Genomewide Motif Identification Using a Dictionary Model

Chiara Sabatti and Kenneth Lange

C. Sabatti is with the Human Genetics and Statistics Departments at UCLA, 695 Charles Young Drive South, Los Angeles, CA 90095-7088, USA. Phone: (310) 794-9567. Fax: (310) 794-5446. E-mail: csabatti@mednet.ucla.edu.

K. Lange is with the Biomathematics, Human Genetics, and Statistics Departments at UCLA, 695 Charles Young Drive South, Los Angeles, CA 90095, USA. Phone: (310) 206-8076. Fax: (310) 825-8685. E-mail: klange@ucla.edu.

Abstract

This paper surveys and extends models and algorithms for identifying binding sites in non-coding regions of DNA. Binding sites control the transcription of genes into messenger RNA in preparation for translation into proteins. The base sequence of most binding sites is not entirely fixed, with the different permitted spellings collectively constituting a "motif." After summarizing the underlying biological issues, we review three different models for binding site identification. Each model was developed with a different type of dataset as reference. We then present a unified model that borrows from the previous ones and integrates their main features. In our unified model, one can identify motifs and their unknown positions along a sequence. One can also fit the model to data using maximum likelihood and maximum a posteriori algorithms. These algorithms rely on recursive formulas and the maximization/minorization principle. Finally, we conclude with a prospectus of future data analyses and theoretical research.

Keywords

Genomic sequence; expectation-maximization algorithm; maximum a posteriori; text segmentation.

I. Introduction

Computational genomics has many different goals and profits from many different scientific perspectives. One obvious goal is to find all of the genes within a genome and how they operate. This task is complicated by the segmentation of genes into exons and introns. After a gene is transcribed into mRNA, its introns are spliced out of the message. Many genes display alternative splicing patterns that eliminate some of the underlying exons as well. Regulatory regions upstream of a gene determine when and in what tissues a gene is transcribed. A second goal of genomics is to use the amino acid content of each message to deduce the structure and function of the encoded protein. A third goal is to understand how genes and gene products interact in space and time. Each of these goals benefits from the pattern recognition principles widely used in computer science and statistics. At the same time, the peculiarities of genetics demand special techniques in addition to general methods. Because the information housed in a genome is written in a distinct language, it is tempting to transfer ideas from mathematical linguistics to genomics. In our view, such a transfer is apt to be more successful for semantics than for grammar. The current paper surveys and develops a dictionary model for locating binding sites in regulatory regions. In the dictionary model, a DNA sequence is viewed as a random concatenation of words with alternative spellings.

A. The Biological Problem

DNA, the molecule that encodes genetic information, is a long polymer whose structure can be effectively be described by a sequence of letters of four types—A, C, G, and T—corresponding to the four nucleotides (or bases) adenine, cytosine, guanine, and thymine. The vast majority of human DNA is organized into 46 linear chromosomes stored in the cell nucleus. Except for the X and Y sex chromosomes, the remaining 44 chromosomes come in 22 pairs of nearly identical homologous chromosomes. The total length of the 22 consensus autosomes and the two sex chromosomes is approximately three billion bases. By comparison, the genome of the bacterium E. Coli consists of a single circular strand, five and a half million bases long. In the past decade, the complete genomes of hundreds of organisms have been sequenced, and last year a rough draft of the human genome was announced [1],[2]. These remarkable achievements make it possible to undertake whole genome analysis and compare genomes of different species.

In eukaryotes, the higher organisms with a cell nucleus, genes occupy only a small fraction of the total genome. For example in humans, recent estimates suggest that coding DNA amounts to only 1.5% of the genome. The function of the remaining portion of DNA is not entirely understood, but it is clear that it plays an important role in evolution and in the regulation of gene expression. In this paper, we focus on non-coding DNA, in particular, on regions immediately upstream of genes. These regions are often involved in regulation of transcription, the process of copying genes in preparation for their translation into proteins. In order for the transcription machinery to operate on a given gene at a given time, regulatory proteins typically must bind or unbind to specific locations upstream of the gene. Most organisms possess multiple interacting regulatory proteins, and each regulatory protein typically influences the expression of many genes. Thus, one can expect to find far fewer regulatory proteins than genes. For example, E. coli has about 4200 genes and only about 100 major regulatory proteins.

In this conceptual framework, each regulatory protein recognizes and binds to a series of DNA locations. These locations share a common sequence pattern that is specific to the protein. Because of the variation in different realizations of the same pattern, geneticists have adopted the term “motif” rather than “pattern.” This is consistent with usage in the

visual arts, where motif refers to a virtual archetype that can be rendered in a variety of different ways. Figure 1 presents some experimentally identified binding sites for CRP, a regulatory protein of major importance in E. Coli. This example clearly illustrates both the constancy and variation among realizations of the same DNA motif. All realizations span 22 bases. Although experimentation is the definitive way of identifying and characterizing binding site motifs, geneticists are keenly interested in less labor intensive methods. For that reason, bioinformatics approaches have blossomed. These are the theme of the current paper.

