Project 3 (due October 5): Multiple alignment

This project is about implementing the exact (dynamic programming based) and the 2-approximation algorithm for global sum-of-pairs multiple alignment.

The first part of the project is to implement two programs: sp_exact_3 that implements the exact algorithm for computing an optimal MSA of 3 sequences and its score (decribed on page 408 in BA, or in Section 14.6.1 in Gusfield's book), and sp_approx that implements the 2-approximation algorithm for any number of sequences (described in Section 8.2.2 in BA, or in Section 14.6.2 in Gusfield's book).

For both programs, the objective is to minimize the SP-score using the below score matrix and a gap penalty of 5.

	A	C	G	T
A	0	5	2	5
C	5	0	5	2
G	2	5	0	5
T	5	2	5	0

The second part of the project is to use your MSA programs (or any other MSA programs) to align the 8 sequences of length 200 in brca1-testseqs.fasta

as close to optimum as possible (the sequences corresponds to first 200 nucleotides in the mRNA from the BRCA1 gene in cow, wolf, chicken, human, macaque, mouse, chimpanzee, and rat).

To pass the project your group must upload a report (in pdf-format) and your code (in a zip-archive) via Blackboard cf. assignment below before **Thursday, October 5**, describing your experiments with the approximation algorithn, and give a presentation in class (see below) on **Thursday, October 12**, describing how you have aligned the 8 sequences in the best possible way.

This project will not be graded but you should be able to present your work and the underlying theory and algorithms at the oral exam. The project should be done in groups of 2-3 students.

Report

You should describe your work in a short report containing the following (the report must be uploaded via Blackboard cf. assignment below before **Thursday, October 5, 14:00**):

- Introduction:
  A short status of your work. Does everything work as expected, or are there any problems or unsolved issues.
- Methods:
  An overview of the implemented programs sp_exact_3 and sp_approx. Comment on design/implementation choices that differ significantly from what we have talked about in class. Be

precise in describing how to access and use your programs. Also explain how you have verified the correctness of your programs. The files testdata_short.txt

and testdata_long.txt each contain three sequences and the score of their alignment as computed by my implementation sp_exact_3. You are welcome to use this as (part of) your test data.

- Experiments:
  Answer the following questions:
  - What is the score of an optimal aligment of the first 3 sequences in brca1-testseqs.fasta

    (i.e. brca1_bos_taurus, brca1_canis_lupus and brca1_gallus_gallus) as computed by your program sp_exact_3? How does an optimal alignment look like?
  - What is the score of the alignment of the first 5 sequences in brca1-testseqs.fasta

    (i.e. brca1_bos_taurus, brca1_canis_lupus, brca1_gallus_gallus, brca1_homo_sapiens, and brca1_macaca_mulatta) as computed by your program sp_approx?

    Which of the 5 sequences is choosen as the 'center string'?

    When the center string has been fixed, there are fak(4)=24 different orders in which you can choose to add the remaining 4 sequences. Compute (and report) the score of the alignment that you produce for each of these orders. Comment on the differences (if any) that you observe.
  - Make an experiment comparing the scores of the alignments computed by sp_exact_3 and sp_approx that validates that the approximation ratio of sp_approx is $2(k-1)/k$ for k sequences. i.e 4/3 for three sequences.

    You should use the testdata in testseqs.zip

    that contains 20 fasta files (testseqs_10_3.fasta, testseqs_20_3.fasta, ..., testseqs_200_3.fasta) each containing 3 sequences of lengths 10, 20, ..., 200.

    For each triplet of sequences (i.e. each fasta file), you should compute the optimal score of an MSA using sp_exact_3 and the score of the alignment produced by sp_approx. Make a graph in which you plot the ratio of the computed scores for each sequence length. Comment on what you observe.

  The python script msa_sp_score.py

  (or msa_sp_score_3k.py if you are using Python 3.x) can be used to compute the SP-score of an alignment stored in FASTA format cf. the above distance matrix and gapcost.

Presentation

As explained above, the file brca1-testseqs.fasta

contains 8 sequences of length 200 that correspond to first 200 nucleotides in the mRNA from the BRCA1 gene in: cow, wolf, chicken, human, macaque, mouse, chimpanzee, and rat.

In this part of the project, you must compute a multiple alignment (using column-based sum-of-pairs score based on the score matrix and gap penalty above) with a score as close to optimum as possible of:

- seqs 1-3: brca1_bos_taurus, brca1_canis_lupus, brca1_gallus_gallus
- seqs 1-4: brca1_bos_taurus, brca1_canis_lupus, brca1_gallus_gallus, brca1_homo_sapiens
- seqs 1-5: brca1_bos_taurus, brca1_canis_lupus, brca1_gallus_gallus, brca1_homo_sapiens, brca1_macaca_mulatta
- seqs 1-6: brca1_bos_taurus, brca1_canis_lupus, brca1_gallus_gallus, brca1_homo_sapiens, brca1_macaca_mulatta, brca1_mus_musculus

Finally, you must also compute a multiple alignment (using column-based sum-of-pairs score) with a score as close to optimum as possible of the 8 full length BRCA1 genes in brca1-full.fasta

. Since these 8 sequences are 'real' dna data, it can actually contain other symbols than A, C, G and T due to sequencing, see fx this page on Wikipedia. Inspecting the sequences show that fx the rat sequence (rattus_norvegicus) contains two N's. You must decide how to handle this, e.g. substitute the N by any base you like.

You might be able to using the simple exact method for computing an optimal scoring multiple alignment of three or four sequences, but you will probably need other ways of finding a good scoring alignment of five or more sequences. You can use sp_approx or any other alignment heuristic.

You (i.e. your group) must prepare a short 5 minutes presentation of your work for presentation in class on **Thursday, October 12**. Submit your presentation in pdf-format via Blackboard before the class. If you for some reason cannot give a presentation in class, you must hand in a report covering the topics of the presentations.

The presentation must include:

- Your best scoring alignments for the five cases above, and their scores.
- An explanation of how you have computed your best scoring alignments along with information about the running time and space consumption of your methods.
- A reference to where the alignments can be downloaded in FASTA format such that we can check your score using the python script msa_sp_score.py (or msa_sp_score_3k.py if you are using Python 3.x).