12 septembre 2024

Alignement multiple heuristique par la méthode CLUSTAL

DUONG Karine (22011253) - M2 BI

Abstract

Desmond G. Higgins et al. [1] décrit l'implémentation de la méthode d'alignement par la méthode CLUSTAL avec des algorithmes optimisés, tels que l'algorithme de Wilbur et Lipman, ou encore celui de Gotoh.
Dans ce rapport, nous allons discuter de notre version de l'implémentation de la méthode CLUSTAL en python avec des algorithmes simplifiés et des différences possibles avec le programme CLUSTAL Omega [2], tout en obtenant des résultats appropriés.

Code source : https://github.com/KarinDuong/clustal_alignement

Abréviations: *NW* = Needleman and Wunsch; UPGMA = Unweighted pair group method with arithmetic mean

1. Introduction

L'alignement multiple de séquence (*MSA*) est une technique fondamentale en bioinformatique, utilisée pour trouver les régions conservées, relier la séquence à la structure et à la fonction mais aussi permettre de construire un arbre phylogénétique pour les séquences données.
Parmi les différentes méthodes de MSA existantes, CLUSTAL est la méthode la plus précise et performante. L'article de Desmond G. Higgins et Paul M. Sharp [1] évoquent la stratégie et les différents algorithmes utilisés pour implémenter CLUSTAL.
Ce projet *Clustal_alignement* nous a permis d'implémenter la démarche évoquée dans l'article [1], en utilisant des algorithmes simplifiés.

2. Matériels et méthodes

2.1 Développement

L'implémentation de ce projet a été faite avec `Python (3.11.9)`. Différents modules python ont été utilisés tels que : `argparse (1.1)` pour permettre de définir des arguments dans la ligne de commande; `biopython (1.84)` pour permettre de lire et extraire les données d'un fichier fasta; `numpy (1.26.4)` pour pouvoir manipuler les données sous forme d'array; `pathlib (1.0.1)` pour pouvoir vérifier l'existence et l'extension du fichier fasta; `pandas (2.2.2)` pour pouvoir manipuler les données sous forme de dataframe.

2.2 Implémentations

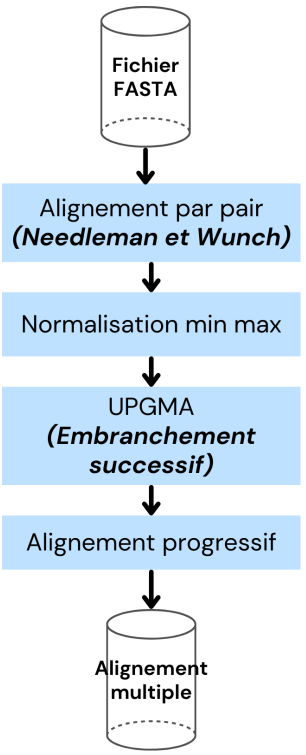


Fig. 1: **Organigramme** des grandes étapes utilisées pour l'implémentation de ce projet.

A partir d'un fichier FASTA, nous récupérons les séquences d'intérêts. Nous alignons ses séquences par paires (à l'aide de l'algorithme d'alignement global NW [Eq.1]) afin d'extraire l'alignement optimal pour ces séquences et le score de cet alignement.

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + \text{score}(a_i, b_j) \\ M_{i,j-1} + \text{score}_{gap} \\ Mi-1,j + \text{score}_{gap} \end{cases}$$

Équation 1 – **Formule utilisée pour déterminer la valeur de chaque case selon les cases voisines**, pour la matrice de score. Où score(ai, bj) correspond à la valeur dans la matrice BLOSUM62 ou Pam pour l'élément i et j sur la séquence A et B.

En récupérant le score de tous les alignements, nous pouvons construire une matrice de score. Cette matrice va être convertie en matrice de distance à l'aide d'une normalisation min max [Eq.2].

$$ValNormalis_{i,j} = \frac{Val_{i,j} - ValMin}{ValMax - VaMin}$$

Équation 2 - **Formule utilisée pour calculer la normalisation de la valeur à une position donnée**.

A partir de cette matrice de distance, nous allons pouvoir construire un arbre guide (à l'aide de l'algorithme UPGMA [Eq.3]), qui va déterminer l'ordre des séquences pour l'alignement progressif.

$$d(AB, C) = \frac{d(A, C) + d(B, C)}{len(numrateur)}$$

Équation 3 – **Formule utilisée pour calculer les distances intermédiaires** de la matrice de distance au cours des itérations. Il s'agit ici d'un exemple pour avoir la distance entre le cluster AB et la séquence C, où d(A,C) et d(B,C) sont les distances entre les séquences A-C et B-C.

L'alignement progressif consiste à ré-appliquer l'algorithme de NW sur les séquences initiales contre un cluster de séquences alignées en suivant l'ordre donné par l'arbre guide. Où le calcul de la diagonale dans la matrice de score prend maintenant en compte la somme du score (dans la matrice blosum62) de tous les acides aminés présents - *à une position donnée* - dans le cluster de séquence à une position donnée et la séquence étudiée.

3. Résultats

Clustal_alignment retourne le résultat de alignements multiples dans le terminal.

