{\gamma}|)} \frac{\Gamma(|\boldsymbol{\gamma}|)}{\Gamma(\boldsymbol{\gamma})} \right\}, \end{aligned}$$

where $\mathbf{c} = \sum_{j=1}^w \mathbf{c}_j$ denotes the word matrix column counts, $\mathbf{N}_{[1:q]} = \{N_1, \dots, N_q\}$ denotes the letter frequencies, and $\boldsymbol{\beta}_{0[1:q]} = \{\beta_{01}, \dots, \beta_{0q}\}$ denotes the prior pseudocounts for the q letters.

When there are $D - q > 1$ motifs (nontrivial stochastic words) in the dictionary, the log(MAP) score can be computed as

$$\begin{aligned} \log \text{MAP}(\mathbf{A}) = & \log \left\{ \frac{\Gamma(\mathbf{N} + \boldsymbol{\beta}_0)}{\Gamma(|\mathbf{N} + \boldsymbol{\beta}_0|)} \frac{\Gamma(|\boldsymbol{\beta}_0|)}{\Gamma(\boldsymbol{\beta}_0)} \right\} \\ & - \log \left\{ \frac{\Gamma(\mathbf{N}_{[1:q]} + \mathbf{c} + \boldsymbol{\beta}_{0[1:q]})}{\Gamma(|\mathbf{N}_{[1:q]} + \mathbf{c} + \boldsymbol{\beta}_{0[1:q]}|)} \frac{\Gamma(|\boldsymbol{\beta}_{0[1:q]}|)}{\Gamma(\boldsymbol{\beta}_{0[1:q]})} \right\} \\ & + \sum_{k=q+1}^D \sum_{j=1}^{w_k} \log \left\{ \frac{\Gamma(\mathbf{c}_{jk} + \boldsymbol{\gamma})}{\Gamma(|\mathbf{c}_{jk} + \boldsymbol{\gamma}|)} \right\} \\ & + \sum_{k=q+1}^D w_k \log \left\{ \frac{\Gamma(|\boldsymbol{\gamma}|)}{\Gamma(\boldsymbol{\gamma})} \right\}, \end{aligned}$$

where now $\mathbf{c} = \sum_{k=q+1}^D \sum_{j=1}^{w_k} \mathbf{c}_{jk}$, and $\mathbf{N}_{[1:q]}$ and $\boldsymbol{\beta}_{0[1:q]}$ are the same as before.

It is of interest to know how the log(MAP) behaves when the sequences are in fact iid generated, containing no motifs at all. We generated 50 datasets, each in three sizes, 1,000, 5,000, and 10,000, from the iid background model. We then set the SDDA algorithm to find motifs of width 6, 10, and 16 in each set. To find the ‘‘optimal’’ alignment, we used an annealing procedure. For each generated dataset, we used 5 independent runs of 1,000 iterations and 4,000 annealing steps, and chose the maximum MAP score in the 5 runs for our study. The observed values of the maximum log(MAP) score in each case are depicted in Figure 2, for the dataset of size 1,000. In all but one of the cases that did not lead to the alignment with no motifs (the *null* alignment), the observed log(MAP) score was lower than that of the null alignment, indicating that the algorithm was stuck in a local chance mode. For the larger datasets of size 5,000 and 10,000, the observed log(MAP) score tended to be generally lower, and it was harder for the algorithm to reach the null alignment, especially for longer motif widths 10 and 16.

From a number of simulated and real datasets (Secs. 5.1 and 5.2), we observed that the MAP score dominates the other components of the Bayes factor and is often more reliable for differentiating the two models than the approximated Bayes factor via importance sampling.

5. NUMERICAL RESULTS

5.1 Case Study I: Simulated Dataset With Background Polynucleotide Repeats

To test the effectiveness of the algorithm, we first constructed sets of data with multiple motif patterns in sequences with varying degrees of polynucleotide repeats. To mimic two types of repeats in biological sequences, monomer repeats (e.g., poly-A) and dimer repeats (e.g., CGCGCG. . .), we generated the background from two types of first-order Markov transition matrices, with increasing degrees of dependence. Motifs were generated based on a known matrix for the transcription factor **narL** found in *E. coli* (Fig. 3). The motif probability matrix, based on 11 known motif sites, is

$$\Theta = \begin{bmatrix} 0 & .55 & .55 & .09 & 0 & 0 & .18 & .46 & .36 & .82 & .09 & .09 & 0 & 0 & 1 \\ 0 & .09 & .45 & .55 & .36 & 1 & .09 & .18 & .09 & .18 & .09 & .09 & 0 & 0 & 0 \\ .09 & .09 & 0 & 0 & 0 & 0 & .27 & .09 & 0 & 0 & .46 & .64 & .91 & 1 & 0 \\ .91 & .27 & 0 & .36 & .64 & 0 & .46 & .27 & .55 & 0 & .36 & .18 & 0 & 0 & 1 \end{bmatrix}.$$

