Project 3 – Multiple Alignment

Group – 7

Shakil Ahmed – 202402609

Al Ashrif Bin Ahamed – 202402610

Introduction:

In this project we have implemented two programs for global multiple sequence alignment (MSA):

- **sp_exact_3:** An exact algorithm for computing the optimal sum-of-pairs (SP) alignment score and alignment for exact three sequences. This method uses a three-dimensional dynamic programming approach and guarantees an optimal solution.
- **sp_approx:** A 2-approximation algorithm for MSA that can handle an arbitrary (>=3) number of sequences. This method selects a "center" sequence based on pairwise alignment costs and then aligns all sequences to the center. Thus, this method does not guarantee an optimal solution.

Overall, both programs work as expected on the test data (***testseqs*** **folder**). The exact algorithm produces the optimal alignment for three sequences (as verified using provided test data in ***testdata_short.txt*** and ***testdata_long.txt***), while the approximate method scales to more sequences. one unsolved challenge is that for relatively short sequences (e.g., 200 nucleotides), our highly optimized ***sp_exact_3*** (compiled with ***Numba***) sometimes runs faster than ***sp_approx*** - this is due to the parallelization and merging overhead inherent in the approximate method. However, as sequence lengths increase or the number of sequences grows, ***sp_approx*** is expected to show its scalability advantages.

Method:

sp_exact_3

Algorithm Overview:

The *sp_exact_3* program aligns three DNA sequences exactly by minimizing the sum-of-pairs (SP) score. It uses a 3D dynamic programming (DP) table to account for every position in each sequence simultaneously. The user can provide sequences either manually or through a FASTA file, customize the scoring scheme with a substitution matrix and a gap penalty, and then receive the final alignment along with its total SP score.

Implementation and Design Choices:

1. **Input:**
    - **Substitution Matrix**: Supplied by default or loaded from a file or entered manually (4×4 format for A, C, G, T).
    - **Gap Penalty**: A single integer for scoring gaps, applied uniformly to each gap position.
    - **Sequence Input (Manual)**: Prompts for three sequences, replacing ambiguous nucleotides with 'A'.
    - **Sequence Input (FASTA)**: Reads sequences from a file. If there are exactly three, they are used directly. If there are more, the user chooses which three to align. If fewer than three, missing sequences are obtained either from more FASTA files or from manual input.
2. **Output:**

   - **Alignment**: Shows three aligned sequences and the computed alignment score. The DP ensures the global optimal SP alignment of three sequences.
   - **Optional Save**: Exports the alignment as FASTA.
   - **Optional Verification**: Compares the computed score with a test function in ***msa_sp_score_3k.py***.
3. **Optimization:**
   - We implemented Numba's JIT compilation to optimize the innermost loops of our DP computation, which reduces constant overhead and enables rapid execution even with cubic complexity.
4. **Visualization:**
   - The program visualizes the substitution matrix and the optimal alignments in the program. Also, for the manual entry of substitution matrix if a user input different phylip-like format than the program also suggests the user with a visualization of supported matrix format.
5. **Verification:**
   - We verified the correctness of ***sp_exact_3*** by testing on the provided ***testdata_short.txt*** and ***testdata_long.txt*** files, ensuring that our computed SP scores match those computed by the instructor's implementation.

Usage:

- Run the *sp_exact_3* module by typing – ***python sp_exact_3.py*** (It will show the CLI-interface).
- We recommend to install **numba** and **rich** (for the beautiful interface) – ***pip install numba, rich.***
- Select, enter or load a substitution matrix.
- Enter a gap penalty integer.
- Choose how to input sequences (manual or FASTA).
- Let the program perform the 3D DP alignment and display results.
- Save the alignment if desired.
- Optionally verify the SP score using the ***msa_sp_score_3k.py*** test function.

sp_approx

Algorithm Overview:

The sp_approx implements a 2-approximation method for multiple sequence alignment, focusing on DNA sequences that may contain ambiguous letters. It automatically replaces any character outside of A, C, G, T with 'A,' preserving alignment feasibility. The key objective is to produce a reasonably good alignment in a more efficient manner compared to an exact 3D approach, especially when dealing with three or more sequences.

Design and Implementation Choices:

1. **Input:**
   - **Manual**: Users can type in an arbitrary number of sequences, each of which is substituted to remove non-standard nucleotides.
   - **FASTA File**: One file can contain multiple sequences, from which the user selects how many to include in the alignment.
2. **Output:**
   - **Computed Alignment**: The program displays the center-star alignment of all chosen sequences and calculates a sum-of-pairs (SP) score.
   - **Optional Save**: Results can be saved to a FASTA file.
   - **Optional Verification**: The script can verify the output against a separate scoring function.
3. **Center Selection:**

    - Identifies a center sequence by computing pairwise alignment (Needleman) scores between every pair of sequences and picking the one with the smallest total distance to others. Then the algorithm aligns each remaining sequences to the chosen center sequence.

4. **Pairwise Scoring:**
    - Uses a gap cost and a substitution matrix mapping each nucleotide pair to an integer cost. (Default scoring matrix). – we did not extend it but we will implement it and update it in my GitHub.
5. **Parallelism:**
    - Pairwise scoring for all sequence pairs runs in parallel with a "***ProcessPoolExecutor***" speeding up the center determination step for larger numbers of sequences.
6. **Verification:**
    - Verification was performed by comparing the SP scores computed by sp_exact_3 and sp_approx on test datasets. We also computed the approximation ratio, which for three sequences should be at most 4/3.

Usage:

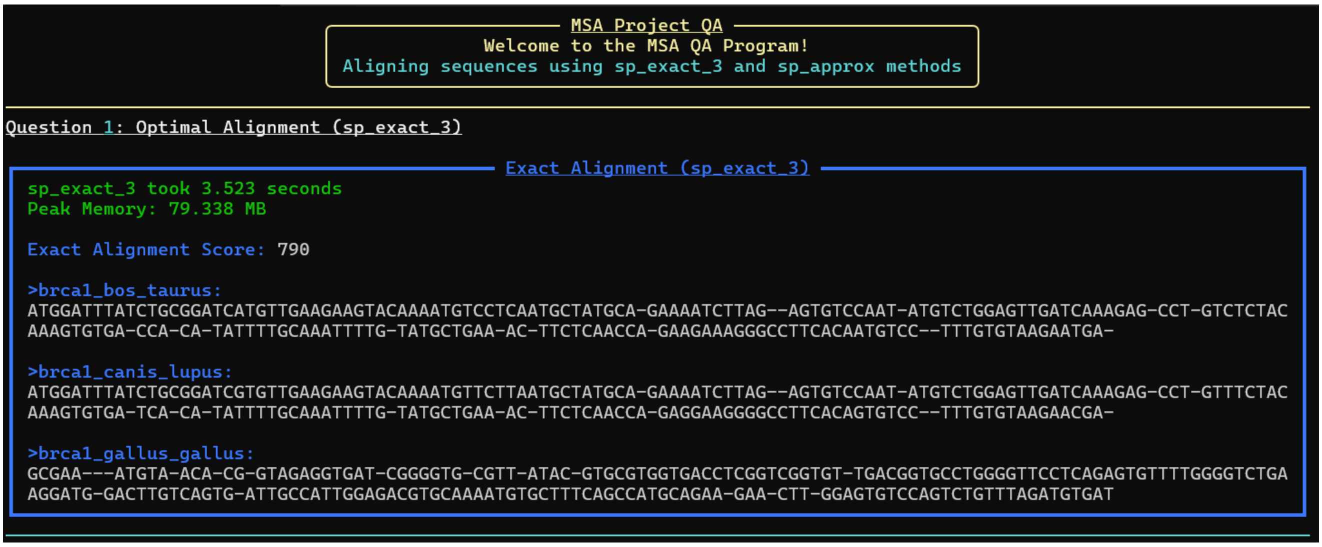
- Run the script (***python sp_approx.py***).
- **Select Input Mode** (1 for manual, 2 for FASTA).
- **Enter or Select** the DNA sequences. If fewer than two exist, alignment cannot proceed.
- **View** the final alignment and SP score.
- **Optionally Save** the alignment to FASTA.
- **Optionally Verify** the alignment score with the ***msa_sp_score_3k.py*** function.