Nous tenons d'abord à préciser que notre programme n'utilise pas les mêmes algorithmes d'optimisation que le programme en ligne Clustal Omega [2], qui est plus complexe. Il utilise par exemple un score de gap affine pour différencier les ouverture et extension de gap, ce qui conduit à des matrices de score différentes et donc un arbre guide différent[Fig. 2A et 3A]. Cela peut expliquer les légères différences entre nos résultats [Fig. 2, 3]. Ces différences sont moins marquées lorsque l'on aligne des séquences avec des régions conservées.
Malgré cette différence pour le score des gap, l'alignement de ces trois séquences d'exemple (*data/sequence_test_court.fasta*) semble similaire à quelques résidus près [Fig 2B et 3B].

En ce qui concerne le temps d'exécution de notre programme, cela dépend du nombre de séquences testées et de leur nombre de résidus [Tab.1]. Il s'agit d'une complexité $O(n^2)$, car …

['QMS45321.1', 'XP_004050475.2', 'NP_000198.1']

(A)

(B)

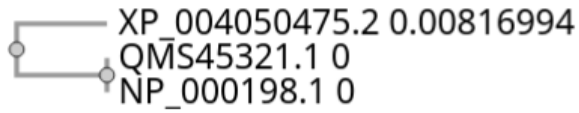
```
QMS45321.1:     MGSETIKPAGAQQPSALQDRLHQKRPSSRSVPRAFASGGLRIPGWLDPRPQLCSREDVAGLVK
XP_004050475.2: MGSETIKPVGTQQPSALQDRLHQKRPSSRSVPRAF_____________________________
NP_000198.1:    ________________________________________________________________

QMS45321.1:     HVGVSPGAPRQGTWPSACLSPACLPDHCPSAMALWMRLLPLLALLALWGPDPAAAFVNQHLCG
XP_004050475.2: _________________A__S_____DHCPSAMALWMRLLPLLALLALWGPDPAAAFVNQHLCG
NP_000198.1:    ___________________________________MALWMRLLPLLALLALWGPDPAAAFVNQHLCG

QMS45321.1:     SHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQC
XP_004050475.2: SHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQC
NP_000198.1:    SHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQC

QMS45321.1:     CTSICSLYQLENYCN
XP_004050475.2: CTSICSLYQLENYCN
NP_000198.1:    CTSICSLYQLENYCN
```

Figure 2: **Comparaison des résultats** produits par notre programme sur le fichier *data/sequence_test_court.fasta.* (A) L'ordre des clusters produit par UPGMA, qui se lit de gauche à droite pour avoir les embranchements successifs c'est-à-dire *((QMS, XP),NP)*. (B) L'alignement de ses trois séquences selon l'ordre donné par les clusters.



(A)

(B)

```
XP_004050475.2      MGSETIKPVGTQQPSALQDRLHQKRPSSRSVPRAFA----------------------      36
QMS45321.1          MGSETIKPAGAQQPSALQDRLHQKRPSSRSVPRAFASGGLRIPGWLDPRPQLCSREDVAG      60
NP_000198.1         ------------------------------------------------------------      0


XP_004050475.2      --------------------------SDHCPSAMALWMRLLPLLALLALWGPDPAAAFV      69
QMS45321.1          LVKHVGVSPGAPRQGTWPSACLSPACLPDHCPSAMALWMRLLPLLALLALWGPDPAAAFV      120
NP_000198.1         ---------------------------------MALWMRLLPLLALLALWGPDPAAAFV      26
                                                     *************************


XP_004050475.2      NQHLCGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSL      129
QMS45321.1          NQHLCGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSL      180
NP_000198.1         NQHLCGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSL      86
                    ************************************************************


XP_004050475.2      QKRGIVEQCCTSICSLYQLENYCN    153
QMS45321.1          QKRGIVEQCCTSICSLYQLENYCN    204
NP_000198.1         QKRGIVEQCCTSICSLYQLENYCN    110
                    ************************
```

Figure 3: **Comparaison des résultats** produits par CLUSTAL Omega sur le fichier *data/sequence_test_court.fasta*. (A) Le dendrogramme produit par l'étape de clustering. (B) L'alignement de ses trois séquences selon l'ordre donné par les clusters.

Nombre de séquences	*Temps d'exécution*
3	0.02 sec
9	0.3 sec
25	4min

Tableau 1: **Récapitulatif du temps d'exécution** pour un nombre de séquences variables et pour des séquences d'environ 150-200 acides aminés.

Discussion

Différents points de notre algorithme peuvent être optimisés, en appliquant par exemple la parallélisation lors des alignements par paires afin de gagner du temps de complexité et d'exécution.
De plus, certaines fonctionnalités aurait pû être ajoutées, en utilisant par exemple un score de gap affine plutôt qu'une constant selon s'il s'agit d'une ouverture de gap ou une extension de gap.

Conclusion

Notre implémentation de l'alignement multiple par la méthode CLUSTAL n'est pas parfaite.
L'utilisation de la méthode d'embranchement successif a été utilisée pour faciliter la partie alignement progressif, en ne traitant qu'un cluster de séquence à la fois et en définissant un ordre d'alignement linéaire.

Références bibliographiques

[1] Desmond G. Higgins , Paul M. Sharp, *Fast and sensitive multiple sequence alignments on a microcomputer,* Bioinformatics, Volume 5, Issue 2, April 1989, Pages 151–153. URL: https://doi.org/10.1093/bioinformatics/5.2.151

[2] Fabio Madeira, Nandana Madhusoodanan, and al. *The EMBL-EBI Job Dispatcher sequence analysis tools framework in 2024.* Nucleic Acids Research, 01 July 2024. URL: https://doi.org/10.1093/nar/gkae241
Tool : https://www.ebi.ac.uk/jdispatcher/msa/clustalo