CS5483 & CS4223 Homework 2

Due: Wed, Feb 26, 11:59pm

Submission via blackboard: one MATLAB source code file (for Q1), and one word document with outputs, figures, and answers to the questions (for both Q1 and Q2).

Problem 1 (40 points): Alignment Statistics

A). (10 points) Write a MATLAB program to calculate a series of scoring matrices for aligning DNA sequences with 60, 70, 80, and 90% of identity, assuming A, C, G and T have equal probabilities. Do not scale the scores to integers (so λ is equal to 1 in all the matrices). Output should be a 4x4 matrix and the row/column order of the nucleotides are A, C, G, T. (For example, the first row of the matrix shows the score of aligning an A again an A, C, G, or T respectively.)

B). (5 points) Assuming that you are using one of the matrices above to perform ungapped local alignment (i.e., gap penalty = minus infinity) between two 100bp-long DNA sequences, and got a score of 20. Using the extreme value distribution, calculate the E-value of the alignment score (assuming K = 0.1). Is this alignment significant?

C). (5 points) Redo (B) for score = 5.

D). (5 points) Redo (B), but you are comparing a sequence of length 100bp to a database of 1 million sequences, each of which is 100bp long, and the most similar sequence has an alignment score 20.

E). (15 points) Take the MATLAB swalign function (type in 'help swalign' for details), or revise your implementation of Smith-Waterman algorithm so that it can align two DNA sequences with different substitution matrix, and only outputs scores (no traceback needed). Use a fixed gap penalty of 5 per gap. (This gap penalty is large enough so you probably would not see any gap in your optimal local alignment). No trace back is necessary. Using the matrices you computed in (A) to align each pair of sequences in the fasta files (e.g., ident_60_seq_1 and ident_60_seq_2 are a pair.) For your information, ident_60 means the two sequences are expected to share something that are 60% similar to each other. Each sequence in the files is 500 bases long. You can use the function fastaread to read the sequences: *seqs = fastaread('hw2q2.fa')*. The variable seqs is a struct array and you can get the i-th sequence with seqs(i).Sequence, and the name of the sequence with seqs(i).Header.

There are 4 pairs of sequences, and 4 scoring matrices, so you will need to calculate 4 x 4 = 16 alignment scores.

Plot the alignment scores (y-axis) against the percent identities (x-axis) of the substitution matrices used to obtain the alignments. (The alignment scores for each pair of sequences should be plotted as one line; you will have a total of 4 lines.)

In addition, compute the p-value of each alignment score and plot the p-values (y-axis) against the percent identities (x-axis). It is best to plot the p-values in logarithm scale. (Use semilogy instead of plot.)

Properly annotate your figures with axis labels and legends. (The following matlab functions may be needed: plot, semilogy, xlabel, ylabel, title, and legend. Type in 'help plot' for usage of the plot function.)

Which substitution matrix gives the high scoring alignment for each sequence pair? Are these alignments significant?

To help you check the correctness of your implementation, the substitution matrix with sequence identity = 60% is 0.8755 for a match and -0.6286 for a mismatch. The alignment score that I got when aligning the two first pair of sequences using the matrix with 60% identity is 18.54.

Problem 2 (30 points) Sequence Databases and BLAST

NCBI is one of the largest and most comprehensive databases belonging to the NIH – National Institutes of Health (USA). Entrez is the search engine of NCBI, and can be accessed at `http://www.ncbi.nlm.nih.gov/`. You can use it to search for genes, proteins, genomes, publications and much more. Each type of information is in a separate database, but can be searched all together using Entrez.

To limit the results returned, you can limit your query to a particular database, and/or combine your query terms with field qualifiers and Boolean operators (AND, OR, NOT). See the help page at `http://www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html#Search_Fields_and_Qualifiers` for all field qualifiers.

Even with these qualifiers, you may still get a lot hits, as some of the database entries are highly redundant, representing essentially the same sequence with different identifiers. For this reason, NCBI has created a sub-database, RefSeq, which contains only non-redundant, highly annotated entries for genomic DNA, transcript (mRNA), and protein sequences.

A). Search the **Protein database** to find the sequence for a human gene called CD4, using the search string *CD4 [GENE] and "Homo sapiens" [ORGN]*, where [GENE] and [ORGN] are field qualifiers, and Homo sapiens is the scientific name for human. You should see more than 10 entries. Using the filters on the left, limit your source databases to RefSeq. You will see 6 entries, corresponding to various isoforms of this protein. (Check Wikipedia page on protein isoforms: http://en.wikipedia.org/wiki/Protein_isoform). Click to view details of the longest isoform, isoform 1 precursor. The sequence is displayed in a format called GenBank (or GenPept for protein), with annotations (features) appearing before the actual sequence. For some explanation of the format, see `http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html`.

What is the accession number (a unique identifier) of this sequence? How many amino acids does this protein have? What is the first five amino acids of this protein? Find out how to change the display to FASTA format, which is one of the simplest and most popular formats. (You've seen this in HW1.) Save the sequence in FASTA format to a text file.

B). Go to NCBI homepage and find the link to the BLAST web-page, and choose the *protein blast* program (blastp). Copy the human CD4 protein sequence (or just its accession number) you just saved to the query window. Pick the RefSeq_protein database as the search set. Down at the very bottom, click on "Algorithm Parameters", change *Max target sequences* to the max value, *Expect threshold* to 1, and the scoring matrix to BLOSUM45, and run blast.

Which five organisms have protein sequences that are most similar to human CD4? Google to find out the common names of the organisms. Among all the hits, can you find one sequence from the chicken? (The scientific name of chicken is Gallus Gallus). Use the chicken-human alignment for the next question.

C). Above the graphic summary section on the result page, you can find a link to "search summary", which shows some statistical parameters used to compute the significance of the alignment. In particular, you can see Lambda and K for *gapped alignment* (second column), and the size of the database. (It used to provide *effective lengths* of query and database, but that function has been removed, unfortunately.) Use these numbers, and the chicken-human CD4 alignment to show (1) how to get the bit score from the raw score; and (2) how to compute the E-value of an alignment using both the bit score and the raw score. (Because the effective lengths are not known, the E-value you computed will only be an approximation of what is shown by blast).

D). Go back to the BLAST homepage, under "BLAST Genomes", click "Human". Copy the human CD4 sequence you just saved to the query window. Select "RefSeq RNA" as the database. Choose an *appropriate program* (among the five programs shown at the top of the page, i.e., blastn, blastp, blastx, etc.) to align the protein sequence to the human reference RNA sequences. The top hit should be the corresponding reference mRNA sequence of this protein. What is the accession number of this reference sequence? How many exons are in the human CD4 gene?

E). Go back to the BLAST homepage, choose the nucleotide blast program. Paste the accession number of reference mRNA sequence you obtained in (D) to the query window. Change database to refseq_rna. On the very bottom of the page, click on "Algorithm parameters". Record the following parameters used by the program: word size, Match/Mismatch Scores, Gap costs. Run BLAST. How many hits are found? Save the following information about the *least significant* hit: sequence accession number, score, E-value, alignment length, percent of identities, and percent of gaps.

F). Repeat the experiment in (E), but change Program Selection to optimize for "somewhat similar sequence". Click on "Algorithm parameters" and compare the parameters with the ones you recorded in (E). Explain the difference. Run BLAST. How many hits are found this time? Is the least significant hit you found in (E) still in your result? If yes, compare its score, E-value, length, percent identities and gaps to the result in (E) and explain the difference.

Bonus (5 points)

How much time did you spend on this homework? Who did you discuss with and what was the discussion about? How is the difficulty level? Do you have any comments about the course?