Chapter 5

Modeling motifs: Position Specific Scoring Matrices

Local multiple sequence alignment involves the discovery, modeling, and recognition of conserved patterns or motifs in multiple (and potentially very many) DNA or protein sequences.

- In *discovery*, we are given *unlabeled* sequences. The task is to identify one or more shared, conserved motifs in these sequences. In machine learning terms, this is equivalent to *labeling* the sequences. For example, each symbol in a sequence might be labeled "1" if it is in the conserved pattern and "0" if it is not. More complex labeling schemes representing more than one motif or different substructures within a motif are also possible.

- In *modeling*, we are given a local multiple alignment as input. In machine learning terms, we are given *labeled* sequences. The task is to construct a probabilistic model that represents the properties of each column in the alignment (i.e., the symbols we are likely to observe at that position) in an efficient manner and that can be used for searching for new instances of the pattern.

- In *recognition*, we are given a new unlabeled sequence containing zero, one, or more than one instances of the motif of interest. A probabilistic model of the motif is used to search the unlabeled sequence for instances of the motif. The location and extent of each motif identified are reported.

We will first discuss *Position Specific Scoring Matrices* (PSSM's), a formalism for modeling local multiple alignments, and the Gibbs Sampler, a discovery method that uses the PSSM formalism. PSSM's and the Gibbs sampler are suitable for ungapped motifs only. The *Hidden Markov Model (HMM)* is a formalism that can be used for both modeling and

discovery of patterns that contain gaps. We will discuss HMM's immediately following the Gibbs sampler.

In these notes, PSSM's and Gibbs sampler are presented in terms of amino acid motifs. Both formalisms can be equally well applied to patterns in nucleic acid sequences. In fact, discovering and modeling transcription factor binding sites in DNA sequences is a common application of the Gibbs sampler in bioinformatics.

5.0.6 Position Specific Scoring Matrices

PSSM's are a formalism for *modeling* ungapped local alignments. Like scoring matrices used for pairwise alignments, PSSM's are based on a log-odds formalism. Recall that both the PAM and BLOSUM matrices are defined in terms of an alternate hypothesis, H_a, that a pair of sequences are related at a given evolutionary divergence and a null hypothesis, H_0, that the sequences are unrelated and any observed similarity is due to chance. In such matrices, $S[x, y]$ is the logarithm of the likelihood ratio:

$$S[x, y] = \log_2 \frac{P[x \text{ aligned with } y | H_a]}{P[x \text{ aligned with } y | H_0]}. \tag{5.1}$$

A PSSM is similarly defined in a log-likelihood framework. In this case, the scoring matrix is used to score a candidate instance of a motif in a single sequence. The focus is on the probability of observing of a particular amino acid at a particular position in the motif. A PSSM is constructed from training data representing examples of the pattern. Given an ungapped local alignment representing k instances of a motif of width w, we derive a *propensity* matrix, P, representing the likelihood ratio

$$P[x, i] = \frac{q[x, i]}{p_x}, \tag{5.2}$$

where $q[x, i]$ is the probability of observing amino acid x at position i under the alternate hypothesis that this is an instance of the motif. The probability of the same event under the null hypothesis is p_x, the background distribution of amino acid a. The log odds scoring matrix is

$$S[x, i] = \log_2 P[x, i]. \tag{5.3}$$

Note that both P and S have $|\Sigma|$ rows and w columns, where Σ is the alphabet (in this case the 20 amino acids).

To complete the definition of $P[x, i]$, we need an expression for the numerator, $q[x, i]$. The frequency of amino acid $x \in \Sigma$ at position i in the alignment is

$$q[x, j] = \frac{c[x, i] + b}{k + b \cdot |\Sigma|}, \tag{5.4}$$

where $c[x, i]$ is the number of x's at position i and b is a pseudocount.