Experiments:

By running the "***python QA.py"*** program we can get the answers of these experimental questions -

- **Question 1: What is the score of an optimal alignment 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?

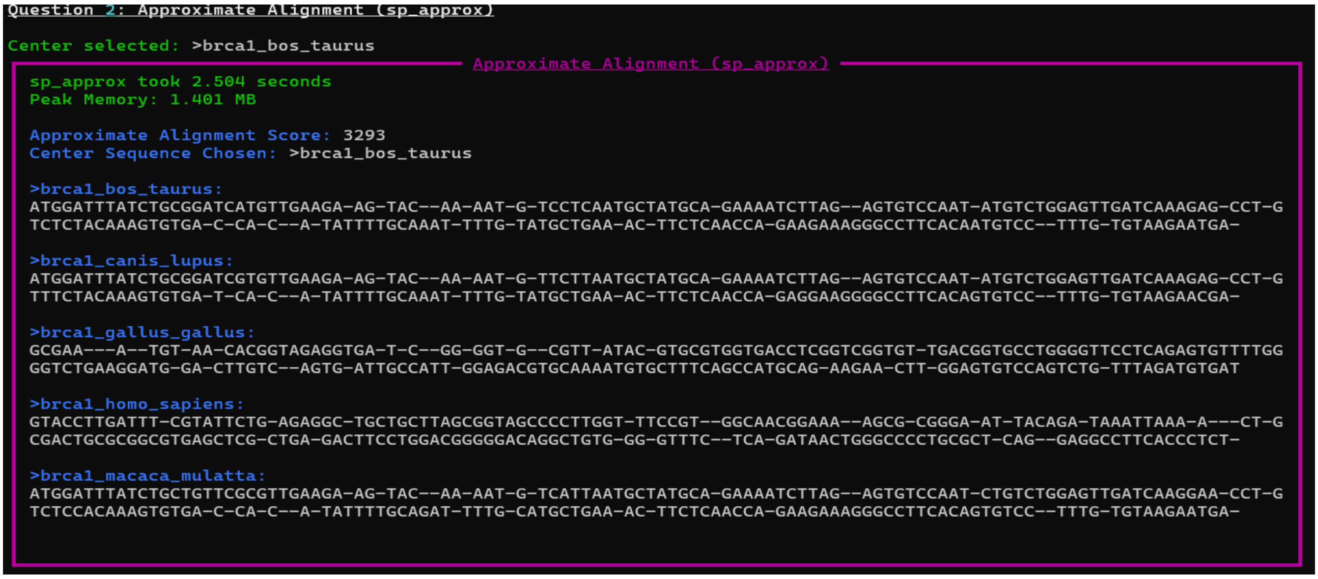
Answer:



```
                        MSA Project QA
                 Welcome to the MSA QA Program!
      Aligning sequences using sp_exact_3 and sp_approx methods

Question 1: Optimal Alignment (sp_exact_3)

                           Exact Alignment (sp_exact_3)
  sp_exact_3 took 3.523 seconds
  Peak Memory: 79.338 MB

  Exact Alignment Score: 790

  >brca1_bos_taurus:
  ATGGATTTATCTGCGGATCATGTTGAAGAAGTACAAAATGTCCTCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTCTCTAC
  AAAGTGTGA-CCA-CA-TATTTTGCAAATTTTG-TATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAATGTCC--TTTGTGTAAGAATGA-

  >brca1_canis_lupus:
  ATGGATTTATCTGCGGATCGTGTTGAAGAAGTACAAAATGTTCTTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTTTCTAC
  AAAGTGTGA-TCA-CA-TATTTTGCAAATTTTG-TATGCTGAA-AC-TTCTCAACCA-GAGGAAGGGGCCTTCACAGTGTCC--TTTGTGTAAGAACGA-

  >brca1_gallus_gallus:
  GCGAA---ATGTA-ACA-CG-GTAGAGGTGAT-CGGGGTG-CGTT-ATAC-GTGCGTGGTGACCTCGGTCGGTGT-TGACGGTGCCTGGGGTTCCTCAGAGTGTTTTGGGGTCTGA
  AGGATG-GACTTGTCAGTG-ATTGCCATTGGAGACGTGCAAAATGTGCTTTCAGCCATGCAGAA-GAA-CTT-GGAGTGTCCAGTCTGTTTAGATGTGAT
```

- **Question 2:** 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'?

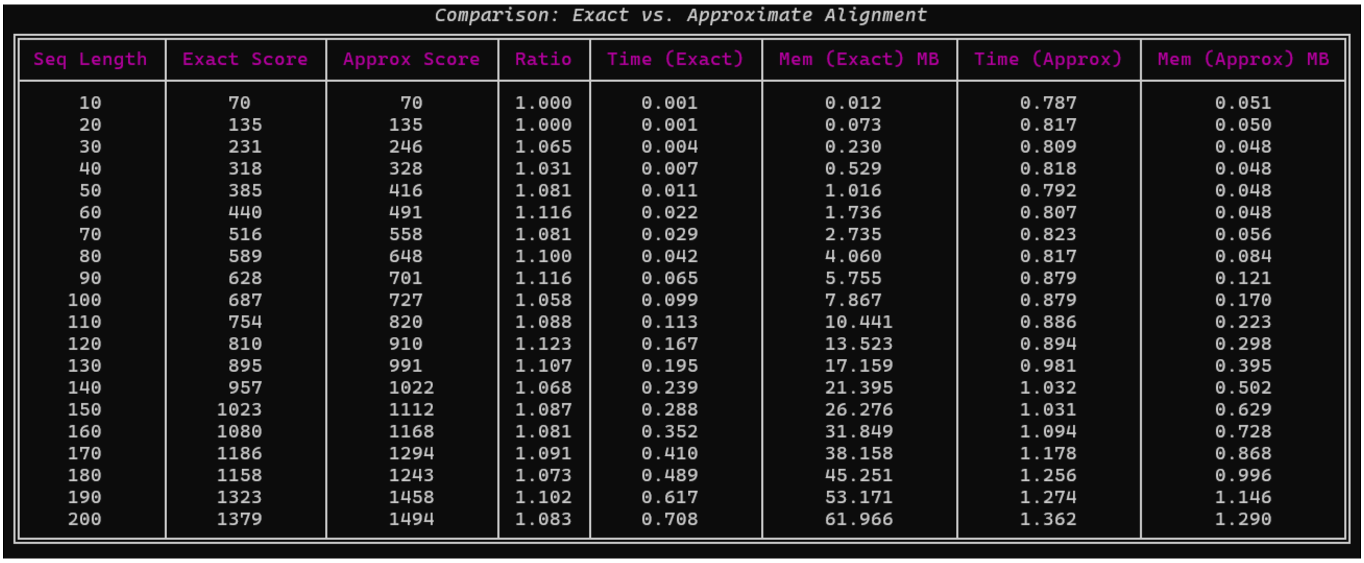
Answer:



```
Question 2: Approximate Alignment (sp_approx)

Center selected: >brca1_bos_taurus
                                       Approximate Alignment (sp_approx)
  sp_approx took 2.504 seconds
  Peak Memory: 1.401 MB

  Approximate Alignment Score: 3293
  Center Sequence Chosen: >brca1_bos_taurus

  >brca1_bos_taurus:
  ATGGATTTATCTGCGGATCATGTTGAAGA-AG-TAC--AA-AAT-G-TCCTCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-G
  TCTCTACAAAGTGTGA-C-CA-C--A-TATTTTGCAAAT-TTTG-TATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAATGTCC--TTTG-TGTAAGAATGA-

  >brca1_canis_lupus:
  ATGGATTTATCTGCGGATCGTGTTGAAGA-AG-TAC--AA-AAT-G-TTCTTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-G
  TTTCTACAAAGTGTGA-T-CA-C--A-TATTTTGCAAAT-TTTG-TATGCTGAA-AC-TTCTCAACCA-GAGGAAGGGGCCTTCACAGTGTCC--TTTG-TGTAAGAACGA-

  >brca1_gallus_gallus:
  GCGAA---A--TGT-AA-CACGGTAGAGGTGA-T-C--GG-GGT-G--CGTT-ATAC-GTGCGTGGTGACCTCGGTCGGTGT-TGACGGTGCCTGGGGTTCCTCAGAGTGTTTTGG
  GGTCTGAAGGATG-GA-CTTGTC--AGTG-ATTGCCATT-GGAGACGTGCAAAATGTGCTTTCAGCCATGCAG-AAGAA-CTT-GGAGTGTCCAGTCTG-TTTAGATGTGAT

  >brca1_homo_sapiens:
  GTACCTTGATTT-CGTATTCTG-AGAGGC-TGCTGCTTAGCGGTAGCCCCTTGGT-TTCCGT--GGCAACGGAAA--AGCG-CGGGA-AT-TACAGA-TAAATTAAA-A---CT-G
  CGACTGCGCGGCGTGAGCTCG-CTGA-GACTTCCTGGACGGGGGACAGGCTGTG-GG-GTTTC--TCA-GATAACTGGGCCCCTGCGCT-CAG--GAGGCCTTCACCCTCT-

  >brca1_macaca_mulatta:
  ATGGATTTATCTGCTGTTCGCGTTGAAGA-AG-TAC--AA-AAT-G-TCATTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-CTGTCTGGAGTTGATCAAGGAA-CCT-G
  TCTCCACAAAGTGTGA-C-CA-C--A-TATTTTGCAGAT-TTTG-CATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAGTGTCC--TTTG-TGTAAGAATGA-
```