The background transition matrices were of the forms

$$\begin{aligned} \text{(a)} \quad & \begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix} \quad \text{and} \\ \text{(b)} \quad & \begin{pmatrix} \beta & \alpha & \alpha & 1-2\alpha-\beta \\ \alpha & \beta & 1-2\alpha-\beta & \alpha \\ \alpha & 1-2\alpha-\beta & \beta & \alpha \\ 1-2\alpha-\beta & \alpha & \alpha & \beta \end{pmatrix}, \quad (9) \end{aligned}$$

where values of α in (a) were taken as $\{.01, .07, .13, .19\}$. The degree of dependence of the background was represented by the magnitudes of the second-largest eigenvalue (EVAL2), which were $(.24, .48, .72, .96)$. In matrix type (b), with both one- and two-letter dependence, values of (α, β) were taken as $\{(.01, .65), (.07, .55), (.13, .45), (.19, .35)\}$, which incidentally turned out to have the same set of values for EVAL2. The simulated datasets were 2,000 nucleotides long (10 sequences of 200 nucleotides each), and 20 datasets were independently constructed under background distributions characterized by the 8 matrices described in (9). The number of motif sites ranged from zero to two in each sequence.

We compared the performance of the SDDA method with two popular existing algorithms that are Gibbs sampling based but with additional adaptations to tackle polymer repeats. BioProspector (Liu, Brutlag, and Liu, 2001) uses a third-order Markov chain to model the background, whereas AlignAce (Roth, Hughes, Estep, and Church 1998) uses a masking approach to remove previously found words from its search space. In both algorithms, we set the maximum number of motifs to be detected as five. A single run of each algorithm was used for each dataset, no gaps were allowed in the motif pattern, and the motif width was set to be the known width of 16. In each case, the algorithm was considered to have found the correct pattern if there was at least 60% site overlap between the true and found motifs for at least 80% of the true sites. For SDDA, to allow repeat-based words to be included in the dictionary, we set the following empirical stopping criterion to limit the number of words in the dictionary: (a) No more new words were sampled after at least two random starts of the algorithm, or (b) the MAP score decreased twice in succession after sampling a new word. We restricted our search to the top five words found. In almost all cases, when the true word was found, it was within the first four selected.

The results are shown in Table 1. The performance of each algorithm is compared in terms of success rates in finding the motif pattern and the percentage of false-positive results. The SDDA procedure shows superior performance in distinguishing the true motifs for increasing degrees of background dependence in terms of success rates of finding the motif pattern. AlignAce has a very low success rate in this setting, and the proportion of false-positives is high among the found patterns. BioProspector performs relatively well for monomer repeats, possibly due to the Markovian background assumption, but in the presence of the higher-order repeats it fails more often. In the SDDA algorithm, repeat patterns occasionally accounted for the first or second pattern found in the

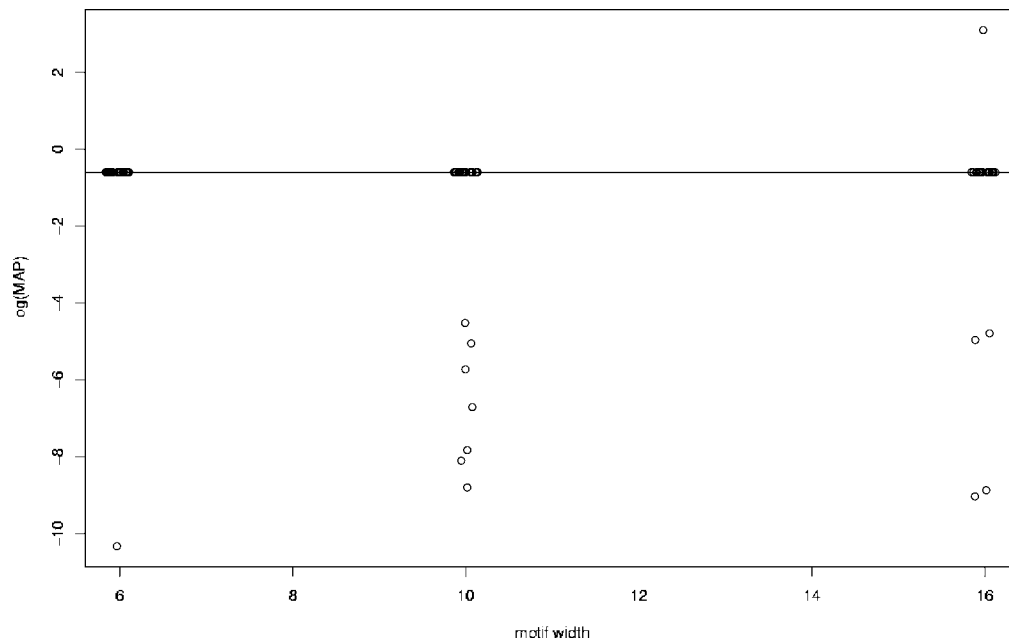


Figure 2. The $\log(\text{MAP})$ Values Found in Randomly Generated Sequences for Motif Width Settings of 6, 10, and 16 for a Dataset of Size 1,000. The horizontal line corresponds to the log-posterior probability when the alignment with no motifs (the null alignment) is chosen. The points have been jittered to show the relative frequency.

datasets with high monomer or dimer repeat contents, but after adding these to the dictionary, the algorithm was successful in detecting the true motif. This simulation study is by no means exhaustive, but it presents some empirical evidence of the robustness of the algorithm to model misspecification when the assumption of background independence is violated.