Pseudocounts are introduced to account for examples of the pattern that are not represented in the training data. It is possible that a particular amino acid, x, does sometimes occur at position i in this motif, but that this case does not arise in any of the sequences in the alignment. If $P[x, i] = 0$, then the resulting PSSM will assign a score of zero to any sequence with an x at position i, preventing the future discovery of this variant of the motif. To account for this, pseudocounts are used to give every amino acid a small, but non-zero probability at every position in the motif. The normalization in the denominator is adjusted accordingly by the term $b|\Sigma|$. In this class, we typically use $b = 1$. However, there are more complex approaches to selecting a pseudocount. For those interested in exploring this further, a more general treatment of pseudocounts is given in Section 5.6 of Durbin's book. This is not required for this course.

Given a new, unlabeled sequence, t, of length n, we can search for an instance of the motif in t by scoring the w residues starting at each position o+1 in the sequence as follows:

$$\mathcal{S}(t, o) = \sum_{i=1}^{w} S[t[o+i], i], \tag{5.5}$$

where the *offset*, o, ranges from 0 to n-w. The offset refers to the position *before* the first symbol in the motif; for a given value of o, the motif is the subsequence from position o+1 to o+w. The offset with the highest score is most likely to be an instance of the motif. To be convincing, the score must also be high in an absolute sense, not just higher that the scores associated with other offsets. In cases where there may be more than one instance of the motif in t, offsets with near optimal scores should also be considered.

Note that the score of a window of length w following offset o in t is a log likelihood ratio

$$\mathcal{S}(t, o) = \log_2 \frac{\Pr(\text{motif at offset } o|H_a)}{\Pr(\text{motif at offset } o|H_0)}, \tag{5.6}$$

where H_a is the alternate hypothesis that t contains the motif at position o+1 and H_0 is the null hypothesis that there is no instance of the pattern at this location and the residues

in the window occur with typical background frequencies. To see this, consider that

$$\begin{aligned}
\mathcal{S}(t,o) &= \sum_{i=1}^{w} S[t[o+i],i] \\
&= \sum_{i=1}^{w} \log_2 \frac{q[t[o+i],i]}{p_{t[o+i]}} \\
&= \log_2 \prod_{i=1}^{w} \frac{q[t[o+i],i]}{p_{t[o+i]}} \\
&= \log_2 \frac{\prod_{i=1}^{w} q[t[o+i],i]}{\prod_{i=1}^{w} p_{t[o+i]}}.
\end{aligned}$$

The numerator is the probability that the w residues starting at position $o+1$ in t represent an instance of the motif. The denominator is the probability of observing those residues under the null hypothesis.

5.0.7 Gibbs sampler for motif discovery

A PSSM provides a compact, probabilistic representation of an ungapped motif, based on a training data set consisting of k representative sequences of length w. But how do we discover a motif in unlabeled sequences, when the motif is not known in advance?

The *Gibbs sampler* is an algorithm for *discovery* of ungapped local alignments that uses the PSSM formalism as its basic data structure. The Gibbs sampler takes as input k sequences, $t_1 \ldots t_k$, of lengths $n_1 \ldots n_k$, that share an ungapped motif of length w. The underlying assumption of the Gibbs Sampler is that each sequence contains exactly one instance of the motif. The length of the motif, w, must be supplied by the user. The output is a set of k subsequences of length w, one in each input sequence, that are "most similar" to each other. Here, our measure of "most similar" is a likelihood function derived from the propensity matrix, P, defined in Equation 5.2.

Sequences $t_1 \ldots t_k$ contain $O(n^k)$ candidate motifs, corresponding to all possible sets of offsets $\{o_1 \ldots o_k\}$, where $0 \leq o_z \leq n_z - w$ is the offset in sequence t_z. Since the motif is ungapped, for a fixed width, w, each candidate is completely defined by a set of offsets $\{o_1 \ldots o_k\}$ that specify starting points of the subsequences in the k sequences. Each set of offsets defines a local alignment consisting of the k subsequences of length w,

$$A = \begin{cases} t_1[(o_1+1)\ldots(o_1+w)] \\ t_2[(o_2+1)\cdots(o_2+w)] \\ \cdots \\ t_k[(o_k+1)\cdots(o_k+w)], \end{cases}$$

where the notation $t[u \cdots v]$ denotes the substring of t, starting at position u, up to and including position v.