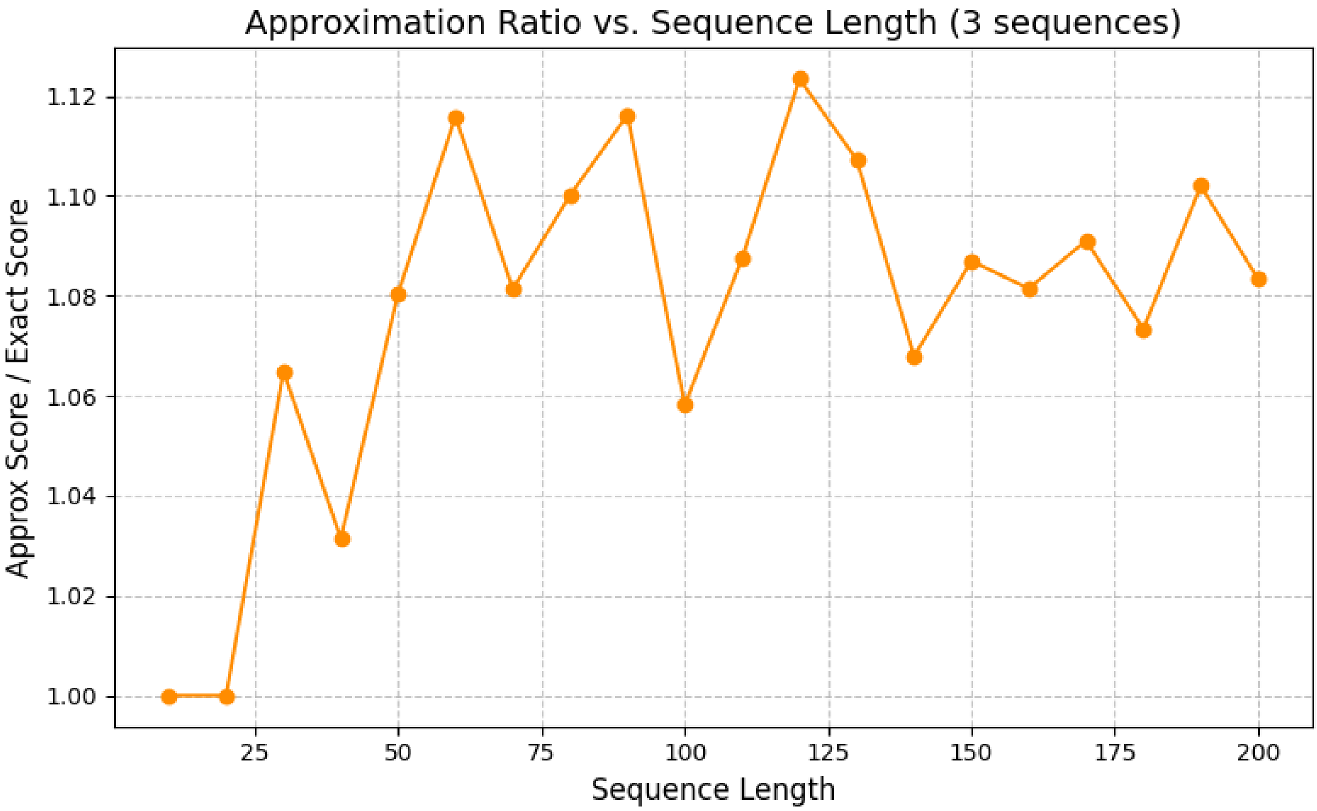
- **Question 3:** Make an experiment comparing the scores of the alignments computed by sp_exact_3 and sp_approx that validates that the approximation ratio of sp_approx is 2(k-1)/k for k sequences. i.e 4/3 for three sequences. For each triplet of sequences (i.e. each fasta file), you should compute 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.

Answer:



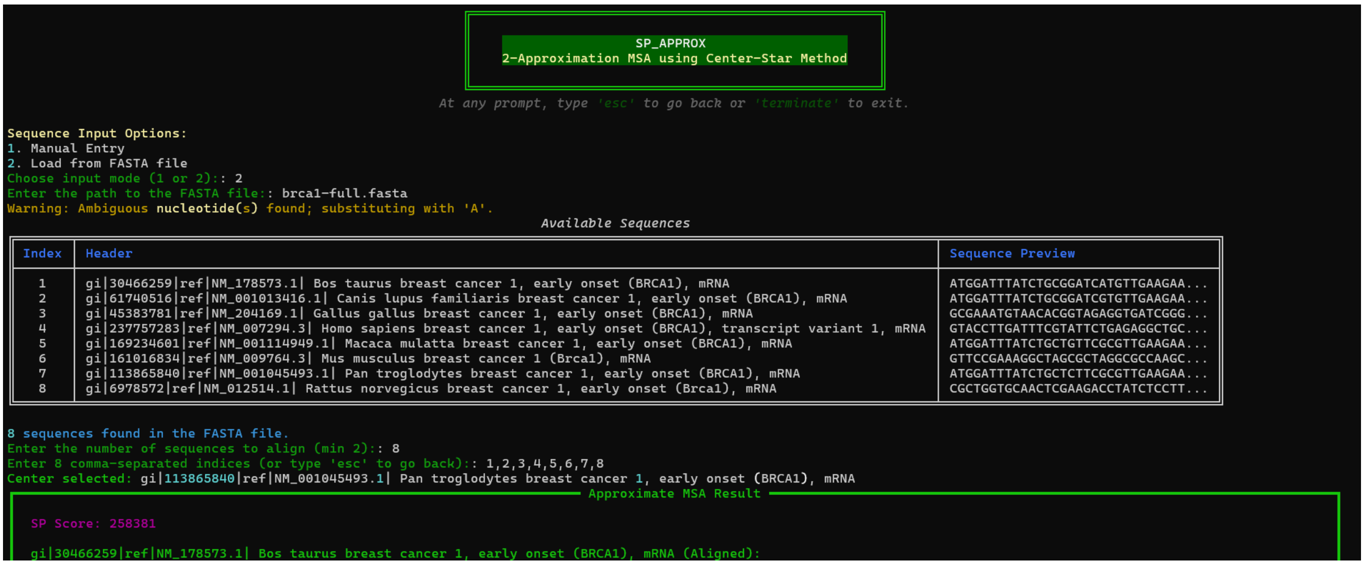
```
                          Comparison: Exact vs. Approximate Alignment
 ╔════════════╦═════════════╦══════════════╦═══════╦══════════════╦═══════════════╦══════════════╦════════════════╗
 ║ Seq Length ║ Exact Score ║ Approx Score ║ Ratio ║ Time (Exact) ║ Mem (Exact) MB ║ Time (Approx) ║ Mem (Approx) MB ║
 ╠════════════╬═════════════╬══════════════╬═══════╬══════════════╬═══════════════╬══════════════╬════════════════╣
 ║     10     ║      70     ║       70     ║ 1.000 ║     0.001    ║      0.012     ║     0.787     ║      0.051      ║
 ║     20     ║     135     ║      135     ║ 1.000 ║     0.001    ║      0.073     ║     0.817     ║      0.050      ║
 ║     30     ║     231     ║      246     ║ 1.065 ║     0.004    ║      0.230     ║     0.809     ║      0.048      ║
 ║     40     ║     318     ║      328     ║ 1.031 ║     0.007    ║      0.529     ║     0.818     ║      0.048      ║
 ║     50     ║     385     ║      416     ║ 1.081 ║     0.011    ║      1.016     ║     0.792     ║      0.048      ║
 ║     60     ║     440     ║      491     ║ 1.116 ║     0.022    ║      1.736     ║     0.807     ║      0.048      ║
 ║     70     ║     516     ║      558     ║ 1.081 ║     0.029    ║      2.735     ║     0.823     ║      0.056      ║
 ║     80     ║     589     ║      648     ║ 1.100 ║     0.042    ║      4.060     ║     0.817     ║      0.084      ║
 ║     90     ║     628     ║      701     ║ 1.116 ║     0.065    ║      5.755     ║     0.879     ║      0.121      ║
 ║    100     ║     687     ║      727     ║ 1.058 ║     0.099    ║      7.867     ║     0.879     ║      0.170      ║
 ║    110     ║     754     ║      820     ║ 1.088 ║     0.113    ║     10.441     ║     0.886     ║      0.223      ║
 ║    120     ║     810     ║      910     ║ 1.123 ║     0.167    ║     13.523     ║     0.894     ║      0.298      ║
 ║    130     ║     895     ║      991     ║ 1.107 ║     0.195    ║     17.159     ║     0.981     ║      0.395      ║
 ║    140     ║     957     ║     1022     ║ 1.068 ║     0.239    ║     21.395     ║     1.032     ║      0.502      ║
 ║    150     ║    1023     ║     1112     ║ 1.087 ║     0.288    ║     26.276     ║     1.031     ║      0.629      ║
 ║    160     ║    1080     ║     1168     ║ 1.081 ║     0.352    ║     31.849     ║     1.094     ║      0.728      ║
 ║    170     ║    1186     ║     1294     ║ 1.091 ║     0.410    ║     38.158     ║     1.178     ║      0.868      ║
 ║    180     ║    1158     ║     1243     ║ 1.073 ║     0.489    ║     45.251     ║     1.256     ║      0.996      ║
 ║    190     ║    1323     ║     1458     ║ 1.102 ║     0.617    ║     53.171     ║     1.274     ║      1.146      ║
 ║    200     ║    1379     ║     1494     ║ 1.083 ║     0.708    ║     61.966     ║     1.362     ║      1.290      ║
 ╚════════════╩═════════════╩══════════════╩═══════╩══════════════╩═══════════════╩══════════════╩════════════════╝
```

N.B: While the approximate method's theoretical complexity is O($n^2 \cdot L^2$) (with k being the number of sequences), its overhead (especially for small sequences) can be higher than sp_exact_3 due to parallelization and the merging process.



The sp_approx method, which leverages parallelized pairwise alignment and a center heuristic, offers a scalable approximation for larger datasets, with a theoretical guarantee on the approximation ratio (2(k–1)/k) or for 3 sequence its 4/3. As sequence length increases, the approximation ratio remains within the theoretical bound (approximately 1.333 for three sequences), validating the theoretical guarantee of the center–star algorithm.

Also, we consider to test **brca1-full.fasta,** and we successfully did it with our program.



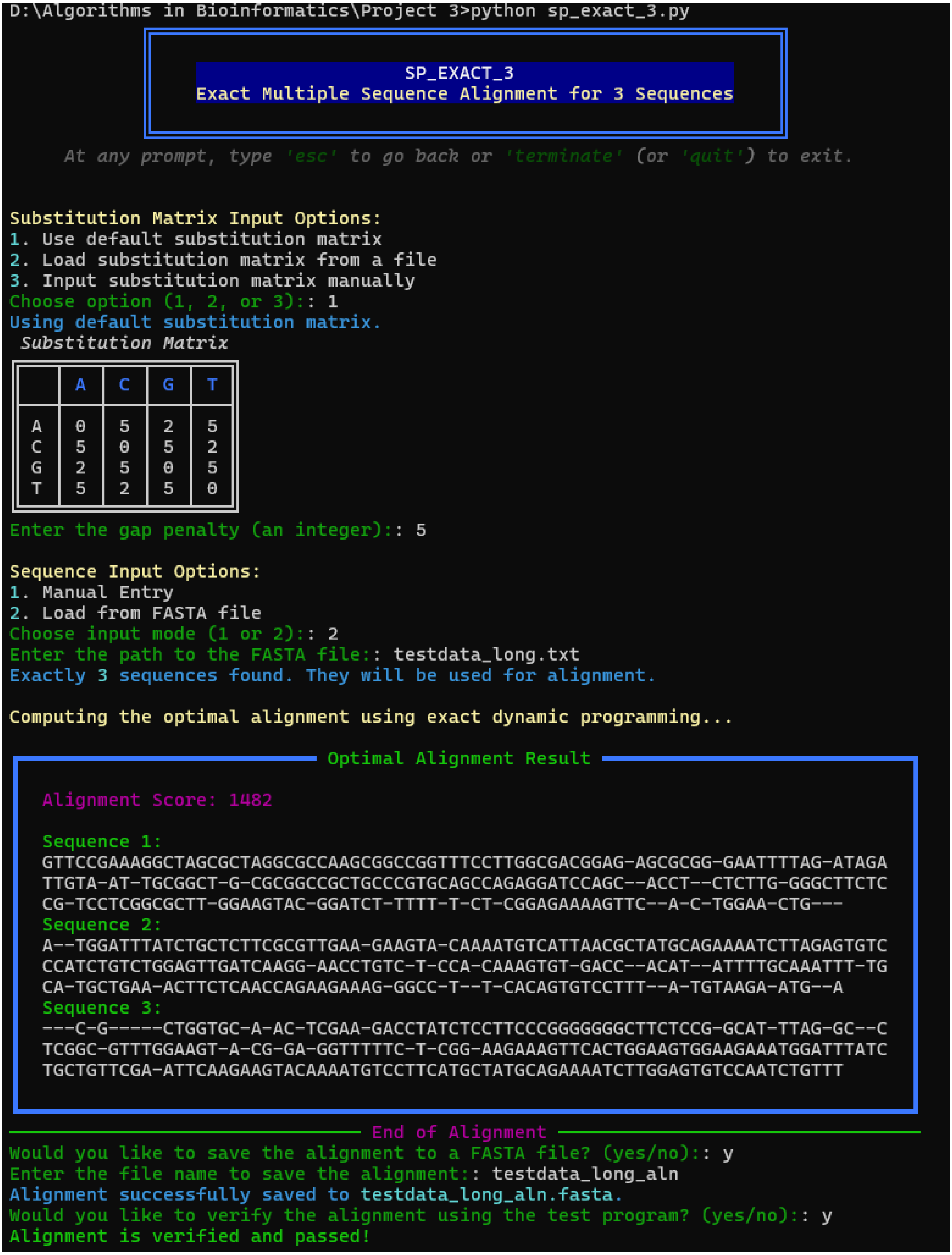
```
                                   SP_APPROX
                  2-Approximation MSA using Center-Star Method

            At any prompt, type 'esc' to go back or 'terminate' to exit.

Sequence Input Options:
1. Manual Entry
2. Load from FASTA file
Choose input mode (1 or 2):: 2
Enter the path to the FASTA file:: brca1-full.fasta
Warning: Ambiguous nucleotide(s) found; substituting with 'A'.
                                   Available Sequences
 Index  Header                                                                        Sequence Preview
   1    gi|30466259|ref|NM_178573.1| Bos taurus breast cancer 1, early onset (BRCA1), mRNA            ATGGATTTATCTGCGGATCATGTTGAAGAA...
   2    gi|61740516|ref|NM_001013416.1| Canis lupus familiaris breast cancer 1, early onset (BRCA1), mRNA  ATGGATTTATCTGCGGATCGTGTTGAAGAA...
   3    gi|45383781|ref|NM_204169.1| Gallus gallus breast cancer 1, early onset (BRCA1), mRNA          GCGAAATGTAACACGGTAGAGGTGATCGGG...
   4    gi|237757283|ref|NM_007294.3| Homo sapiens breast cancer 1, early onset (BRCA1), transcript variant 1, mRNA  GTACCTTGATTTCGTATTCTGAGAGGCTGC...
   5    gi|169234601|ref|NM_001114949.1| Macaca mulatta breast cancer 1, early onset (BRCA1), mRNA     ATGGATTTATCTGCTGTTCGCGTTGAAGAA...
   6    gi|161016834|ref|NM_009764.3| Mus musculus breast cancer 1 (Brca1), mRNA                       GTTCCGAAAGGCTAGCGCTAGGCGCCAAGC...
   7    gi|113865840|ref|NM_001045493.1| Pan troglodytes breast cancer 1, early onset (BRCA1), mRNA    ATGGATTTATCTGCTCTTCGCGTTGAAGAA...
   8    gi|6978572|ref|NM_012514.1| Rattus norvegicus breast cancer 1, early onset (Brca1), mRNA       CGCTGGTGCAACTCGAAGACCTATCTCCTT...

8 sequences found in the FASTA file.
Enter the number of sequences to align (min 2):: 8
Enter 8 comma-separated indices (or type 'esc' to go back):: 1,2,3,4,5,6,7,8
Center selected: gi|113865840|ref|NM_001045493.1| Pan troglodytes breast cancer 1, early onset (BRCA1), mRNA
──────────────────────────────── Approximate MSA Result ────────────────────────────────

  SP Score: 258381

  gi|30466259|ref|NM_178573.1| Bos taurus breast cancer 1, early onset (BRCA1), mRNA (Aligned):
```