5.2 Case Study II: CRP Binding Motifs Data

This dataset contains 18 DNA fragments of length 105 each known to contain CRP binding sites (Stormo and Hartzell 1989). CRP is a transcription factor in *E. coli* that modulates the transcription activity of many genes in response to glucose levels. High glucose reduces the concentration of the cyclic AMP (cAMP) molecule within the cell, whereas glucose starvation leads to its increase, allowing it to bind to CRP. The CRP is then activated to bind to the regulatory sites, attracting the RNA polymerase to initiate the transcription of the down-

stream gene into messenger RNA. Previous experimental studies (footprinting and crystallography-based) (de Crombrugge, Busby, and Buc 1984) have identified 24 potential TF binding sites of (hypothesized) length 22 each in these sequences. The variation in the degree of conservation of each position is illustrated in Figure 4. A number of these sites have been located by computational methods, the EM algorithm (Lawrence and Reilly 1990) and the GMS (Neuwald et al. 1995). Apart from locating the motif sites, we are also interested in assessing the significance of the motif pattern and determining the width of the motif pattern.

The prior Dirichlet pseudocounts (i.e., $\beta_{01}, \dots, \beta_{04}$) for the frequencies of the four single-letter words were chosen to be around 10% of the total nucleotide counts of each type. Slight variations from this do not appear to significantly affect the results. The prior pseudocount for the motif frequency was set at one site per sequence times 10%. The pseudocounts

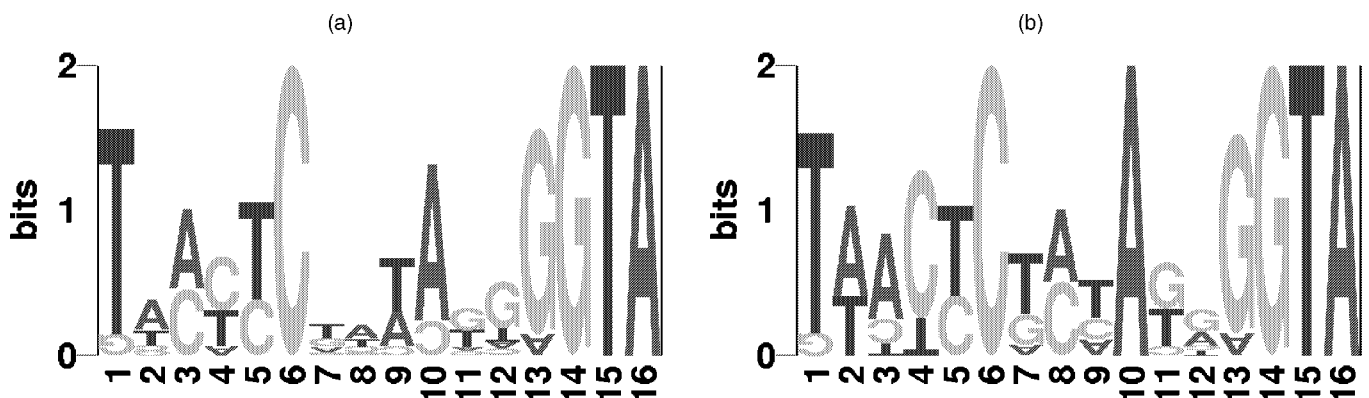


Figure 3. Sequence Logo for the True *narL* Matrix (a) and Discovered Motifs (b) in a Generated Dataset. The letter height is proportional to its frequency at the site, and the stack height reflects the total information content. Inverted letters represent those observed less often than expected under the background model (see Schneider et al. 1986).

Table 1. Relative Performance of the SDDA Method Compared With BioProspector (BP) and AlignAce (AA) for Background Distributions Characterized by Matrix Types (a) and (b)

EVAL2	Stochastic dictionary		BioProspector		AlignAce	
	Success	FP	Success	FP	Success	FP
(a) .24	1.0	(.07)	1.0	(.02)	.3	(.43)
.48	.6	(.06)	.7	(0)	.0	(—)
.72	.5	(.12)	.1	(0)	.0	(—)
.96	.7	(.02)	.0	(—)	.0	(—)
(b) .24	1.0	(.03)	1.0	(.09)	.1	(.52)
.48	.9	(.12)	.7	(.01)	.1	(.62)
.72	.9	(.05)	.6	(0)	.1	(.36)
.96	1.0	(.03)	.0	(—)	.0	(—)

NOTE: Success represents the proportion of runs the true motif pattern was found under the set conditions, False-positive (FP) error rates were calculated only for the sets in which the true pattern was found.

for every column of the stochastic word weight matrix were .25 for each nucleotide type. The algorithm was initially run to find ungapped motifs of width fixed at 22 nucleotides. As shown in Figure 4, the motif pattern found by our algorithm agrees extremely well with the experimentally determined pattern. Nineteen of the experimental sites were sampled in more than 50% of the DA iterations.

We computed the approximated Bayes factor and MAP ratio for evaluating the significance of the one-motif model (of width 22). The log(MAP) score was obtained as $-.4986$, identical in 20 independent DA runs with simulated annealing steps. For the importance sampling approximation to the BF, the trial distribution was chosen to be a $(2k + 1)$ -component mixture as in (8), with component parameters corresponding

to the found optimal alignment and shifts of up to k positions in either direction. We tested $k = 1, \dots, 6$. For small k , the estimates were clearly biased downward. For $k \geq 3$, the logarithm of the Bayes factor estimates stabilized at a value between .64 and .70 (each with 50,000 importance samples).

