Nikola	Ivanović	2105038

Midterm Test 1

03 / 11 / 2023

Introduction

Please answer all the questions below and submit this document in **PDF format** by **10:30** on **November 17, 2023** (two weeks) to damiano.piovesan@unipd.it. Please rename the file as **midterm1_<surname>_<name>.pdf**.

Each student is assigned a different **DNA sequence** and a protein superfamily from the **CATH** database with 10 representative sequences. The first part of the test is based on the analysis of the DNA sequence, and the second part pertains to the analysis of the superfamily.

For each question, concisely explain all the steps (**maximum of 5 rows**) necessary to reproduce the results, including parameters, database queries, algorithms, etc. If relevant, you can provide source code, but it is not necessary.

Assignments (input)

1. DNA sequence and superfamily CATH identifiers are available in the Students' sheet.
2. Superfamily sequences are available here (Columns: PDB ID, chain ID, PDB domain start, PDB domain end, domain sequence).

Questions (part 1)

Paste below your assigned DNA sequence.

GGGACCACGAAAGAAGTCCACTCGTCCGAGCTCCGCGTCGACGAGGACCTCGCGATGGGTCTCACG
CCCTTGGACGAAGACGCCGGACCGCCCCTACCGTTCCTGTCACACAGACAGTGGTGGGCCGTCTAAG
AGTTACCCAGAGGGCACCAGTTTGTGATGTTTCACTTCGCCCGTCCAAGATTCATACACTAGCTGCACC
TTCTGGGCAAAAGGACGGGGAGCGACCTTCGACATCAGTTGATAAAACACTGCGTGTGGTTCTCCCGC
GACCAGGGGAAGAACAAC

1. Paste below the correct translation of your DNA sequence.
   Hint: *Test all possible frames (3 shifts x 2 forward/complement x 2 directions) as the sequence could be a fragment, complement, or inversion of a gene.* ***Warning***: *Making a mistake at this step can affect the entire exam.*

   PWCFLQVSRLEAQLLLERYPECGNLLLRPGGDGKDSVSVTTRQILNGSPVVKHYKVKRAGSKYV
   IDVEDPFSCPSLEAVVNYFVTHTKRALVPFLL

The BioPython *translate* function was executed against the DNA sequence for all possible frames (12 in total). The parameter *to_stop* was set to *True*, thus the resulting protein sequences were translated until the first stop. When examining the results, several of the longest sequences had equal or similar lengths, thus the correct translation was obtained as the one that gave optimal results using *BLAST*.

2. Align the amino acid sequence against the SwissProt database using the BLAST service.
   ***Hint**: If the search against SwissProt does not provide any significant hits, try using a larger database like UniRef50.*

   The search was performed by selecting the *SwissProt* database, setting the maximum number of match alignments to 1000 and alignment views to *BLASTXML*. Additionally, the exponential threshold (*EXP.THR*) was set to *1e-2 (0.01)* in order to obtain only the significant hits.

   a. How many significant hits?

      There are **10** hits with an E value lower than *0.01*. When searching with the default setting (*EXP.THR = 10*), the results contain 38 hits.

   b. What is the coverage of the query sequence (your input) with the best matched sequence?

      The coverage of the query sequence is indicated by the *Identities* column of the result table and in this case it is **100%** - fully covered.

   c. What is the coverage of the best matched sequence with your input sequence?

      The coverage of the best matched sequence can be obtained from the result as: *(subject end - subject start + 1) / alignment length* from the downloaded *xml* results. In this case the coverage equals **35.29%**, which is visible in the result visual output.

   d. According to the BLAST results, is your input sequence a fragment or a full protein?

      The input sequence is a **fragment** as it only covers a relatively small part of the best matched sequence.

3. Compare your amino acid sequence with the best match found in the previous BLAST by using Needleman-Wunsch and Smith-Waterman algorithms.
   ***Hint**: You can retrieve the full sequence of a matched protein from UniProtKB using the protein ID or Accession code available in the BLAST output.*

   The algorithms were executed using *BioPython* and the *Align* module. The mode was set to *global* for *Needleman-Wunsch* and *local* for the *Smith-Waterman* algorithm. Both algorithms were executed with *open gap score = -1*, *extend gap score = -1*, using the *BLOSUM62* substitution matrix for scoring. Similarity was calculated as the ratio of exact matches and the length of the alignment: *similarity = identities/alignment length*

   a. Provide identity, number of gaps, similarity, and score for the two alignments generated with the two different algorithms.

      Needleman-Wunsch: identity: 96, gaps: 315, similarity: 23.36%, score: 190.0

      Smith-Waterman: gaps: identity: 96, gaps: 0, similarity: 100.00%, score: 505.0

   b. Which algorithm gives the best alignment? Why?

The *Smith-Waterman* (*local*) alignment gives better results because the query sequence is only a fragment of the target sequence.

c. Which algorithm between BLAST and Smith-Waterman gives the best alignment? Why?

Both algorithms give the optimal alignment because the query and target sequences are very similar.

