Project 3 Algorithms in Bioinformatics

Bjarke Meyer Pedersen

Genoma Torruella Maseras

Paula Rodrigo Martín

20-03-2023

Introduction

In this project we implement 2 different ways to perform a multiple sequence alignment. The 2 approaches differ in the way the sequences are used. The first one aligns all the sequences, whereas the latter uses a ¨center string¨, which is a sequence to align all the others too.

Our implementation works and passes the tests (as described in section methods) with the sequences given for this exercise and there are not unsolved issues.

Methods

The algorithms for both solutions have been implemented in the file *alignment.py* and we can execute an interactive program from the command line by executing the file *alignments_main.py*:

```
nonatorruella@DESKTOP-NONA-9QDQ7G6:/mnt/c/Users/genon/uni/master/semestre2/algorithms/AiB projects/P3$ python3 alignments_main.py
Multiple alignment: do you want to align sequences with the exact method (e) or approx (a)?
e
Enter the path of the fasta file with the sequences:
input/sequences.fasta
Do you want to use the default settings for the substitution matrix and gap penalties? (y/n)
y
Do you want to see an optimal alignment and store it in a file? (y/n)
y
---T--G-CATGCTGAAACTTCTCAACCA
--AT--G-GAT-TT-AT-CTGCTC-TTCG
GT-TCCGAAAGGCTAGCGCTAGGC-GCC-
--Alignment saved in  output/generated_alignment_exact.fasta --
The optimal score:  198.0
Do you want to align more other sequences? (Y/N)
n
Vi ses!!
```

The program will ask to input a fasta file with the sequences and by default it will use the substitution matrix in the examples file. We can also give it the path of a file with the gap cost and the substitution matrix in this specified format, where the first line is the gap cost:

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

We can also execute both solutions in the command line by running sp_exact_3.py or sp_approx.py with the following arguments:

```
python3 sp_exact_3.py input/sequences.fasta input/subst_matrix.txt output.fasta

python3 sp_approx.py input/sequences.fasta input/subst_matrix.txt output.fasta
```

Experiments

- **What is the score of an optimal aligment of the first 3 sequences in brca1-testseqs.fasta as computed by your program sp_exact_3? How does an optimal alignment look like?**

  The optimal score is 790 and it retrieves the following alignment:

```
The optimal score is:  790.0
GCGAA---ATGTA-ACA-CG-GTAGAGGTGAT-CGGGGTG-CGTT-ATAC-GTGCGTGGTGACCTCGGTCGGTGT-TGACGGTGCCTGGGGTTCCTCAGAGTGTTTTGGGGTCTGAAGGATG-GACTTGTCAGTG-ATTGCCATTGGA
GACGTGCAAAATGTGCTTTCAGCCATGCAGAA-GAA-CTT-GGAGTGTCCAGTCTGTTTAGATGTGAT
ATGGATTTATCTGCGGATCGTGTTGAAGAAGTACAAAATGTTCTTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTTTCTACAAAGTGTGA-TCA-CA-TATTTTGCAAATTTT
G-TATGCTGAA-AC-TTCTCAACCA-GAGGAAGGGGCCTTCACAGTGTCC--TTTGTGTAAGAACGA-
ATGGATTTATCTGCGGATCATGTTGAAGAAGTACAAAATGTCCTCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTCTCTACAAAGTGTGA-CCA-CA-TATTTTGCAAATTTT
G-TATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAATGTCC--TTTGTGTAAGAATGA-
```

- **What is the score of the alignment of the first 5 sequences in brca1-testseqs.fasta as computed by your program sp_approx? Which of the 5 sequences is chosen as the 'center string'?**