The approximate Bayes factor was still not convincingly high to decisively favor the one-motif model over the pure background model. To see whether the estimated Bayes factor represented a significant departure from a random case, we simulated 100 control sets of random shuffles of the original data set and computed their approximated Bayes factors. Figure 5 shows a large separation between the criteria values for true motifs and random motifs found in the shuffled datasets. The log-posterior probability corresponding to the alignment with no motifs [denoted as $\log(\text{null})$], is shown in Figure 5(b). It can be observed that most alignments found by the algorithm give a log(MAP) score lower than $\log(\text{null})$, indicating that algorithm has not yet reached the global optimum. The annealing procedure used does not appear to significantly improve the alignment toward a higher posterior value, which is consistent with our observations in simulation studies for randomly generated datasets of this size and motif width specification. On one hand, the results clearly show that the algorithm exhibits a strong “stickiness” when running on randomly simulated data with no motif present. On the other hand, both the importance sampling estimate and the log(MAP) score showed a clear separation between the true motif alignment in the original data and chance motif alignments in the shuffled data. The fact that the presence of true motifs can help the algorithm pull off from a local chance optimum is both encouraging and perplexing.

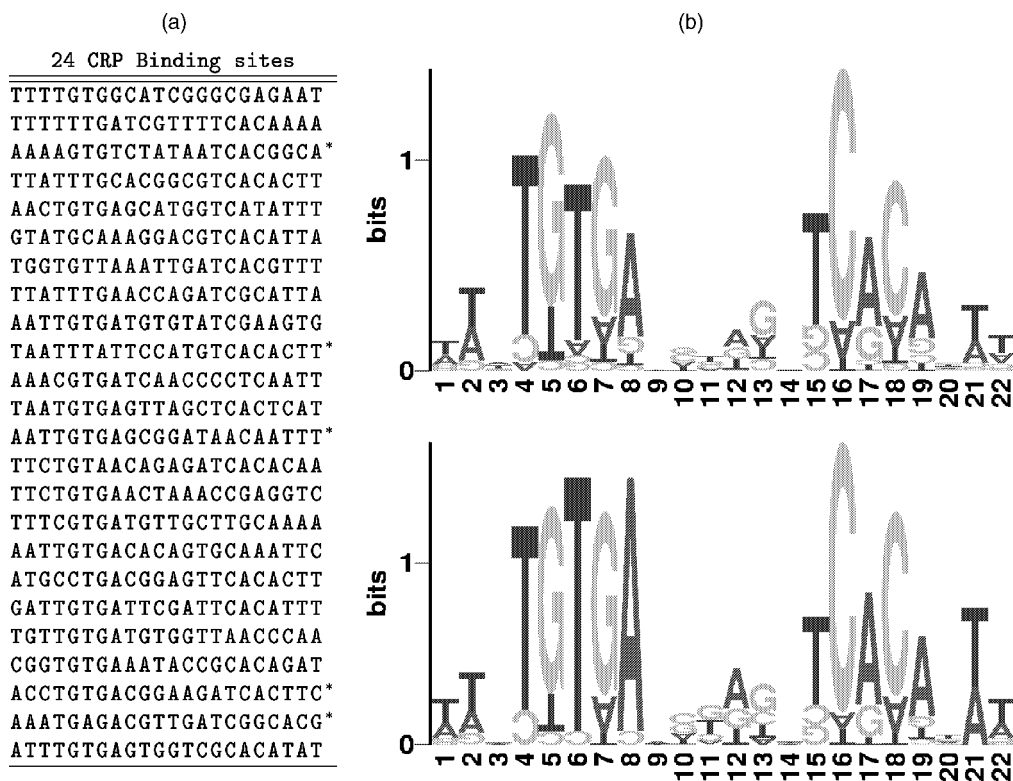


Figure 4. (a) The 24 Experimentally Hypothesized Sites in the CRP Data (Lawrence and Reilly 1990), and (b) Sequence Logo for Hypothesized Sites (24) and Sites for Best Posterior Alignment (19). The undetected sites are marked with *.

