Finding motifs in biological sequence data

Joe Herman*

This practical builds on topics you have studied over the course of the last couple of weeks, including Markov chains, Monte Carlo simulation, parameter estimation, Bayesian statistics and model comparison.

In groups of three, you will be implementing an algorithm for detecting regulatory motifs in DNA sequences, and applying this to the datasets we have provided. To give you a head-start, we will be distributing skeleton code written in MATLAB. On Friday each group will be giving a 10-15 minute presentation; this will be an opportunity for you to demonstrate your understanding of the problem, as well as presenting results obtained from simulations. The assessment will be based on each group's progress on the Thursday, as well as the presentation.

This document contains the necessary background material for the practical, along with the set of tasks that we will be asking you to attempt; we recommend you read this thoroughly before diving into the MATLAB code. Some of the more technical details are placed in the Appendix at the end of this document, for the enjoyment of the more mathematically inclined reader. Don't worry if you don't manage to finish all the tasks; the main objective is for you to demonstrate that you have understood the key concepts of the approach. Equally, if you finish everything, feel free to explore some additional ideas of your own!

1 Regulatory motifs

Gene expression is controlled by a wide range of regulatory mechanisms and feedback loops, allowing organisms to respond to their changing environments. The expression level of a particular gene may be increased or decreased by the binding of regulatory proteins, including transcription factors, to specific target sequences within the *promoter* region of the gene. Promoters are non-coding DNA regions located near the gene, typically upstream, which act together with other regulatory regions (enhancers, silencers etc.) to influence the level of expression.

The identification of such regulatory regions is an essential step in understanding the mechanisms of gene regulation, and this has been the target of much research over the past couple of decades. With the advent of huge genomic databases, we now have access to a wealth of sequence data to aid with discovery of functionally important patterns. However, dealing with such large databases requires the development of sophisticated statistical techniques for distinguising between signal and random noise.

*herman@stats.ox.ac.uk

Given that we expect functionally important regions of the genome to evolve at a slower rate than non-functional regions, one approach to predicting transcription factor binding-sites and other regulatory regions is to look for highly conserved segments in a set of orthologous genes. A *motif* is a short (5-30 bp) pattern of nucleotides that is found to occur near genes. Many such motifs have been identified as transcription-factor binding sites. However, the probability of observing a motif of length K by random chance becomes quite high when K is small, such that motifs of less than around 10bp are often unlikely to be statistically significant.

1.1 Position weight matrices

Although motifs are typically highly conserved, there are often bases within the motif that can vary without significantly affecting the strength of transcription factor binding. As such, rather than a being a fixed sequence, a motif is best characterised by a *position weight matrix* (PWM), which is a $4 \times K$ matrix, M; each column of this matrix contains the probabilities of finding each of the four bases A, C, G, T at a particular site in the motif.

To assess the non-randomness of a particular PWM, we can compute its *information* content, which gives the divergence of the position weight matrix from a given background distribution, which we denote by G. The information for a site k is defined as

$$I^{(k)}(M \mid\mid G) = H^{(k)}(G) - H^{(k)}(M) \tag{1}$$

where

$$H^{(k)}(M) = -\sum_{\chi=A,C,G,T} M_{\chi}^{(k)} \log M_{\chi}^{(k)} \tag{2}$$

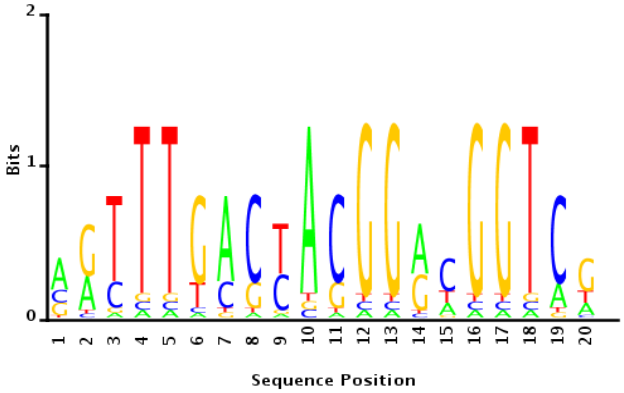
is the *Shannon entropy* of the profile at site k, and $M_{\chi}^{(k)}$ is the probability of finding base χ at position k of the motif. The background entropy will be constant for each site, and is given by

$$H^{(k)}(G) = -\sum_{\chi=A,C,G,T} G_{\chi} \log G_{\chi} \tag{3}$$

where G_{χ} is the background probability of nucleotide χ.

