# BTRY 4840 / 6840 / CS 4775
# Computational Genetics and Genomics

# Problem Set 5

## Problem 1: Unsupervised Learning and EM Estimation

In problem set 3, you considered a two-state HMM and were asked to find the maximum likelihood path given parameter settings; you were also asked to take a known path and to find maximum likelihood parameters. For this problem, suppose you have been given neither the parameters nor the path and you want to find both. You have reason to believe that the sequence consists of G+C-rich and G+C-poor regions, in roughly equal parts, so you plan to use a two-state HMM parameterized as in problem set 3. You will estimate the parameters iteratively by EM, after initializing the emission distribution for the $h$ state to reflect moderately high G+C content (say, $\theta_h = 0.6$) and the emission distribution for the $l$ state to reflect moderately low G+C content (say, $\theta_l = 0.4$). You will initialize the parameter $\mu$ to a noncommittal value of 0.05.

(a) (35 points) Write a program to estimate $\mu$, $\theta_l$, and $\theta_h$ by EM. You should be able to reuse your implementation of the forward/backward algorithm from problem set 3, but you will have to adapt it to compute the expected number of state transitions of each type, and you will have to embed it (probably as a subroutine) within a loop that repeats an E step and an M step until convergence. Observe that the M step will exactly mirror your solution for the complete-data case (the supervised learning problem in problem set 3), but in this case you will use expected rather than actual values of each of the six counts of interest.

To monitor convergence, compute the log likelihood on each iteration of the EM algorithm (it will be a natural byproduct of your forward/backward routine), and terminate the algorithm when the difference in log likelihood between two successive iterations is less than some threshold $\delta$ (a reasonable value would be $\delta = 0.01$). Your program should output the final parameter estimates and you should turn these in. In addition, have your program output the log likelihood on each iteration and turn in a plot of log likelihood in each iteration with the X-axis being the iteration number and the Y-axis the log likelihood.

(b) (8 points) Try at least three alternative initializations of $\mu$, $\theta_l$, and $\theta_h$, including at least one case with $\mu > 0.5$. Turn in the final parameter estimates and a plot of the log likelihood across iterations for each one. How sensitive is the algorithm to its starting point?

## Problem 2: Gibbs Sampling

In this problem, you will perform Gibbs sampling in order to identify sequence motifs in simulated sequence data. Given $t$ sequences $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$, the Gibbs sampler will estimate the start positions $s_i$ of the motifs in each sequence $\mathbf{x}_i, 1 \leq i \leq t$. As discussed in lecture, Gibbs sampling is a Markov chain Monte Carlo (MCMC) approach that estimates a multivariate probability distribution by sampling from the probability of each univariate element conditional on assignments of all other variables. In the context of this problem, you will implement a Gibbs sampler that operates on

$P(s_1, s_2, \ldots, s_t|\mathbf{x}_1, \ldots, \mathbf{x}_n)$ by successively sampling $s_i$ for each $1 \leq i \leq t$ from $P(s_i|\mathbf{x}_i, \boldsymbol{\theta}, \hat{\boldsymbol{\pi}})$. The latter distribution implicitly conditions on the other sequences and start positions since $\hat{\boldsymbol{\pi}}$ is estimated from these positions and sequences.

Given a motif length $k$ and a set of start positions $\mathbf{s}_{-i} = \{s_a : a \neq i\}$ of motif instances, one can count the number of times each base occurs at each position in the motif, which is sometimes called the *position frequency matrix*. Assuming these motif instances are correct, we compute the probability $\hat{\pi}_{j,b}$ of observing a given base $b \in \{A, C, G, T\}$ at a given position $j \in [1, k]$ in our motif from these counts as follows. If $x_{l,p}$ is the base in sequence $\mathbf{x}_l$ at position $p$, then

$$\hat{\pi}_{j,b} = f_j(b|\mathbf{s}_{-i}, \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_t) = \frac{\sum_{l \neq i} I[x_{l,s_l+j-1} = b] + \epsilon}{t - 1 + 4\epsilon},$$

where $\epsilon$ is a pseudocount to be specified (see below), and $I[]$ is the indicator function used to count whether a given sequence has base $b$ at the corresponding position in its motif instance. The above gives us a probability model of the current estimated motif (i.e., $\hat{\boldsymbol{\pi}}$ is estimated conditional on a set of start positions, which will vary throughout the run of the Gibbs sampler).