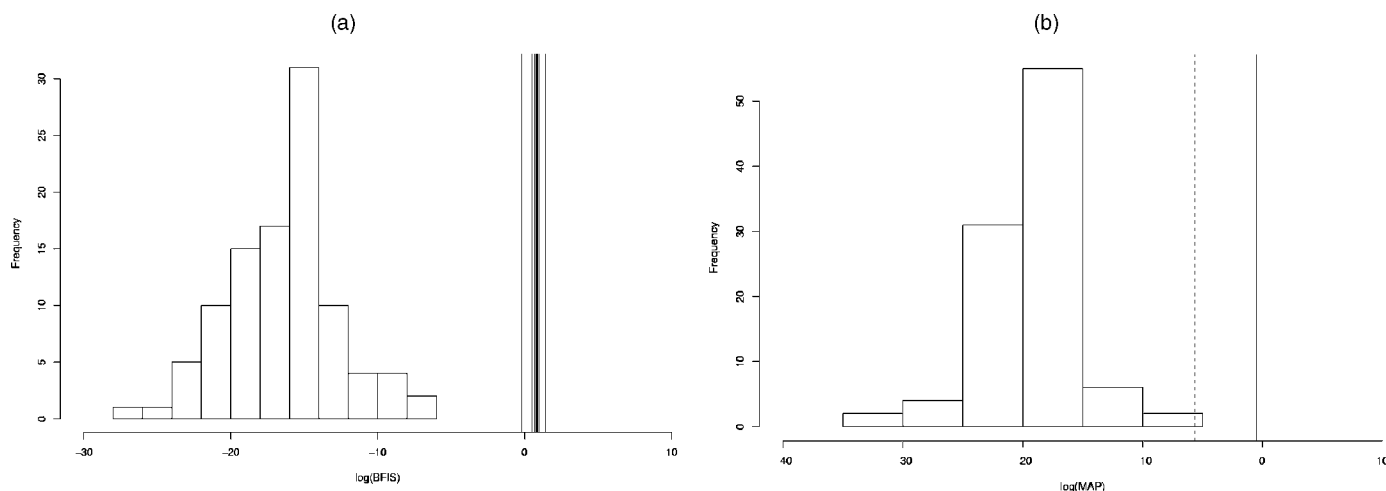


Figure 5. Histograms of the Importance Sampling Estimator for (a) the Bayes Factor and (b) the $\log(\text{MAP})$ for 100 Random Shuffles of the CRP Dataset. The vertical lines represent the estimated values in the true dataset. The broken line in the (b) corresponds to the $\log(\text{MAP})$ value if no positions were selected as motif sites.

We also experimented starting with a motif width different from the “known” one of 22 bases, and using the Gibbs–Metropolis width update (still considering ungapped motifs). For starting widths ranging from 15 to 30, the sampler had the highest observed MAP for motif width 24 with 11 motif sites, but the posterior distribution of widths (Fig. 6) indicates that the most likely width of the motif is 19. The hypothesized width of 22 appears to be slightly suboptimal.

The relative lack of conservation in the middle positions (i.e., six residues in positions 9 to 14) and at the two ends of the motif prompted us to use the stochastic alignment algorithm to detect possible gapped motifs. We specified the maximum number of positions corresponding to gaps as 6, and the motif width (for nongap positions) as 19. We kept the other parameters unchanged. The algorithm now succeeded in finding a previously undetected motif site in the 17th sequence (*cat*) that had the first two nucleotides missing from the sequence (i.e., a *deletion*) but was shown to be a possible site in biological experiments. The posterior distribution of motif site positions is shown in Figure 7; the dip in the motif site probability within a well-conserved region (i.e., around position 45 in sequence 13, and between positions 2 and 7

in sequence 17) represent the likely gap positions within the motif blocks. The algorithm also suggests the existence of a possible site in the eighth sequence (*ilv B*) starting at position 20, which has not been detected experimentally. The sequence logo of the 19 aligned positions (excluding gaps) now shows a more highly conserved set of motifs (Fig. 8).

6. DISCUSSION

The stochastic dictionary model presents a tractable framework in which to search for unknown and stochastically varying patterns in sequences. Bayesian modeling in this complex data setting provides a flexibility that is difficult to achieve through many heuristics-based procedures. The data augmentation procedure demonstrates the use of recursive techniques that reduce computational complexity and increase efficiency. For a dataset of N nucleotides with k distinct motif types, the order of computation per iteration is approximately $N(k+1)$, and in most of the applications we found that 2,000 or fewer iterations were sufficient for detecting the motif pattern. In the CRP data example (dataset size 1,890), running 2,000 iterations took approximately 3 minutes on a Sun Ultra 5 workstation. For a maximum motif size of $w+g$, (where w is the motif width and g is the maximum gap size), the order of computation is approximately $N(k+1)[3(w+g)^2 + (w+g)]$ for k motif types. By progressively updating a stochastic dictionary, our approach reduces the inherent arbitrariness present in “low-complexity” masking procedures for data preprocessing. Our proposed strategy differs significantly from the MobyDick algorithm of Bussemaker et al. (2000), not only in its use of stochastic word matrices and the data augmentation scheme, but also in its method of dictionary build-up and its treatment of low-complexity regions.

The assumption of independence between different positions within the motif may appear to be an oversimplification. However, because the sequence conservation in the regulatory binding motif sites are mainly due to the requirement of a favorable energetic interaction between the motif site and its recognition transcription factor, there seems to be no *a priori* reason why two different positions in a motif should covary.

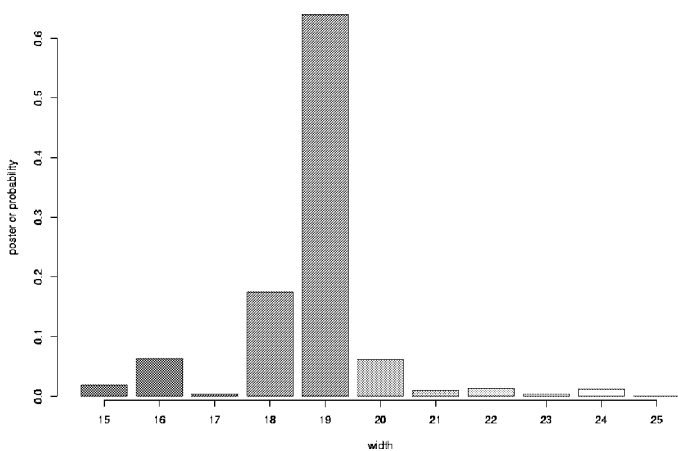


Figure 6. The Posterior Probability Distribution for Motif Width for CRP Data Indicates that the Most Likely Width is 19.