A brute force approach to identifying the true motif is exhaustive enumeration of all candidates. A score can be assigned to each candidate ungapped alignment that captures the extent to which columns in the alignment reflect similar sequence features. A PSSM, $S[\cdot, \cdot]$, is constructed as specified in Equations 5.2 - 5.4, and used to score each sequence in the alignment:

$$\mathcal{S}(t_z, o_z) = \sum_{i=1}^{w} S[t_z[o_z+i], i]. \tag{5.7}$$

The sum of these scores, $\mathcal{S} = \sum_{z=1}^{k} \mathcal{S}(t_z, o_z)$, gives the overall score of the candidate motif.

In most cases, the PSSM will be meaningless in the sense that the individual subsequences will be unrelated to each other, and the residues in each column will not reflect a significant degree of conservation. In this case, the individual subsequences will not receive scores that are better than chance. Only a candidate alignment that does, in fact, correspond to a conserved motif will produce a PSSM that yields good $\mathcal{S}(t_z, o_z)$ scores.

The computational cost of this brute force approach is prohibitive for all, but the smallest problem instances. The Gibbs Sampler is a more efficient approach to searching the space of candidate motifs that does not require that all possible alignments be considered. The Gibbs Sampler uses an iterative approach in which a new estimate of the motif is generated from the current estimate of the motif at each iteration. The Gibbs Sampler has a robust theoretical basis and can be proven to converge to the best estimate. This convergence guarantee results from the specific features of the procedure for generating a new estimate from the current estimate. First, the new estimate must be different from the current estimate, but not too different. Second, the new estimate must be chosen in such a way that the algorithm can find the global optimum (i.e., the best estimate) and not just a local optimum. (A local optimum is an estimate that is better than all estimates that could be obtained in one step of the algorithm; that is, an estimate that has a higher score than all estimates resulting from a single modification of the current estimate.)

We introduce these ideas in two steps. We first consider a Hill Climbing algorithm (Algorithm 1) for motif discovery that has the same iterative structure as the Gibbs sampler, but is not guaranteed to converge to the global optimum. The term "hill-climbing" reflects the fact that this algorithm finds an estimate with a higher score at each step and stops when it reaches the top of the hill (i.e., when it has found a local optimum). This Hill Climbing algorithm has the same data structures as the Gibbs Sampler and exemplifies the procedure for generating new candidate estimates. We then extend this algorithm to obtain a full Gibbs Sampler (Algorithm 2) that has a more sophisticated procedure for generating a new estimate, which allows it to converge to a global optimum.

Both algorithms generate a new estimate from the current estimate by modifying the contribution of one of the k sequences, called t^* to A, and holding the subsequences from the

remaining $k-1$ sequences fixed. The current estimate is represented by $(k-1) \times w$ matrix, A' which holds the subsequences of length w that represent the current best estimate in the $k-1$ sequences that will not change in this iteration. A PSSM is then constructed from A' and used to score all candidate offsets in t^*. Based on these scores, a new offset o^* is selected for sequence t^*. A new estimate of the motif is then constructed by removing one row in A' and replacing it with the new subsequence of length w with offset o^* in t^*. In order to simplify the book keeping associated with selecting a sequence at random from the $k-1$ sequences represented in the current iteration of A', we introduce an array called `index` that contains the indices of the sequences currently in A'. The row to be removed is selected by generating a random number, r, between 1 and $k-1$; the index of the sequence to be removed is `index[r]`.

