BMI/CS 776 Spring 2015
Homework #1

Prof. Colin Dewey

Due Thursday, February 12th, 2015 by 11:59pm

The goal of this assignment is to become more familiar with the classic methods for discovering motifs within biological sequences.

To turn in your assignment, copy all relevant files to the directory:
`/u/medinfo/handin/bmi776/hw1/USERNAME`
where `USERNAME` is your account name for the BMI network. You must submit a file named `README` to this directory, which gives directions on how to compile (if necessary) and run your programs. For each question below, the `README` file should list the files relevant to that question (e.g., code, other files with written answers).

1. Write a program, `LearnMotif`, that takes as input a set of DNA sequences and a width W, and learns an OOPS motif model for a motif of width W. Depending on your choice of language, the command line to run your program should be one of:

   ```
   LearnMotif sequences_file width model_file positions_file
   java LearnMotif sequences_file width model_file positions_file
   python LearnMotif.py sequences_file width model_file positions_file
   perl LearnMotif.py sequences_file width model_file positions_file
   ```

   where `sequences_file` is the file containing the input sequences, `width` is the width of the motif model to learn, `model_file` is the name of a file to which you will output the learned motif model, and `positions_file` is the name of a file to which you will output the predicted location of the motif in each sequences. You may add other arguments as you see fit (e.g., maximum number of iterations, random number generator seed, etc.). Please document these extra arguments in your `README` file.

   The input `sequences_file` will contain the DNA sequences, with one sequence per line.

   The output `positions_file` should simply contain a list of the best position for the motif in each sequence, one position per line. The `model_file` should contain a tab-delimited profile matrix, with the background frequencies in the first column.

You may use either the EM algorithm (i.e., MEME OOPS), or the Gibbs sampling algorithm discussed in class for learning the motif model. For the EM algorithm, you should calculate the log likelihood, $\log P(X|\theta)$, after each iteration (this can be efficiently computed using intermediate values generated during the E step) and stop the algorithm when this value no longer increases above a fixed threshold. For the Gibbs sampling algorithm you should calculate the log complete-data likelihood, $\log P(X, Z|\theta)$ (with θ estimated from *all* sequences), after each iteration and keep track of the motif positions that maximized this value.

For both the Gibbs and EM algorithms, you must implement multiple starting points. Simple pseudocounts (e.g., $d_{c,k} = 1$) should be used for both as well. However, you need not implement any of the fancier aspects: dirichlet mixtures, phase shift correction, etc..

2. With `LearnMotif`, discover the motif of width 14 hidden in sequences in the file: `http://www.biostat.wisc.edu/bmi776/hw/hw1_hidden_motif.txt`.

3. Construct a *sequence logo* for the predicted motif sequences from (2) by using the WebLogo Web form (`http://weblogo.berkeley.edu/`). You will need to extract the subsequences corresponding to the motif occurrences to provide the proper input to the WebLogo service.

4. Search the JASPAR transcription factor binding profile database (`http://jaspar.genereg.net`) using the profile matrix that you learned in (2). Which transcription factor binds to this motif? You should be able to use the contents of the `model_file` for this search, after deleting the first (background) column.