ClustAlignPy

oumayma meskini

September 2024

1 Introduction

Multiple sequence alignment (MSA) is a fundamental bioinformatics technique used to arrange sequences of DNA, RNA, or proteins in such a way that regions of similarity can be identified. These similarities can indicate functional, structural, or evolutionary relationships between the sequences. MSA plays a crucial role in many biological applications, including the identification of conserved motifs, phylogenetic analysis, and the prediction of protein structure and function. By aligning multiple sequences, it becomes possible to study the evolutionary distances between species, detect genetic variations, and highlight conserved regions that are critical for biological functions.

In this project, the goal is to perform a multiple sequence alignment of proteins, using dynamic programming through the Needleman-Wunsch algorithm. The process involves several steps, including the use of methods like UPGMA for phylogenetic tree construction, inspired from [1].

2 Materials & Methods

As mentionned above, this program performs multiple sequence alignment using dynamic programming through the Needleman-wunsch algorithme, executing several keys steps.

First, pairwise alignments are performed on the input sequences to construct a score matrix, where each cell represents the alignment score between two sequences. This score matrix is then transdormed into a distance matrix by normalizing the scores between 0 and 1, using the formula : $\frac{\text{score}-\text{score}_{\min}}{\text{score}_{\max}-\text{score}_{\min}}$ The normalized values are then substructed from 1 to convert scores into distances, such that a higher alignment score (closer sequences) results in a smaller distance, and vice versa.

The resulting distance matrix is utilized to perform clustering and generate a phylogenetic tree, ordered from the closest sequences to the most distant. The UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm is employed for this purpose, progressively clustering the sequences by minimizing distances at each step, using this formula for distance calculation :

$$d_{(C_i,C_j)} = \frac{|C_i|\cdot d_{(C_i,C_k)}+|C_j|\cdot d_{(C_j,C_k)}}{|C_i|+|C_j|}$$

- $d_{(C_i,C_j)}$: The distance between clusters C_i and C_j.
- $d_{(C_i,C_k)}$: The distance between cluster C_i and another cluster or sequence C_k.
- $d_{(C_j,C_k)}$: The distance between cluster C_j and another cluster or sequence C_k.
- $|C_i|$: The number of sequences in cluster C_i.
- $|C_j|$: The number of sequences in cluster C_j.

The method ensures that the sequences are grouped and merged hierarchically, reflecting their evolutionary proximity.

Once an order for the sequences is determined from the UPGMA tree, the sequences are progressively aligned based on their proximity, starting with the closest pair and expanding to larger clusters, using a fixed gap penalty of -8. This step by step alignement follows the structure dictated by the phylogenetic tree, ensuring an optimal multiple sequence alignment.

3 Results

To ensure the accuracy and functionality of our alignment algorithm, we first tested it by aligning a sequence (glucagon protein 36aa) with it self. As shown in Fig1, this results in a perfect alignment as expected, and confirm that the algorithm computes scores correctly and produces a maximal alignment score.

```
sp|P81026-0|GLUC1_ORENI       HSEGTFSNDYSKYLEDRKAQDFVRWLMNNKRSGAAE 36
sp|P81026-1|GLUC1_ORENI       HSEGTFSNDYSKYLEDRKAQDFVRWLMNNKRSGAAE 36
```

Figure 1: Glucagon protein alignment with it self using ClustAlignPy.

Next, we aligned Tropomyosin alpha-1 chain protein (TPM1) with two of its isoforms, as illustrated in Fig2. We observe that the majority of the protein sequences are highly conserved and well-aligned. Isoforms 2 and 3