The maximum possible information is when each site in the motif has no uncertainty associated with it, meaning that each column of M contains a single 1 and three zeros. In this case the entropy of each site is zero, and the information is equal to the background entropy (which is $2 \log 2$ for a uniform background distribution on nucleotides).

A graphical representation that is commonly used to represent a position weight matrix is the *sequence logo*, in which the height of a stack of symbols at each position corresponds to the information at that site, while each symbol is scaled according to its relative frequency. In MATLAB, sequence logos can be generated from position weight matrices using the function `seqlogo_fig.m` provided.



2 Detecting motifs: a Bayesian approach

The problem of finding motifs in a set of sequences is very similar to that of multiple sequence alignment, in that it is necessary to find the corresponding locations at which the motif starts in each sequence. Here we will explore a classic algorithm (originally proposed by Lawrence et al. in 1993) for simultaneously inferring the PWM for a motif (denoted by M), and its location in a set of sequences of interest (denoted by s). We will be addressing this question within a Bayesian framework, such that the main object of inference is the *posterior distribution* of the unknown parameters given the sequence data (denoted by X). Using Bayes' rule, we have

$$p(s, M \mid X) = \frac{p(X \mid s, M)p(s, M)}{p(X)} \tag{4}$$

where $p(s, M)$ is the prior distribution on the PWM and motif starting locations, and $p(X \mid s, M)$ is the likelihood of the data as a function of s and M. We will examine specific forms for the likelihood and prior later.

2.1 Computational inference: Gibbs sampling

For many problems of interest, the posterior distribution is too complex to be represented explicitly, meaning that we must use computational techniques in order to sample from the distribution. Here we will be using Markov chain Monte Carlo (MCMC) simulation, which works by setting up a Markov chain whose stationary distribution is the target posterior distribution of interest.

One way of achieving this is through the use of a *Gibbs sampler*, which involves cycling through the parameters of interest, drawing a new sample for each parameter from its *full conditional distribution*, i.e. the posterior distribution given the current values of the other parameters. In our case, this means first sampling the starting locations for the motif in the sequences of interest from the full conditional $p(s \mid X, M)$, and then sampling the PWM for the motif from its full conditional $p(M \mid X, s)$. If this process is repeated long enough, the Markov chain will converge, and the samples of s and M will be from the

desired posterior distribution $p(s, M \mid X)$. Part of the art of MCMC involves determining exactly what is 'long enough', as well as exploring ways of speeding up the convergence process.

3 Likelihood

The PWM can be used to define a model for generating motifs, whereby the probability of observing a particular motif is simply given by the product of the relevant entries in the PWM.

$$p(X \mid M) = \prod_{k=1}^{K} M_{X_k}^{(k)} \tag{5}$$

where X_k is the kth character of X.

3.1 Motifs within larger sequences

We are interested in the case where the motif forms a subsequence in a larger sequence. We will model the non-motif regions of the sequence using the background PWM, denoted by G. The joint likelihood of all the observed sequences given the starting locations of the motif in each sequence, $s = s_1, \ldots, s_N$, the position weight matrix, M, and background distribution G, can be written in terms of the following product

$$\begin{aligned} p(X^{(1)}, \ldots, X^{(N)} \mid s, M, G) &= \prod_{i=1}^{N} p(\text{Sites } s_i \text{ to } s_i + K - 1 \text{ in sequence } i \text{ come from } M) p(\text{Other sites come from } G) \\ &= \prod_{i=1}^{N} \underbrace{\prod_{j=1}^{s_i - 1} p(X_j^{(i)} \mid G)}_{\text{not motif}} \underbrace{\prod_{j=s_i}^{s_i + K - 1} p(X_j^{(i)} \mid M)}_{\text{motif}} \underbrace{\prod_{j=s_i + K}^{L_i} p(X^{(i)} \mid G)}_{\text{not motif}} \end{aligned}$$

4 Prior distributions

The prior distribution $p(s, M)$ for the unknown parameters reflects our initial state of belief about the distribution of these parameters. Unless there is reason to suspect otherwise, it is likely that our initial belief is that the motifs are equally likely to be found in any region of the sequences of interest, such that the prior on s will be uniform.