4. Evaluate your amino acid sequence against the Pfam database of HMM models.
***Hint**: Use the HMMER web services to evaluate HMM models. Use the UniProtKB website (advanced search) to search Pfam proteins.*

The amino acid was evaluated using *PHMMER*, against SwissProt in order to discover its Pfam domain. After obtaining the domain, it was searched against UniProtKB. To accomplish this, an advanced search was performed, selecting Domain [FT] in the search form. The number of matches from Swiss-Prot is indicated under *Status -> Reviewed (Swiss-Prot).*

a. Which Pfam domain(s) match your sequence?

Exact match with query architecture: **SH2**

b. Is your sequence fully covered by Pfam domains?

No, since the alignment starts at position 2 and ends in position 83 (out of 96), with several gaps, according to the *PHMMER* search results.

c. How many proteins in SwissProt have the same domain? (Consider only one Pfam ID if your protein is multi-domain.)

454 results.

Questions (part 2)

Paste below your assigned superfamily identifier.

3.40.109.10

5. Compare the sequences of your superfamily provided in the assignment file by performing an all-vs-all pairwise sequence alignment.
***Hint**: You can use BioPython and the Align module.*

The *BioPython Align* module was used to compare the proteins against each other, using a *global* alignment with *open gap score = -1*, *extend gap score = -1*, using the *BLOSUM62* substitution matrix. The percent identity score was used as it seemed appropriate: *score = identities / alignment length.* Determining the sequence most similar to all others was done by computing the average score per protein and choosing the one with the greatest value.

a. Paste below a 10 x 10 matrix where cells represent the pairwise sequence identity. Provide sequence identifiers in the matrix tick labels.

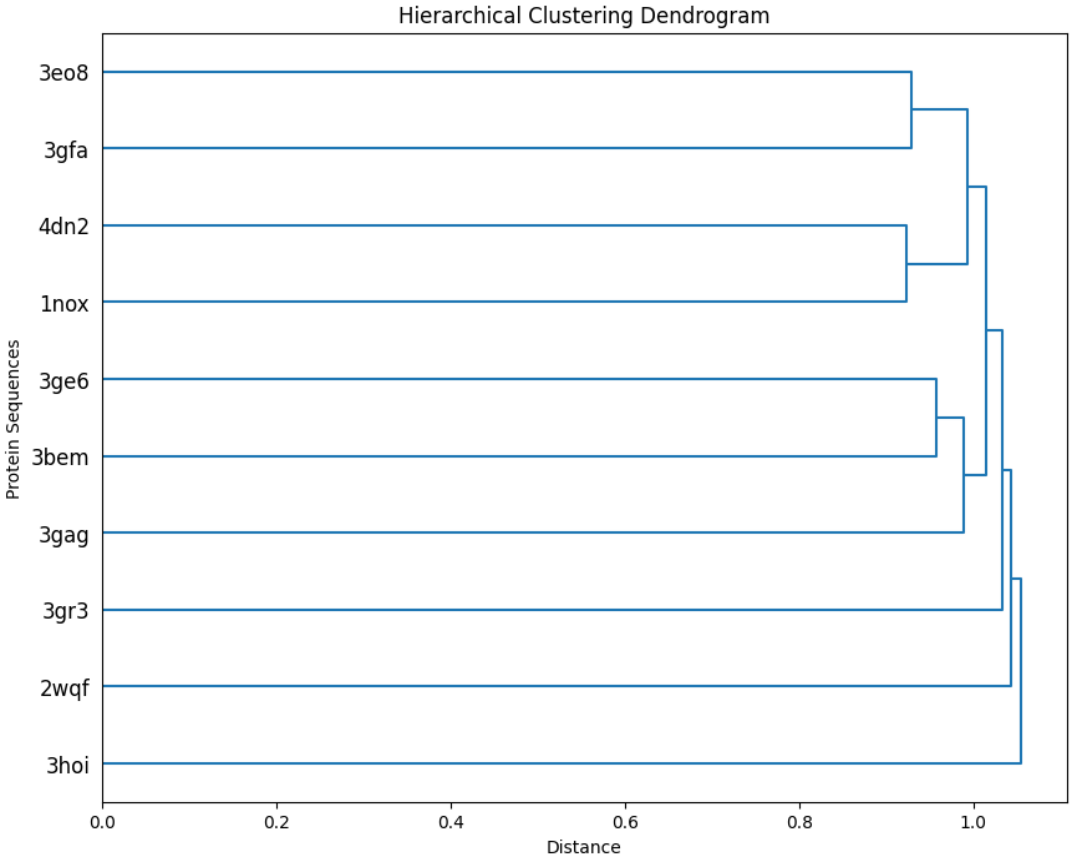
	3gfa	2wqf	1nox	3bem	3gr3	4dn2	3gag	3hoi	3ge6	3eo8
3gfa	100.00%	28.24%	30.50%	27.61%	27.21%	29.55%	24.81%	26.17%	30.08%	34.39%
2wqf	28.14%	100.00%	26.04%	25.81%	24.91%	28.24%	24.91%	24.62%	27.37%	27.24%
1nox	30.38%	25.93%	100.00%	29.48%	28.00%	34.78%	30.47%	28.62%	32.06%	29.45%
3bem	27.61%	25.36%	29.21%	100.00%	27.27%	27.27%	28.83%	27.14%	32.58%	25.54%
3gr3	26.86%	24.91%	27.80%	26.32%	100.00%	28.36%	25.89%	25.18%	26.33%	27.80%
4dn2	29.55%	28.12%	35.04%	26.72%	28.36%	100.00%	27.17%	24.71%	32.56%	30.55%
3gag	24.81%	24.91%	29.96%	28.83%	26.33%	28.74%	100.00%	22.01%	31.66%	26.95%
3hoi	26.17%	24.81%	28.15%	27.04%	24.91%	25.58%	21.72%	100.00%	26.62%	27.37%
3ge6	29.81%	27.47%	32.44%	32.58%	26.95%	32.30%	31.78%	26.62%	100.00%	29.24%
3eo8	34.39%	27.60%	29.56%	25.54%	28.14%	30.77%	27.50%	27.64%	28.78%	100.00%