Download:

For download the alignment
https://drive.google.com/drive/folders/1NduCVqLIBLg_Kmk3o98lABS5mWwc9Lv6?usp=drive_link

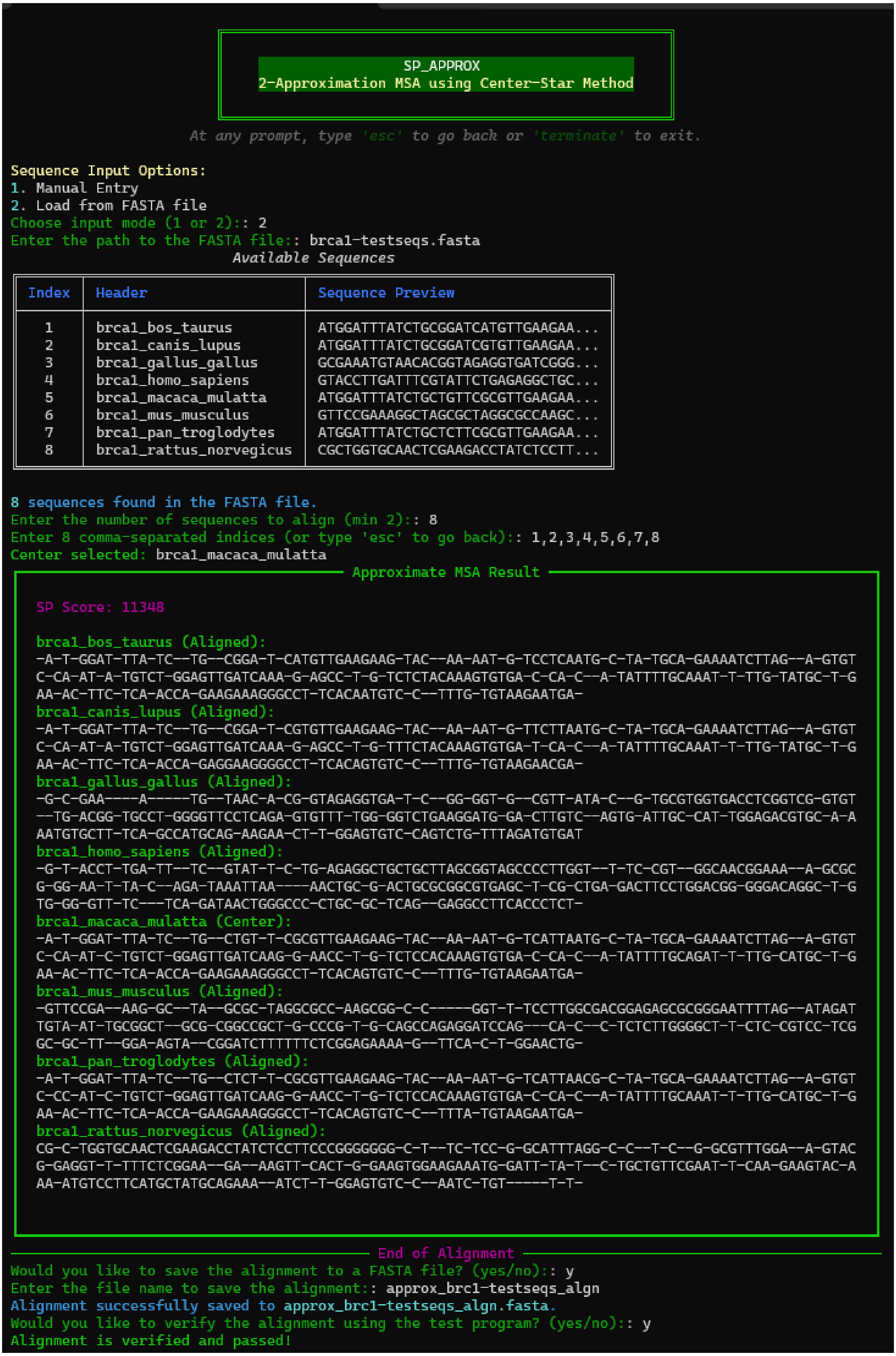
Appendix:

1. Run sp_exact_3.py



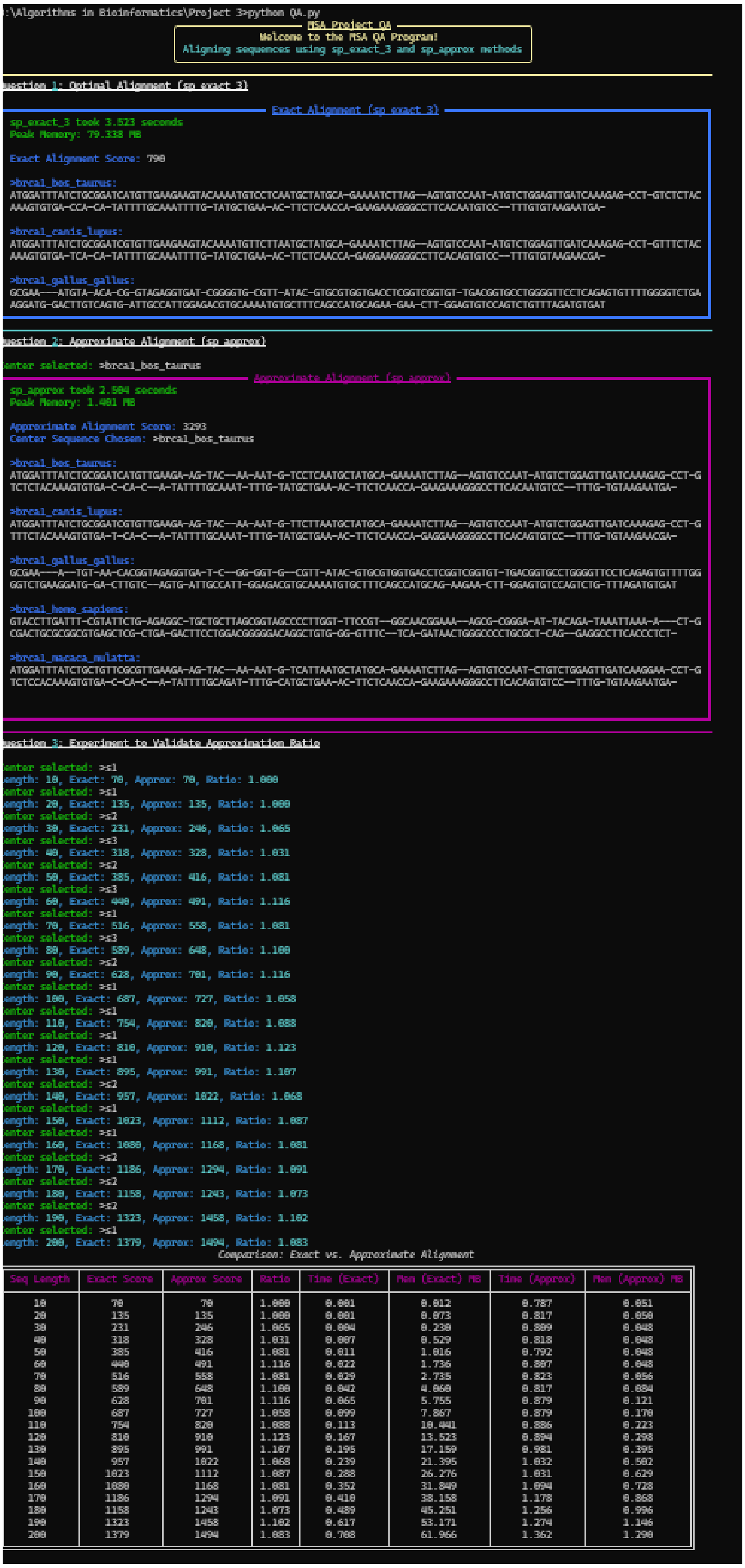
```
D:\Algorithms in Bioinformatics\Project 3>python sp_exact_3.py

                         SP_EXACT_3
           Exact Multiple Sequence Alignment for 3 Sequences

     At any prompt, type 'esc' to go back or 'terminate' (or 'quit') to exit.


Substitution Matrix Input Options:
1. Use default substitution matrix
2. Load substitution matrix from a file
3. Input substitution matrix manually
Choose option (1, 2, or 3):: 1
Using default substitution matrix.
 Substitution Matrix

    |  A |  C |  G |  T
 ---+----+----+----+----
  A |  0 |  5 |  2 |  5
  C |  5 |  0 |  5 |  2
  G |  2 |  5 |  0 |  5
  T |  5 |  2 |  5 |  0

Enter the gap penalty (an integer):: 5

Sequence Input Options:
1. Manual Entry
2. Load from FASTA file
Choose input mode (1 or 2):: 2
Enter the path to the FASTA file:: testdata_long.txt
Exactly 3 sequences found. They will be used for alignment.

Computing the optimal alignment using exact dynamic programming...

                            Optimal Alignment Result

   Alignment Score: 1482

   Sequence 1:
   GTTCCGAAAGGCTAGCGCTAGGCGCCAAGCGGCCGGTTTCCTTGGCGACGGAG-AGCGCGG-GAATTTTAG-ATAGA
   TTGTA-AT-TGCGGCT-G-CGCGGCCGCTGCCCGTGCAGCCAGAGGATCCAGC--ACCT--CTCTTG-GGGCTTCTC
   CG-TCCTCGGCGCTT-GGAAGTAC-GGATCT-TTTT-T-CT-CGGAGAAAAGTTC--A-C-TGGAA-CTG---
   Sequence 2:
   A--TGGATTTATCTGCTCTTCGCGTTGAA-GAAGTA-CAAAATGTCATTAACGCTATGCAGAAAATCTTAGAGTGTC
   CCATCTGTCTGGAGTTGATCAAGG-AACCTGTC-T-CCA-CAAAGTGT-GACC--ACAT--ATTTTGCAAATTT-TG
   CA-TGCTGAA-ACTTCTCAACCAGAAGAAAG-GGCC-T--T-CACAGTGTCCTTT--A-TGTAAGA-ATG--A
   Sequence 3:
   ---C-G-----CTGGTGC-A-AC-TCGAA-GACCTATCTCCTTCCCGGGGGGGCTTCTCCG-GCAT-TTAG-GC--C
   TCGGC-GTTTGGAAGT-A-CG-GA-GGTTTTTC-T-CGG-AAGAAAGTTCACTGGAAGTGGAAGAAATGGATTTATC
   TGCTGTTCGA-ATTCAAGAAGTACAAAATGTCCTTCATGCTATGCAGAAAATCTTGGAGTGTCCAATCTGTTT

                                End of Alignment
Would you like to save the alignment to a FASTA file? (yes/no):: y
Enter the file name to save the alignment:: testdata_long_aln
Alignment successfully saved to testdata_long_aln.fasta.
Would you like to verify the alignment using the test program? (yes/no):: y
Alignment is verified and passed!
```