As discussed in the Appendix, the posterior probability of the motif starting at site s_i is then given by the ratio of probabilities of generating the data at sites s_i to $s_i + K - 1$ from the motif model, M, versus the background model, G

$$p(s_i = n \mid M, G, X) \propto \prod_{k=n}^{n+K-1} p(X_k^{(i)} \mid M) / p(X_k^{(i)} \mid G)$$

In the case of M, we will use a *conjugate prior*, meaning that the prior multiplied by the likelihood has the same form as the prior, simplifying the

mathematical analysis. The conjugate prior for the multinomial distribution is the *Dirichlet distribution*.

As discussed in the Appendix, this yields the following posterior for each column in the motif

$$p(M^{(k)} \mid X^{(1)}, \ldots, X^{(N)}, s) = Dirichlet(M \mid \alpha + f^{(k)}(s, X))$$

where α is a length-four vector that reflects our prior belief about the probability of each nucleotide, and $f^{(k)}$ is a length-four vector with entries corresponding to the number of times each nucleotide is seen at site k of the motif, given the current starting locations, s. Note that α need not be the same as the background distribution, G. In the absence of any other information, we might set $\alpha_i = a$, where a is a constant.

Since it is easy to sample directly from a Dirichlet distribution (for example as done in the code `dirichlet.m`, which you have been provided with), we can use this to sample each column from its full conditional, given the starting locations, s, and assuming that the columns are independent.

5 The Gibbs sampling algorithm

Based on the ideas discussed above, we can define the following algorithm for detecting motifs in a set of sequences:

Algorithm 1: *Gibbs sampler for inference on s and M*

1. Choose a random motif starting location, s_i, for each sequence.
2. Sample a PWM, M, from the full conditional, $p(M \mid s, X)$
3. For some number of iterations:

   For each sequence $X^{(i)}$ of length L_i (for $i = 1, \ldots, N$):

   i. For each position $n = 1, \ldots, L_i - K + 1$ in the sequence, compute the ratio of probabilities of observing the characters from n to $n + K - 1$ from the motif, M, versus from the background distribution, G:

   $$\pi_n = \prod_{k=n}^{n+K-1} p(X_k^{(i)} \mid M) / p(X_k^{(i)} \mid G) \qquad (6)$$

   ii. Set $s_i = n$ with probability proportional to π_n.

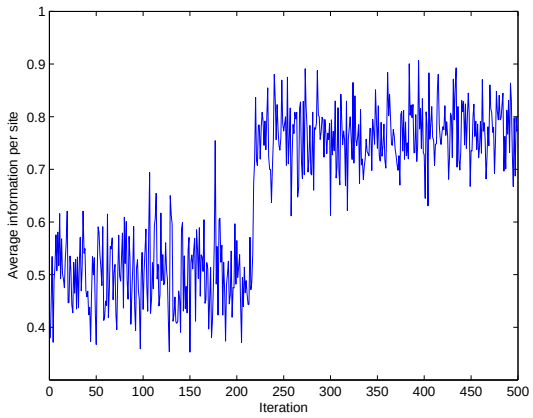
   iii. Sample M from the full conditional $p(M \mid s, X)$

(The derivation of steps 3.*i*. and 3.*ii*. is explained in the Appendix.)

5.1 Convergence

As with other MCMC techniques, the Markov chain underlying this Gibbs sampling strategy will take some time (known as the *burn-in period*) to converge to the stationary distribution, at which point the samples will represent true

draws from the joint posterior of all the parameters. One way of diagnosing the convergence of the chain in this case is to monitor summary statistics such as the average information per site as the iterations proceed. In the case that the algorithm detects a set of motifs, the uncertainty in the inferred PWM will decrease over time (and hence the information will increase), until it converges on a steady state of some kind.



The information content may also appear constant if motifs are being randomly selected from the background distribution. As shown above, when a motif is found, the average information per site will increase, and reach a new steady state. However, with all MCMC techniques it is very important to re-run the sampler from a different starting configuration (for example with different motif starting locations, or a different prior distribution) to see if it ends up in the same area of the parameter space.

5.2 Summarising the posterior

After converging, the Gibbs sampler will produce samples from the posterior for as long as the chain continues to run. When the parameters of interest are multidimensional, it is often convenient to examine summary statistics of this posterior in order to assess the form of the distribution.

