

CS5263 Homework 2

Due: Wed, Oct 15, 7:00pm

Problem 1 (20 points): Alignment Statistics

- A). Write a program to compute a series of scoring matrices for aligning DNA sequences with 55, 60, 65, 70, 75, 80, 85, 90, and 95% 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).
- B). Implement the Smith-Waterman algorithm to align two sequences. The input of your program should be a file containing two lines of DNA sequences, and a file containing the substitution matrix. 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 alignment). Your implementation should also include a trace-back, and output the following five numbers for an optimal alignment: alignment score (S), length of alignment (L), number of gaps (G), number of matched columns (M), the percent identity of the alignment ($100M/L$), and the E-value of the alignment (assume $K = 0.1$). Using the matrices you computed in (1) to align each pair of sequences in the five files from http://cs.utsa.edu/~jruan/teaching/cs5263_fall_2008/hw/seqs.zip. Each sequence in the files is 1000 bases long.

Plot the alignment scores against the percent identities of the substitution matrices used to obtain the alignments. Which substitution matrix has given the best alignment for each sequence file? What can you say about the relationships between the sequences in each file? Are these alignments significant?

To help you check the correctness of your implementation, here are the numbers I got when aligning the two sequences in seq_25.txt using the matrix with 55% identity: $S = 13.1$, $L = 76$, $G = 0$, $M = 40$.

Problem 2 (30 points) Sequence Databases and BLAST

NCBI is one of the largest and most comprehensive databases belonging to the NIH – national institute 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). Use Entrez to search the protein sequence for a human gene called CD4, using field qualifiers [GENE] and [ORGN]. You should see 10 entries. Right below the query input boxes, you

should see a RefSeq tab; clicking on it gives you the RefSeq entry of this protein. 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. Save the sequence to a file in this format.

- B). Go to NCBI homepage and find the link to the BLAST web-page, and choose the *protein blast* program. Copy the human CD4 protein sequence you just saved to the query window. Pick RefSeq as the selected database. Down at the very bottom, click on “Algorithm Parameters”, change the expect threshold to 1 and the scoring matrix to BLOSUM80, 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 the hits, can you find one sequence from the chicken? (The scientific name of chicken is Gallus Gallus). Save the chicken-human alignment for the next question.

- C). At the bottom of the result page, you can find some statistical parameters used to compute the significance of the alignment. In particular, you can see Lambda and K for *gapped alignment*, and the *effective lengths* of the query and database. 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 the bit score.
- D). Go back to the BLAST homepage, under “Blast assembled genome”, choose “human”. Copy the human CD4 sequence you just saved to the query window. Select “RefSeq RNA” as the database. Choose an *appropriate program* 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.

Problem 3 (10 points) KMP and Failure Links

Compute $SP(i)$ for the following pattern: *actaactc*. Show the failure links that are constructed using the SP values. Illustrate how the failure links can be used to speed up the search of the pattern in the following text: *actaactaactc*. How many comparisons are needed by the naive algorithm to find the pattern? How many comparisons are needed by the KMP algorithm?

Problem 4 (10 points) Suffix Tree

- A). Draw a suffix tree for the string *taataaataa*. Label the edges and terminal nodes explicitly.
- B). **Longest nonoverlapping repeat.** A *nonoverlapping repeat* in a string S is a string w such that $S = xwywz$ (where x , y , and z are possibly empty strings). Show how to find a longest nonoverlapping repeat in a string S in $O(n)$ time, where n is the length of S , using a suffix tree

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?