b. Which sequence is the most similar to all other sequences?

The sequence most similar to all others is **3ge6** with average identity **36.92%**

c. Based on sequence identity values, are there sequences that can be grouped in the same (sub)family?

Sequences with high identity values potentially belong to the same (sub)family because they share a higher degree of sequence similarity. Observing the matrix, no proteins seem to belong in the same (sub)family. However, executing a hierarchical clustering algorithm over the matrix gives better insight into protein similarities.



6. Create a multiple sequence alignment (MSA) starting from the domain sequences available in the assignment file using EBI ClustalOmega.

   The MSA was created using the Clustal Omega web service, selecting Pearson/FASTA as output format. The sequence was then read and stored as a sequence of characters. For every column, occupancy was computed and different amino acids were counted in order to determine the column entropy. The results indicate that the same columns were the most conserved in both aspects when the parameters are adjusted in a specific way.

   a. Which columns are the most conserved when looking at the amino acid composition?

      Columns with occupancy above **0.5** and predominantly containing a single amino acid (above **0.8** of the column):

```
     pos  occupancy   entropy            counts
26    26        1.0  0.000000         [(R, 10)]
28    28        1.0  0.108515  [(S, 9), (T, 1)]
63    63        1.0  0.108515  [(A, 9), (S, 1)]
64    64        1.0  0.000000         [(P, 10)]
65    65        1.0  0.108515  [(S, 9), (T, 1)]
68    68        1.0  0.000000         [(N, 10)]
70    70        1.0  0.000000         [(Q, 10)]
139  139        0.7  0.136900  [(D, 6), (K, 1)]
201  201        0.9  0.116443  [(G, 8), (S, 1)]
215  215        1.0  0.108515  [(G, 9), (R, 1)]
256  256        1.0  0.000000         [(G, 10)]
```

   b. Which columns are the most conserved when looking at the column entropy?

      Columns with occupancy above **0.5** and entropy below **0.15**:

```
     pos  occupancy   entropy            counts
26    26        1.0  0.000000         [(R, 10)]
28    28        1.0  0.108515  [(S, 9), (T, 1)]
63    63        1.0  0.108515  [(A, 9), (S, 1)]
64    64        1.0  0.000000         [(P, 10)]
65    65        1.0  0.108515  [(S, 9), (T, 1)]
68    68        1.0  0.000000         [(N, 10)]
70    70        1.0  0.000000         [(Q, 10)]
139  139        0.7  0.136900  [(D, 6), (K, 1)]
```

```
201  201       0.9  0.116443  [(G, 8), (S, 1)]

215  215       1.0  0.108515  [(G, 9), (R, 1)]

256  256       1.0  0.000000        [(G, 10)]
```

7. Use the MSA generated before to perform a PSI-BLAST and an HMMER search against human proteins (or SwissProt if the search against human returns nothing).
   Hint*: For the PSI-BLAST search, you can use the NCBI web service and provide a PSSM generated with the PSI-BLAST command line. For HMMSEARCH you can provide directly the MSA.*

   Using the *PSI-BLAST* command line, the *PSSM* was generated using the *FASTA* file containing the protein superfamily along with the *FASTA* file that contains the *MSA* from the previous task. Then, using *NCBI BLAST*, a search was performed by uploading the *PSSM* file, selecting the appropriate database and organism, selecting *PSI-BLAST* as the algorithm and setting the expect threshold to 0.01. The *HMMSEARCH* was performed by uploading the *MSA* file and selecting the appropriate database, with the significance E-value of 0.01.

   a. How many significant hits are returned by the two methods?

      SwissProt:

      PSI-BLAST: 2 Sequences with E value better than threshold

      HMMER Search: 146 Significant Query Matches in *swissprot* (v.2021_04)

      Ensembl Human:

      PSI-BLAST: No significant similarity found

      HMMER Search: 6 Significant Query Matches in *ensembl* (v.104)

The code for this solution is available at: https://github.com/ivanovicnikola/biological-data-midterm-1