In our case, we are interested in inference about the PWM, and the starting locations of the motif in each sequence. One way of summarising the posterior uncertainty in the PWM is to compute the *posterior mean*, $\bar{M} = \frac{1}{T}\sum_{t=1}^{T} M(t)$, where $M(t)$ is the estimate of the PWM at the t^{th} iteration of the sampler, and T is the total number of samples taken.

Since means can be a poor summary in multimodal distributions, we may wish to instead record some kind of modal value. In particular, we may wish to store the PWM that has the maximum likelihood ratio, a_n, as described in equation (13) (equivalent to the posterior mode), along with the associated motif starting locations; alternatively, if we suspect that the motif is going to be highly conserved, we might be more interested in the PWM with the highest information per site (equivalent to the minimum entropy).

5.3 When some of the sequences don't contain the motif

One of the weaknesses of the original Gibbs sampling approach of Lawrence *et al.* is that it presumes that all of the sequences contain a copy of the motif. In realistic situations this may not be known in advance, especially considering that the object of the analysis may be to detect an as-yet-unknown motif.

In order to allow for some of the sequences to have zero copies of the motif, we can make a simple adaptation to the original scheme. We will introduce a parameter μ that describes the probability of any sequence containing a motif, and an indicator variable z_i that is equal to one if sequence i contains a motif, and zero otherwise. When $\mu = 1$, all the sequences contain a motif, but for $\mu < 1$ some of the sequences may end up with $z_i = 0$, i.e. no motif. In the Appendix we show the form of the joint full conditional for the starting locations and the presence/absence vector z. The Gibbs sampling algorithm now contains extra steps to sample z, given a fixed value for μ:

Algorithm 2: *Gibbs sampler for inference on s, z and M*

1. Choose a random motif starting location, s_i, for each sequence.
2. Randomly initialise the indicator variables z_i for each sequence.
3. Sample a PWM, M, from the full conditional, $p(M \mid s, z, X)$
4. For some number of iterations:

   For each sequence $i = 1, \ldots, N$:

   i. For each position $n = 1, \ldots, L_i - K + 1$ in the sequence, compute the ratio of probabilities of observing the characters from n to $n + K - 1$ from the motif, M, versus from the background distribution, G:

   $$\pi_n = \prod_{j=n}^{n+K-1} p(X_j^{(i)} \mid M)/p(X_j^{(i)} \mid G)$$

   ii. Set z_i to zero with probability

   $$\frac{(L_i - K + 1)(1 - \mu)}{(L_i - K + 1)(1 - \mu) + \mu \sum_n \pi_n}$$

   iii. If $z_i = 1$, set $s_i = n$ with probability proportional to π_n.

   iv. Sample M from the full conditional $p(M \mid s, z, X)$.

(The reasoning behind step 4.*ii*. is explained in the Appendix.)

To help judge convergence when only some of the sequences contain the motif, it may be useful to also visualise which sequences contain the motif at each MCMC iteration (this information is contained in the `S` variable in the accompanying code).

5.4 Bayesian inference on the motif probability

A clear issue with the approach in Algorithm 2 is that the choice of value for μ could make a big difference to the results, but we may not know how many sequences are likely to contain the motif. To deal with this issue, we can treat μ as another random variable, and carry out Bayesian inference on this at the same time as the other unknown variables.

Again, we can use a conjugate prior, in this case a beta distribution (which is the same as a two-dimensional Dirichlet). As discussed in the Appendix, the posterior for μ is then also beta

$$p(\mu \mid z) = \text{Beta}\left(\mu \;\middle|\; \beta_0 + \sum_i z_i, \beta_1 + N - \sum_i z_i\right) \tag{7}$$

where β is a length-two vector that represents our prior belief about the number of sequences with $z_i = 0$ and $z_i = 1$. (Note that the posterior for μ is independent of M, s and X, given z.)

We can therefore modify the original Gibbs sampling scheme by adding in some extra steps (highlighted in blue) to sample a new μ from the appropriate beta distribution after each update of the PWM, conditional on the current values of z:

Algorithm 3: *Gibbs sampler for inference on s, z, μ and M*

1. Choose a random motif starting location, s_i, for each sequence.

2a Choose a random starting value for μ in the range $[0, 1]$.