The selection of a new offset, o^*, in t^* is a crucial aspect of the convergence of this algorithm. In the Hill-Climbing algorithm (Algorithm 1), the subsequence with the highest score is selected. Initially, selecting the subsequence with the highest score might seem an attractive strategy, but this could trap the algorithm in a local optimum. In fact, this is why the Hill Climbing algorithm is not guaranteed to converge to a global optimum.

The Gibbs Sampler (Algorithm 2), instead, selects a window in t^* at random from all windows starting at offsets ranging from 0 to n^*-w. The probability of selecting a particular offset, o, is biased by the probability of the subsequence at that offset so that higher scoring windows have a greater chance of being selected. The probability of the subsequence starting at offset $o+1$ is defined in terms of its propensity with respect to the current estimate is:

$$pdf(o) = \frac{\prod_{j=1}^{w} P[t^*[o+j], j]}{\sum_{l=0}^{n^*-w} \prod_{j=1}^{w} P[t^*[l+j], j]}. \tag{5.8}$$

The numerator is the propensity of the subsequence starting at $o+1$. The denominator is the sum of propensities for all possible offsets, from $o=0$ to $o=n^*-w$. This is a normalization factor that ensures that $\sum_{l=0}^{n^*-w} pdf(o) = 1$. Note that here we score offsets using the propensity matrix, P, and not the log odds scoring matrix, S. Since $pdf(o)$ is a probability it must always have a non-negative value; S cannot be used because S can be negative.

Selecting a value of o with probability $pdf(o)$ requires a method for obtaining a random number conditioned on an arbitrary probability distribution. This random number can be obtained by calculating the cumulative distribution function

$$cdf(o) = \sum_{o=0}^{n^*-w} pdf(o),$$

a monotonically increasing function with domain $[0, n^*-w]$ and range $[0,1]$. Its inverse, $cdf^{-1}(r)$, is a function defined on the domain $[0,1]$ with range $[0, n^*-w]$. The offset of the new subsequence is defined to be $o^* = cdf^{-1}(r)$, where r is a uniformly distributed random

```
Algorithm:  Hill-Climbing
Input:
  Sequences t_1,...,t_k of lengths n_1,...,n_k.

Initialization:
  z = 1                                    # index of special sequence.
  t* = t_z, n* = n_z                       # t1 is the special sequence, initially.
  for (j = 2 to k) {
     index[j-1] = j                        # index of non-special sequences
     o_j = rand(1, n_j - w)                # Guess starting offsets
     A'[j-1, 1...w] = t_j[(o_j+1)...(o_j + w)]
  }
  Calculate P[x,i], the propensity matrix of A' with pseudocounts

Search for motif:
  Repeat
  {
     o* = argmax_o{S(t*, o)}               # Select starting offset in t*
     r = rand(1, k-1)                      # Select new special sequence
     A'[r, 1...w] = t*[(o*+1)...(o* + w)]  # Replace new special with t* in A'
     y = index[r]; index[r] = z; z = y     # store ptr to t* in index
     t* = t_z; n* = n_z                    # initialize new t*
     Calculate P[x,i], the propensity matrix of A' with pseudocounts
     S[x,i] = log_2 P[x,i]
  } until(P[·,·] stops changing)
  Obtain A by adding t*[(o*+1)...(o* + w)] to A'
  Compute the log odds scoring matrix, S, from A.

Output:
  Local multiple sequence alignment A with scoring matrix S.
```

Algorithm 1: **Hill Climbing** The matrices P and S are the propensity and log odds matrices defined in Equations 5.2 and 5.3. Note that A' and P are $(k-1) \times w$ matrices, whereas the output matrices A and S are $k \times w$ matrices. The use of pseudocounts when calculating P and S is recommended to ensure all symbols in the alphabet are represented.

number in the interval $[0, 1]$. The index o^* generated by this procedure is a random number with distribution $pdf(o)$.