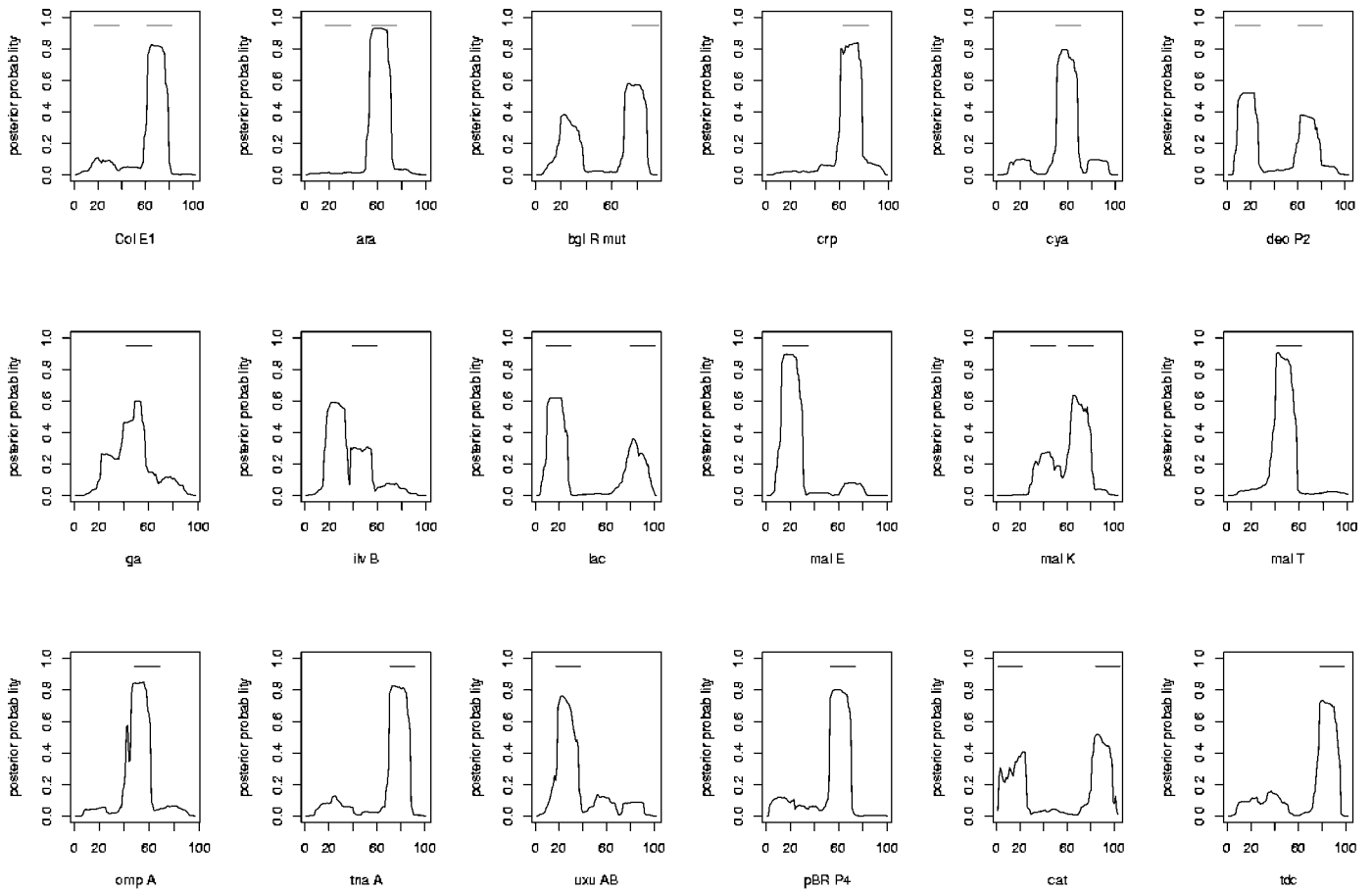


Figure 7. Posterior Motif Site Probability for 18 Sequences of CRP Data Resulting From the Gapped Motif Search Version of SDDA. Each horizontal line indicates the extent of an experimentally verified site.

There is some biological evidence that transcription factor binding sites sometimes have palindromic patterns; for example, in the consensus pattern for CRP, TGAGANNNNNNT-CACA, the last five nucleotides are the reverse complement of the first five. Palindromicity may be easily accommodated by the algorithm without introducing additional complexity, by modifying the corresponding columns of the stochastic word matrix Θ and setting $\theta_{ij} = \theta_{5-i,w-j}$ ($i = 1, \dots, 4; j = 1, \dots, w$). This would also reduce the number of parameters to be estimated. Palindromicity does not necessarily imply “dependence” between columns of the motif, however. From

measures of dependence calculated for known motifs in a number of datasets, the correlations generally appear too weak to be useful in detecting motif sites. Hence we have not yet modified the model to encompass this form of dependence.