B. Previous Methods of Motif Recognition

As promised, we now briefly review three different approaches for identifying binding sites in DNA. Although this overview is hardly exhaustive, it does demonstrate the steady evolution of the models toward greater complexity and biological realism.

In 1990 Lawrence and Reilly [3] proposed a successful motif model in which the binding sites for a regulatory protein are assumed to have a constant length k. While this assumption is not always true, it is the rule because the usual lock and key argument of molecular biology requires all binding domains to fit into the same physical portion of the regulatory protein. At each motif position i, any of the four letters A, C, G, and T may occur. The relative frequencies of occurrence are described by a distribution $\ell_i = (\ell_{iA}, \ell_{iC}, \ell_{iG}, \ell_{iT})$ specific to position i. The letters appearing at different positions are independent. In statistical language, a motif is distributed as a product of multinomials. Motifs are contrasted to "background" sequence, where letters are chosen independently from a common distribution $\ell_0 = (\ell_{0A}, \ell_{0C}, \ell_{0G}, \ell_{0T})$. In a typical data set, each observed upstream sequence is assumed to harbor a single instance of the motif, but its exact location is unknown. Lawrence and Reilly [3] turned this missing data feature to their advantage and devised an EM algorithm for estimating both the parameter vectors $\ell_i,\ i = 1, \ldots, k$, and the locations of the motif within each upstream sequence. Later Lawrence et al. [4] elaborated a Bayesian version of the model and applied Gibbs sampling to estimate parameters and motif locations. Their Gibbs algorithm can be run on the internet at the site `http://www.bayesbio.html`.

A different type of input data motivated the research of Robison et al. [5]. Instead of

starting with a small set of sequences known to harbor the same unknown motif, they considered the entire genome of E. Coli relative to a collection of experimentally identified binding sites involving 55 regulatory proteins. Their goal was to identify all of the other binding sites for these proteins. The computational strategy in [5] is nonparametric and heuristic. A scoring function is defined for each motif. The mean m and variance v of the score values from a set of experimentally certain binding sites are recorded. The scoring function is then evaluated at each genome position, and the locations that lead to a score higher than $m - 2\sqrt{v}$ are considered putative binding sites for the protein under study. Results of this study can be viewed at `http://arep.med.harvard.edu/ecoli_matrices/`. The most appealing feature of the Robison et al. approach is its genomewide nature. One of its least appealing features is its relatively uninformative description of the binding site.

Bussemaker et al. [6] propose a third, and very different, approach to motif recognition. In their model, DNA sequence data is viewed as a concatenation of different words, each word randomly selected from a dictionary with specified probabilities. Words of length 1 play substantially the same role as background sequence in [3]. Longer words may represent binding sites. Bussemaker at al. [6], [7] describe algorithms that estimate the probabilities of all of the words in a fixed dictionary and sequentially build a dictionary from data. Their algorithms have been tested on the first ten chapters of the novel *Moby Dick* with all punctuation signs and blanks between words removed. The results are encouraging, though occasionally identified words are concatenations of two English words. A similar approach can be applied to DNA to identify regulatory sites. One defect of the model is its dubious assumption that each word has a unique spelling. If we take misspellings into account, then constructing a dictionary from scratch appears overly ambitious, particularly with a 4 letter alphabet.

In the rest of this article, we develop a model that borrows some elements from all the above approaches: (a) our description of a motif substantially coincides with that in [3]; (b) in common with [5], we seek to identify the binding sites of a predetermined set of regulatory proteins for which some experimental evidence exists; and (c) we use a likelihood description for DNA similar to that in [6]. Note that databases such as the TRANSFAC database at `http://transfac.gbf.de/TRANSFAC/`) warehouse sequence information on

experimentally identified binding sites for a variety of proteins across many organisms.

II. A Unified Model

The model we propose describes a DNA sequence as a concatenation of words, each independently selected from a dictionary according to a specific probability distribution. For us, a word is simply an irreducible semantic unit, or in the genetic context, a motif. Each word may have more than one spelling. Thus, in English, "theater" and "theatre" represent the same word. Two different words may share a spelling. For instance, "pot" may refer either to a cooking utensil or something to smoke.

In our model, a word w always has the same number of letters $|w|$. Hence, alternative spellings such as "night" and "nite" with different number of letters are disallowed. For reasons that will soon be apparent, it is convenient to group words according to their lengths and to impose a maximum word length $k_{\max}$ on our dictionary. It may be that no words of a given length $k \leq k_{\max}$ exist. For example, in the Lawrence et al. model [3] for the CPR binding site, only words of length 1 and length 22 appear. A random sequence S is constructed from left to right by concatenating random words, with each word and each spelling selected independently. The letters of a word are independently sampled from different multinomial distributions. This is known as product multinomial sampling.