display gaps in the initial portion of their sequences and are found to be the most closely related to each other. This alignment, involving 3 sequences of approximately 250 amino acids each, took 3.08 seconds as we can see in Fig 2.

```
sp|P06753-2|Isoform_2         -MAGI-TTIEAVK-RKIQVL-Q-QQAD-D---AEERA---E-RL---QREVEG---E--R-----R-A-- 41
sp|P06753-3|Isoform_3         -MAGI-TTIEAVK-RKIQVL-Q-QQAD-D---AEERA---E-RL---QREVEG---E--R-----R-A-- 41
sp|P06753|TPM3_HUMAN          MMEAIKKKMQMLKLDKENALDRAEQAEAEQKQAEERSKQLEDELAAMQKKLKGTEDELDKYSEALKDAQE 70


sp|P06753-2|Isoform_2         R-E--Q-----AEAEVASLNRRIQLVEEELDRAQERLATALQKLEEAEKAADESERGMKVIENRALKDEE 103
sp|P06753-3|Isoform_3         R-E--Q-----AEAEVASLNRRIQLVEEELDRAQERLATALQKLEEAEKAADESERGMKVIENRALKDEE 103
sp|P06753|TPM3_HUMAN          KLELAEKKAADAEAEVASLNRRIQLVEEELDRAQERLATALQKLEEAEKAADESERGMKVIENRALKDEE 140


sp|P06753-2|Isoform_2         KMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEGDLERTEERAELAESRCREMDEQIRLMDQNLKCLSA 173
sp|P06753-3|Isoform_3         KMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEGDLERTEERAELAESRCREMDEQIRLMDQNLKCLSA 173
sp|P06753|TPM3_HUMAN          KMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEGDLERTEERAELAESKCSELEEELKNVTNNLKSLEA 210


sp|P06753-2|Isoform_2         AEEKYSQKEDKYEEEIKILTDKLKEAETRAEFAERSVAKLEKTIDDLEDKLKCTKEEHLCTQRMLDQTLL 243
sp|P06753-3|Isoform_3         AEEKYSQKEDKYEEEIKILTDKLKEAETRAEFAERSVAKLEKTIDDLEERLYSQLERNRLLSNELKLTLH 243
sp|P06753|TPM3_HUMAN          QAEKYSQKEDKYEEEIKILTDKLKEAETRAEFAERSVAKLEKTIDDLEDELYAQKLKYKAISEELDHALN 280


sp|P06753-2|Isoform_2         DLNEM 248
sp|P06753-3|Isoform_3         DLCD- 247
sp|P06753|TPM3_HUMAN          DMTSI 285


Execution time: 3.08 seconds
```

Figure 2: TPM1 alignment with isoform 2 and 3 using ClustAlignPy.

