MASSACHVSETTS INSTITVTE OF TECHNOLOGY

Problem Set 2
Due February 24

The following problems all deal with motif-finding, and the yeast transcription factor RCS1, which is believed to target a seven nucleotide motif. The course website has a zip-file containing all the sequence data you need, as well as partial implementations of algorithms in Matlab.

Problem 1

Begin by completing the given implementation of the EM algorithm. Missing code in the provided implementation is marked with a `MISSING` tag throughout; parts of at least four functions are missing some code. Make sure you've filled in the missing parts of `expectation.m`, `log_point_expectation.m`, `motif_log_prob.m`, and `expected_counts.m`.

Once you've completed the partial implementation, use it to try to find a motif for the binding site of the *S. cerevisiae* transcription factor RCS1. We've included a small set of sequences known to contain the RCS1 binding site in the accompanying file, `rcs1_genes.fasta`. Seed the algorithm with each of the following motif matrices, and one of your own choosing. For each seed, obtain 5 motifs by masking out each optimum alignment after converging. Report the five final motifs produced from each seed, then answer the discussion questions below. Several helper functions have been provided to ease the loading of all the sequences, the creation of the requisite motif-matrices, and the parsing of the results. Once you've filled in the missing code, you should be able to run the whole process in just a few lines of Matlab; see your TA if you are confused about how to load the input data, or view the results.

A

	Motif Position	1	2	3	4	5	6	7	B
Nucleotide	A	.25	.25	.25	.25	.25	.25	.25	.25
	C	.25	.25	.25	.25	.25	.25	.25	.25
	G	.25	.25	.25	.25	.25	.25	.25	.25
	T	.25	.25	.25	.25	.25	.25	.25	.25

B

	Motif Position	1	2	3	4	5	6	7	B
Nucleotide	A	.24	.25	.28	.24	.25	.25	.24	.25
	C	.24	.25	.24	.28	.25	.25	.24	.25
	G	.24	.25	.24	.24	.25	.25	.28	.25
	T	.28	.25	.24	.24	.25	.25	.24	.25

C

	Motif Position	1	2	3	4	5	6	7	B
Nucleotide	A	.24	.24	.24	.28	.24	.24	.24	.25
	C	.28	.24	.28	.24	.24	.24	.24	.25
	G	.24	.28	.24	.24	.24	.28	.28	.25
	T	.24	.24	.24	.24	.28	.24	.24	.25

Position "B" is the position-independent background. (We have provided functions in the code provided for building these matrices automatically).

Starting with the tasks outlined above, please answer the following questions:

a. Give working versions of the missing code in each of the five files listed above.

b. Report the 20 motifs (5 motifs from each of 4 seeds) obtained using your completed code.

c. Provide short written answers to each of the following questions concerning the use of EM for motif discovery:

(i) You'll notice that the top-level `em()` function provides a way to specify `pcounts`, or "pseudo-counts," for the updating of the motif-parameters at each iteration. *Pseudo-counts* represent (in a Bayesian sense) weak prior knowledge about the "shape" of the motif-matrix. Practically, they serve to smooth the results of the M-step, enabling us to prevent learning degenerate motif-matrices with probabilities of 0.0 in any position. Try your motif-discovery tool both *with* and *without* a weak pseudo-count matrix. What effect, if any, does this have on your results?

(ii) When using an EM algorithm for motif discovery, if we want to discover multiple motifs we need to *mask out* previously-discovered versions of the motif so that we don't find ourselves with the same result over and over again (EM is a deterministic algorithm). The code provided performs this masking in a specific way: it finds the top-scoring sites in each sequence, and masks them out *completely*. Briefly describe (without implementation) at least one other reasonable way of masking out discovered motifs. Can this procedure of sequential discovery and masking affect whether or not the algorithm ever settles on the "right" motif?

Problem 2

This problem is required only for graduate students who are taking the class for credit.

We selected the sequences for input to your EM algorithm in Part 1 by taking a subset of the regions at which RCS1 binding was measured in a whole-genome binding experiment. You've discovered a set (20) of motifs, one of which may characterize that binding event. How can you determine which, if any, of those motifs is the "right" one?

For your convenience, we have provided a scanning function, `scan_fasta_file.m`; it only requires a motif model and a threshold for determining matches, both in log-scale. (You'll also notice it uses the method `motif_log_prob.m` that you completed in the first problem.) Furthermore, the file `ut5_sc_1000.fasta` contains the sequences for 1000 bp in the upstream region of every ORF in the yeast genome.

This problem asks you to develop a way of using the hypergeometric distribution, discussed in class, to score how well a motif captures an underlying "bound" set.

a. Develop and implement a method, using the hypergeometric distribution, to score how well a discovered motif captures the underlying "bound" set of regions. Your method should assign a p-value to each motif, based on how likeliy it is that a randomly-selected set of regions (of equivalent size) would have captured as many of the bound regions. *Hint: You may find that your implementation has numerical problems if you calculate the hypergeometric distribution naively, given the large numbers involved. It maybe be useful to remember that* $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ *and that* $n! = \Gamma(n+1)$ *for integer values of* n*; furthermore, MATLAB contains a useful function,* `gammln()` *to help here, too.*

b. Provide your list of discovered motifs from Part 1, ranked by their hypergeometric score.

c. The hypergeometric score we've suggested here is based on a simple "keyword" approach to a motif occurrences: any sequence that contains *even one* occurrence of the motif is said to unequivocally have the site, and is included in the "true" set of the hypergeometric score calculation. However, we can see that this ignores some reasonable features of motifs – shouldn't a motif be more significant, if every sequence that contains a site has *at least* two of those sites? What if the motif has some consistent spatial arrangement or ordering, with respect to other motifs or sequence features? Briefly describe (without implementation) at least one alternative to the hypergeometric distribution for scoring motifs (perhaps even just a *different way* of using that distribution) that captures one of these count-based or arrangement-based phenomena.