2b Randomly initialise the indicator variables z_i for each sequence.

3. Sample a PWM, M, from the full conditional, $p(M \mid s, z, X)$

4. For some number of iterations:

   For each sequence $i = 1, \ldots, N$:

   i. For each position $n = 1, \ldots, L_i - K + 1$ in the sequence, compute the ratio of probabilities of observing the characters from n to $n + K - 1$ from the motif, M, versus from the background distribution, G:

   $$\pi_n = \prod_{j=n}^{n+K-1} p(X_j^{(i)} \mid M)/p(X_j^{(i)} \mid G)$$

   ii. Set z_i to zero with probability

   $$\frac{(L_i - K + 1)(1 - \mu)}{(L_i - K + 1)(1 - \mu) + \mu \sum_n \pi_n}$$

   iii. If $z_i = 1$, set $s_i = n$ with probability proportional to π_n.

   iv. Sample M from the full conditional $p(M \mid s, z, X)$.

   v. Sample μ from the full conditional $p(\mu \mid z)$.

In this modified scheme, it is important to also check that the samples for μ have also converged, as well as checking the convergence of the motif PWM and the starting locations.

Your task

You will be given several datasets containing a number of DNA sequences, some or all of which contain an unknown regulatory motif. The aim of this exercise is to detect the motifs, and build up a position weight matrix describing their distribution. The main part of this analysis will involve implementing and using the Gibbs sampling approach as described above, making use of the skeleton MATLAB code provided.

As part of the assessment, each group will be asked to submit the PWM for the motifs detected in each dataset, and there will be a prize given to the group that achieves the highest preduction accuracy[1].

Task 1: Implementing and testing the Gibbs sampler

You have been given the file `find_motifs.m`, along with some auxiliary functions. This code contains most of what is needed to implement the Gibbs sampler for motif detection, but you will need to fill in the body of the functions `sample_M`, `sample_s`, `sample_mu`, and `likelihood` in order to get the code to work. These functions are defined mathematically in this document, so the implementation should be a relatively straightforward exercise.

You may also want to re-implement the function `compute_background`, which sets up the background model, G. Currently it assumes a uniform background distribution, but it would be more sensible to use the relative frequencies of each nucleotide as observed in the data.

In order to check that you are able to successfully get the algorithm working, you should apply it to the ten sequences contained in the file `data1.fasta`, which contain an easily locatable motif of length 10, with a highly conserved consensus sequence of $AATTCGAATT$ (as well as some slight variations on this, such as $TTCGAATTCC$).

Run the code with `mu = 1` and `mu_unknown = 0`, meaning that each sequence is forced to contain one copy of the motif.

Run the sampler for as long as it takes to converge[2] (as monitored by the appropriate summary statistics). You should also familiarise yourself with analysing the output of the Gibbs sampler, including quantities such as the average information per site, and the minimum entropy profile.

Use the function `seqlogo_fig.m` to visualise the PWM(s) that you discover.

Also briefly explore the effect of the choice of the prior parameter a; this determines the strength of the uniform prior on the motif, and when a is larger, there is less difference between the posterior probability for different starting

[1] We define accuracy as the average Kullback-Liebler divergence between the columns of the submitted motif and the true PWM. If the proposed PWM is longer than the true motif, then we take the best contiguous length-K subset of columns. If the proposed PWM is shorter, then we take the length-K subset of the true motif that yields the highest score.

[2] A few hundred iterations should be enough for convergence in this case.

locations. Hence, one might expect that a larger a could lead to faster convergence. Is this the case? What happens when a is very large (e.g. larger than the number of sequences)?

Explore the effect of changing K – is it better if you search for slightly longer motifs, e.g. $K = 12$?

Task 2: Application to promoter data

Once you have successfully got the algorithm up and running on the test dataset, you should now apply it to the sequences in the file `data2.fasta`, which contains twenty gene sequences, each of which has a promoter region containing a particular motif. Since you do not know the length of the motif, you must consider whether to search through different motif lengths, as well as exploring the effect of the prior. When you are satisfied that your sampler has converged to a particular position weight matrix, store the posterior mean PWM, and the PWMs giving the maximum likelihood ratio, and minimum entropy.