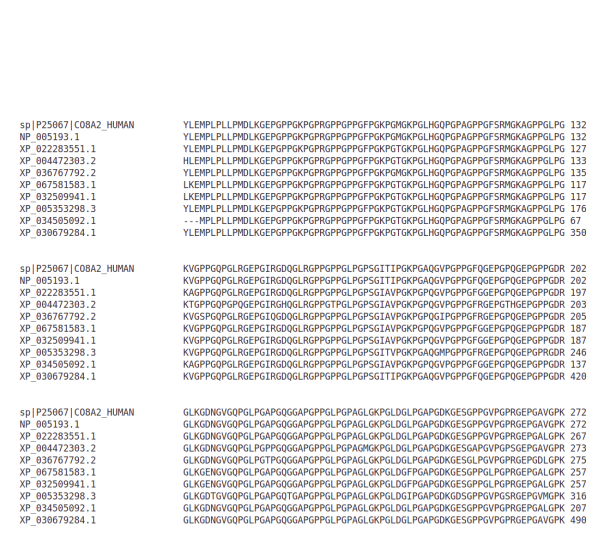
```
sp|P06753|TPM3_HUMAN       -----MMEAIKKKMQMLKLDKENALDRAEQAEAEQKQAEERSKQLEDELAAMQKKLKGTE 55
sp|P06753-2|Isoform_2      MAGITTIEAVKRKIQVLQQQADD------------------------------------ 23
sp|P06753-3|Isoform_3      MAGITTIEAVKRKIQVLQQQADD------------------------------------ 23
                                 :**:*:*:*:*: : ::

sp|P06753|TPM3_HUMAN       DELDKYSEALKDAQEKLELAEKKAADAEAEVASLNRRIQLVEEELDRAQERLATALQKLE 115
sp|P06753-2|Isoform_2      -----AEERAERLQREVEGERRAREQAEAEVASLNRRIQLVEEELDRAQERLATALQKLE 78
sp|P06753-3|Isoform_3      -----AEERAERLQREVEGERRAREQAEAEVASLNRRIQLVEEELDRAQERLATALQKLE 78
                                 .*  :  *.::*  .:   :*********************************

sp|P06753|TPM3_HUMAN       EAEKAADESERGMKVIENRALKDEEKMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEG 175
sp|P06753-2|Isoform_2      EAEKAADESERGMKVIENRALKDEEKMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEG 138
sp|P06753-3|Isoform_3      EAEKAADESERGMKVIENRALKDEEKMELQEIQLKEAKHIAEEADRKYEEVARKLVIIEG 138
                           ************************************************************

sp|P06753|TPM3_HUMAN       DLERTEERAELAESKCSELEEELKNVTNNLKSLEAQAEKYSQKEDKYEEEIKILTDKLKE 235
sp|P06753-2|Isoform_2      DLERTEERAELAESRCREMDEQIRLMDQNLKCLSAAEEKYSQKEDKYEEEIKILTDKLKE 198
sp|P06753-3|Isoform_3      DLERTEERAELAESRCREMDEQIRLMDQNLKCLSAAEEKYSQKEDKYEEEIKILTDKLKE 198
                           **************:* *::*::: : :***.*.*  **********************

sp|P06753|TPM3_HUMAN       AETRAEFAERSVAKLEKTIDDLEDELYAQKLKYKAISEELDHALNDMTSI   285
sp|P06753-2|Isoform_2      AETRAEFAERSVAKLEKTIDDLEDKLKCTKEEHLCTQRMLDQTLLDLNEM   248
sp|P06753-3|Isoform_3      AETRAEFAERSVAKLEKTIDDLEERLYSQLERNRLLSNELKLTLHDLCD-   247
                           **********************:.* .   .    .. *. :* *: .
```

Figure 3: TPM1 alignment with isoform 2 and 3 using Clustal 0(1.2.4).

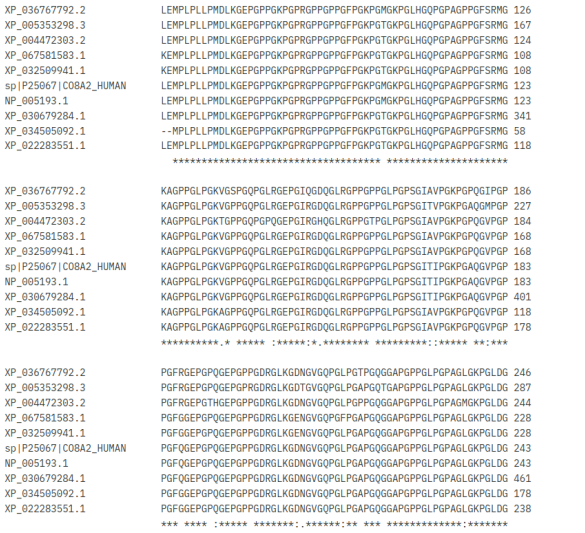
Comparing the alignment produced with our algorithm to Clustal 0(1.2.4) (Fig3), we first see that the conserved regions are aligned similarly. In addition, isoforms present as well gaps in the initial portion of their sequences, however, gaps are often extended over a broader region in the Clustal 0(1.2.4) alignment. In our program, the gap penalty is fixed, which tends to result in gaps appearing individually within the aligned sequences, while Clustal use an affine gap penalty, where the penalty for opening a gap is higher than the penalty for extending it. This method better reflects biological reality, as gaps in real sequences typically occur over continuous regions rather than as isolated insertions or deletions. This difference in gap handling explains the observed variation between our program and Clustal's results.