In summary, our DNA model requires a static dictionary with a list of alternative spellings and probability distributions determining which words and spellings are selected. The parameters describing the model are as follows:

1. The probability of choosing a word of length k is q_k. Here k ranges from 1 to $k_{\max}$, and $\sum_{k=1}^{k_{\max}} q_k = 1$. If there are no words of length k, then $q_k = 0$.

2. Conditional on choosing a word of length k, a particular word w with $|w| = k$ is selected with probability r_w. Hence, $\sum_{|w|=k} r_w = 1$.

3. The letters of a word w follow a product multinomial distribution with success probabilities

$$\ell_{wi} \;=\; (\ell_{wiA}, \ell_{wiC}, \ell_{wiG}, \ell_{wiT})$$

for the letters A, C, T, and G at position i of w.

A randomly chosen word of length k exhibits the spelling $s = (s_1, \ldots, s_k)$ with probability

$$p(s) \;=\; \sum_{|w|=k} r_w \prod_{i=1}^{k} \ell_{wis_i}. \tag{1}$$

If some letters are missing, for instance when sequencing quality is poor, then formula (1) fails. To force its validity in the presence of missing data, we represent missing letters by question marks and introduce the additional letter probability $\ell_{wi?} = 1$ for each word w and position i within w. This missing letter convention will be used later to describe the probability of partially observed words that overlap the edges of a sequence.

An observed sequence generally contains more than one word, with unknown boundaries separating the words. Missing word boundaries are more vexing than missing letters. We will call the portion of a sequence between two consecutive word boundaries a "segment" and the set of word boundaries dividing a sequence an "ordered partition" of the sequence. For theoretical purposes, the probability of a sequence is best evaluated by conditioning on its ordered partition and then averaging the resulting conditional probability over all partitions. In numerical practice, we implement this strategy recursively via forward and backward algorithms similar to those used with hidden Markov chains.

We consider two stochastic models for generating a random sequence S by concatenating words. These models differ in how they treat edge effects. The model proposed by Bussemaker et al. [6], which we will call full text model, assumes that a sequence starts and ends with full words. This is reasonable if the sequence represents a DNA strand in its entirety, or the sequence coincides with a well delimited and biologically meaningful region such as an exon. We propose an alternative model, which we call the equilibrium model, in which the first (or last) letter of an observed sequence need not be the first (or last) letter of a word. In this model we observe a random fragment of text from an infinitely long sequence. The equilibrium model is more realistic for randomly selected DNA sequences of predetermined length.

To describe the probability of an observed sequence s under these two models, we now introduce some necessary index notation. A vector of consecutive indices

$$\sigma \;=\; (i, i+1, \ldots, j-1, j) \;=\; (i:j)$$

is called a compatible block if its length $|\sigma| = j - i + 1$ does not exceed the maximum word length $k_{\max}$. An ordered partition π of a sequence s divides the indices of s into a vector of compatible blocks $\pi = (\pi_1 \ldots, \pi_{|\pi|})$ subject to two conditions. Condition (a) applies to both models and says that if block π_i ends with index j, then block π_{i+1} begins with index $j + 1$. Condition (b) applies only to the full text model and requires the first block π_1 to begin with index 1 and the last block $\pi_{|\pi|}$ to end with the last index $|s|$ of s. Condition (c) applies only to the equilibrium model and requires the first block π_1 merely to contain index 1 and the last block $\pi_{|\pi|}$ merely to contain the last index $|s|$ of s. Each block π_i of π determines a segment $s[\pi_i]$ of s.

For instance, the ordered partition π with blocks $\pi_1 = (1, 2)$, $\pi_2 = (3, 4, 5)$, and $\pi_3 = (6)$ divides the sequence $(s_1, \ldots, s_6)$ into the three segments

$$\begin{aligned} s[\pi_1] &= (s_1, s_2) \\ s[\pi_2] &= (s_3, s_4, s_5) \\ s[\pi_3] &= (s_6). \end{aligned}$$

This particular partition is consistent with both models. The collection $\mathcal{F}$ of partitions compatible with the full text model is smaller than the collection $\mathcal{E}$ of partitions compatible with the equilibrium model. For example, the ordered partition $\pi \in \mathcal{E} \setminus \mathcal{F}$ with blocks $\pi_1 = (-1, 0, 1, 2)$, $\pi_2 = (3, 4, 5)$, and $\pi_3 = (6, 7)$ divides the sequence $(s_1, \ldots, s_6)$ into the three segments

$$\begin{aligned} s[\pi_1] &= (s_{-1}, s_0, s_1, s_2) &= (?, ?, s_1, s_2) \\ s[\pi_2] &= (s_3, s_4, s_5) \\ s[\pi_3] &= (s_6, s_7) &= (s_6, ?). \end{aligned}$$

Here we have padded s with missing letters on its left and right ends. In general, we have the constraints $\sum_{i=1}^{|\pi|} |\pi_i| = |s|$ for $\pi \in \mathcal{F}$ and $\sum_{i=1}^{|\pi|} |\pi_i| \geq |s|$ for $\pi \in \mathcal{E}$ on the sum of the segment lengths.