For this assignment, you may assume a background model of $\theta_b = \frac{1}{4}$ for all bases $b$. (Note that in practice, the background model is more than just single base frequencies and can encode dinucleotide frequencies or other patterns that frequently occur but unlikely to be part of a biologically meaningful motif.)

(a) (45 points) Write a program in Python, R, C, or C++ that uses Gibbs sampling to identify sequence motifs. The length $k$ of the motif should be a parameter to your program, but you may choose to hard code this for the analyses below. Be sure to use appropriate pseudocounts when calculating $\hat{\pi}_{j,b}$ in order to avoid 0 probabilities. The pseudocount value $\epsilon$ to add is left to you to determine. (When calculating $\hat{\boldsymbol{\pi}}$, each sequence contributes a frequency count of 1, and since the pseudocount gets added to each base at each position, you should consider how large $\epsilon$ should be relative to the amount of data you have.)

As noted above, the $\hat{\boldsymbol{\pi}}$ matrix is calculated conditional on the start positions of the other sequences $\mathbf{s}_{-i}$ and the data. This model assumes that there is one motif in each sequence, which suffices for this problem, although in real data, we would ideally relax this assumption.

In this problem we are interested in the maximum likelihood sequence motif. Therefore, rather than storing information derived from each MCMC iteration, you will keep track of the maximum likelihood motif encountered in any MCMC iteration. At the conclusion of each iteration (i.e., after sampling $s_i$ for all $1 \leq i \leq t$), compute the likelihood of the model by calculating $\hat{\boldsymbol{\pi}}$ from the motif instances at all $t$ starting points (modifying the calculation of $\hat{\boldsymbol{\pi}}$ as necessary because this uses all $t$ sequences instead of $t - 1$) and then calculating the likelihood $P(\mathbf{x}_1, \ldots, \mathbf{x}_t, |\mathbf{s}, \boldsymbol{\theta}, \hat{\boldsymbol{\pi}}) = \prod_{i=1}^{t} P(x_i|s_i, \boldsymbol{\theta}, \hat{\boldsymbol{\pi}}) \propto \prod_{i=1}^{t} \prod_{j=1}^{k} \frac{P(x_{i,s_i+j-1}|\boldsymbol{\pi}_j)}{P(x_{i,s_i+j-1}|\boldsymbol{\theta})}$. Keep track of the $\hat{\boldsymbol{\pi}}$ with the highest likelihood that you have encountered in any iteration and print this motif model when the program terminates.

Determine the criteria for terminating the MCMC by experimenting with different approaches. One good possibility for this problem would be to keep track of the number of iterations that have passed since the current maximum likelihood value $\hat{\boldsymbol{\pi}}$ was found, and terminate after some fixed number of iterations is reached. Another possibility is to decide ahead of time how many iterations you wish to run. Given the size of the data and the model definitions, you will not need to run for more than 3,000 iterations in total.

Turn in your source code and also comment on your choice of the pseudocount value $\epsilon$ and your termination criteria.

(b) (4 points) Apply your program with $k = 10$ to identify a motif in the 10 sequences in the file motif1.fa on the course webpage. Write down what you believe the consensus motif is. In this case, all motif instances are identical to the consensus. Run your program multiple times until you are sure you have found the motif.

(c) (4 points) Apply your program with $k = 10$ to identify a degenerate motif in the 10 sequences in the file motif2.fa on the course webpage. What is the consensus sequence? Here again, you should run the program more than once until you are sure you have found the motif.

(d) (4 points) Comment on the results you obtained in parts (b) and (c). Did the sampler find the same motif in each of its runs? If not, why might it have found something else?