Algorithm Description

CS466 Miniproject
Edward W Huang, Hanchen Huang, Xinzhou Zhao

In our project, we utilized Gibbs sampling in order to implement our motif finder. Our algorithm's input is a set of p strings and a motif length k. We wish to find, across the p strings, the most mutually similar substring of length k.

First, we randomly initialize a set of p integers within the size of our input strings. Each integer represents a random starting position for each input string.

Then, we update each of the starting positions. To do this, for each starting position i, we build a profile matrix P using the k-length sequences at all other starting positions. Then, we score each possible subsequence in the current sequence using the following equation.

$$\text{Score}(x) = \sum_{i=1}^{k} \log\left(\frac{e_i(x_i)}{0.25}\right) \tag{1}$$

where $e_i(x_i)$ is the probability of observing the ith character in substring x according to the profile matrix P. 0.25 is simply the probability of observing any character at random, and is used to correct for expectation by chance.

After scoring every possible k-length subsequence in the current sequence, we pick the subsequence with the highest score by equation (1), and update the starting position to the one corresponding to the highest score subsequence. We continue updating for every starting index until a full iteration converges.

The motif output by the algorithm is the profile matrix that matches the final converged starting indices for each of the input strings.