We now derive the likelihood of a sequence s under the full text model. Let F be the event that randomly concatenating words gives a sequence with a word boundary at position $|s|$. Because the probability of a partition $\pi \in \mathcal{F}$ is proportional to the product

of the probabilities of the lengths of the segments constituting it, we have

$$\Pr(\pi|F) \;=\; \frac{\prod_{i=1}^{|\pi|} q_{|\pi_i|}}{\sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|}}.$$

The normalizing constant here is difficult to evaluate analytically, but it can be rewritten as

$$\Pr(F) \;=\; \sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} \;=\; \sum_{m\in\mathcal{M}} \binom{m_1+\cdots+m_{k_{\max}}}{m_1\ldots m_{k_{\max}}} \prod_{k=1}^{k_{\max}} q_k^{m_k},$$

where $\mathcal{M}$ denotes the set of vectors $m = (m_1, \ldots, m_{k_{\max}})$ of nonnegative integers with weighted sum $\sum_{k=1}^{k_{\max}} k m_k = |s|$. Here m_k is the number of blocks of length k. The likelihood of the sequence under the full text model boils down to

$$\begin{aligned}
\mathcal{L}_F(s) &= \Pr(S = s|F) \\
&= \frac{\sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} \Pr(s[\pi_i] \mid \pi)}{\sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|}} \\
&= \frac{\sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} p(s[\pi_i])}{\sum_{\pi\in\mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|}}.
\end{aligned}$$

Bussemaker et al. [7] give an algorithm for computing the numerator of this likelihood, but none for computing the denominator $\Pr(F)$. They assert that it is sufficiently close to 1 for practical purposes. While this may be true in their specific context, we have observed substantial variation in $\Pr(F)$ as a function of $q = (q_1, \ldots, q_{k_{\max}})$. For example, for a dictionary containing only words of length 1 and 10 and a sequence of 800 bases, $\Pr(F)$ varies between 1 and 0.02. This makes us uncomfortable in equating it to 1. Later we will derive an efficient algorithm for computing the value of $\Pr(F)$.

Over the enormous stretches of DNA seen in all genomes, it is reasonable to suppose that the process of concatenating words has reached equilibrium at the start of any small sequence s. The equilibrium model makes it possible to assign a probability to the first segment generated by a partition $\pi \in \mathcal{E}$ covering s. Indeed, the probability that a randomly chosen position along the genome is covered by a word of length j is the ratio $jq_j/\bar{q}$, where $\bar{q} = (\sum_{k=1}^{k_{\max}} kq_k)$ denotes the length of an average word. In particular, the probability $jq_j/\bar{q}$ applies to position 1 of s. The conditional probability that position 1 of s coincides with a particular position of a covering word of length j is $1/j$. It follows that the jth index

π_{1j} of π_1 covers position 1 of s with probability $q_{|\pi_1|}/\bar{q}$. Similar considerations apply to the last block of π if we consider concatenating words from right to left rather than from left to right. In either case, we can express the probability of $\pi \in \mathcal{E}$ under the event E of equilibrium as

$$\Pr(\pi \mid E) \;=\; \frac{\prod_{i=1}^{|\pi|} q_{|\pi_i|}}{\bar{q}}.$$

It is a relatively simple exercise to check that $\sum_{\pi\in\mathcal{E}} \Pr(\pi \mid E) = 1$.

For readers dissatisfied with this intuitive explanation of equilibrium, it may help to consider a Markov chain on an infinite sequence of letters constructed by randomly concatenating words. The state of the chain X_n at position n of the sequence is a pair of integers (i, j) with $1 \le i \le j \le k_{\max}$. The integer j gives the length of the word covering position n, and the integer i gives the position of n within that word. The actual letter at n is irrelevant. It is easy to prove that this finite-state chain is irreducible and, provided there is at least one single-letter word, aperiodic. Let λ_{nij} be the probability that the chain occupies state (i, j) at position n. Elementary reasoning yields the one-step recurrence

$$\lambda_{nij} \;=\; 1_{\{i>1\}}\lambda_{n-1,i-1,j} + 1_{\{i=1\}} \sum_{k=1}^{k_{\max}} \lambda_{n-1,kk} q_j,$$

and standard theory for a Markov chain says that the limits $\lim_{n\to\infty} \lambda_{nij} = \lambda_{ij}$ exist and do not depend on the initial distribution of the chain. Because the probability distribution $\lambda_{ij} = q_j/\bar{q}$ obviously satisfies the one-step recurrence, this validates our claimed equilibrium model.