The center string used for this algorithm is the first one given in the file and the optimal score is 4345 and it retrieves the following alignment:

```
The optimal score is:  4345
ATGGATTTATCTGCGGATCATGTTGAAGA-AG-TAC--AA-AAT-GTCC-TCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAATA-TGTCTGGAGTTGATCAAAGAGCCT-GTC-TCTACAAAGTGTGA-C-C-A-C--A-TATT
TTGCAAATTTTGTAT-G-C-TG-AAAC-T--TCTCAACCA-GAAGAAAGGGCCT-T-CAC--AAT-GTCC--TTTGTGTAAGAATGA
ATGGATTTATCTGCGGATCGTGTTGAAGA-AG-TAC--AA-AAT-GTTC-TTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAATA-TGTCTGGAGTTGATCAAAGAGCCT-GTT-TCTACAAAGTGTGA-T-C-A-C--A-TATT
TTGCAAATTTTGTAT-G-C-TG-AAAC-T--TCTCAACCA-GAGGAAGGGGCCT-T-CAC--AGT-GTCC--TTTGTGTAAGAACGA
GCGAA---AT--GTA-A-CACGGTAGAGGTGA-T-C--GG-GGT-G-CG-TTA-TAC-GTGCGTGGTGACCTCGGTCGGTGTT-GACGGTGCCTGGGGTTCCTCAGAGTGTTTTGGGGTCTGAAGGATG-GA-C-TTGTC--AGTGAT
T-GCCA-TTGGAGAC-G-TGCA-AAATGTGCTTTCAGCCATGCAGAA-GAAC-T-T-GG---AGT-GTCCAGTCTGTTTAGATGTGA
GTACCTTGATTT-CGTATTCTGA-GAGGC-TGCTGCTTAGCGGTAGCCCCTTGGT-TTCCGTG-GCAA-CGGAAA--AGCGCGGGA-A-T-TACAGA-TAAATTAAA-A-C-T-GCG-ACTGCGCGGCGTGAGCTC-G-CTGA-GACT
TCCTGGACGGGGGACAGGC-TGTGGGG-T--T-TC--TCA-GATAACTGGGCCCCTGCGCTCAGGAGGCC--TTCACCCTC---T--
ATGGATTTATCTGCTGTTCGCGTTGAAGA-AG-TAC--AA-AAT-GTCA-TTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAATC-TGTCTGGAGTTGATCAAGGAACCT-GTC-TCCACAAAGTGTGA-C-C-A-C--A-TATT
TTGCAGATTTTGCAT-G-C-TG-AAAC-T--TCTCAACCA-GAAGAAAGGGCCT-T-CAC--AGT-GTCC--TTTGTGTAAGAATGA
```

- **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.**

seqSize	approx_score	exact_score	ratio
10	60.0	70.0	0.857143
20	140.0	135.0	1.037037
30	259.0	231.0	1.121212
50	428.0	385.0	1.111688
70	565.0	516.0	1.094961

In the table to the left we can see the sequence size, the scores for both the exact and approximation algorithms and their ratio (approx/exact). The ratio

increases with the sequence size (also shown in Fig 1) because for longer sequences, the exact algorithm performs better in aligning the sequences, since we retrieve the minimal score for the alignment. The approximation algorithm performs poorer for sequences longer than 25 nucleotides. With this figure we also validate that the approximation ratio of sp_approx is 2(k-1)/k for k sequences, in this case, 4/3 because we are using 3 sequences.

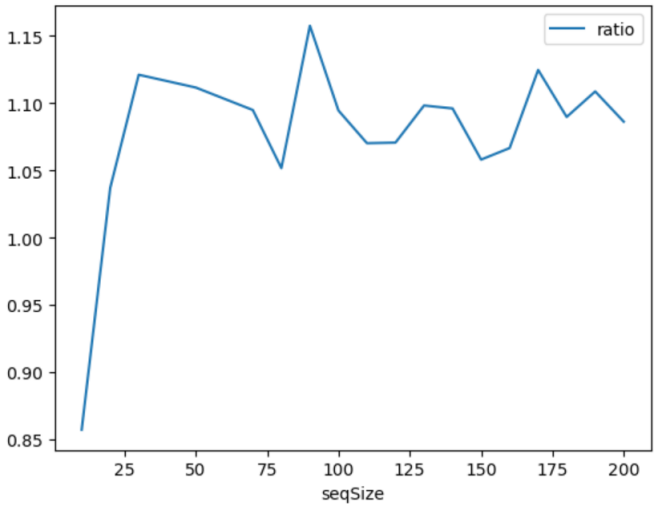

Figure 1: ratio of the scores for the approximation and exact algorithms plotted against the size of the sequences.