For the *background*, the independence model is often inappropriate because of the frequent occurrences of low-complexity repeats (e.g., poly-A’s, GCGCGC. . .) and regions with different nucleotide composition. The relative robustness of the dictionary model compared with other existing pattern-finding algorithms in the presence of low-complexity regions arises from the progressive update mechanism that provides a “control” against these polynucleotide repeats. Some of our simulation studies indicate that including overrepresented oligonucleotide blocks decreases the tendency of the sampler to be trapped in polynucleotide repeats.

The proposed method has shown preliminary success in finding multiple patterns of unknown widths and patterns with varying lengths of insertions and deletions. However, certain issues relating to convergence and robustness still require attention. For simulation studies, an approximate number of iterations for convergence was determined at a point where 5 independent runs converged to the same pattern and the log-likelihood varied within a limit of 10 for at least 200 iterations.

When multiple motif types (i.e., distinct stochastic words) are present, it is of interest to know how the input of the pattern widths affect the final results. From our simulation studies, it appears that the algorithm will tend to pick out

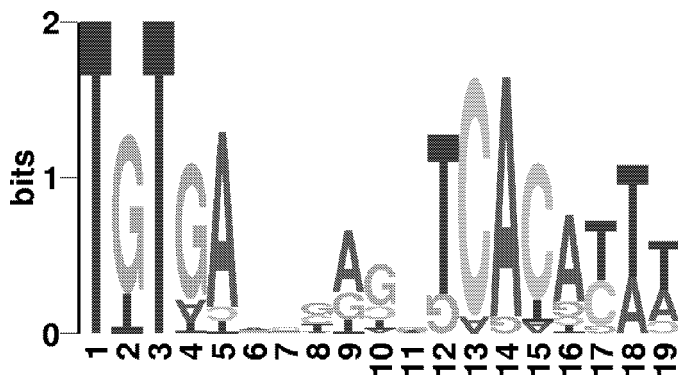


Figure 8. Sequence Logo for Alignment of CRP Binding Sites Allowing Gaps.

the “strongest” or most highly conserved motif (which can be measured in terms of its “entropy” or Kullback–Leibler divergence from the background distribution) irrespective of small differences in the motif width inputs. The length updating step of the algorithm then may be used to determine the most likely length of each found motif. The heterogeneity of background composition between parts of the DNA sequence suggests using a segmentation model (Liu and Lawrence 1999). Using a first-order Markov model with segmentation according to the heterogeneity of the background may very well be the next step toward achieving a more sensitive algorithm for finding motif patterns. It would also be of interest to see whether greater sensitivity can be achieved by introducing a such a set of stochastic “grammar rules” in addition to the stochastic dictionary.

APPENDIX: METROPOLIS UPDATE FOR MOTIF WIDTH

As before, we let $N = (N_1, \dots, N_D)$ denote the word counts for D words in the dictionary (with q being the number of single-letter words), $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_w)$ denote the column count matrix for the single word of current width w , and w_0 denote the prior mean of the width distribution. $\boldsymbol{\beta}_0 = (\beta_{01}, \dots, \beta_{0q})$ and $\mathbf{B} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_w)$ denote the prior parameters of the word count distribution and word count matrix columns. For any vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)$, $\boldsymbol{\alpha}_{[1:k]}$ denotes the vector of the first k components. We next derive the Metropolis ratio to be used in the update of motif width from w to $w + \delta$ ($\delta = \pm 1$).

The posterior predictive distribution with motif width fixed at w is

$$P(W = w | \mathbf{N}, \mathbf{c}) = \frac{\Gamma(\mathbf{N} + \boldsymbol{\beta}_0)}{\Gamma(\mathbf{N} + \boldsymbol{\beta}_0)} \frac{\Gamma(|\boldsymbol{\beta}_0|)}{\Gamma(\boldsymbol{\beta}_0)} \times \prod_{i=1}^w \frac{\Gamma(\mathbf{c}_i + \boldsymbol{\beta}_i)}{\Gamma(|\mathbf{c}_i + \boldsymbol{\beta}_i|)} \frac{\Gamma(|\boldsymbol{\beta}_i|)}{\Gamma(\boldsymbol{\beta}_i)} \frac{e^{-w_0} w_0^w}{w!}.$$

For $\delta = 1(-1)$, when the motif width is changed from w to $w + \delta$, we include (exclude) a column whose count vector is denoted by $\mathbf{H} = (h_1, \dots, h_q)^T$. Then

$$P(W = w + \delta | \mathbf{N}, \mathbf{c}) = \frac{\Gamma(\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H}) \Gamma(N_D + \beta_{0D})}{\Gamma(|\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H}| + N_D + \beta_{0D})} \times \frac{\Gamma(|\boldsymbol{\beta}_0|)}{\Gamma(\boldsymbol{\beta}_0)} \prod_{i=1}^{w+\delta} \frac{\Gamma(\mathbf{c}_i + \boldsymbol{\beta}_i)}{\Gamma(|\mathbf{c}_i + \boldsymbol{\beta}_i|)} \frac{e^{-w_0} w_0^{w+\delta}}{(w + \delta)!}.$$