By allowing missing letters and partitions that straddle the ends of s, we can write the likelihood of s under the equilibrium model as

$$\begin{aligned} \mathcal{L}_E(s) &= \Pr(S = s|E) \\ &= \frac{1}{\bar{q}} \sum_{\pi\in\mathcal{E}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} p(s[\pi_i]). \end{aligned}$$

Again, this formula is ill adapted to computing. It is noteworthy, however, that the normalizing constant is vastly simpler. Furthermore, the likelihood under the full text model can be viewed as a conditional probability in the equilibrium model in the sense that $\mathcal{L}_F(s) = \Pr(S = s \mid E, F)$.

III. Algorithms for Likelihood Evaluation

Our likelihood algorithms resemble Baum's forward and backward algorithms from the theory of hidden Markov chains [8], [9]. For the sake of simplicity, we first consider the full text likelihood of s. Let B_i be the event that a word ends at position i. The forward algorithm updates the joint probabilities

$$f_i \;=\; \Pr(S[1:i] = s[1:i], B_i),$$

and the backward algorithm updates the conditional probabilities

$$b_i \;=\; \Pr(S[i:n] = s[i:n] \mid B_{i-1})$$

for $n = |s|$.

The forward algorithm initializes $f_0 = 1$ and iterates according to

$$f_i \;=\; \sum_{k=1}^{\min\{k_{\max},i\}} f_{i-k} q_k p(s[i-k+1:i])$$

in the order $i = 1, \ldots, n$. At the last step, f_n equals the numerator of $\mathcal{L}_F(s)$, that is $\sum_{\pi \in \mathcal{F}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} \Pr(s[\pi_i] \mid \pi)$. The forward algorithm for computing the denominator is similar except that it iterates via

$$f_i \;=\; \sum_{k=1}^{\min\{k_{\max},i\}} f_{i-k} q_k,$$

ignoring the letter content of the sequence. The backward algorithm begins with $b_{n+1} = 1$ and updates

$$b_i \;=\; \sum_{k=1}^{\min\{k_{\max},n+1-i\}} b_{i+k} q_k p(s[i:i+k-1])$$

in the reverse order $i = n, \ldots, 1$. At the last step, we recover the numerator of $\mathcal{L}_F(s)$ as b_1. Finally, the backward algorithm for the denominator iterates via

$$b_i \;=\; \sum_{k=1}^{\min\{k_{\max},n+1-i\}} b_{i+k} q_k.$$

To derive these updates, we simply concatenate an additional segment to one of the current partial sequences, assuming that the entire sequence starts and ends with full words.

Bussemaker et al. [7], [6] give the backward and forward algorithms for the numerator but omit the algorithms for the denominator of $\mathcal{L}_F(s)$.

The forward and backward algorithms for the equilibrium likelihood are similar but more complicated. The forward algorithm commences with $f_i = 1/\bar{q}$ for $i = 1-k_{\max}, \ldots, 0$. This expanded set of initial values reflects the variety of starting points for segments containing position 1. The remaining joint probabilities are determined by

$$f_i = \sum_{k=\max\{1,i+1-n\}}^{k_{\max}} f_{i-k} q_k p(s[i-k+1:i])$$

for $i = 1, \ldots, n + k_{\max} - 1$. This is precisely the update used for the numerator of the full text likelihood when $i \leq n$. When $i > n$, the requirement that the last word must contain position n limits the range of summation of k to $i - k < n$. The equilibrium likelihood amounts to $\mathcal{L}_E(s) = f_n + \cdots + f_{n+k_{\max}-1}$. The backward algorithm begins with $b_i = 1$ for $i = n+1, \ldots, n+k_{\max}$ and iterates according to

$$b_i = \sum_{k=\max\{1,2-i\}}^{k_{\max}} b_{i+k} q_k p(s[i:i+k-1])$$

for $i = n, \ldots, 2 - k_{\max}$. In this case, $\mathcal{L}_E(s) = (b_{2-k_{\max}} + \cdots + b_1)/\bar{q}$.

As a trivial example, consider $s = (s_1)$ and $k_{\max} = 2$. Then the updates

$$\begin{aligned}
f_1 &= f_{-1} q_2 \sum_{|w|=2} r_w \ell_{w2s_1} + f_0 q_1 \sum_{|w|=1} r_w \ell_{w1s_1} \\
f_2 &= f_0 q_2 \sum_{|w|=2} r_w \ell_{w1s_1} \\
b_1 &= b_2 q_1 \sum_{|w|=1} r_w \ell_{w1s_1} + b_3 q_2 \sum_{|w|=2} r_w \ell_{w1s_1} \\
b_0 &= b_2 q_2 \sum_{|w|=2} r_2 \ell_{w2s_1}
\end{aligned}$$

both lead to the equilibrium likelihood

$$\mathcal{L}_E(s) = \frac{1}{q_1 + 2q_2} \left[q_1 \sum_{|w|=1} r_w \ell_{w1s_1} + q_2 \sum_{|w|=2} r_w (\ell_{w1s_1} + \ell_{w2s_1}) \right].$$