2. Run ***sp_approx.py***



```
                           SP_APPROX
          2-Approximation MSA using Center-Star Method

           At any prompt, type 'esc' to go back or 'terminate' to exit.

Sequence Input Options:
1. Manual Entry
2. Load from FASTA file
Choose input mode (1 or 2):: 2
Enter the path to the FASTA file:: brca1-testseqs.fasta
                           Available Sequences
╔═══════╦═════════════════════════╦══════════════════════════════════╗
║ Index ║ Header                  ║ Sequence Preview                 ║
╠═══════╬═════════════════════════╬══════════════════════════════════╣
║   1   ║ brca1_bos_taurus        ║ ATGGATTTATCTGCGGATCATGTTGAAGAA...║
║   2   ║ brca1_canis_lupus       ║ ATGGATTTATCTGCGGATCGTGTTGAAGAA...║
║   3   ║ brca1_gallus_gallus     ║ GCGAAATGTAACACGGTAGAGGTGATCGGG...║
║   4   ║ brca1_homo_sapiens      ║ GTACCTTGATTTCGTATTCTGAGAGGCTGC...║
║   5   ║ brca1_macaca_mulatta    ║ ATGGATTTATCTGCTGTTCGCGTTGAAGAA...║
║   6   ║ brca1_mus_musculus      ║ GTTCCGAAAGGCTAGCGCTAGGCGCCAAGC...║
║   7   ║ brca1_pan_troglodytes   ║ ATGGATTTATCTGCTCTTCGCGTTGAAGAA...║
║   8   ║ brca1_rattus_norvegicus ║ CGCTGGTGCAACTCGAAGACCTATCTCCTT...║
╚═══════╩═════════════════════════╩══════════════════════════════════╝

8 sequences found in the FASTA file.
Enter the number of sequences to align (min 2):: 8
Enter 8 comma-separated indices (or type 'esc' to go back):: 1,2,3,4,5,6,7,8
Center selected: brca1_macaca_mulatta
┌─────────────────────────── Approximate MSA Result ───────────────────────────┐
│                                                                              │
│ SP Score: 11348                                                              │
│                                                                              │
│ brca1_bos_taurus (Aligned):                                                  │
│ -A-T-GGAT-TTA-TC--TG--CGGA-T-CATGTTGAAGAAG-TAC--AA-AAT-G-TCCTCAATG-C-TA-TGCA-GAAAATCTTAG--A-GTGT
│ C-CA-AT-A-TGTCT-GGAGTTGATCAAA-G-AGCC-T-G-TCTCTACAAAGTGTGA-C-CA-C--A-TATTTTGCAAAT-T-TTG-TATGC-T-G
│ AA-AC-TTC-TCA-ACCA-GAAGAAAGGGCCT-TCACAATGTC-C--TTTG-TGTAAGAATGA-
│ brca1_canis_lupus (Aligned):
│ -A-T-GGAT-TTA-TC--TG--CGGA-T-CGTGTTGAAGAAG-TAC--AA-AAT-G-TTCTTAATG-C-TA-TGCA-GAAAATCTTAG--A-GTGT
│ C-CA-AT-A-TGTCT-GGAGTTGATCAAA-G-AGCC-T-G-TTTCTACAAAGTGTGA-T-CA-C--A-TATTTTGCAAAT-T-TTG-TATGC-T-G
│ AA-AC-TTC-TCA-ACCA-GAGGAAGGGGCCT-TCACAGTGTC-C--TTTG-TGTAAGAACGA-
│ brca1_gallus_gallus (Aligned):
│ -G-C-GAA----A-----TG--TAAC-A-CG-GTAGAGGTGA-T-C--GG-GGT-G--CGTT-ATA-C--G-TGCGTGGTGACCTCGGTCG-GTGT
│ --TG-ACGG-TGCCT-GGGGTTCCTCAGA-GTGTTT-TGG-GGTCTGAAGGATG-GA-CTTGTC--AGTG-ATTGC-CAT-TGGAGACGTGC-A-A
│ AATGTGCTT-TCA-GCCATGCAG-AAGAA-CT-T-GGAGTGTC-CAGTCTG-TTTAGATGTGAT
│ brca1_homo_sapiens (Aligned):
│ -G-T-ACCT-TGA-TT--TC--GTAT-T-C-TG-AGAGGCTGCTGCTTAGCGGTAGCCCCTTGGT--T-TC-CGT--GGCAACGGAAA--A-GCGC
│ G-GG-AA-T-TA-C--AGA-TAAATTAA----AACTGC-G-ACTGCGCGGCGTGAGC-T-CG-CTGA-GACTTCCTGGACGG-GGGACAGGC-T-G
│ TG-GG-GTT-TC---TCA-GATAACTGGGCCC-CTGC-GC-TCAG--GAGGCCTTCACCCTCT-
│ brca1_macaca_mulatta (Center):
│ -A-T-GGAT-TTA-TC--TG--CTGT-T-CGCGTTGAAGAAG-TAC--AA-AAT-G-TCATTAATG-C-TA-TGCA-GAAAATCTTAG--A-GTGT
│ C-CA-AT-C-TGTCT-GGAGTTGATCAAG-G-AACC-T-G-TCTCCACAAAGTGTGA-C-CA-C--A-TATTTTGCAGAT-T-TTG-CATGC-T-G
│ AA-AC-TTC-TCA-ACCA-GAAGAAAGGGCCT-TCACAGTGTC-C--TTTG-TGTAAGAATGA-
│ brca1_mus_musculus (Aligned):
│ -GTTCCGA--AAG-GC--TA--GCGC-TAGGCGCC-AAGCGG-C-C-----GGT-T-TCCTTGGCGACGGAGAGCGCGGGAATTTTAG--ATAGAT
│ TGTA-AT-TGCGGCT--GCG-CGGCCGCT-G-CCCG-T-G-CAGCCAGAGGATCCAG---CA-C--C-TCTCTTGGGGCT-T-CTC-CGTCC-TCG
│ GC-GC-TT--GGA-AGTA--CGGATCTTTTTTCTCGGAGAAAA-G--TTCA-C-T-GGAACTG-
│ brca1_pan_troglodytes (Aligned):
│ -A-T-GGAT-TTA-TC--TG--CTCT-T-CGCGTTGAAGAAG-TAC--AA-AAT-G-TCATTAACG-C-TA-TGCA-GAAAATCTTAG--A-GTGT
│ C-CC-AT-C-TGTCT-GGAGTTGATCAAG-G-AACC-T-G-TCTCCACAAAGTGTGA-C-CA-C--A-TATTTTGCAAAT-T-TTG-CATGC-T-G
│ AA-AC-TTC-TCA-ACCA-GAAGAAAGGGCCT-TCACAGTGTC-C--TTTA-TGTAAGAATGA-
│ brca1_rattus_norvegicus (Aligned):
│ CG-C-TGGTGCAACTCGAAGACCTATCTCCTTCCCGGGGGGG-C-T--TC-TCC-G-GCATTTAGG-C-C--T-C--G-GCGTTTGGA--A-GTAC
│ G-GAGGT-T-TTTCTCGGAA--GA--AAGTT-CACT-G-GAAGTGGAAGAAATG-GATT-TA-T--C-TGCTGTTCGAAT-T-CAA-GAAGTAC-A
│ AA-ATGTCCTTCATGCTATGCAGAAA--ATCT-T-GGAGTGTC-C--AATC-TGT-----T-T-
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
─────────────────────────────── End of Alignment ───────────────────────────────
Would you like to save the alignment to a FASTA file? (yes/no):: y
Enter the file name to save the alignment:: approx_brc1-testseqs_algn
Alignment successfully saved to approx_brc1-testseqs_algn.fasta.
Would you like to verify the alignment using the test program? (yes/no):: y
Alignment is verified and passed!
```

