BMI/CS 776 Spring 2015
Homework #3

Prof. Colin Dewey

Due Tuesday, March 10th, 2015 by 11:59pm

The goal of this assignment is to become familiar with the maximal dependence decomposition (MDD), interpolated Markov models (IMMs), and position weight matrix (PWM) models for short sequence motifs.

To turn in your assignment, copy all relevant files to the directory:
`/u/medinfo/handin/bmi776/hw3/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. You code must be able to run on the submission servers and may not use any third-party libraries (i.e., you may only use the standard library for your programming language of choice). 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, `pwm_predict`, that learns PWM models from training data and outputs scores for test data. The syntax for running the program should be:

   ```
   pwm_predict train_real train_false test test.scores
   ```

   where `train_real` is a file containing a set of positive training examples, `train_false` is a file containing a set of negative training examples, `test` is a file containing a set of test sequences for which you are to output scores, and `test.scores` is the file into which the scores will be written.

   Given the training data, you should learn two PWMs, one for the positive examples, $model_p$, and one for the negative examples, $model_n$. When estimating the parameters of the PWMs, use a pseudocount of one to avoid zeros. For each model, you should be able to calculate the likelihood of a sequence, $P(sequence|model)$. We will score each test sequence with the log likelihood ratio for the two models:

   $$score(sequence) = \log\left(\frac{P(sequence|model_p)}{P(sequence|model_n)}\right)$$

The output file (`test.scores`) will be a newline-separated list of the scores for each sequence in the test set. For example, if the test set consists of three sequences, the output file might look like:

```
0.013409
-1.767870
4.515226
```

2. Write a program, `mdd_predict`, that is the same as that in problem 1, but uses MDD models instead of PWM models. For stopping criteria, use 400 as the minimum number of sequences in a subset and 16.3 as the minimum χ^2 statistic to be considered significant (this corresponds to a p-value of 0.001 for a χ^2 test with 3 degrees of freedom). For your reference, the χ^2 statistic for a $n \times m$ contingency table, O, is:

$$\chi^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

where $E_{i,j}$ is the expected number of counts for the (i, j) entry:

$$E_{i,j} = \frac{R_i C_j}{N}$$

where R_i is the sum of the ith row, C_j is the sum of the jth column and N is the sum of the counts in the table. The degrees of freedom for a χ^2 test of such a contingency table is $(n-1)(m-1)$.

As you did for Problem 1, use a pseudocount of one when estimating the parameters of the PWMs at each node of the MDD tree. However, *do not* use pseudocounts when constructing the contingency table of counts for χ^2 testing. When computing the χ^2 statistic, ignore any cells of the contingency table that have zero expected value.

3. Write a program, `imm_predict`, that is the same as that in problem 1, but that uses an inhomogenous, 5th order interpolated Markov model. By inhomogenous, I mean that every position in the sequences will have its own 5th order IMM. Use the IMM approach of GLIMMER, with a count threshold of 40 instead of 400. As you did for Problems 1 and 2, use a pseudocount of one when estimating the parameters of the IMM, but **do not** use pseudocounts for the χ^2 contingency tables. For calculating p-values of the χ^2 test (which will have degrees of freedom = 3), your program may use the p-value table file provided at: `http://www.biostat.wisc.edu/bmi776/hw/hw3_chisquare_df3_pvalues`. (simply round your calculated χ^2 statistic to the nearest tenth and find the p-value from this table). Note that with the GLIMMER approach $d = 1 - p$, where p is the p-value of the χ^2 test.

4. Run your `pwm_predict`, `mdd_predict`, and `imm_predict` programs on the training and test files found on the HW Web site (these are real and false donor splice sites from the *Drosophila melanogaster* genome). Output scores for both the positive and negative test sets. By setting a threshold, T, for a score, we can use these programs as classifiers. Those sequences that score above T will be predicted as real, and those that don't will be predicted as false. For each program, sweep T from its minimum to maximum values and compute the *precision* and *recall* of the classifier for each T. Plot these points to obtain a *precision-recall* curve for each program. Does one model dominate the other?

5. Draw the structure of the MDD model obtained from the positive training examples. How does it compare to the structure of the MDD model used by Genscan?