For long sequences, one has to rescale to prevent underflows. Rescaling is a general device that applies to linear iteration. Suppose x^i is a vector sequence generated by the recurrence $x^{i+1} = M^i x^i$ for matrices M^i. In rescaling we replace this sequence by another

sequence y^i starting with $y^0 = x^0$ and satisfying $y^{i+1} = c_i^{-1} M^i y^i$. The positive constant c_i is typically taken to be $\|y^i\|$ for some norm. One can easily show by induction that $x^i = (\prod_{j=0}^{i-1} c_j) y^i$. If want the logarithm of some positive inner product $v^* x^i$, then we compute the logarithm of the positive inner product $v^* y^i$ and add the compensating sum $\sum_{j=0}^{i-1} \ln c_j$. Readers can supply the details of how this applies to computing loglikelihoods under the forward and backward algorithms.

Intermediate values from the forward and backward algorithms are stored for a variety of reasons. For instance under the equilibrium model, we may want the conditional probability that the sequence s contains a segment extending from index i to index j. This probability can be expressed as

$$\kappa_{ij} \;=\; \frac{f_{i-1} q_{j-i+1} p(s[i:j]) b_{j+1}}{\mathcal{L}_E(s)}. \tag{2}$$

The restriction that a particular word w fills this segment has conditional probability

$$\rho_{ij}(w) \;=\; \frac{f_{i-1} q_{j-i+1} r_w \prod_{k=1}^{j-i+1} \ell_{w k s_{i+k-1}} b_{j+1}}{\mathcal{L}_E(s)}. \tag{3}$$

These particular conditional probabilities are pertinent to estimation of the parameter vectors q, r, and ℓ describing the model.

IV. Parameter Estimation via the MM Algorithm

A Bayesian approach to parameter estimation is attractive because it allows the incorporation of prior information on experimentally identified binding sites. The application of a 0-1 loss function in similar classification problems suggests that we maximize the posterior density. This is proportional to the product of the prior density and the likelihood. There is no harm in selecting the prior density from a convenient functional family provided we match its parameters to available prior data. Since the presence of the prior adds little complexity to optimization of the likelihood itself, we will first discuss maximum likelihood estimation and then indicate how it can be modified to accommodate a prior.

To maximize the complicated likelihood function $\mathcal{L}_E(s \mid q, r, \ell)$, we resort to an MM algorithm [10]. This iterative optimization principle maximizes a target function $f(x)$ by taking a current iterate x^m and constructing a minorizing function $g(x \mid x^m)$ in the sense that $g(x \mid x^m) \leq f(x)$ for all x and $g(x^m \mid x^m) = f(x^m)$. The next iterate x^{m+1} is chosen

to maximize $g(x \mid x^m)$. This choice of x^{m+1} guarantees that $f(x^{m+1}) \geq f(x^m)$. For the MM strategy to be successful, maximization of $g(x \mid x^m)$ should be easy.

The best known class of MM algorithms consists of the EM algorithms. All EM algorithms revolve around the notion of missing data. In the current setting, the missing data are the partition π segmenting the sequence and the words assigned to the different segments of s generated by π. In the E step of the EM algorithm, one constructs a minorizing function to the loglikelihood by taking the conditional expectation of the complete data loglikelihood with respect to the observed data. For the equilibrium model, the complete data likelihood is

$$\frac{1}{\bar{q}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} r_{w_i} \prod_{j=1}^{|w_i|} \ell_{w_i j s_{\pi_{ij}}},$$

where segment $s[\pi_i]$ is assigned word w_i, and π_{ij} denotes the jth index of π_i. Let M_k be the number of segments of length k, N_w be the number of appearances of word w, and L_{wjt} be the number of letters of type t occurring at position j of the segments assigned word w. In this notation, the complete data loglikelihood is expressed as

$$\sum_{k=1}^{k_{\max}} M_k \ln q_k + \sum_w N_w \ln r_w + \sum_{w,i,j} L_{wij} \ln \ell_{wij} - \ln \bar{q}.$$

The conditional expectations of the counts M_k, N_w, and L_{wij} given $S = s$ are readily evaluated as

$$\begin{aligned}
\mathrm{E}\left(M_k \mid S = s, q, r, \ell\right) &= \sum_{i=-k+2}^{|s|} \kappa_{i,i+k-1} \\
\mathrm{E}\left(N_w \mid S = s, q, r, \ell\right) &= \sum_{i=-|w|+2}^{|s|} \rho_{i,i+|w|-1}(w) \\
\mathrm{E}\left(L_{wjt} \mid S = s, q, r, \ell\right) &= \sum_{i=-|w|+2}^{|s|} 1_{\{s_{i+j-1}=t_j\}} \rho_{i,i+|w|-1}(w)
\end{aligned}$$

using equations (2) and (3).