Finally, we aligned 10 collagen proteins from different organisms and different isoforms, from 640 to 920

amino acids, that took 8 min (Fig4). The alignment produced shows some conserved regions (Fig 4) as well as variable regions (Fig 5). When comparing our alignment with the one provided by Clustal tool, we can see that the conserved regions are similar, while they are differences in the variable regions containing gaps. This again can be explained by the gap handling. We can note that the ordering or proximity of sequences is also different between ClustAlignPy alignment and Clustal tool alignment.

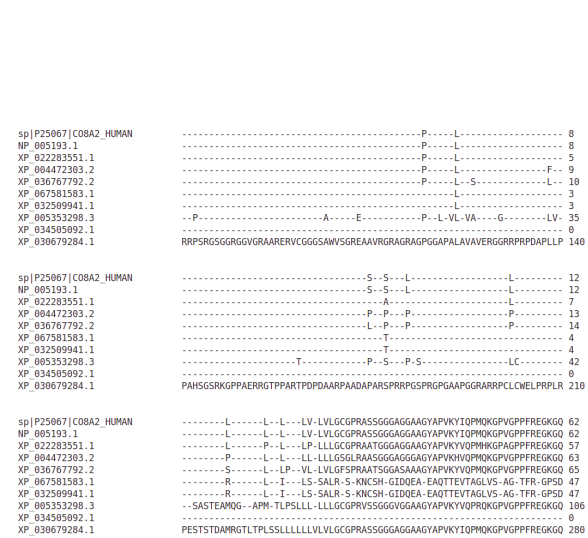


(a) 10 Collagen proteins alignement by ClustalignPy, conserved regions

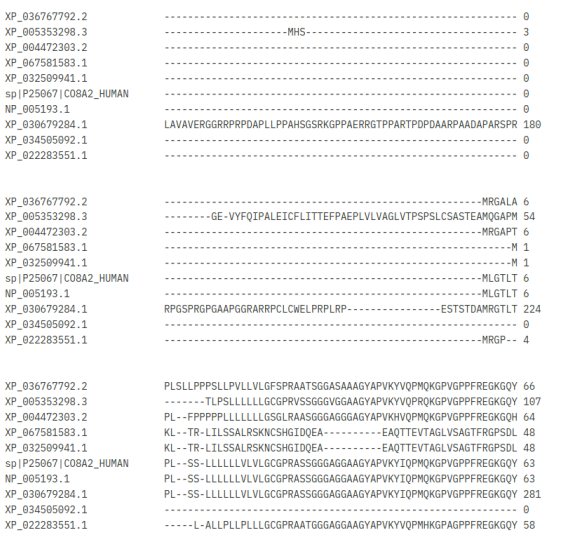


(b) 10 Collagen proteins alignement by Clustal 0(1.2.4), conserved regions

Figure 4: Conserved regions for aligned collagen proteins.



(a) 10 Collagen proteins alignement by ClustalignPy, non conserved regions



(b) 10 Collagen proteins alignement by Clustal 0(1.2.4), non conserved regions

Figure 5: Non conserved regions for collagen aligned proteins.

4 Discussion

The program could be improved by incorporating an affine gap penalty instead of a fixed penalty. Affine gaps better reflect biological reality, as insertions and deletions often occur in contiguous regions rather than isolated events. This adjustment would lead to more accurate alignments, similar to those produced by tools like Clustal.

Additionally, implementing parallelization could significantly reduce execution time. By distributing computations, such as pairwise alignments and distance matrix calculations, across multiple processors, the program

would become more efficient, especially when handling larger datasets. These enhancements would improve both biological relevance and computational performance.

Bibliography

[1] Desmond G Higgins and Paul M Sharp. "Fast and sensitive multiple sequence alignments on a microcomputer". In: *Comput Appl Biosci* 5.2 (Apr. 1989), pp. 151–153. DOI: 10.1093/bioinformatics/5.2.151.