File 6: The Ahab algorithm: fitting multiple weightmatrices to sequence

The algorithm is based on a probabilistic segmentation of a sequence S in terms of background and binding sites for a given set of transcription factors. The sequence motifs recognized by the factors are modeled by weight matrices. There are many ways to partition a sequence and one needs a systematic procedure to weight all possible tilings. Our model is that each tile (background and weight matrices) has a probability p_w and that the various drawings needed to cover the whole sequence in a given tiling T are independent. The set of probabilities p_w's is fixed by maximizing the likelihood to observe the sequence.

Specifically, let us denote by $N(T)$ the number of tiles needed to cover S in the tiling T. Its likelihood is given by

$$P(T) = \prod_{k=1}^{N(T)} p_{w_k} m(s|w_k)\,, \tag{1}$$

The weight m for a match between a subsequence $s = (n_1, \ldots n_l)$ and a weight matrix of the same length follows from the very definition of weight matrix. Denoting by $f_j(n|w)$ the normalized frequency of the nucleotide n at the j-th position in the weight matrix, we have

$$m(s|w) = \prod_{j=1}^{l} f_j(n|w). \tag{2}$$

Sequence which is not be covered by binding sites should be modelled as background sequence. Obviously, background sequence could still be functional, for example it could be recognized and bound by other transcription factors which are not part of our set. In any case one should assume that background sequence has retained some correlations and use a Markov model of order M to model it. First, we count the frequencies $f^{(B)}$ of all $m+1$ tuples in the sequence. These counts determine also the frequencies of all $1, ..., m$ tuples. Second, we calculate the weight $m(n_1|B)$ for a nucleotide n_1 to be drawn from background as the probability of n_1 conditional to the couple preceding it. For example, for a Markov model of order 3, $n_1 = C$ and the precedent couple AA we have

$$m(n_1|B) = f^{(B)}(AAC)/[f^{(B)}(AAA) + \ldots f^{(B)}(AAT)]. \tag{3}$$

If n_1 happens to be the beginning base of the sequence (or if there are unknown nucleotides in the couple preceding it), the match is calculated similarly but using the frequencies of couples or single-nucleotides. Note that the length L of S sets bounds on the order M of the Markov model because one should have $L >> 4^{m+1}$.

The likelihood Z to observe the sequence S given the set of probabilities p_w is simply the sum of (1) for all possible tilings:

$$Z = \sum_T P(T)\,. \tag{4}$$

The optimal assignment of the p_w's is found by maximizing the likelihood Z or, equivalently, by minimizing the free energy $F = -\log Z$. The specific method which we have used is a conjugate gradient method and therefore involves the calculation of the function F and its first derivatives.

Readers familiar with hidden Markov models (HMM) (see (1) and references therein) may find it enlightening that Ahab (and Mobydick (2)) can be formulated as HMMs. The states of the model are background and all possible positions within the various weight matrices. The transition probabilities between the states are as follows. If the present state is the background or the final position of a weight matrix, the next state will be the background or the beginning of a weight matrix with probabilities given by the p_w's. If the present state is within a weight matrix, the next one is bound to be the successive position within the same weight matrix. The emission probabilities are simply given by the weight matrix frequencies and the conditional probabilities discussed above.

The number of possible segmentations of a sequence S increases factorially with its length L and it would therefore be impossible to calculate (4) by brute force for any relevant length. However, the Markovian nature of the model hints that dynamic programming techniques ("transfer matrix"

methods in statistical mechanics) are possible. Indeed, the likelihood for the sequence up to i obeys the following recursion relation:

$$Z(1,i) = \sum_w p_w \, m(s|w) \, Z(1, i - l_w) \, , \tag{5}$$

where the sum w is over the background and each weight matrices and l_w is the width of the wth matrix in the sum. For each w the sequence s appearing in $m(s|w)$ includes the bases $(i, i-1, \ldots, i-l_w+1)$ from the data being fit. The likelihood for S is $Z = Z(1, L)$ and the initial condition is $Z(1, i \leq 0) = 1$. Equation (5) is the analog of the forward algorithm for HMMs. The only *caveat* in (5) is that the sum is numerically prone to underflows. The well-known remedy is to use ratios, e.g. to numerically implement the recursion for $Z(1,i)/Z(1,i-1)$. The free energy F is finally calculated in a time which scales linear with L.

For the computation of the gradients the most convenient procedure is to calculate first the derivatives with respect to the probabilities of all the relevant subsequences appearing in S and then use the chain rule. Specifically, let the probability of a subsequence s be denoted by $q_s = \sum p_w \, m(s|w)$. The derivative with respect to the q's is calculated as:

$$\frac{\partial F}{\partial q_s} = -\frac{1}{Z}\frac{\partial Z}{\partial q_s} = -\sum_{i=1}^{L} G(i, l_s)\delta_{s,s_i} \tag{6}$$

where the δ restricts the sum to the positions in the sequence where s is present and $G(i, l_s) = Z(1, i - l_s)Z(i + 1, L)/Z(1, L)$. The backward likelihoods obey a recursive relation analogous to (5) (but the sequence s appearing in $m(s|w)$ now includes the bases $(i, i+1, \ldots, i+l_w-1)$):

$$Z(i, L) = \sum_w p_w \, m(s|w) \, Z(i + l_w, L) \, . \tag{7}$$

This allows to calculate the G appearing in (6) and thus all the derivatives in $O(L \times l_{max})$ operations, where l_{max} denotes the maximum length among the weight matrices. The derivatives with respect to the p_w's are finally calculated using the chain rule:

$$\frac{\partial F}{\partial p_w} = \sum \frac{\partial F}{\partial q_s} m(w, s) \, . \tag{8}$$

Numerically, it is convenient to precompute the subsequences and their matches $m(s|w)$. The sums are restricted to subsequences and weight matrices having the same length.

The free energy and its derivatives are the input for the conjugate gradient method of optimization (see (3) for a simple discussion of the method and the subroutines used). In some cases, it turns out to be numerically convenient to start with a few preliminary steps in the direction of steepest descent. Note that our optimization problem is constrained, i.e. the p_w's are restricted between zero and one and normalized. However, these restrictions are easily lifted by the parametrization:

$$p_w = \frac{\exp(-\beta_w)}{\sum \exp(-\beta_w)} \tag{9}$$

and by setting one the β's to zero (for example for background).

The outcome of the optimization procedure is the converged set of probabilities p_w and the corresponding value of the free energy F. The latter is used to rate the density and the quality of transcription factor binding sites by the log-score $Q = F_B - F$. Here, F_B is the free energy in the absence of weight matrices and is the log likelihood that S comes only from background. The converged set of p_w's is used in posterior decoding. Finally, at each position $i = (1, \ldots, L)$ in S we calculate the posterior probability for the j-th position within the tile w (weight matrix or background) as:

$$\mathcal{P}_i(w, j|S) = \frac{Z(1, i - j) \, p_w \, m(s|w) \, Z(i + l_w + 1 - j, L)}{Z(1, L)} . \tag{10}$$

References

[1] Durbin R., Eddy S., Krogh A. & Mitchison, G (1998) *Biological Sequence Analysis*, (Cambridge Univ. Press).

[2] Bussemaker H. J., Li H. & Siggia E. D. (2000) *Proc. Natl. Acad. Sci. USA* **97**, 10096-100.

[3] Press W. H. *et al.* (1993) *Numerical Recipes in C :The Art of Scientific Computing* (Cambridge Univ. Press).