The EM algorithm for hidden multinomial trials updates a success probability by equating it to the ratio of the expected number of successes to the expected number of trials given the observed data and the current parameter values [11]. This recipe translates into

the iterates

$$\begin{aligned} r_w^{m+1} &= \frac{\mathrm{E}\left(N_w \mid S = s, q^m, r^m, \ell^m\right)}{\mathrm{E}\left(M_{|w|} \mid S = s, q^m, r^m, \ell^m\right)} \\ \ell_{wjt}^{m+1} &= \frac{\mathrm{E}\left(L_{wjt} \mid S = s, q^m, r^m, \ell^m\right)}{\mathrm{E}\left(N_w \mid S = s, q^m, r^m, \ell^m\right)}. \end{aligned}$$

Updating the segment probabilities q_k is more problematic. Because the surrogate function created by the E step separates the q_k parameters from the remaining parameters, it suffices to maximize the function

$$g(q \mid q^m) = \sum_{k=1}^{k_{\max}} \mathrm{E}\left(M_k \mid S = s, q^m, r^m, \ell^m\right) \ln q_k - \ln\Big(\sum_{k=1}^{k_{\max}} k q_k\Big)$$

subject to the constraints $q_k \geq 0$ and $\sum_{k=1}^{k_{\max}} q_k = 1$. To our knowledge, this problem can not be solved in closed form. It is therefore convenient to undertake a second minorization exploiting the inequality $\ln x \leq \ln y + x/y - 1$. Application of this inequality produces the minorizing function

$$h(q \mid q^m) = \sum_{k=1}^{k_{\max}} \mathrm{E}\left(M_k \mid S = s, q^m, r^m, \ell^m\right) \ln q_k - \ln\Big(\sum_{k=1}^{k_{\max}} k q_k^m\Big) - c^m \sum_{k=1}^{k_{\max}} k q_k + 1$$

with $c^m = 1/(\sum_{k=1}^{k_{\max}} k q_k^m)$.

The function $h(q \mid q^m)$ still resists exact maximization, but at least it separates the different q_k. To maximize $h(q \mid q^m)$, we employ the method of Lagrange multipliers. This entails finding a stationary point of the Lagrangian

$$h(q \mid q^m) + \lambda\Big(\sum_{k=1}^{k_{\max}} q_k - 1\Big).$$

Differentiating the Lagrangian with respect to q_k yields the equation

$$0 = \frac{e_k^m}{q_k} - c^m k + \lambda,$$

where

$$e_k^m = \mathrm{E}\left(M_k \mid S = s, q^m, r^m, \ell^m\right).$$

The components

$$q_k = \frac{e_k^m}{c^m k - \lambda}$$

of the stationary point involve the unknown Lagrange multiplier λ. Fortunately, λ is determined by the constraint

$$1 \;=\; \sum_{k=1}^{k_{\max}} q_k \;=\; \sum_{k=1}^{k_{\max}} \frac{e_k^m}{c^m k - \lambda}.$$

The right hand side of the second of these two equations is strictly monotone in λ and equals 1 at exactly one point. Any of a variety of numerical methods will yield this point. In practice, we use bisection, which is easy to program and highly reliable. Its relatively slow rate of convergence is hardly noticeable amid the other more computationally intensive tasks.

We now briefly describe how a slight modifications of these algorithms permit maximization of the posterior density. The general idea is to put independent priors on q, r, and ℓ. Because Dirichlet densities are conjugate priors for multinomial densities, it is convenient to choose Dirichlet priors. Therefore, consider a Dirichlet prior

$$\frac{\Gamma(\sum_{k=1}^{k_{\max}} \alpha_k)}{\prod_{k=1}^{k_{\max}} \Gamma(\alpha_k)} \prod_{k=1}^{k} q_k^{\alpha_k - 1}$$

for q, say. In selecting the prior parameters $\alpha_1, \ldots, \alpha_{k_{\max}}$, is helpful to imagine a prior experiment and interpret $\alpha_k - 1$ as the number of successes of type k in that experiment. In this imaginary setting, there is nothing wrong with specifying a fractional number of successes. The sum $\sum_{k=1}^{k_{\max}} \alpha_k - k_{\max}$ gives the number of trials in the prior experiment and hence determines the strength of the prior. If little or no prior information is available, then one can set all $\alpha_k = 1$. This yields a posterior density that coincides with the likelihood. Setting all $\alpha_k = 2$ regularizes estimation and deters estimates of q_k from approaching the boundary value 0.

In summary, adding a Dirichlet prior to a multinomial likelihood corresponds to adding $\alpha_k - 1$ pseudo-counts to category k of the observed data. Hence, if we focus on estimating q, then in the MM algorithm just described we replace M_k by $M_k + \alpha_k - 1$. Everything else about the algorithm remains the same. Similar considerations apply to estimation of the parameter vectors r and ℓ except we deal with product multinomials rather than multinomials. This distinction entails substituting products of independent Dirichlet priors for a single Dirichlet prior.