Task 3: When only some of the sequences contain the motif

As discussed earlier, one of the weaknesses of the original Gibbs sampling strategy as proposed by Lawrence *et al.* is that it forces each sequence to have a copy of the motif. In this section of the analysis you will examine a set of sequences (in the file `data3.fasta`) where only a subset contain the motif, and the rest may contain other signals or none at all.

First run the sampler with the variable `mu` set to 1 and `mu_unknown = 0` (as before). You will probably find that the sampler does not converge on a particularly informative profile, and that the result varies with each iteration.

Now experiment with setting `mu` to different values less than one (still keeping `mu_unknown = 0`), and observe the effect on the convergence of the average information, as well as looking at which sequences are deselected via the variable `Z`. Does the sensitivity to the prior parameter a change when $\mu < 1$? What is the maximum value of a that still results in convergence to an informative PWM?

When you appear to be getting robust results, record the posterior mean and maximum likelihood-ratio PWMs, as well as the corresponding maximum likelihood-ratio starting locations for the motifs.

For a particular value of μ, choose values for β_0 and β_1 that will give a prior distribution with this value as its mean, and re-run the analysis, this time setting `mu_unknown = 1`. Plot the posterior distribution of μ. Does it appear to have converged?

Task 4: Homologous sequences

As discussed earlier, the Lawrence method assumes that each sequence represents an independent sample from the same distribution. In the case where the

sequences are only distantly related to each other, this may be a reasonable approximation. However, for sequences that are descended from a common ancestor, if the evolutionary divergence is low then assumption of independence may be very poor, since the sequences will share many features due to their common heritage. As such, the Liu-Lawrence method may struggle to find conserved motifs, as it will be fooled by the many other conserved regions.

To illustrate this point, try to detect a motif in the highly conserved set of sequences in `data4.fasta`. You will almost certainly observe that the sampler will converge quickly to something, but not the true motif, since there are too many other highly conserved regions.

Can you think of any ways of dealing with this problem? How might the likelihood be modified to account for the non-indepdendence of the sequences? You may wish to look at the file `felsenstein_likelihood.m` for some ideas.

You will have an opportunity to present any ideas during your presentation on Friday.

Task 5: Varying K

If you are feeling particularly adventurous, you might wish to consider how it would be possible to do Bayesian inference on K within the Gibbs sampling framework. What would be a suitable prior for K?

References

[1] Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A., & Wootton, J. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science,* **262** (5131), 208–14.

Appendix

Likelihood: multinomial model

The PWM can be used to define a model for generating motifs, whereby the probability of observing a particular motif is simply given by the product of the relevant entries in the PWM. For a sequence fragment of length K generated by a PWM, M, this yields:

$$p(X \mid M) = \prod_{k=1}^{K} M_{X_k}^{(k)} \tag{8}$$

where X_k is the kth character of X. For several such sequence fragments, the powers can be collected, and written as a product of multinomial likelihoods for each site in the motif

$$p(X^{(1)}, \ldots, X^{(N)} \mid M) = \prod_{i=1}^{N} \prod_{k=1}^{K} M_{X_k^{(i)}}^{(k)} \tag{9}$$

$$= \prod_{k=1}^{K} \prod_{\chi=A,C,G,T} (M_\chi^{(k)})^{f_\chi^{(k)}} \tag{10}$$

where $f_\chi^{(k)}$ is the number of occurences of nucleotide χ at site k in the motif, summed over all the observed sequences, and, as before, $M_\chi^{(k)}$ is the probability of observing nucleotide χ at site k.

Prior distributions

The conjugate prior for the multinomial distribution is the *Dirichlet distribution*, which in our four-dimensional setting (A, C, G, T) has the form

$$\mathcal{D}(M^{(k)} \mid \alpha) = \frac{1}{Z(\alpha)} \prod_{\chi=A,C,G,T} (M_\chi^{(k)})^{\alpha_\chi - 1} \tag{11}$$

where $\alpha = (\alpha_A, \alpha_C, \alpha_G, \alpha_T)$ is a vector or prior counts for each nucleotide, and $Z(\alpha)$ is a normalising constant involving Gamma functions.