Potential pitfalls: There are various potential pitfalls associated with the Gibbs sampler, as with any algorithmic attempt to discover biological truth. For one thing, you could find a statistically significant or biologically meaningful motif that is not the motif you are looking for. In addition, problems can arise if one or more sequences have no copy of the motif or have more than one copy.

Using this algorithm to obtain meaningful solutions requires a number of decisions that are not programmatically determined and require *ad hoc* solutions, possibly guided by the user's biological intuition:

- Selecting the window size, w
- Selecting the starting configuration
- Selecting values for pseudocounts
- Termination condition: how should the algorithm decide when to stop?

These issues are discussed in greater detail in Lawrence et al. (1993), which is available via the "optional readings" column of the syllabus.

Convergence: The Gibbs sampler models the search for an optimal local alignment as a Markov Chain, in which each state is a set of k subsequences of length w. It can be shown that this Markov Chain has a stationary distribution and that the state corresponding to the most likely motif has high probability in that distribution. In theory, this process is guaranteed to converge to the optimal solution, given "enough time." In practice, the sampler can get bogged down in local optima for long periods of time. An approach to avoiding this problem is to run the procedure several times with different starting configurations. This is discussed in greater detail in the materials listed under "optional reading."

Background: The Gibbs sampler is a general method for estimating a joint probability distribution by repeated calculations of a conditional distribution using a Markov Chain Monte Carlo (MCMC) approach. The application of the Gibbs sampler for motif finding in biomolecular sequences was proposed first by Chip Lawrence[1] and his colleagues in 1993. For those interested in more theoretical aspects, Ewens and Grant discuss the Gibbs sampler for biomolecular motif discovery in the MCMC framework in their book (Section 10.5 in

[1] *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment.* Lawrence et al., Science. 1993 262(5131):208-14.

```
Algorithm:  Gibbs Sampler
Input:
  Sequences t_1,...,t_k of lengths n_1,...,n_k.

Initialization:
  z = 1                                    # index of special sequence.
  t* = t_z, n* = n_z                       # t1 is the special sequence, initially.
  for (j = 2 to k) {
     index[j-1] = j                        # index of non-special sequences
     o_j = rand(1, n_j - w)                # Guess starting offsets
     A'[j-1, 1...w] = t_j[(o_j+1)...(o_j + w)]
  }
  Calculate P[x,i], the propensity matrix of A' with pseudocounts

Search for motif:
  Repeat
  {
     for (o = 0 to (n*-w))
     {
        pdf(o) = ∏_{j=1}^{w} P[t*[o + j], j] / Σ_{l=0}^{n*-w} ∏_{j=1}^{w} P[t*[l + j], j]
     }
     With probability pdf[o], o* = o       # Select starting offset in t*
     r = rand(1, k-1)                      # Select new special sequence
     A'[r, 1...w] = t*[(o*+1)...(o* + w)]  # Replace new special with t* in A'
     y = index[r]; index[r] = z; z = y     # store ptr to t* in index
     t* = t_z; n* = n_z                    # initialize new t*
     Calculate P[x,i], the propensity matrix of A' with pseudocounts
  } until(P[·,·] stops changing)
  Obtain A by adding t*[(o*+1)...(o* + w)] to A'
  Compute the log odds scoring matrix, S, from A.

Output:
  Local multiple sequence alignment A with scoring matrix S.
```

Algorithm 2: Gibbs Sampler

the first edition). A general introduction to the Gibbs sampler[2] in a statistical context can be found under "optional readings" on the syllabus page. These readings are not required for the course.

Another probabilistic search procedure called *Expectation Maximization (EM)* can also be used to identify conserved, ungapped motifs. We will discuss EM briefly in the context of HMM's later in the course. EM is discussed in detail in 03-712.

[2] *Explaining the Gibbs sampler*, G. Casella & E. I. George, The American Statistician, 46:167-174, 1992