V. Discussion and Conclusion

In the current paper, we have explored some of the conceptual issues involved in applying the dictionary model to motif finding. A clearer understanding of these issues is crucial in formulating algorithms that make statistical sense. Limited space and an impending writing deadline do not permit us the luxury of data analysis. However, we have coded the MM algorithm for the equilibrium model in Fortran 95. The code performs well on sample problems, but more extensive testing is necessary.

The model of Lawrence and Reilly [3] and its extensions [4] successfully identify sites in short sequences locally enriched for a given binding site. As the whole genomes of more species become available, there is a growing interest in global methods of motif identification. The work of Robison et al. [5] and Bussemaker et al. [6] is motivated by this aim. Our current synthesis points in the same direction. In the long run, comparison of homologous sequences from related species is apt to provide the best leverage in identifying binding sites [12]. Adaptation of the dictionary model to this purpose is a natural research goal.

Other more modest theoretical extensions come to mind. For example, one could search for protein motifs by substituting amino acids for bases. In noncoding regions of DNA, it might be useful to model binding site motifs that are palindromes. This puts constraints on the parameters in the product multinomial distributions for letters within a give word. The independent choice of letters in a word is also suspect. A Markov chain model might be more appropriate in some circumstances. Finally, our model assumes that consecutive words are selected independently. However, it is reasonable to posit that multiple proteins interact in regulating expression. This assumption translates into the co-occurrence of binding sites. Co-occurrence can be investigated within the framework of the unified model by monitoring the posterior probabilities of binding sites and checking whether these tend to be cross correlated as a function of position along a sequence.

We have assumed a static dictionary. Bussemaker at al. [7], [6] tackle the problem of dictionary construction. Although their methods are elegant, it is unclear how well they will perform in the presence of alternative spellings. One of the virtues of the unified model is that it encourages exploration of alternative spellings and estimation of letter

frequencies within words.

Many interesting probabilities can be computed in the unified model. For example, suppose we want to compute the probability that a particular word w is missing from a sequence s. Let $\mathcal{L}_{uw}(s)$ be the result of applying the forward or backward algorithms with w omitted throughout in the updates of the f_i or b_i. The ratio $o_w(s) = \mathcal{L}_{uw}(s)/\mathcal{L}(s)$ then supplies the required conditional probability. This suggests defining a motif distance $d_M(s,t)$ between two sequences s and t by the equation

$$d_M(s,t)^2 \;=\; \sum_w [o_w(s) - o_w(t)]^2.$$

This definition makes it possible to compare vastly different sequences with an emphasis on uncommon regulatory elements and a de-emphasis on random background. It might be especially illuminating for cross-species comparisons between homologous regions of human and mouse DNA. It would also be useful for correlating expression profiles from genes residing on different sequences.

References

[1] International human genome sequencing consortium, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 2001.

[2] J. C. Vender et al. "The sequence of the human genome," *Science*, vol. 291, pp. 1304–1351, 2001.

[3] C. E. Lawrence, and A. A. Reilly, "An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins*, vol. 7, pp. 41–51, 1990.

[4] C. E. Lawrence, S. F. Altschul, M. S. Bogouski, J. S. Liu, A. F. Neuwald, and J. C. Wooten, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," *Science*, vol. 262, pp. 208–214, 1993.

[5] K. Robison, A. M. McGuire, and G. M. Church, "A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete Escherichia coli K12 genome," *Journal of Molecular Biology*, vol. 284, pp. 241–254, 1998.

[6] H. J. Bussemaker, H. Li, and E. D. Siggia, "Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis," *PNAS*, vol. 97, pp. 10096–10100, 2000.

[7] H. J. Bussemaker, H. Li, and E. D. Siggia, "Regulatory element detection using a probabilistic segmentation model," in *Proc. Int. Conf. Intell. Syst. Mol. Biol. 8*, 2000, pp. 67–74.

[8] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[9] P. A. Devijver, "Baum's forward-backward algorithm revisited," *Pattern Recognition Letters*, vol. 3, pp. 369–373, 1985.

[10] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions (with discussion)," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 1–59, 2000.

[11] K. Lange, *Mathematical and Statistical Methods for Genetic Analysis*, 2nd ed. New York: Springer-Verlag, 2002.

[12] McCue, L. A. W. Thompson, C. S. Carmack, M. P. Ryan, J. S. Liu, V. Derbyshire, and C. E. Lawrence, "Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes," *Nucleic Acids Research* vol. 29, pp 774–782, 2001.

attcgtgatagctgtcgtaaag
ttttgttacctgcctctaactt
aagtgtgacgccgtgcaaataa
tgccgtgattatagacactttt
atttgcgatgcgtcgcgcattt
taatgagattcagatcacatat
taatgtgacgtcctttgcatac
gaaggcgacctgggtcatgctg
aggtgttaaattgatcacgttt
cgatgcgaggcggatcgaaaaa
aaattcaatattcatcacactt

Fig. 1. Experimentally identified binding sites for CRP mentioned at the website: `http://arep.med.harvard.edu/ecoli_matrices/`. Each row represent one binding site of length 22.