3. Run ***python QA.py***



```
:\Algorithms in Bioinformatics\Project 3>python QA.py
                                  ╭──────────────── MSA Project QA ────────────────╮
                                  │           Welcome to the MSA QA Program!        │
                                  │ Aligning sequences using sp_exact_3 and sp_approx methods │
                                  ╰─────────────────────────────────────────────────╯
──────────────────────────────────────────────────────────────────────────────────────────
uestion 1: Optimal Alignment (sp_exact_3)

─────────────────────────────────── Exact Alignment (sp_exact_3) ─────────────────────────
 sp_exact_3 took 3.523 seconds
 Peak Memory: 79.338 MB

 Exact Alignment Score: 798

 >brca1_bos_taurus:
 ATGGATTTATCTGCGGATCATGTTGAAGAAGTACAAAATGTCCTCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTCTCTAC
 AAAGTGTGA-CCA-CA-TATTTTGCAAATTTTG-TATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAATGTCC--TTTGTGTAAGAATGA-

 >brca1_canis_lupus:
 ATGGATTTATCTGCGGATCGTGTTGAAGAAGTACAAAATGTTCTTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-GTTTCTAC
 AAAGTGTGA-TCA-CA-TATTTTGCAAATTTTG-TATGCTGAA-AC-TTCTCAACCA-GAGGAAGGGGCCTTCACAGTGTCC--TTTGTGTAAGAACGA-

 >brca1_gallus_gallus:
 GCGAA---ATGTA-ACA-CG-GTAGAGGTGAT-CGGGGTG-CGTT-ATAC-GTGCGTGGTGACCTCGGTCGGTGT-TGACGGTGCCTGGGGTTCCTCAGAGTGTTTTGGGGTCTGA
 AGGATG-GACTTGTCAGTG-ATTGCCATTGGAGACGTGCAAAATGTGCTTTCAGCCATGCAGAA-GAA-CTT-GGAGTGTCCAGTCTGTTTAGATGTGAT
──────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────────────────────────────
uestion 2: Approximate Alignment (sp_approx)

enter selected: >brca1_bos_taurus
─────────────────────────────── Approximate Alignment (sp_approx) ────────────────────────
 sp_approx took 2.504 seconds
 Peak Memory: 1.401 MB

 Approximate Alignment Score: 3293
 Center Sequence Chosen: >brca1_bos_taurus

 >brca1_bos_taurus:
 ATGGATTTATCTGCGGATCATGTTGAAGA-AG-TAC--AA-AAT-G-TCCTCAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-G
 TCTCTACAAAGTGTGA-C-CA-C--A-TATTTTGCAAAT-TTTG-TATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAATGTCC--TTTG-TGTAAGAATGA-

 >brca1_canis_lupus:
 ATGGATTTATCTGCGGATCGTGTTGAAGA-AG-TAC--AA-AAT-G-TTCTTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-ATGTCTGGAGTTGATCAAAGAG-CCT-G
 TTTCTACAAAGTGTGA-T-CA-C--A-TATTTTGCAAAT-TTTG-TATGCTGAA-AC-TTCTCAACCA-GAGGAAGGGGCCTTCACAGTGTCC--TTTG-TGTAAGAACGA-

 >brca1_gallus_gallus:
 GCGAA---A--TGT-AA-CACGGTAGAGGTGA-T-C--GG-GGT-G--CGTT-ATAC-GTGCGTGGTGACCTCGGTCGGTGT-TGACGGTGCCTGGGGTTCCTCAGAGTGTTTTGG
 GGTCTGAAGGATG-GA-CTTGTC--AGTG-ATTGCCATT-GGAGACGTGCAAAATGTGCTTTCAGCCATGCAG-AAGAA-CTT-GGAGTGTCCAGTCTG-TTTAGATGTGAT

 >brca1_homo_sapiens:
 GTACCTTGATTT-CGTATTCTG-AGAGGC-TGCTGCTTAGCGGTAGCCCCTTGGT-TTCCGT--GGCAACGGAAA--AGCG-CGGGA-AT-TACAGA-TAAATTAAA-A---CT-G
 CGACTGCGCGGCGTGAGCTCG-CTGA-GACTTCCTGGACGGGGGACAGGCTGTG-GG-GTTTC--TCA-GATAACTGGGCCCCTGCGCT-CAG--GAGGCCTTCACCCTCT-

 >brca1_macaca_mulatta:
 ATGGATTTATCTGCTGTTCGCGTTGAAGA-AG-TAC--AA-AAT-G-TCATTAATGCTATGCA-GAAAATCTTAG--AGTGTCCAAT-CTGTCTGGAGTTGATCAAGGAA-CCT-G
 TCTCCACAAAGTGTGA-C-CA-C--A-TATTTTGCAGAT-TTTG-CATGCTGAA-AC-TTCTCAACCA-GAAGAAAGGGCCTTCACAGTGTCC--TTTG-TGTAAGAATGA-


──────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────────────────────────────
uestion 3: Experiment to Validate Approximation Ratio

enter selected: >s1
ength: 10, Exact: 70, Approx: 70, Ratio: 1.000
enter selected: >s1
ength: 20, Exact: 135, Approx: 135, Ratio: 1.000
enter selected: >s2
ength: 30, Exact: 231, Approx: 246, Ratio: 1.065
enter selected: >s3
ength: 40, Exact: 318, Approx: 328, Ratio: 1.031
enter selected: >s2
ength: 50, Exact: 385, Approx: 416, Ratio: 1.081
enter selected: >s3
ength: 60, Exact: 440, Approx: 491, Ratio: 1.116
enter selected: >s1
ength: 70, Exact: 516, Approx: 558, Ratio: 1.081
enter selected: >s3
ength: 80, Exact: 589, Approx: 648, Ratio: 1.100
enter selected: >s2
ength: 90, Exact: 628, Approx: 701, Ratio: 1.116
enter selected: >s1
ength: 100, Exact: 687, Approx: 727, Ratio: 1.058
enter selected: >s1
ength: 110, Exact: 754, Approx: 820, Ratio: 1.088
enter selected: >s1
ength: 120, Exact: 810, Approx: 910, Ratio: 1.123
enter selected: >s1
ength: 130, Exact: 895, Approx: 991, Ratio: 1.107
enter selected: >s2
ength: 140, Exact: 957, Approx: 1022, Ratio: 1.068
enter selected: >s1
ength: 150, Exact: 1023, Approx: 1112, Ratio: 1.087
enter selected: >s1
ength: 160, Exact: 1080, Approx: 1168, Ratio: 1.081
enter selected: >s2
ength: 170, Exact: 1186, Approx: 1294, Ratio: 1.091
enter selected: >s2
ength: 180, Exact: 1158, Approx: 1243, Ratio: 1.073
enter selected: >s2
ength: 190, Exact: 1323, Approx: 1458, Ratio: 1.102
enter selected: >s1
ength: 200, Exact: 1379, Approx: 1494, Ratio: 1.083
                             Comparison: Exact vs. Approximate Alignment
╔════════════╦═════════════╦══════════════╦═══════╦══════════════╦════════════════╦═══════════════╦═════════════════╗
║ Seq Length ║ Exact Score ║ Approx Score ║ Ratio ║ Time (Exact) ║ Mem (Exact) MB ║ Time (Approx) ║ Mem (Approx) MB ║
╠════════════╬═════════════╬══════════════╬═══════╬══════════════╬════════════════╬═══════════════╬═════════════════╣
║     10     ║      70     ║       70     ║ 1.000 ║    0.001     ║     0.012      ║     0.787     ║      0.051      ║
║     20     ║     135     ║      135     ║ 1.000 ║    0.001     ║     0.073      ║     0.817     ║      0.050      ║
║     30     ║     231     ║      246     ║ 1.065 ║    0.004     ║     0.230      ║     0.809     ║      0.048      ║
║     40     ║     318     ║      328     ║ 1.031 ║    0.007     ║     0.529      ║     0.818     ║      0.048      ║
║     50     ║     385     ║      416     ║ 1.081 ║    0.011     ║     1.016      ║     0.792     ║      0.048      ║
║     60     ║     440     ║      491     ║ 1.116 ║    0.022     ║     1.736      ║     0.807     ║      0.048      ║
║     70     ║     516     ║      558     ║ 1.081 ║    0.029     ║     2.735      ║     0.823     ║      0.056      ║
║     80     ║     589     ║      648     ║ 1.100 ║    0.042     ║     4.060      ║     0.817     ║      0.084      ║
║     90     ║     628     ║      701     ║ 1.116 ║    0.065     ║     5.755      ║     0.879     ║      0.121      ║
║    100     ║     687     ║      727     ║ 1.058 ║    0.099     ║     7.867      ║     0.879     ║      0.170      ║
║    110     ║     754     ║      820     ║ 1.088 ║    0.113     ║    10.441      ║     0.886     ║      0.223      ║
║    120     ║     810     ║      910     ║ 1.123 ║    0.167     ║    13.523      ║     0.894     ║      0.298      ║
║    130     ║     895     ║      991     ║ 1.107 ║    0.195     ║    17.159      ║     0.981     ║      0.395      ║
║    140     ║     957     ║     1022     ║ 1.068 ║    0.239     ║    21.395      ║     1.032     ║      0.502      ║
║    150     ║    1023     ║     1112     ║ 1.087 ║    0.288     ║    26.276      ║     1.031     ║      0.629      ║
║    160     ║    1080     ║     1168     ║ 1.081 ║    0.352     ║    31.849      ║     1.094     ║      0.728      ║
║    170     ║    1186     ║     1294     ║ 1.091 ║    0.410     ║    38.158      ║     1.178     ║      0.868      ║
║    180     ║    1158     ║     1243     ║ 1.073 ║    0.489     ║    45.251      ║     1.256     ║      0.996      ║
║    190     ║    1323     ║     1458     ║ 1.102 ║    0.617     ║    53.171      ║     1.274     ║      1.146      ║
║    200     ║    1379     ║     1494     ║ 1.083 ║    0.708     ║    61.966      ║     1.362     ║      1.290      ║
╚════════════╩═════════════╩══════════════╩═══════╩══════════════╩════════════════╩═══════════════╩═════════════════╝
```