Multiplying the prior $p(M^{(k)} \mid \alpha) = \mathcal{D}(M^{(k)} \mid \alpha)$ by the likelihood in equation (10), we see that the posterior for each column also has the form of a Dirichlet distribution

$$\begin{aligned} p(M^{(k)} \mid X^{(1)}, \ldots, X^{(N)}, \alpha) &= p(X_k^{(1)}, \ldots X_k^{(N)} \mid M^{(k)}) p(M^{(k)} \mid \alpha) / p(X_k^{(1)}, \ldots, X_k^{(N)}) \\ &\propto \prod_{\chi=A,C,G,T} (M_\chi^{(k)})^{f_\chi^{(k)}} (M_\chi^{(k)})^{\alpha_\chi - 1} \\ &\propto \prod_{\chi=A,C,G,T} (M_\chi^{(k)})^{f_\chi^{(k)} + \alpha_\chi - 1} \\ &= Dirichlet(M \mid \alpha + f^{(k)}) \end{aligned}$$

The Bayesian inference problem here involves determining the posterior distribution of s and M given the sequences, X. Following the Gibbs sampling

strategy, we split this question into two parts. First we consider the posterior of s given M and X

$$p(s \mid M, X) = \frac{p(X \mid s, M)p(s)}{\sum_s p(X \mid s, M)} \tag{12}$$

where $p(s)$ is the prior on the starting locations. In the absence of any other information, we will assume this is uniform. Substituting the expression for $p(s_i \mid M, G, X)$, and normalising, this yields

$$p(s \mid G, M, X) = \prod_{i=1}^{N} \frac{\prod_{j=s_i}^{s_i+K-1} p(X_j^{(i)} \mid M)/p(X_j^{(i)} \mid G)}{\sum_{n=1}^{L_i-K+1} \prod_{k=n}^{n+K-1} p(X_k^{(i)} \mid M)/p(X_k^{(i)} \mid G)} \tag{13}$$

After sampling the starting locations, we can then sample M from its full conditional

Full conditional for general μ

The above analysis assumes that every sequence contains a motif. As discussed in the main text, we can introduce an additional variable, μ, that represents the probability of any particular sequence containing a motif:

$$p(s, z \mid X, M, \mu) = \prod_{i=1}^{N} \frac{\mu^{z_i}(1-\mu)^{1-z_i} \prod_{j=s_i}^{s_i+K-1} p(X_j^{(i)} \mid M)^{z_i} p(X_j^{(i)} \mid G)^{1-z_i}/p(X_j^{(i)} \mid G)}{\sum_{z_i \in \{0,1\}} \sum_{n=1}^{L_i-K+1} \mu^{z_i}(1-\mu)^{1-z_i} \prod_{j=n}^{n+K-1} p(X_j^{(i)} \mid M)^{z_i} p(X_j^{(i)} \mid G)^{1-z_i}/p(X_j^{(i)} \mid G)} \tag{14}$$

$$= \prod_{i=1}^{N} \frac{1}{\mathcal{Z}_i} \left((1-z_i) \underbrace{(1-\mu)}_{\text{not motif}} + z_i \underbrace{\mu \prod_{j=s_i}^{s_i+K-1} \frac{p(X_j^{(i)} \mid M)}{p(X_j^{(i)} \mid G)}}_{\text{motif}} \right) \tag{15}$$

where $\mathcal{Z}_i$ is the appropriate normalising constant.

Full conditional when motif probability μ is unknown

Since μ enters the likelihood in the form of a Binomial density (i.e. $p(z \mid \mu) \propto \mu^{\sum_i z_i}(1-\mu)^{\sum_i (1-z_i)}$), the conjugate prior for μ is a beta distribution

$$p(\mu \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1}(1-\mu)^{b-1} \tag{16}$$

where a and b are hyperparameters that define the shape of the prior distribution. In the absence of any prior information regarding the probability of a motif, one might set $a = b = 1$, which results in a uniform prior on μ. Using Bayes' rule, the posterior for μ conditional on z is then

$$
\begin{aligned}
p(\mu \mid z, a, b) &= p(z \mid \mu) \times p(\mu \mid a, b)/p(z) \\
&\propto \text{Bin}(\sum_i z_i \mid \mu) \times \text{Beta}(\mu \mid a, b) \\
&\propto \frac{1}{f(z, N)} \mu^{\sum_i z_i} (1-\mu)^{N - \sum_i z_i} \times \frac{1}{g(a, b)} \mu^{a-1} (1-\mu)^{b-1} \\
&= \text{Beta}\left(\mu \;\middle|\; a + \sum_i z_i, b + N - \sum_i z_i\right)
\end{aligned}
$$