Under the assumption that all of the columns of the matrix are generated independently from the same prior distribution (i.e., $\beta_{il} = \gamma_l$ for $i = 1 \dots w$, $l = 1 \dots 4$), the Metropolis ratio reduces to

$$\frac{P(W = w + \delta | \mathbf{N}, \mathbf{c})}{P(W = w | \mathbf{N}, \mathbf{c})} = \frac{\Gamma(\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H})}{\Gamma(|\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]} - \delta \mathbf{H}| + N_D + \beta_{0D})} \times \frac{\Gamma(|\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]}|)}{\Gamma(\mathbf{N}_{[1:D-1]} + \boldsymbol{\beta}_{0[1:D-1]})} \times \left\{ \frac{w_0}{(w + \frac{\delta+1}{2})} \frac{\Gamma(\mathbf{H} + \boldsymbol{\gamma})}{\Gamma(|\mathbf{H} + \boldsymbol{\gamma}|)} \frac{\Gamma(|\boldsymbol{\gamma}|)}{\Gamma(\boldsymbol{\gamma})} \right\}^\delta.$$

This can be reduced to a computationally more efficient formula (because computing gamma functions may be time-consuming) using Stirling’s approximation,

$$\frac{P(W = w + \delta | \mathbf{N}, \mathbf{c})}{P(W = w | \mathbf{N}, \mathbf{c})} \approx \frac{\prod_{j=1}^{D-1} (N_j + \beta_{0j} - \delta H_j)^{(N_j + \beta_{0j} - \delta H_j)} (N_D + \beta_{0D})^{(N_D + \beta_{0D})}}{(\sum_{j=1}^{D-1} (N_j + \beta_{0j} - \delta H_j) + N_D + \beta_{0D})^{(\sum_{j=1}^{D-1} (N_j + \beta_{0j} - \delta H_j) + N_D + \beta_{0D})}} \times \frac{(\sum_{j=1}^D (N_j + \beta_{0j}))^{(\sum_{j=1}^D (N_j + \beta_{0j}))}}{\prod_{j=1}^{D-1} (N_j + \beta_{0j})^{(N_j + \beta_{0j})}} \times \left\{ \frac{w_0}{(w + \frac{\delta+1}{2})} \frac{\Gamma(\mathbf{H} + \boldsymbol{\gamma})}{\Gamma(|\mathbf{H} + \boldsymbol{\gamma}|)} \frac{\Gamma(|\boldsymbol{\gamma}|)}{\Gamma(\boldsymbol{\gamma})} \right\}^\delta.$$

[Received May 2002. Revised October 2002.]

REFERENCES

- Bailey, T., and Elkan, C. (1994), “Fitting a mixture model by expectation maximization to discover motifs in biopolymers,” in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, eds. R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, AAAI Press, Menlo Park, California, pp. 28–36.
- Bussemaker, H. J., Li, H., and Siggia, E. D. (2000), “Building a Dictionary for Genomes: Identification of Presumptive Regulatory Sites by Statistical Analysis,” *Proceedings of the National Academy of Science USA*, *97*, 10096–10100.
- de Crombrughe, B., Busby, S., and Buc, H. (1984), “Cyclic AMP Receptor Protein: Role in Transcription Activation,” *Science*, *224*, 831–838.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Ser. B*, *39*, 1–38.
- Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993), “Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment,” *Science*, *262*, 208–214.
- Lawrence, C. E., and Reilly, A. A. (1990), “An Expectation-Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Biopolymer Sequences,” *Proteins*, *7*, 41–51.
- Liu, J. S., and Lawrence, C. (1999), “Bayesian Inference on Biopolymer Models,” *Bioinformatics*, *15*, 38–52.
- Liu, J. S., Neuwald, A. F., and Lawrence, C. E. (1995), “Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies,” *Journal of the American Statistical Association*, *90*, 1156–1170.
- Liu, J. S., Wong, W. H., and Kong, A. (1994), “Covariance Structure of the Gibbs Sampler With Applications to the Comparisons of Estimators and Augmentation Schemes,” *Biometrika*, *81*, 27–40.
- Liu, X., Brutlag, D. L., and Liu, J. S. (2001), “BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes,” in *Pacific Symposium on Biocomputing*, eds. R. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, River Edge, NJ: World Scientific Press, pp. 127–138.
- Meng, X.-L., and Wong, W. (1996), “Simulating Ratios of Normalising Constants via a Simple Identity: A Theoretical Exploration,” *Statistica Sinica*, *6*, 831–860.
- Neuwald, A. F., Liu, J. S., and Lawrence, C. E. (1995), “Gibbs Motif Sampling: Detection of Bacterial Outer Membrane Protein Repeats,” *Protein Science*, *4*, 1618–1632.
- Roth, F. R., Hughes, J., Estep, P., and Church, G. M. (1998), “Finding DNA Regulatory Motifs Within Unaligned Non-Coding Sequences Clustered by Whole-Genome mRNA Quantitation,” *Nature Biotechnology*, *10*, 939–945.
- Schneider, T. D., Stormo, G. D., Gold, L., and Ehrenfeucht, A. (1986), “Information Content of Binding Sites in Nucleotide Sequences,” *Journal of Molecular Biology*, *188*, 415–431.
- Stormo, G. D., and Hartzell, G. W. (1989), “Identifying Protein-Binding Sites From Unaligned DNA Fragments,” *Proceedings of the National Academy of Science USA*, *86*, 1183–1187.
- Tanner, M., and Wong, W. H. (1987), “The Calculation of Posterior Distributions by Data Augmentation,” *Journal of the American Statistical Association*, *82*, 528–550.